PAPERS
IN COMPUTATIONAL LEXICOGRAPHY
COMPLEX '96

Edited by
Ferenc Kiefer, Gábor Kiss and Júlia Pajzs

LINGUISTICS INSTITUTE
HUNGARIAN ACADEMY OF SCIENCES, BUDAPEST

PAPERS IN COMPUTATIONAL LEXICOGRAPHY
COMPLEX '96

PAPERS IN COMPUTATIONAL LEXICOGRAPHY COMPLEX '96

Edited by
Ferenc Kiefer, Gábor Kiss and Júlia Pajzs

RESEARCH INSTITUTE FOR LINGUISTICS
HUNGARIAN ACADEMY OF SCIENCES, BUDAPEST
1996

Proceedings of the 4th International Conference on
Computational Lexicography, COMPLEX '96
Budapest, Hungary

All correspondence should be sent to
Research Institute for Linguistics
Hungarian Academy of Sciences
Department of Lexicography and Lexicology
Budapest P.O. Box 19
Hungary 1250

Cover design by Gábor Kiss
Technical assistence: Judit Pais

ISBN 963 8461 98 5

Hozott anyagról sokszorosítva

Contents

Preface

The COMPLEX conference on Computational Lexicography and Text Research is organized for the fourth time by the Research Institute of Linguistics of the Hungarian Academy of Sciences and the Laboratoire d'Automatique Documentaire et Linguistique Université Paris 7. The papers included in the proceedings cover most of the "hot topics" of this field. The connection of corpus research and electronic dictionaries is described from different viewpoints: in some projects the corpus serves as a basis for creating new dictionaries, in other cases the corpus is analysed by using already existing electronic dictionaries. One common problem in each case is the handling of multi-word lexemes, which is addressed by several papers in the collection. Morphological analysis is still an interesting and unsolved task for many languages, projects for creating algorithms and morphological lexicons are presented in a few articles.

We hope that we succeeded again in collecting present-day papers on this field. In the process of selection we were greatly helped by the members of the program committee: *Gregory GREFENSTETTE* Rank Xerox, Grenoble, *Maurice GROSS* Université Paris 7, *Ferenc KIEFER* Hungarian Academy of Sciences, *Ramesh KRISHNAMURTHY* COBUILD, *Tamás VÁRADI* Hungarian Academy of Sciences.

Júlia Pajzs

Preface

The COMPLEX conference on Computational Lexicography and Text Research is organized for the fourth time by the Research Institute of Linguistics of the Hungarian Academy of Sciences and the Laboratoire d'Automatique Documentaire et Linguistique Université Paris 7. The papers included in the proceedings cover most of the "hot topics" of this field. The connection of corpus research and electronic dictionaries is described from different viewpoints: in some projects the corpus serves as a basis for creating new dictionaries, in other cases the corpus is analysed by using already existing electronic dictionaries. One common problem in each case is the handling of multi-word lexemes, which is addressed by several papers in the collection. Morphological analysis is still an interesting and unsolved task for many languages; projects for creating algorithms and morphological lexicons are presented in a few articles.

We hope that we succeeded again in collecting present-day papers on this field. In the process of selection we were greatly helped by the members of the program committee: *Gregory GREFENSTETTE*, Rank Xerox, Grenoble, *Maurice GROSS* Université Paris 7, *Ferenc KIEFER* Hungarian Academy of Sciences, *Ramesh KRISHNAMURTHY* COBUILD, *Tamás VÁRADI* Hungarian Academy of Sciences.

Júlia Pajzs

MultiWord Lexical Units in EUSLEM, a lemmatiser-tagger for Basque

ALDEZABAL I. – ARTOLA X. – EZEIZA N. –
URIZAR R. – ADURIZ I.

Abstract

A lemmatiser-tagger must not only lemmatise word-forms consisting of a single lexical element but it must also be able to detect complex units. In this paper we try to delimit linguistically which complex words deserve to be lemmatised as a unit. Then, we propose a formal description for MultiWord Lexical Units (MWLU) in Basque —resulting of a conscientious analysis of their syntactic and morphological behaviour. Based on that formal description we propose a simple logical formalism to represent those MWLUs so that they can be automatically processed.

1. Introduction

A lemmatiser-tagger is a computational tool used for assigning the correct lemma and grammatical category to each token of a corpus. It is a basic device for corpus analysis, automatic indexation, syntactic and semantic analyses etc. For example, the lemmatiser-tagger for Basque (EUSLEM) (Aduriz *et al.*, 96a) is essential for the second phase of the Systematic Compilation of Modern Basque[1] (EEBS) project (Urkia *et al.*, 91.) being carried out by UZEI[2].

The EUSLEM project is being designed by a group from the Computer Science Faculty of the University of The Basque Country and UZEI. The knowledge in computational linguistics of the former added to the large experience in lexicography of the latter is turning out to be vital for the completion of this project. Some previous outcomes resulted from their collaboration can be considered both the basis and antecedent of this work:

- The Lexical Database for Basque (EDBL) (Agirre *et al.*, 95), which is the basis for many other applications, consists of (i) dictionary entries (the same you can find in any conventional dictionary), (ii) verb forms and (iii) dependent morphemes, all of them with their respective morphological information.
- A tagset system has been also developed. The one we have chosen for Basque is a three level system which the user can parametrise when using the programme. In the first level seventeen general categories are included (noun, adjective, verb, etc). In the second one each category tag is further refined by subcategory tags. The last level includes other interesting morphological information (case, number, etc.).
- With regard to lemmatisation, a morphological analyser was needed, specially for Basque, which is an agglutinative and morphologically complex language. With this purpose in mind, a morphological analyser was developed, based on the Two-Level Morphology (Koskenniemi, 83). Due to the fact that the process of normalisation of Basque is still in progress, the morphological processor had to be able to analyse linguistic variants and to distinguish between standard and non-standar lemmas. For this purpose, the treatment of variant errors was carried out using a two-level subsystem made up of (i) 1,000 items linked to the corresponding correct ones and (ii) twenty rules to cover the most common competence errors.
- If the lexicon-based analyser produced no valid analysis, we needed a lexicon-free lemmatiser that wouldn't let any token unlemmatised or untagged, so that the system might be robust enough. Among the different systems we studied, we chose a two-level mechanism based on the idea used in speech synthesis (Black *et al.*, 1991).

The morphological analyser at that step gives as a result *all* the possible analyses of each *token* in the text. So, on one hand, we must discard as many wrong analyses as possible. For that purpose we will apply a combination of two different strategies: the Constraint Grammar (CG) formalism (Karlsson *et al.*, 95; Aduriz *et al.*, 96b) based on linguistic knowledge and a procedure based on statistics and developed within MULTEX project (Amstrong *et al.*, 1995). For the moment, both methods are being tested separately. On the other hand, it is obviously not enough to lemmatise just the word-forms between two blanks. There are several MultiWord Lexical Units that must be lemmatised as a single unit.

Next we will try to delimit linguistically which word combinations deserve to be lemmatised as a unit. Then, we will propose a formal description for Basque MWLUs and a simple logical formalism so that they can be automatically processed. Eventually, the conclusions of this work will be expounded.

[1] During the 1987-1992 period UZEI manually compiled and lemmatised a three million word corpus of twentieth century's Basque texts, which is annually renewed.

[2] UZEI is a cultural association created in 1977 the aim of which is to promote Basque lexicon's modernisation within the normalisation process of this language.

2. Delimiting lexical units

Giving an exact definition for *word* is not an easy task at all. At text level one solution could be to define a *word* as "any string of characters between two blanks" (Fontenelle et al., 94). There are, of course, some lexical units that fit that definition (*etxe* 'house', *zuri* 'white'...). Even a great number of idioms that in non-inflected languages would be multiword, in Basque, which is a highly inflected language, constitute a single typographic unit (*ziurrenik* 'most probably', *aurrerantzean* 'from now on', *aurretiaz* 'in advance',...). But a different definition is obviously necessary for many other idiomatic expressions as well as for several MWLUs (*lan egin* 'to work', *beste bat* 'another', *hutsaren hurrengoa* 'the lowest of the low',...). We have mostly relied on the experience that UZEI gathered in the EEBS project, in which a wider perspective was taken and an extensive range of MWLUs was lemmatised as a unit. We have used those complex units as a basis for our analysis. Next, we will explain what we consider a MWLU.

First of all, we will focus on the treatment of compounds, for they constitute quite a special group within the MultiWord Lexical Units.

2.1. Compounds

Compounds in Basque can appear written mainly in four different ways (Euskaltzaindia, 92):

- attached (*idazmakina* 'writing machine', *plazagizon* 'public man',...)
- hyphenated (*datu-base* 'database', *begi-nini* 'pupil',...)
- separated by a blank and with the first component of the compound phonologically transformed (*Euskal Etxea* 'Basque House', *itsas armada* 'naval forces',... where *euskal* and *itsas* are the respective variants in composition and derivation of *euskara* 'Basque' and *itsaso* 'sea') and
- separated but with the components unaltered (*bake ituna* 'peace treaty',...).

For lemmatising purposes attached compounds are identical to single words and therefore are included in the lexical database as such.

Some of the hyphenated compounds that are considered to be lexicalised are included in our database as if they were single words (*botoi-zulo* 'buttonhole',...). Some others, being freely generated, (*mahai-hanka* 'table leg',...), don't appear in any dictionary and so they are not in our database either. However, the morphological analyser is capable of detecting them. In our database we consider the hyphen a lexical element, as the Two-Level Morphology renders it possible. Therefore, the different hyphens[3], can be followed, like any other lexical element in the database, by a particular set of morphemes and lexicons, and so certain types of compounds can be recognised.

Among compounds in which the first component has been modified there are also lexicalised compounds and freely generated ones. This kind of compounds will only be included in our database as MWLUs when they are lexicalised (*Boli Kosta* 'Ivory Coast'). Nevertheless, all compounds of that type will be detected in a later stage since phonologically transformed components are easy to detect:

* among the words that lose their final *-a*, there are, on one hand, those ending in *-ia* (*filologia* 'philology', *energia* 'energy', *psikologia* 'psychology',...) which are easily recognised by a simple rule. On the other hand, there is a reduced set of exceptions that can also lose their final *-a* in composition (***natura*** 'nature', ***literatura*** 'literature', ***kultura*** 'culture', ***burdina*** 'iron', *eliza* 'church' and *hizkuntza* 'language').

* the rest of the cases constitute quite a restricted set (*itsas, erret,...* variants of *itsaso* 'sea', *errege* 'king',...).

[3] In the database there is more than one type of hyphen because it has other uses distinct from composition, e.g. declension of foreign words (*Shakespeare-engandik*, 'from Shakespeare').

As for compounds separately written and not hyphenated, the only way of locating them is introducing them into the database. The freely generated compounds, though, cannot be located for the moment. Nonetheless, a computational tool that will retrieve lexical collocations automatically from corpora is being designed. That information will be later used to enrich the MWLU database.

2.2. MultiWord Lexical Units

Sometimes **syntactic patterns** (*zenbat eta ...-ago ... orduan eta ...-ago*, 'the ... -er ... the ... -er') are also lemmatised as MWLUs. Since Basque is an inflected language, frequently the boundary between syntax and morphology is quite vague. We think that those patterns are easier to treat at syntax level so we have left them for a later stage.

To distinguish between **lexical collocation** and **idioms** authors generally use a semantic criterion (Heid, 94). Idioms can hardly be interpreted in terms of the meanings of their constituents (*adarra jo* 'pull s.o.'s leg' ≠ *adarra* 'horn' + *jo* 'to play'), while in collocations their components (or at least one of them) keep their original sense (*zarata atera* 'make noise').

However, it is very difficult to outline the boundary between idioms and lexical collocations since from totally opaque idioms to open lexical collocation there is a wide range of word-combination arranged along a scale or continuum (Cowie, 90):

- *Pure idioms*. These are totally opaque idioms, that is, idioms *strictu sensu*. Historically, opaque idioms are "the end-point of a process in which word combinations first establish themselves through constant re-use, then undergo figurative extension and finally petrify or congeal" (Cowie, 90). Among opaque idioms we can find "pure lexical idioms" (*ahuntzaren gauerdiko eztula* 'trifle') as well as those we call "grammatical" (*harik eta* 'until', *hala eta guztiz ere* 'nevertheless',...).
- *Figurative idioms*. These word-combinations are idiomatic in the sense that variations are seldom found but for the speakers the primitive "literal" sense is not as distant as those in the opaque idioms (*hutsaren hurrengoa* 'the lowest of the low').
- *Restricted collocations*. In these combinations, also called *semi-idioms*, one word has a figurative sense not found outside that limited context. The other elements are used in their usual literal sense (*bideak urratu* 'to make a way').
- *Open collocations*. Each element of these word-combination is used in its ordinary literal sense (*hego haizea* 'south wind').

As we mentioned above, we have relied on UZEI's criterion and taken a wider perspective when deciding what to consider a MWLU. We deal with both types of idioms, pure and figurative, as well as with restricted collocations. Open collocations, though, have only been taken into account if they express a particular concept and so they deserve an entry in Basque modern dictionaries (*Euskal Herria* 'The Basque Country', *Amerikako Estatu Batuak* 'United States of America',...).

We also deal with **foreign words** such as *in situ, in fraganti, a priori,...* those MWLUs can be somehow considered idiomatic as well since the meaning of their constituents is totally opaque in Basque.

However, we do not consider **proverbs**, although among them there are also idiomatic (*txakur zaunkaria ez da horzkaria* 'his bark is worse than his bite') and non-idiomatic ones (*San Bizente hotza, neguaren bihotza* 'The cold St. Vincent 's day is the middle of the winter'. Locating **catch-phrases** (*Hauxe behar genuen!* 'that's all we needed') and **similes** (*berakatz-atala baino finagoa* '(as) busy as a clove of garlic') is not our purpose either.

3. Representation of MWLUs.

After delimiting linguistically what we consider MWLUs, we needed a formal description so that they could be automatically processed. For that, we made a conscientious analysis of the syntactic and morphological behaviour of those lexical units. From that analysis we deduced the relevant formal features that had to be considered. In order to describe MWLUs in Basque, we have functionally established the following features:

- Sure/ambiguous: We say a MWLU is *sure* when its components can only be analysed as a whole lexical unit and therefore no other interpretation is possible (*behin eta berriro* 'over and over again'). In that case, the analyses that don't belong to the MWLU interpretation are eliminated (see Fig. 1).

```
/<behin>/
      ("behin  eta  berriro"   ADB  ADO  HAT  1/3)
/<eta>/
      ("behin  eta  berriro"   ADB  ADO  HAT  2/3)
/<berriro>/
      ("behin  eta  berriro"   ADB  ADO  HAT  3/3)
/<begiratzen>/
      ("begira"  ADI SIN + ASP EZBU)
      ("begira"  ADI SIN + LOT MEN KONP @-JADNAG_MP @-JADLAG_MP)
/<nuen>/
      ("*edun"  ADL B1 NR_HU NK_NI + LOT MEN @+JADNAG_MP @+JADLAG_MP)
      ("*edun"  ADL B1 NR_HU NK_NI + LOT MEN ERLT @+JADNAG_IZLG> @+JADLAG_IZLG>)
      ("*edun"  ADL B1 NR_HU NK_NI)
      ("*edun"  ADT B1 NR_HU NK_NI + LOT MEN @+JADNAG_MP @+JADLAG_MP)
      ("*edun"  ADT B1 NR_HU NK_NI + LOT MEN ERLT @+JADNAG_IZLG> @+JADLAG_IZLG>)
      ("*edun"  ADT B1 NR_HU NK_NI)
```

Fig. 1 Morphological analysis of the sentence ***behin eta berriro** begiratzen nuen*... 'I was looking **over and over again**...'.

- contiguous/dispersed: We say a MWLU is dispersed when its components do not necessarily occur one after another. In that case, the processing gets more complicated since we have to seek the components in subsequent words (see Fig. 2). If a MWLU has more than two components, some of them may be contiguous and some others may not.

```
/<gure>/
      ("gu"   IOR PER NUMP + DEK GEN @IZLG> @<IZLG + DEK ABS MG @OBJ @SUBJ)
      ("gu"   IOR PER NUMP + DEK GEN @IZLG> @<IZLG)
/<sukalderaino>/
      ("sukalde"  IZE ARR + DEK NUMS MUGM + DEK ALA @ADLG + DEK ABU @ADLG)
/<heldu>/
      ("hel"  ADI SIN + ASP PART + DEK ABS MG @OBJ @SUBJ)
      ("hel"  ADI SIN + ASP PART)
/<nahi>/
      ("nahi  izan"   ADI  ADP  HAT  1/2)
      ("nahi"  ADI ADP)
      ("nahi"  ADI SIN)
      ("nahi"  IZE ARR + DEK ABS MG @OBJ @SUBJ)
      ("nahi"  IZE ARR)
/<baldin>/
      ("baldin"  PRT)
/<bada>/
      ("nahi  izan"   ADI  ADP  HAT  2/2)
      ("izan"  ADB ADO + ADL A1 NR_HU)
      ("izan"  ADB ADO + ADT A1 NR_HU)
      ("izan"  AUR BALD + ADL A1 NR_HU)
      ("izan"  AUR BALD + ADT A1 NR_HU)
      ("bada"  LOT LOK @LOK)
```

Fig. 2 Morphological analysis of the sentence *gure sukalderaino heldu **nahi** baldin bada*... 'If you **want** to reach our kitchen...'.

- <u>ordered/order-free</u>: Regardless of it being continuous or dispersed the elements of a MWLU may not necessarily keep an order[4] (see Fig. 3). A clear example of this are verb periphrasis such as *lo egin* 'to sleep' *min eman* 'to hurt' etc. since their constituents usually shift their positions, e.g. in imperative clauses (we say *<u>lo</u> <u>egin</u> dut* 'I have slept', but *<u>egizu</u> <u>lo</u>* 'sleep').

```
/<Ez>/
      ("ez"  ADB ADO)
      ("ez"  IZE ARR + DEK ABS MG @OBJ @SUBJ)
      ("ez"  IZE ARR)
/<zitzaion>/
      ("falta  izan"   ADI  ADP  HAT  2/2)
      ("izan"  ADL B1 NR_HU NI_HU + LOT MEN @+JADNAG_MP @+JADLAG_MP)
      ("izan"  ADL B1 NR_HU NI_HU + LOT MEN ERLT @+JADNAG_IZLG> @+JADLAG_IZLG>)
      ("izan"  ADL B1 NR_HU NI_HU)
/<korbata>/
      ("korbata"  IZE ARR + DEK ABS MG @OBJ @SUBJ)
      ("korbata"  IZE ARR + DEK ABS NUMS MUGM @OBJ @SUBJ)
      ("korbata"  IZE ARR)
/<eramatea>/
      ("eraman"  ADI SIN + LOT MEN KONP @-JADNAG_MP @-JADLAG_MP)
      ("eraman+te"  ADI SIN + ASP IZE + DEK ABS NUMS MUGM @OBJ @SUBJ)
/<besterik>/
      ('beste'  DET DZG + DEK PAR MG @OBJ @SUBJ)
/<falta>/
      ("falta  izan"   ADI  ADP  HAT  1/2)
      ("falta"  ADI ADP)
      ("falta"  IZE ARR + DEK ABS MG @OBJ @SUBJ)
      ("falta"  IZE ARR + DEK ABS NUMS MUGM @OBJ @SUBJ)
      ("falta"  IZE ARR)
```

Fig. 3 Morphological analysis of the sentence *Ez **zitzazion** korbata eramatea besterik **falta**...*, 'All he **needed** was to wear a tie...'.

- <u>Inflected/invariable</u>: The components of a MWLU may either appear in an invariable form (*hurrenez hurren* 'respectively') or inflect. For instance, both constituents of the verb periphrasis *bizi izan* 'to live' can inflect and therefore the possible combinations are countless (*bizi naiz* 'I live', *biziko banintz* 'if I lived', *bizi izanik* 'living'...). In the case that the components of the MWLU can inflect, they can accept either any inflection or just a restricted set of inflected forms. Thus, restrictions are needed for components accepting just a few inflected forms. Moreover, lack of restrictions, especially in "dispersed + order-free" lexical units, may increase the ambiguity rate considerably. The more restrictions we make the less ambiguity we get.

Summing up, when defining a MWLU we must give the following information:

- We must declare whether each component is invariable or can inflect and specify the restrictions of inflection when necessary. We use a logical formalism to represent these restrictions.

Examples:	(1)	adarra	adar	((ABS NUMS MUGM) or (PAR MG)) /
		jo	jo	
		+SEG		
		ADI ADK		
	(2)	hala	-	
		ere	-	
		+SEG		
		LOT LOK		

[4] However, the "order-free + contiguous" combination is not necessary. The analysis of the syntactic and morphological behaviour of MWLUs —previous to their formal description— led us to the conclusion that in Basque whenever a MWLU is order-free, it is also dispersed.

(3)	lan	-	
	egin	egin	
	-SEG		
	ADI ADK		
(4)	beste	beste	*
	bat	bat	
	-SEG		
	DET DZG		

In the example (1) above —*adarra jo* 'to pull s.o.'s leg'— the lemma *adar* can only take the singular absolutive ("ABS NUMS MUGM") or the partitive ("PAR MG"). The lemma *jo*, on the contrary, accepts any possible inflection (*jo, jotzen, joko, jotzeko, jotzera,...*). In the example (2) —*hala ere* 'nevertheless'— both components are invariable and in (3) —*lan egin* 'to work'— the first component cannot vary but the second one can inflect freely.

- When two components of the MWLU may appear dispersed but are not order-free we mark with the symbol * between them (see example (4)).
- When two components can be both dispersed and order-free we mark with the symbol / between them (see example (1)).
- When a MWLU is sure we mark it +SEG and when it is ambiguous we mark it -SEG (see examples above).
- Finally the category of each MWLU is specified (see examples above).

4. Conclusions

In this paper we have tried to show that it is not sufficient to lemmatise just the word-forms between two blanks. There are several Multiword Lexical Units such as idioms, restricted collocations, foreign words etc. that must be lemmatised as a single unit. Nevertheless, processing those complex lexical units involves some additional difficulties. For instance, the constituents of the MWLUs do not always occur one after another or in the same order and, although some components have an invariable form, some others can inflect. Thus, it is essential to develop a formal representation so as to be able to describe in detail the MWLU.

Currently we are testing a prototype for the treatment of MWLUs. A few significative examples of its output are also shown in this paper (see Fig. 1, 2 and 3).

5. Bibliography

Aduriz, I.; Aldezabal, I.; Alegria, I; Artola, X; Ezeiza, N.; Urizar, R. (1996a): "EUSLEM: A lemmatiser/tagger for Basque", Computational Lexicology and Lexicography. *Euralex'96*. Göteborg.

Aduriz, I.; Arriola, J.M.; Artola, X.; Díaz De Ilarraza, A.; Gojenola, K.; Maritxalar, M.; Urkia M. (1996b): *Euskararako murriztapen-gramatika: lehen urratsa.* Department of Languages and Computing Systems, Computer Science Faculty, University of The Basque Country.

Agirre, E.; Arregi, X.; Arriola, J.M.; Artola, X.; Díaz De Ilarraza, A.; Insausti, J.M.; Sarasola, K. (1995): "Different issues in the design of a general-purpose Lexical Database for Basque". *First Workshop on application of Natural Language to Data Bases, NLDB'95.*

Alegria, I. (1995): *Euskal morfologiaren tratamendu automatikorako tresnak.* Ph. D. thesis. University of The Basque Country.

Amstrong, S, Russel, G., Petitpierre, D., Robert, G. (1995): "An open Architecture for Multilingual Text Processing", in: Proceedings of the 5th Conference of the EACL, Volume 1, 101-106.

Black, A., van de Plassche, J., Williams, B. (1991): "Analysis of Unknown words through Morphological Decomposition", in: *Proceedings of the 5th Conference of the EACL,* Volume 1, 101-106.

Clausen, U.; Lyly, E. (1994): "Criteria for identifying and Representing Idioms in a Phraseological Dictionary", The way words work / combinatorics, *Euralex'94*. Amsterdam. 258-262.

Cowie, A.P.; Mackin R.; McCaig I.R. (1990): *Oxford Dictionary of Current Idiomatic English.* v2.

Euskaltzaindia (1992): *Hitz elkartuen osaera eta idazkera.* Hitz-elkarketa / 4. LEF batzordea.

Fontenelle, T.; Adriaens, G.; De Braekeleer, G. (1994): "The Lexical Unit in the Metal® MT System", *MT*.The Netherlands. v9. 1-19.

Heid, U. (1994): "On Ways Words Work Together - Topics In Lexical Combinatorics", The way words work together / combinatorics, *Euralex'94*. Amsterdam. 226-257.

Izagirre, K. (1981): *Euskal lokuzioak. Espainolezko eta frantsesezko gida-zerrendarekin.*

Karlsson, F.; Voutilainen, A.; Heikkila, J.; Anttila, A. (1995): *Constraint Grammar: Language-independent System for Parsing Unrestricted Text.* Mouton de Gruyter.

Koskenniemi, K. (1983): *Two-level Morphology: A general Computational Model for Word-Form Recognition and Production.* Ph D. thesis, University of Helsinki.

Longman (1989): *Dictionary of English Idioms.*

Segond, F.; Tapainen, P. (1995): "Using a Finite-State Formalism to Identify and Generate Multiword Expressions", *MLTT*. Grenoble.

Urkia, M. (Forthcoming): *Euskal morfologiaren analisi automatikorantz.* Ph. D. thesis. University of The Basque Country.

Urkia, M.; Sagarna, A. (1991): "Terminología y lexicografía asistida por ordenador. La experiencia de UZEI". *Actas del VII Congreso de SEPLN.* v9. 193-202.

Study and propositions of specific categories through the lemmatization of an orthographic-phonetic data base of spoken French: BDPho

R. BELRHALI – D. DUJARDIN – L-J. BOË –
J. COURTIN

Abstract

The aim of this paper is to study specific categories through the lemmatization of a spoken French database BDPho. To do so, each form is morphologically analyzed and for each homograph a lemma is delivered. The study of all the forms shows that 33% of them present homographs. We have thus obtained eight families of homographs. They have been studied with a morphological approach. Finally, several categories have emerged as specific spoken language categories.

1. INTRODUCTION

The BDPHO database [Boë, 1992a] elaborated by the *École Nationale Supérieure des Télécommunications* and the *Institut de la Communication Parlée de Grenoble* is based on a corpus of recorded speech (about 10 hours), transcribed by expert phoneticians. There were about 30 speakers, over 300,000 sounds and 102,000 words. BDPHO was constituted by restitution of 7,597 orthographic forms (corresponding to 7,221 phonetic forms, including 1,386 variants).
From this database, the *De À à Zut* dictionary [Boë, 1992b] was produced. This dictionary delivers the number of occurrences of each form, the different pronunciations and the structure of the phonetic words presented as a cohort string (CV, CVCV...). In this study, the automatic lemmatization of this phonetic French spoken lexicon was processed by the PILAF system [Courtin, 1992] and analyzed in our laboratories.

2. DATA

2.1. Corpus

2.1.1. Origin and description

The BDPHO Phonetic Database is constituted of three parts, including 304,752 sounds.
• Corpus n° 1 (86,360 sounds), made up of recordings of radio programmes, was constituted at the *Institut de Phonétique de Grenoble* under the supervision of R. GSELL.
• Corpus n° 2 (201,281 sounds), the most important one, was put together by A. MALÉCOT at the *University of California, Santa Barbara*. It contains 50 hours and a half of free conversations on various subjects with members of the Paris "intelligentsia" (professors, lawyers, doctors, artists...).
• Corpus n° 3 (17,111 sounds) was collected by J. VAN EIBERGEN [1985] at the *Institut de la Communication Parlée, Grenoble*. It includes the transcriptions of 16 short conversations by 16 speakers of various linguistic and socio-professional origins (four teenagers, eight adults aged from 20 to 60 and four from 60 to 80). These are simple spontaneous conversations in informal situations.

2.1.2. Contents

The corpus contains 102,137 lexical occurrences, and 7,221 different phonetic forms. It has been statistically processed in terms of occurrences of sounds and combinations of them. It is well known that all sequences of sounds are inexistent in language, due to phonotactic constrains. Bisounds (the sequence of two sounds) are very numerous (87% of the theoretical possibilities), whereas trisounds and quadrisounds are few (respectively 30% et 3.8%). Some words and sequences of words are very frequent (*alors, parce que, il y a, de la, avec, et par, crois, voir...*).
Liaisons produced by [d n p ʀ t z] sounds represent about 6% of the occurrences of words in the corpus and are very unequally distributed, three of them [n t z] cover over 90%.
The comparison of the first 50 occurrences in the corpus with those published by G. ENGWALL [1984] and the *Listes Orthographiques de Base* (LOBO) by N. CATACH [1984] shows strong similarities (over 30 common shapes or entries), and the emergence of some words which are specific to spoken language (*ça, y, alors, très, oui, enfin, parce que, moi, quand, puis, euh*). Forms like *pour, est, c', pas, on, ça, ce, y, bien, alors, très, oui, enfin, parce que, fait, si, même, là, euh* are more frequent in spoken language whereas the negative form *ne* is more frequent in written language. Regarding cohorts, 25% of them cover 90% of the possibilities; and 44 cohorts represent 95% of the total.
In order to handle the database [Belrhali, 1995] we decided to use the HyperCard environment:

- HyperCard is a flexible programmation environment and is accessible to non-computer experts; data sets can be easily modified;
- It brings very few commercial constraints as regards to distribution (it requires no licence as there are HyperCard stand alone version);
- It offers the API symbols, of good typographical quality, for input and output (vectorized symbols)

– HyperCard is an evoluted, standardized and user friendly hardware.
– The HyperCard environment together with the HyperTalk language facilitates the call of external softwares (in Pascal or C);

The corpuses appear in three HyperCard stacks. Each stack is made out of a group of cards with 150 numbered orthographic lines, and, for each line, its phonetic transcription.

790G C'est ce qui se passe / souvent // Alors il est parti / au cap de Bonne Espérance /
sɛ s ki s ‘pɑːs / su‘vɑ̃ // alɔʀ il ɛ paʀ‘ti / o kap də bɔn ɛspe‘ʀɑ̃ːs /

791G et il a élevé / des autruches / puis il a perdu / le reste
e il a ɛl‘ve / de-z o‘tʀyʃ / pɥi-z il a pɛʀ‘dy / lə ʀɛstə

792G de sa fortune / il a perdu / le reste de sa fortune / alors il
də sa fɔʀ‘tyn / il a pɛʀ‘dy / lə ʀɛstə də sa fɔʀ‘tyn / alɔʀ il

Figure 1 : Extract of a card of the Gsell's corpus.

3. The PILAF system

The PILAF system (*Procédures Interactives Appliquées au Français*) is a user-friendly system for parsing French. It was developped by the TRILAN team (*Traitement Informatique de la Langue Naturelle*) [Courtin, 1992]. It is a part of a linguistic toolbox implemented on microcomputers. It is parametrable and portable. It also has the advantage of being easily integrated to different systems.
For the lexical level, PILAF proposes modules for morphological parsing, generation of flexional forms from a root and a lemmatizer. It relies on a database composed of two dictionaries and linguistic data including a validation-saturation grammar defined by a set of rules, as well as lists of models, of lexical categories and variables. All these data are manipulated by means of specialized editors.
Morphological parsing is done by using a reversible finite state transducer that:
– segmentates a character string in order to obtain its components;
– associates to each of these strings a set of linguistic informations.
To do so, the dictionnary determines a contiguous part of the entry string. It is processed by a grammar which decides whether the concatenation of these different elements is valid or not. The grammar is a "Validation-Saturation grammar" (GVS) which is simply a more concise formulation of a finite state grammar.
For any substring recognized as an enter character-string, the morphological parsing computes, a base, lexical category, and grammatical variables.

Example : "The cats mew" will give :

the	determiner	
cats	common noun	animate, plural
mew	verb	present, not third person singular
mew	infinitive	

The use of generation would have given all the flexional forms which derive from any base or any word, indicating for each its lexical category and the corresponding grammatical variables . Naturally, this list could have been "filtered" to select certain categories or variables. For instance, if the entry-word is `looks`, filtered by the categories `verb` and `past-participle`, generation will deliver : `look, looks and looked.`
The PILAF dictionary completed with the aid of the LADL electronic lexicon [Gross, 1993], with 35,000 bases allows recognition and generation of more than 250,000 flexional forms.

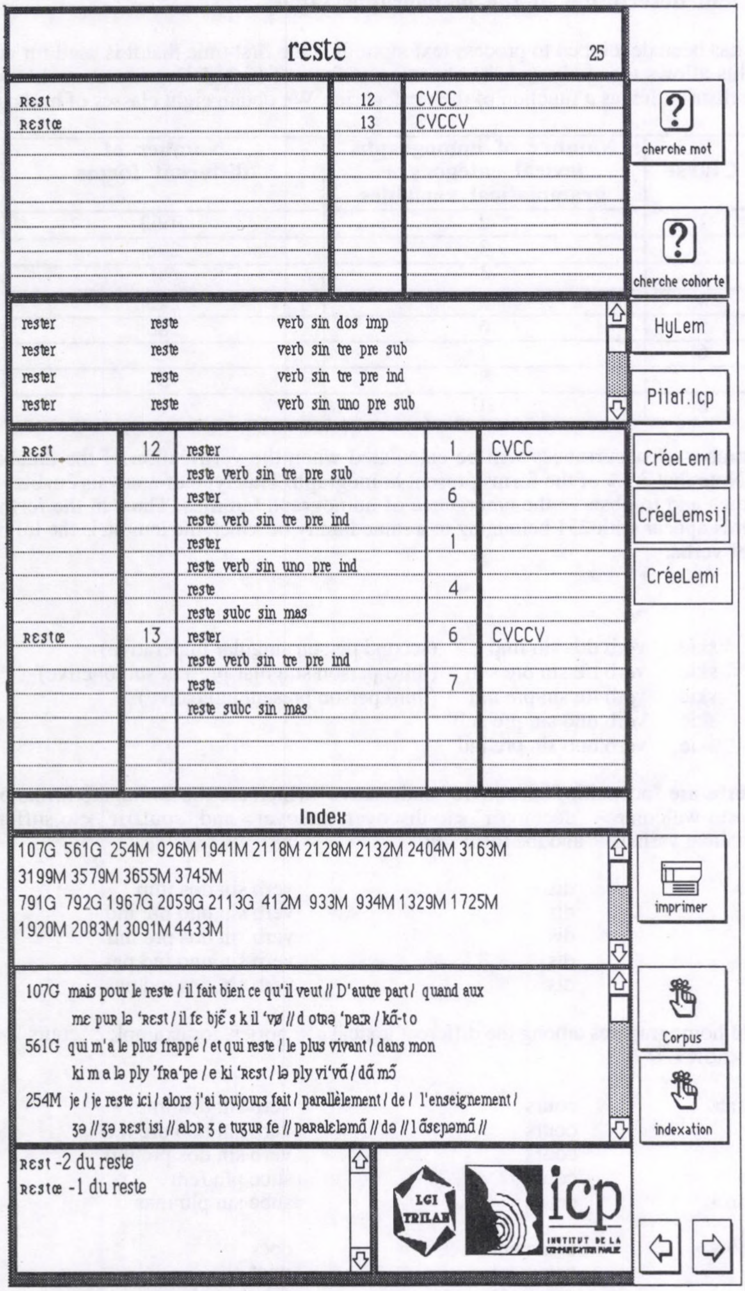
4. The lemmatization

The lemmatization consisted in linking a word or a group of words to a representant of this word. The lemma is defined according to the grammatical characteristic of the studied word.
For all the conjugated forms "chante, chantes…" <sing> of verb "chanter" <to sing> the lemma is the infinitive form "chanter" <to sing>. For the forms "chant, chants" <a song, songs> from the substantive, the lemma would be the form "chant" <song> from the singular substantive. The lemma can also be a representant of a group of words semantically linked : "diplomatie" <diplomamacy> can be the lemma of the words "diplomatie, diplomate, diplomatique, diplomatiquement" <diplomacy, diplomat, diplomatic, diplomatically>, whereas "diplomate" <diplomat> can be the lemma of "diplomate" <trifle> (cake with candied fruits and cream).

4.1. The lemmatization of BDPho

The aim of the first step was to link each BDPHO orthographic entry to a lemma, and each lemmà to one lexical category. In this way, for the word "reste" (to stay, a rest…), two lemmas are possible
– the lemma "rester" (to stay) for the conjugate forms of the verb,
– the lemma "reste" (a remaining) corresponds to a singular masculine substantive.
To obtain this lemmatization, each orthographic entry is morphologically analyzed and for each homograph, a lemma is delivered. We had to be flexible for simple lexical units as well as complex ones (compound nouns, locutions). This modification has led to change the frequencies of the original constituents [Dujardin, 1995]. This lemmatization has been performed with a new version of BDPho database which allows processing of the old version and a direct interaction with the corpus and PILAF.
A reference system and automatic indexation allow the access to both orthographic and phonetic contexts for each orthographic form.

The last version presents for each orthographic generic term: each phonetic variant with the total frequency, cohort representation, different lemma pairs, occurrences with their frequency (corresponding to different homographs) which are associated not only to the lexical category but also to variables.
In the stack BDPho, each card corresponds to an orthographic entry of the lexicon in which are located the informations about the card (cf. Figure 2):
« 25 » : total number of occurences
CVCC, CVCCV : cohorts
[ʀɛst], [ʀɛstœ] : phonetical variants
107G --> 3745M ; 794G --> 4433M : references of the lines of the corpus.
BDPho consists of:
825 orthographic entries with a frequence >10,
2951 orthographic entries with a frequence between 1 and 10,
3821 orthographic entries with a frequence = 1.
The lemmatization can be realised automatically on a part, the whole of the data.



reste 25

ʀɛst	12	CVCC
ʀɛstœ	13	CVCCV

rester	reste	verb sin dos imp
rester	reste	verb sin tre pre sub
rester	reste	verb sin tre pre ind
rester	reste	verb sin uno pre sub

ʀɛst	12	rester	1	CVCC
		reste verb sin tre pre sub		
		rester	6	
		reste verb sin tre pre ind		
		rester	1	
		reste verb sin uno pre ind		
		reste	4	
		reste subc sin mas		
ʀɛstœ	13	rester	6	CVCCV
		reste verb sin tre pre ind		
		reste	7	
		reste subc sin mas		

Index

107G 561G 254M 926M 1941M 2118M 2128M 2132M 2404M 3163M
3199M 3579M 3655M 3745M
791G 792G 1967G 2059G 2113G 412M 933M 934M 1329M 1725M
1920M 2083M 3091M 4433M

107G mais pour le reste / il fait bien ce qu'il veut // D'autre part / quand aux
mɛ puʀ lə 'ʀɛst / il fɛ bjɛ̃ s k il 'vø // d otʀə 'paʀ / kɑ̃-to
561G qui m'a le plus frappé / et qui reste / le plus vivant / dans mon
ki m a lə ply 'fʀa'pe / e ki 'ʀɛst / lə ply vi'vɑ̃ / dɑ̃ mɔ̃
254M je / je reste ici / alors j'ai toujours fait / parallèlement / de / l'enseignement /
ʒə // ʒə ʀɛst isi // alɔʀ ʒ e tuʒuʀ fe // paʀalɛləmɑ̃ // də // l ɑ̃sɛɲəmɑ̃ //

ʀɛst -2 du reste
ʀɛstœ -1 du reste

Figure 2 : Example of the card of the stack BDPho

5. Listing and description of the homographs family

PILAF has been developed to process text input. It is the first time that it is used for an organized database. This allows the study and the checking of forms which deliver the same homographs, as well as their distribution as a function of their categories. We obtain eight classes of homographs:

Classe	Number of homographs lexical category + grammatical variables	Number of different forms
1	2	1663
2	3	298
3	4	70
4	5	243
5	6	288
6	7	75
7	8	5
8	10	1

The total number of occurrences will be calculated when the verification of the database will be completed. Note that 33% of the forms present homograph. This type of analysis enables to perform statistic studies and to observe the emergence of homograph families. Thus, in the forth class, 243 different words appear with 231 belonging to a same family of which the lemma is the infinitive of the first group of verbs.

Example :

skier	skie	verb dos sin imp	(second person singular imperative)
skier	skie	verb tre sin pre sub	(third person singular present subjonctive)
skier	skie	verb tre sin pre ind	(third person present indicative)
skier	skie	verb uno sin pre sub	
skier	skie	verb uno sin pre ind	

The other verbs are "accueille" "découvre" and "sauve" which are the conjugate forms of the verbs "accueillir" <to welcome>, "découvrir" <to discover, uncover> and "souffrir" <to suffer> with the same grammatical variables, and the form "dis"

dire <to say>	dis	verb sin dos imp
dire	dis	verb sin uno pre ind
dire	dis	verb sin dos pre ind
dire	dis	verb sin uno ind pas
dire	dis	verb sin dos ind pas

The last eight homographies among the different lexical categories, for example : "cours " <run> and "comment" <how>.

courir <to run>	cours	verb sin dos imp
courir	cours	verb sin uno pre ind
courir	cours	verb sin dos pre ind
cour <court>	cours	subc plu fem
cours <lesson>	cours	subc sin plu mas

comment <how>	comment	cocs
comment	comment	mint
comment	comment	subc sin plu mas
comment	comment	intj
comment	comment	adv

In the fifth class, on 288 different forms, we have :
62 forms in the family of Verb + Masculine Substantive, and 193 Verb + Feminine Substantive.

rester <to stay>	reste	verb sin dos imp
rester	reste	verb sin tre pre sub
rester	reste	verb sin tre pre ind
rester	reste	verb sin uno pre sub
rester .	reste	verb sin uno pre ind
reste <rest>	reste	subc sin mas

The eigth class, has only one constituent: the form "ouvre" corresponding at two lemmas "ouvrir" <open> and "ouvrer" <to work up>.

It is therefore possible to list the components of each family and to study their realizations proposed in the following extract :

32 Compte	8	compter	8	kɔ̃t	6	Verb tre sin pre ind
<number>					2	Verb uno sin pre ind
	24	compte	23	kɔ̃t	23	Subc mas sin
			1	kɔ̃m	1	Subc mas sin
25 Reste	14	rester	8	ʀɛst	1	Verb tre sin pre sub
<to stay>					6	Verb tre sin pre ind
					1	Verb uno sin pre ind
			6	ʀɛstœ	6	Verb tre sin pre ind
	11	reste	4	ʀɛst	4	Subc mas sin
			7	ʀɛstœ	7	Subc mas sin
13 Voyage	1	voyager	1	vwajaʒ	1	Verb uno sin pre ind
<journey>	12	voyage	12	vwajaʒ	12	Subc mas sin
12 Téléphone	3	téléphoner	3	telefɔn	2	Verb uno sin pre ind
<phone>					1	Verb tre sin pre ind
	9	téléphone	9	telefɔn	9	Subc mas sin
4 Change	4	changer	4	ʃɑ̃ʒ	3	Verb tre sin pre ind
<change>					1	Verb uno sin pre ind
4 Charme	4	charme	3	ʃaʀm	3	Subc mas sin
<charm>			1	ʃaʀmœ	1	Subc mas sin
1 Peuple	1	peuple	1	pœpl	1	Subc mas sin
<people>						

Figure 3 : Extract of the components of the family "reste"

It is also possible to list the components of each family of homograph-orthographic forms for a given category and variables. Then, to associate them to classes and to study them.

avez <have>	176
voulez <want>	134
savez <know>	90
voyez <see>	57
êtes <are>	49
écoutez <listen>	27
allez <go>	24
faites <do>	22
comprenez <understand>	20
dites <say>	20
remarquez <notice>	18
connaissez <know>	16
pouvez <can>	16
pensez <think>	13
mettez <put>	12
prenez <take>	11
parlez <talk>	7
citez <quote>	6

Figure 4: Example of a list containing the occurences of homographs of the verb in the second person plural

6. Study of specific categories of spoken language

The analysis yielded oral speech characteristic words: *alors, ça, y, il y a, très, oui, enfin, parce que, moi, quand, puis, euh.* Morphological classes have thus been adapted to spoken language.
Oral markers have been widely studied [Martinon, 1927, François, 1974, Loufrani, 1990]. However, the creation of the following classes has not been sufficient:

presentatives *c'est, c'était, il y a....*
pauses *euh, bon,, ben...*
oral punctuation marks *quoi, en fait, hein*
speech support *alors, ben, quoi...*

It seems that in informal conversations, the class of pragmatic connectors *ça va, ça va pas, ça y est, c'est ça, c'est bon, c'est fini...* [Fréchet, 1992] is not adequate. Must we consider that a more elaborate classification [Morel, 1989], by taking into account markers such as:structuration of conversation *"alors", "bon", "disons"*, guest of discursive approbation*"voilà", "vous savez que ", "d'accord"*, selfcorrection*"disons", "enfin"*, reformulation, *"c'est-à-dire (que)", "disons (que)"*, logical connexion, *"alors", "mais aussi ", "mais alors"*, association of themes, *"également", "à propos ", "aussi"*? These markers have beeen studied for finalized dialogues (request of information).

In BDPho, the presence of all the contexts facilitates the study of markers. The following example illustrates our approach:

ALORS

Alors	**aloʀ**	**3**
Alors	**alɔʀ**	**520**
Alors	**alɔʀ-z**	**1**

Orthographic form	*phonetic variants*	*occurences number*
Et alors	**/e alɔʀ/**	**21**

80G territoire / et alors / je voudrais dire / de façon très ferme / à ceux qui

tɛʀi'twaːʀ / e a'lɔːʀ / ʒə vudʀɛ 'diːʀ / də fasɔ̃ 'tʀɛ 'fɛʀm / a sø ki

4373M les autres / et / ils ont mis des đnormes / pneus / ɱichelin / et alors / ils flottaient

le-z otʀ // e // il-z ɔ̃ mi de-z enɔʀmə // pnø // miʃəlɛ̃ // e alɔʀ // il flɔte

Et alors **e alɔʀ** **23**

793G est rentré / à X / et alors il faisait / du journalisme / et puis il

ɛ ʀɑ̃'tʀe / a x / e alɔʀ il fə'zɛ / dy ʒuʀna'lism / e pɥi il

Et alors euh **e alɔʀ ə** **9**

235M questions sociales / et alors euh / le Conseil d'Etat / il y a un travail au Conseil

kɛstjɔ̃ sɔsjal // e alɔʀ ə // lə kɔ̃sɛj d eta // il i a œ̃ tʀavaj o kɔ̃sɛj

Alors euh **alɔʀ ə** **16**

177M clair alors euh / qui est imprimé par l'ordinateur avec le montant de la

klɛʀ alɔʀ ə // ki e-t ɛ̃pʀime paʀ l ɔʀdinatœʀ avɛk lə mɔ̃tɑ̃ də la

Alors **/ alɔʀ /** **64**

98G un temps / où la psychanalyse / n'existait pas / alors / ce qui m'intéresse /
œ̃ 'tɑ̃ / u la psikana'liːz / n ɛgzistɛ 'pɑ / a'lɔːʀ / 'sə ki m ɛ̃te'ʀɛs

To assign a grammatical category to each marker, we have recall the categories proposed in *Le Trésor de la Langue Française.*

Regarding verbal lemmas, several categories may be proposed as a function of the specificities they denote in spoken language. The following markers have thus been selected:
reformulation : "Disons"
of confirmation : "Je veux dire que"
suspension : "attends", "attendez"
underline: "vous savez que", "disons que"
call the listener: "si vous voulez", "vous voyez", "écoutez", "remarquez"
expression of the speaker's conviction :"j'avoue que", "je sais pas", "je crois que"
expression of politeness: "si vous permettez", "excusez-moi"

7. Conclusion

The flexibility of the system led to a redefinition of the classification of some connectors, better knowledge of qualitative and quantitative phonetic variants. The presence of contexts allowed the reconsideration of sequences of connectors, and to further study speech markers, and refine the analysis of speech behaviour.

The modifications due to the selection of markers led to the redefinition of new classes in PILAF and the creation of new entries in BDPHo. We will study the consequences of these markers in future reşearch.

The changes we have made will allow us to study the grouping of phonetic variants. For example, the phonetic variant [akɔʀ] accord < 31 > which is completely due to the locution *d'accord* occurs in 15 adverbial locutions and 15 speech supports. The occurrence *ça* < 1069 >, considered as characteristic of spoken language, has only one phonetic variant [sa]. But it is a component which belongs to most of the newly created classes such as *ça serait,* presentative, *ça y est,* pragmatic connector,, *tout ça* pause, *comme ça* support of speech. Among the 90 occurrences *comme ça* three cannot be çonsidered as entities *comme ça vient.*

The estimation of new frequencies will yield new results. Indeed in the case of letter "C" the total number of occrrences in "A à Zut" was 1927. After considering locutions as "c'est pourquoi" and "c'est-à-dire", the number of occurrences decreases to 1797, and to 180 after taking into account "c'était, c'est".

ACKNOWLEDGEMENTS

The authors are very indebted to Sonia Kandel for the translation of this paper.

REFERENCES

Belrhali R., Dujardin D., Courtin J. & Boë L.-J. (1995), *BDPHO : Une base de données lexicales orthographique-phonétique lemmatisée du français parlé.* JADT, III^e Journées Internationales d'Analyse Statistique des Données Textuelles, 11-13 Décembre, Rome, Italie.

Boë L.-J., Tubach J.-P. (1992a), *« De À à Zut » Dictionnaire phonétique du français parlé,* Ellug, Grenoble.

Boë L.-J., Tubach J.-P. (1992b), *BDPHO: une base de données lexicale orthographique-phonétique du français parlé,* Séminaire Lexique du GRECO-PRC Communication Homme-Machine, Toulouse, pp. 111-119.

Catach N. (1984), *Les listes orthographiques de base (LOB),* Nathan, Paris.

Courtin J., Dujardin D., Kowarski I. (1992), *PILAF: Software Tools for Lexicography and Text Research.* COMPLEX'92: 2nd Conference on Computational Lexicography and Text Research, Budapest, Hungary, pp. 93-109.

Dujardin D., Belrhali R., Boë L.-J., & Courtin J. (1995), *Morpho-phonetic relationship and elaboration of De A à Zut a lexicon of spoken french.* August 13-19, 1995, Stockholm, Sweden.

Engwall G. (1984), *Vocabulaire du roman français (1962-1968). Dictionnaire des fréquences,* Almqvist & Wiksel International, Stockholm, Suède.

François D. (1974), *Français parlé. Analyse des unités phoniques et significatives d'un corpus recueilli dans la région parisienne.* S.E.L.A.F. Tome I, II et III.

Fréchet A.L., Morel M.A., Dujardin D., & Caelen J. (1992), *Analyse lexicale d'un langage opératif.* Bulletin de la Communication Parlée N° 2, pp 167-182

Giovannoni D.-C., Savelli M.-J. (1990), *Transcrire, traduire, orthographier le français parlé. De l'impossible copie à la fasification des données orales.* Recherches sur le français, n°10.

Gross M., Silbertzein M (1993), *Outils de traitement linguistique, applications à l'analyse documentaire.* Actes de la troisième école d'été, Lannion.

Léon J., "Formes de discours direct dans les récits oraux" LINX n° 18.

Loufrani C., Roubaud M.N. (1990), *La notion d'approximation : Langage ordinaire, langage pathologique* . Recherches sur le français parlé, n° 10

Martinon Ph. (1927), *Comment on parle en français.* Librairie Larousse, 1927

Morel M.-A. (1989), *Analyse linguistique d'un corpus,* Deuxième corpus : Centre d'Information et d'Orientation de l'Université de Paris V. Paris : Publications de la Sorbonne Nouvelle, 331 p.

Trésor de la Langue Française, Nancy, CNRS.

Van Eibergen J. (1985), *Corpus d'un français vernaculaire à caractère spontané et impératif,* Bulletin de l'Institut de Phonétique de Grenoble, vol.15, pp. 35-74.

Local Grammars for the Description of Multi-Word Lexemes and their Automatic Recognition in Texts

ELISABETH BREIDT – FRÉDÉRIQUE SEGOND –
GIUSEPPE VALETTO

Abstract

Most multi–word lexemes (MWLs) allow certain types of variation. This has to be taken into account for their description to be able to recognize them in texts. We suggest to describe their syntactic restrictions and their idiosyncratic peculiarities with local grammar rules, which at the same time permit to express regularities valid for a whole class of MWLs such as word order variation in German. The local grammars can be written in a very convenient and compact way as regular expressions in the formalism IDAREX which uses a two-level morphology. IDAREX allows to define various types of variables, and to mix canonical and inflected word forms in the regular expressions. The finite–state based dictionary look–up system LOCOLEX/COMPASS uses such local grammars to recognize MWLs in English, German and French on–line texts.

1 Introduction

Most texts are rich in multi-word expressions that cannot be properly understood, let alone be processed in an NLP system if they are not recognized as complex lexical units. We call such expressions *multi-word lexemes (MWL)*. These range from idioms (*to rack one's brains over sth*), over phrasal verbs (*to come up with*) and separated particle verbs (in German or Dutch), lexical and grammatical collocations (*to make love, with regard to*, resp.) to compounds (*on–line dictionary*).

Certain MWLs always occur in exactly the same form, like *out of the blue* or *um Haaresbreite* ('almost', lit. for hair's breadth); these can be easily recognised with simple pattern matching techniques. However, it is well known (see for instance, Gross (1982), Brundage et al. (1992), Nunberg et al. (1994)) that most MWLs cannot be treated in the same way as completely fixed patterns, since they may undergo some variation. Only a *subset* of the variations allowed by general syntax rules is valid though: outside that subset, the expression loses its special, idiomatic meaning, either reverting to its literal meaning or losing any significance altogether. Thus it is important that a representation of MWLs licenses only the allowed variations and excludes variations that would lead to the literal reading of the word sequence.

In certain cases, MWLs can even contradict general syntactic rules, as with *by and large* or G:*von Haus aus* ('originally', lit. from house out), where general rules would require an article between the preposition and the count noun. These latter MWLs have to be recognized before parsing because otherwise the parser would fail.

The identification of MWLs is essential for any natural language processing based on lexical information, ranging from intelligent dictionary look–up over concordancing or indexing to machine translation. Therefore, the restricted lexical and syntactic variability of MWLs and their idiosyncratic peculiarities need to be expressed in the computational lexicon in order to be able to recognize the full range of their occurrences. We propose to use local grammars for this, written as a special type of regular expressions (REs) in the finite–state formalism IDAREX which makes use of a two–level morphological lexicon.

2 Variability of MWLs

Basically, we identify four types of variability (see also Fleischer (1982), Brundage et al. (1992), Van der Linden (1993) and Engelke (1994)) that a description of MWLs, both for NLP and for human use, should cover:

- Morphological variation:
  particular words in the MWL undergo certain inflections.
- Lexical variation:
  one or more words can be substituted by other terms without changing the overall meaning of the MWL. These may be near–synonyms but need not be so.
- Modification:
  one of the MWL's constituents can be modified, preserving the idiomatic meaning.
- Structural flexibility:
  this includes phenomena like passivization, topicalization, scrambling, raising constructions etc.

In the following paragraphs we give English, French and German examples to illustrate the various types of variability. Though we do not always give contrastive examples of MWLs for which the same type of variation is *not* possible, the reader should keep in mind that the illustrated variations only apply on a case by case basis to certain MWLs within one language.

Morphological Variation For instance, in *to kick the bucket*, *kick* can be inflected to any tense, number, person, but the noun may only be used in the singular form to preserve the idiomatic meaning. In F:*faire le fou* ('to play the fool', lit. make the fool) the determiner and the noun can appear in their feminine or plural form, i.e. *faire la folle/les folles.* In G:*durchschlagender Erfolg* ('sweeping success', lit. making its mark success), noun and adjective can vary in case and in number, and comparative and superlative form are possible for the adjective, whereas G:*grüne Welle* ('phased traffic lights', lit. green wave) may only vary in case, but neither in number nor in adjective comparison without loosing its idiomatic meaning.

Lexical Variation For instance, in *to sweep sth under the carpet*, the nouns *rug* or *mat* can be substituted for *carpet*, but not *shag rug* or *stair–carpet.* In F:*perdre la tête* ('to lose one's mind', lit. lose the head), the noun can be substituted by *la boule* (lit. ball, coll. head) or *les pédales* (lit. pedals) without loosing its idiomatic meaning, but not by *la tronche* (lit. mug, coll. head). In G:*sich aufs Ohr legen* ('to hit the hay', lit. onself onto the ear lay), the exchangeable components have no meaning similarities at all: *hauen* (lit. to hit) can be used instead of *legen* (lit. to put/lay down), and *aufs Ohr* (lit. onto the ear) can be replaced by *in die Falle* (lit. into the trap).

Modification For instance, in F:*avoir un coup dans l'aile* ('to be the worse for drink', lit. have a hit in the wing) an adjective can be inserted, and the idiomatic nature of the expression is preserved, as in *elle a un sacré coup dans l'aile* (lit. she has a hell of a hit in the wing). Other adjectives, e.g. *long, rouge* (lit. long, red) cannot be inserted. In G:*den (schönen) Schein wahren* ('to keep up appearances', lit. the (nice) pretence preserve) the presence or absence of the adjective does not change the meaning at all, whereas any modification of G:*das Handtuch werfen* ('to throw in the towel', lit. the towel throw) would lead to a loss of the idiomatic meaning, and the expression would revert to a literal phrase.

Structural Flexibility For instance, *to keep tabs on* can be passivized without loosing its idiomatic meaning. While most verbal MWLs in French do not allow topicalization, F:*chercher midi à quatorze heures* ('to nit pick', lit. search midday at fourteen o'clock) can be topicalized: *Midi, il ne le cherchait pas à quatorze heures.* Similarly in German, topicalization of lexically fixed components is only rarely possible whereas standard word order variation applies to all verbal MWLs. G:*Jan hat dabei den Vogel abgeschossen* ('Jan surpassed everyone in this', lit. Jan has with it the bird shot) can be topicalized as in G:*den Vogel dabei hat dann Jan abgeschossen* ('finally, Jan surpassed everyone in this', lit. the bird with it has then Jan shot).

As said above, a formal description of MWLs should license all the allowed variations but exclude all non–allowed variations. In the next section, we show how to achieve this by writing local grammar rules for MWLs in the form of IDAREX REs.

3 Encoding MWLs with IDAREX

The IDAREX formalism (**ID**ioms **A**s **R**egular **EX**pressions) which uses a two-level morphology has been developed as part of the Finite State Compiler (FSC) at Rank Xerox Research Centre by L. Karttunen, P. Tapanainen and G. Valetto[1].

3.1 Morphological Variation

Because IDAREX uses a two–level morphology, words can be presented either in their base form at the lexical level or in an inflected form at the surface level. The *surface form* is preceded by a colon and restricts occurrences of the word to exactly this form, e.g.

```
:Welle
```

The *lexical form* is followed by an IDAREX morphological variable specifying morphological features of the word, and a colon. The *morphological variable* can be very general, such as `'A'` for any adjectival use, or more specific, such as `Abse` for adjectives that may not be used in comparative form and `Nsg` to restrict nouns to the singular. For example,

```
durchschlagend A:
```

represents any occurrence of the word G:*durchschlagend* (lit. making its mark on sth) as adjective, e.g. G:*durchschlagende, durchschlagender, durchschlagendsten*, whereas in the following example, comparative and superlative form G:*grünere, grünste* are excluded for the adjective:

```
grün Abse:  Welle Nsg:
```

This way, the restricted morpho–syntactic flexibility of MWLs can be expressed in a compact way, relieving the lexicographer from the burden of listing in an exhaustive way all the possible forms in which an MWL may be used.

[1] For a more detailed description of the formalism see Karttunen and Yampol (1993), Tapanainen (1994), Karttunen (1995), and Segond and Tapanainen (1995).

3.2 Modification

If an MWL can be modified with a particular word this is represented as an optional expression in parentheses, as in

```
:den (:schönen) :Schein
```

The definition of *word-class variables* allows to express lexically unrestricted modifications of an MWL such as insertion of any adverb(s) (the Kleene star operator indicates that the item may occur any number of times):

```
perdre V: ADV* :la :tête
```

On the basis of simpler word-class variables more complex ones may be defined for complex syntactic categories such as NP, ADVP or PP. The morphological and word-class variables can be seen as shortcuts for particular regular expressions which cover specific sequences of morphological and lexical information. Their definition depends on the exact form of the morphological analysis.

3.3 Lexical and Structural Variation

The formalism provides a set of *RE operators* to combine the descriptions of single words. Square brackets '[ ]' and the bar '|' are used to describe lexical variants and more complex alternations such as word order variation in German. For instance, for the French example above we write

```
perdre V: ADV* [:la :tête | :la :boule | :les :pédales ] ;
```

To express German verb-front and verb-final word order as in
V1/2: *dabei wahrt er (immer) den Schein* ('in this, he (always) keeps up appearances')
Vend: *um den Schein zu wahren* ('in order to keep up appearances')
we write

```
[ wahren Vfin: (ADV* NPnom) ADV* :den (:schönen) :Schein |
  :den (:schönen) :Schein (:zu) wahren V: ] ;
```

As such REs quickly become very long and complicated, IDAREX allows the definition of *macros* to capture generalisations on the syntactic level.

3.4 Macros

The power of macros resides mainly in their parametrization. Any position in the macro that we want to instantiate differently for each use is indicated by a parameter `$i`. Instantiations of parameters can be single words in lexical or surface form, variables, operators or other macros. A Macro can be inserted within any IDAREX expression by using its name and specifying, between parentheses and separated by spaces, the parameters that must be substituted to the place-holders at the moment the macro is expanded by the IDAREX compiler.

With the macro mechanism, regularities that can be observed for a whole class of MWLs need to be described in detail only once. The defined macro can then be applied to any instance of this class, using a simple and short expression. This offers a lot of flexibility and the possibility to represent complex phenomena in a compact way. General syntactic properties of MWLs that have to be accessible already at the lexical level can also be expressed.

We want to illustrate the usefulness and expressive power of macros with standard word order variation of German verbal MWLs. Three basic orderings are possible, verb-first (V1), verb-second (V2) and verb-final (Vend), and an infinitive construction (Inf+zu). In addition, scrambling of the external argument (*dir/Ute*) with the subject (*Jan*) is possible in all three word orderings.

V1: *Kommt Jan dir dabei in die Quere?*
('does Jan cross your path in this?', lit. comes Jan you with it in the widthways)

V1Scr: *Kommt dir dabei Jan in die Quere?*
('does Jan cross your path in this?', lit. comes you with it Jan in the widthways)

V2: *Jan kam ihr dabei in die Quere.*
('Jan crossed her path', lit. Jan came her with it in the ...)

V2Scr: *Ihr kam dabei Jan in die Quere.*
('Jan crossed her path', lit. her came with it Jan in the ...)

Vend: *Ute war sauer, weil Jan ihr dabei in die Quere kam.*
('Ute was angry because Jan had crossed her path')

VendScr: *Ute war sauer, weil ihr Jan dabei in die Quere kam.*

Inf+zu: *Jan versucht, Ute dabei nicht in die Quere zu kommen.*
('Jan tries not to cross Ute's path')

All these variations of G:*in die Quere kommen* ('to cross sb's path', lit. in the widthways come) are covered with the following RE using the macro `WOV1ARG` with three parameters, one for the type of the external argument (which could be a dative or accusative object or a PP), the second one for the lexically fixed components of the MWL, and the third one for the verbal component:

```
WOV1ARG(NPdat fix3(:in :die :Quere) kommen)
```

The instantiation of the second parameter uses another macro, `fix3`. This is one of a group of auxiliary macros `fix_i` that we use because we want to be able to instantiate the second parameter of `WOV1ARG` with expressions of variable length:

```
fix5: $1 $2 $3 $4 $5      fix3: $1 $2 $3      etc.
```

`WOV1ARG` is defined as

```
[ $3 Vfin: 1ARG1($1) $2 | $2 (Vaux ( 1ARGEND($1) )) (:zu) $3 V: ]
```

using further macros `1ARG1` and `1ARGEND`. The first part of the disjunction covers V1 and V2 order, the second part matches verb-final order and the infinitival construction and in addition topicalization of the fixed MWL component (*in die Quere*). `1ARG1` and `1ARGEND` capture regular syntactic properties of German and can be reused in other macro definitions: they take care of scrambling of subject and external argument for V1/V2 and Vend ordering, respectively (V1 vs. V1Scr, V2 vs. V2Scr, and Vend vs. VendScr in the examples above).

`1ARG1` is defined as

```
[ (ADV* NPnom) ADV* $1 | (ADV* $1) ADV* NPnom ] ADV*
```

`1ARGEND` is defined as

```
ADV* [ NPnom ADV* $1 | $1 ADV* NPnom ] ADV*
```

In both definitions, $1 is instantiated with the variable for the subcategorized idiom–external complement, i.e. NPakk, NPdat, or with a macro for PPs[2]. In 1ARG1, the first component in each part of the disjunction is made optional because in V2 order, this component is moved to topic position to the left of the verb.

The macro WOV1ARG covers word order variation of verbal MWLs with one external argument. For verbal MWLs with two external arguments (e.g. G:*jdm etw in Rechnung stellen* ('to charge sb for sth', lit. sb sth in bill put)), another macro has to be used. The above example for the use of WOV1ARG leads to the following RE when the macro is compiled:

```
WOV1ARG(NPdat fix3(:in :die :Quere) kommen);
```

expands to

```
[ kommen Vfin: [(ADV* NPnom) ADV* NPdat | (ADV* NPdat) ADV* NPnom] ADV*
:in :die :Quere | :in :die :Quere (Vaux ([ NPnom ADV* NPdat | NPdat ADV*
NPnom ] ADV*)) (:zu) kommen V: ] ;
```

For German, further macros are defined for verbal MWLs where the fixed component can be scrambled with an idiom-external PP argument, for MWLs with a reflexive verb, for prefix verbs in their separated form, and for MWLs with a separable prefix verb. In French, macros describe for example the verb complex for reflexive verbs and constructions with the pronoun *en*.

4 Discussion

4.1 Comparison With Other Approaches

NLP treatments of MWLs in so–called high level grammar formalisms have for example been proposed in Abeillé and Schabes (1989) in the framework of lexicalised TAGs, in Erbach (1992) and Copestake (1994) for HPSG, in Van der Linden (1993) for CG. These approaches to our knowledge cannot satisfactorily represent lexical variants, nor the restricted flexibility and modifiability of MWLs. As MWLs are not simply lexical patterns but also show syntactic behaviour, their existence pleads for a tighter interaction between syntax and lexicon than has often been assumed. We have illustrated how this can be achieved using a finite–state approach and local grammar rules formalized as regular expressions.

Instead of using a high–level grammar formalism we describe MWLs with finite–state local grammars. Although finite–state techniques are known to be unable to represent all the dependencies found in natural language, they have the advantage of allowing a very efficient treatment of a great number of phenomena and the implementation of robust, large–scale NLP systems. Within the finite state approach to NL parsing, local grammars are finite state networks that are matched against (parts of) a NL string. Informally speaking, if the string corresponds to one traversal of the network, it is recognised by the local grammar represented by that network. However, the use of these techniques is usually hampered by the unwieldiness in notation that they lead to. The development of adequate notational devices and corresponding compilers is therefore essential.

The presented approach overcomes this problem: instead of having to specify local grammars directly as finite state networks or as graphs as in work done by the 'Parisian School' (e.g. Maurel 1993, Roche 1993 and Silberztein 1993), they are compiled from descriptions which take the form of REs. IDAREX REs provide a convenient way to mix inflected and uninflected word

[2] The PP macro can be used to express the generally possible alternation of a PP with a pronominal adverb, e.g. *mit etw* vs. *damit.*

forms, morphological features and complete word classes, thus greatly relieving lexicographers from the burden of explicitly listing all the possible forms. With macros, generalizations about patterns that can occur for a whole class of MWLs can be expressed. This compactness and flexibility are, as far as we know, specific to our approach.

Encoding the local grammars as REs instead of encoding them directly as networks does of course not change the expressive power of the formalism, but it conveniently abstracts the handling of MWLs from the graph manipulation level, allowing to develop and employ devices that operate on string representations and map them to the underlying finite state networks. As we have shown above, this simplifies considerably the description of the different patterns of variation occuring in MWLs.

Another interesting property of the finite state approach in general is its modularity. This is an advantage our approach shares with the Parisian notion of local grammars. The local grammars we propose encode MWL restrictions to the general syntax, but they are not meant to replace the general syntax. For instance, in a local grammar rule such as

```
prendre V: (POSCLITIC) ADV * (NP) :sous POSS :bonnet ;
```

which stands for F: *prendre qch sous son bonnet* ('to make something one's concern', lit. take sth under one's hat) the inserted NP and adverbial constituents are external to the MWL and as such should be handled by the general syntax. If desired, the general syntax of an NP can again be a local grammar as for instance proposed in Silberztein (1993) where all linguistic phenomena are treated as local grammars; but it need not to be used in this way. In fact, contrary to the Parisian philosophy, we intend to investigate the integration of local FS grammars with other more powerful grammar formalisms.

4.2 Limitations of Local Grammars

Our matching approach differs from high level parsing where the goal is to constrain the grammar to match all and nothing but grammatical input. The local grammar rules we write are formulated as generally as possible, allowing for overgeneration. They cover at most sentence–length patterns. Although more specific and restrictive rules could be written, this is unnecessary as long as the input is grammatical. If the input were not grammatical the most absurd matches could be obtained. E.g. the pattern for G:*in die Quere kommen* defined in section 3.4 will also match the following ungrammatical strings of German:

* *Kamen oft dem Mann in die Quere.*
(lit. came-pl often the-DAT man in the widthways)

* *Das Kind kam aber dem Mann nicht das Wetter in die Quere.*
(lit. the child came but the-DAT man not the weather in the widthways)

* *In die Quere wollten ihm unter Umständen ich in aller Eile zu kommen.*
(lit. in the widthways wanted him under circumstances I in all hurry to come)

In fact, even grammatical input can sometimes be matched wrongly. To take a very simple example, in F:*cela vous coûtera 3 francs en plus* ('that will cost you 3 francs more') the pattern matched by the RE '`:en :plus`' is associated with the translation 'extra, more, too many'. In F:*la vie est de plus en plus chère* ('life is more and more expensive') the same RE matches part of the longer expression *de plus en plus* ('more and more') which is represented as '`:de :plus :en :plus`'. For this reason, applications employing our system to match their input against IDAREX representations of MWLs should always take care to select the longest among multiple matches. It is not sure, however, that the relation between two matching MWLs is always that of simple inclusion of the one in the other.

The technique has its limits: productive variations of MWLs cannot be systematically captured with our approach because such variations are by their very nature unforeseeable. Examples of this are ad–hoc compound formations, or combinations of metaphors and idioms, as in

> G:*das bißchen Kopf, das sie noch haben, zerbrechen sie sich mit ...*
> (from Fleischer 1982)
>
> ('with that little intelligence they still have they rack their brains over ...',
> lit. the little head that they still have break they themselves with ...)
>
> ← G:*sich den Kopf zerbrechen* ('to rack one's brains', lit. oneself the head break)
> + G:*Köpfchen haben/etwas im Kopf haben* ('to have brains', lit. little head have/sth in the head have)

While it is possible to extend the technique so that the system matches also the approximate *das bißchen Kopf haben* and *sich das bißchen Kopf zerbrechen* to the two expressions involved, it will continue to fail on sentences like the one used in last year's French presidential campaign where the candidate of the extreme right dismissed his more to the left competitors by saying F:*c'est bonnet rouge et rouge bonnet* (lit. it is bonnet red and red bonnet). This variation of the standard expression F:*c'est bonnet blanc et blanc bonnet* ('it's six of one and half a dozen of the other', lit. it is bonnet white and white bonnet) is prompted by the association of *red* with 'to the political left'. Of course, the exploitation of this type of information is not only beyond the scope of the approach that we have presented here but also beyond that of more powerful mechanisms.

5 Application: Context-Sensitive Dictionary Look–Up

The presented approach is successfully used in the COMPASS project[3] whose main aim is the development of an on–line foreign language comprehension tool. For this, standard bilingual dictionaries have been converted into special dictionary databases. The COMPASS system, based on the LOCOLEX engine developed at RXRC (see Bauer, Segond and Zaenen (1995) for a description), allows look–up of words in the bilingual dictionary database directly out of an on–line text. When the user clicks on an unknown word in the foreign language, LOCOLEX evaluates the sentence context of the queried word. Currently, the system determines the word's part of speech and whether the word is part of an MWL. In the latter case, the translation for the entire MWL is returned, otherwise a selection of translations for the correct part of speech. A more detailed description of the COMPASS system and the recognition of MWLs can be found in Breidt and Feldweg (1996).

In order for COMPASS to recognize MWLs these are represented as IDAREX REs in the dictionary database. Unfortunately, the dictionary conversion and adaptation cannot be done fully automatically, and in particular the formal representation of the many MWLs as REs is quite a time–consuming task. However, once the MWLs listed in the dictionary have been manually changed into their *canonical base form* — which includes possible lexical variants and modifiers and indicates morphologically flexible components and the scope of alternative components — the IDAREX REs describing all possible contexts in which the MWLs can occur can be produced automatically[4]. For instance, the following canonical forms correspond to

[3] 'Adapting bilingual dictionaries for COMPrehension ASSistance', LRE-62-080.

[4] The automatic RE production from canonical forms is so far only implemented for German, but the mechanisms can just as well be applied for the other languages.

some of the examples from section 2:

out of the blue
faire° le° fou°
grüne° Welle (sg)
avoir° un (sacré) coup dans l'aile
das Handtuch werfen°

kick° the bucket
durchschlagender° Erfolg°
perdre° ˆla tête/la boule/les pédalesˆ
den (schönen) Schein wahren°
T: den Vogel (bei etw) abschießen°

Such canonical base forms, somewhat similar in spirit to the notation used in Longman's 'Dictionary of English Idioms' (1979), do not only form the basis for the automatic processing and recognition of MWLs. Human users as well would profit from a careful description of the variability of MWLs, so it should be worthwhile to also include the canonical forms in dictionaries for human users.

6 Conclusion

We believe the contribution of IDAREX to the problem of recognizing and handling MWLs in NLP systems is twofold. On the one hand, at the description level, the formalism allows to represent idiosyncratic properties as well as regularities of MWLs in a convenient, compact and general way. Its syntax is based on and exploits the well-known concept of REs, supplemented by extensions such as variables and macros. With this notation, it is possible to write local grammar rules that capture a great deal of different variations of MWLs. IDAREX can be easily employed by lexicographers, even with little specific training, and allows to represent large numbers of MWLs with a reasonable effort. So far, we have successfully applied this approach to more than 15,000 English, French and German MWLs (see also Segond and Breidt 1995). Moreover, the production of IDAREXREs can be supported by simple semi-automatic procedures, as we briefly described above.

On the other hand, at the operational level, IDAREX constitutes an efficient and modular mechanism for MWL recognition that can be employed in real life systems, due to the underlying two-level morphology and finite state technology. The transformation of regular expressions into finite state networks is completely disjoint from lexicographic work, and is carried out by the FSC compiler. Moreover, computation of matches between instances of MWLs found in texts and their finite state representations has in general a very good performance.

The successful use of IDAREX in a multi-lingual context as part of the COMPASS comprehension assistant has greatly contributed to the construction, optimization and validation of our paradigm and of the underlying finite state mechanism. The satisfactory results and the observations coming from such a fairly large scale field test are encouraging to further investigate and continue perfecting our approach to MWL recognition.

Acknowledgements

This is an extended version of the poster presented at COLING 96. We thank Annie Zaenen, Ted Briscoe, and Irene Maxwell for their comments on an earlier draft of this paper. The work described here is part of the LOCOLEX/COMPASS project (LRE research project 62-080). It presupposes work done by other members of the LOCOLEX/COMPASS team, especially Lauri Karttunen and Pasi Tapanainen. The dictionaries used in the project are the Oxford-Hachette English-French dictionary (1994) and the HarperCollins German-English dictionary (1991). We are grateful to the publishers to give us the right to use them for research purposes.

References

Abeillé, Anne and Yves Schabes. 1989. Parsing idioms in lexicalized TAGs. In *Proceedings of the 4th EACL*, Manchester, UK.

Bauer, Daniel, Frédérique Segond, and Annie Zaenen. 1995. LOCOLEX: the translation rolls off your tongue. In *Proceedings of the ACH-ALLC conference*, pages 6–8, Santa Barbara, California.

Breidt, Elisabeth and Helmut Feldweg. 1996. Accessing Foreign Languages with COMPASS. *The MT Journal*, special issue on tools for machine–aided translation, edited by Kenneth Church and Pierre Isabelle. To appear.

Brundage, Jennifer, Maren Kresse, Ulrike Schwall, and Angelika Storrer. 1992. Multiword Lexemes: A Monolingual and Contrastive Typology for NLP and MT. *IWBS Report 232*, IBM TR-80.92-029, IBM Deutschland GmbH, Institut f"ur Wissensbasierte Systeme, Heidelberg, September.

Copestake, Anne. 1994. Idioms in general and in HPSG. Presentation given at the Workshop 'The Future of the Dictionary', Uriage-Les-Bains, France, September.

Engelke, Sabine. 1994. *Eigenschaften von Phraseolexemen: Eine Untersuchung zur syntaktischen Variabilität und internen Modifizierbarkeit von somatischen verbalen Phraseolexemen.* Masters Thesis, Universität Tübingen, Germany, April.

Erbach, Gregor. 1992. Head-Driven Lexical Representation of Idioms in HPSG. In M. Everaert et al., editors, *Proceedings of the International Conference on Idioms*, Tilburg, NL, September.

Fleischer, Wolfgang. 1982. *Phraseologie der deutschen Gegenwartssprache.* VEB Bibliographisches Institut, Leipzig, Germany.

Gross, Maurice. 1982. *Une classification de phrases figées français.* Revue Qébécoise de Linguistique, Vol. 11, No. 2, Montreal.

HarperCollins and Klett. 1991. *German-English dictionary.* P. Terell,`V. Schñorr, W. V. A. Morris, R. Breitsprecher, editors. 2nd Edition. HarperCollins Publishers, Glasgow, UK.

Karttunen, Lauri and Todd Yampol. 1993. Interactive Finite-State Calculus. *Technical Report ISTL-NLTT-1993-04-01*, Xerox Palo Alto Research Center, California.

Karttunen, Lauri. 1995. The Replace Operator. *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL-95)*, Boston, MA.

van der Linden, Erik-Jan. 1993. *A Categorial Computational Theory of Idioms.* OTS Dissertation Series, Utrecht, NL.

Maurel, Denis. 1993. Passage d'un automate avec tables d'acceptabilité à un automate lexical. In *Actes du colloque Informatique et langue naturelle*, pages 269–279, Nantes, France.

Nunberg, Geoffrey, Thomas Wasow, and Ivan Sag. 1994. Idioms. *Language*, 70/3:491–538.

Oxford-Hachette. 1994. *English-French Dictionary.* Oxford University Press, Oxford, UK.

Roche, Emmanuelle. 1993. *Analyse syntaxique transformationnelle du français par transducteurs et lexique-grammaire.* Thèse de doctorat, Université Paris 7.

Segond, Frédérique and Tapanainen Pasi. 1995. Using a finite-state based formalism to identify and generate multiword expressions. *Technical Report MLTT-019*, Rank Xerox Research Centre, Grenoble, France, July.

Silberztein, Max. 1993. *Dictionnaires électroniques et analyse automatique de textes – Le système INTEX.* Masson, Paris, France.

Tapanainen, Pasi. 1994. RXRC Finite-State rule Compiler. *Technical Report MLTT-020*, Rank Xerox Research Centre, Grenoble, France.

Transformations in Dictionary Resources Accumulation – Towards a Generic Approach

DOAN NGUYEN HAI

ABSTRACT *This article proposes a generic viewpoint on the problem of accumulation of dictionary resources. Accumulation consists of two principal kinds of activity: acquisition and transformation; the importance of the latter is emphasized by the fact that it helps produce new valuable lexical sets. A generic approach to the problem is proposed which includes a framework for the accumulation process and a transformation formalism.*

Key-words: transformation, acquisition, reutilization, accumulation, dictionary resources, lexical database, computational lexicography, computational linguistics.

1. INTRODUCTION

The problem of acquisition, manipulation and reutilization of dictionary resources has been considered very important since a long time when one recognized the essential role of the lexicon in almost every computational linguistic application. It has been studied in many projects (for example, [Dymetman 86], [Madsen 86], [Byrd & al 87]). But they all seem to treat the problem specifically or to concentrate only on some particular aspects of the problem. In this article, we present a generic approach to the problem. Concrete work on a specific dictionary has suggested to us the general idea that dictionary resources accumulation consists of two principal activities: acquisition and transformation; moreover, with transformation new valuable lexical sets can be cheaply obtained. We then present a framework for the accumulation process and a specialized language for transformation specification and execution.

2. WORK ON A RUSSIAN-FRENCH DICTIONARY

Particular acquisition and transformation work was carried out on a Russian-French (R-F) dictionary accessed by Russian lemmas. (This dictionary was produced from coded data used in a machine translation system - the ARIANE of GETA [Nédeau 94].) The work consisted of acquiring the dictionary to a SGML format, producing a R-F dictionary of Russian "lexical units" (LU), and producing an inverse dictionary, a French-Russian (F-R) dictionary of French LU's.

Note: A *lexical unit (LU)* is understood here to be a derivational family of lemmas. "compare", "comparison", "comparable", "incomparable" belong to a LU for which we can choose "compare", for example, as the denotation.

2.1. The source dictionary

The source dictionary files, contained in an IBM-9221/130, are structured in lines (in total, there are about 22000 lines), the lines are structured in fields of fixed width (Fig. 1). A Russian lemma together with its corresponding information (its LU denotation, grammatical code, sense numbers, French translations of each sense with their LU denotations, etc) is contained in one or more lines. In general, each line corresponds to one French translation (among several) of one sense of a Russian lemma, the Russian part being identical for all lines of the same sense. When this translation is a phrase, one constructs as many almost identical lines as the number of words in the phrase so that each line contains information related to each word. When the Russian lemma is an acronym/abbreviation, there will be additional lines each of which gives a cross-reference to a component of the original phrase of the acronym/abbreviation. Thus, on the one hand information is so scattered, on the other hand there is a great data redundancy in the lines. Moreover, this structuration of the dictionary seems not favourable for further computational linguistic applications.

2-23	*24*	*26-72*	*73*	*74-*		*84-126*	*128*	*129-*	*139-180*
Russian LU	*(1)*	*Russian lemma*	*(2)*	*80*		*French lemma*	*(3)*	*138*	*French LU*
				code				*code*	
garmonika	*1.*	*garmoµnika*		*$f*		*barmonique*		*$f*	*barmonique*
garmonika	*2.*	*garmoµnika*		*$f*		*accordéon*		*$m*	*accordéon*
garmonika	*2.*	*garmoµnika (gubnaµya --)*	*L*	*$f*		*barmonica*		*$m*	*barmonica*
garmonika		*garmoniµst*		*$f2*	*1.*	*barmoniste*		*$f2*	*barmonie*
garmonika		*garmoniµst*		*$f2*	*2.*	*accordéoniste*		*$f2*	*accordéon*
GPS	*1.*	*GPS*		*$S1*		*==> Generaµtor*			
GPS	*1.*	*GPS*		*$S1*		*==> Promezbuµtokbnoj*			
GPS	*1.*	*GPS*		*$S1*		*==> Seµtki*			
GPS	*1.*	*GPS*		*$m*		*générateur de réseau de*	*11*	*$v2*	*générer*
GPS	*1.*	*GPS*		*$m*		*générateur de réseau de*	*12*	*$m.2*	*réseau*
GPS	*1.*	*GPS*		*$m*		*fréquences intermédiaires*	*21*	*$fp.2*	*fréquence*
GPS	*1.*	*GPS*		*$m*		*fréquences intermédiaires*	*22*	*$a*	*intermédiaire*
GPS	*2.*	*GPS*		*$S1*		*==> Giµbkaya*			
GPS	*2.*	*GPS*		*$S1*		*==> Proizvoµdstvennaya*			
GPS	*2.*	*GPS*		*$S1*		*==> Sisteµma*			
GPS	*2.*	*GPS*		*$f*		*atelier flexible*	*1*	*$m*	*atelier*
GPS	*2.*	*GPS*		*$f*		*atelier flexible*	*2*	*$v3*	*fléchir*

(1) Sense number of the Russian lemma
(2) "L" if the Russian lemma is "locution"
(3) Position number of the French LU in the dependency tree representing the French lemma phrase

Russian words are transliterated according to GETA's convention (in which µ represents the accent mark for Russian words).

Fig.1. Some excerpts from the source dictionary.

2.2. Acquisition

In general, the purpose of the *acquisition* work is to transfer the data into a new "convenient" formalism - a predefined format (eg. SGML), a database formalism, etc. It is understood here that acquisition does not change the linguïstic structure of the dictionary entry. Acquisition methods have been well studied, for example, in [Byrd & al 87]; they consist principally of executing an analysis procedure on the resources to identify its information elements.

In our case, the acquisition process is reduced to a "LISPification" of the line entries of the source dictionary, which puts parentheses surrounding information fields. (This was done easily with the column manipulation functions of the XEDIT editor on IBM.) Our intention here is not only to profit from the LISP analyzer of identifying the information fields of the dictionary but also to carry on further manipulations in an object-oriented LISP environment (in detail, the Common LISP Object System - CLOS) on Macintosh. By some simple programming, we have produced the SGML form and the object-oriented form of the source dictionary from its LISPified form. The object-oriented form which will serve for later transformation work is contained in a persistence structure with index called DOP databases. (DOP = Dictionary Object Protocol [Lafourcade & Sérasset 95], based on WOOD (William's Object Oriented Databases) [StClair 95]).

2.3. Transformation: Producing a R-F lexical database of Russian LU's.

The first *transformation* work is to produce a R-F lexical database (LDB) of Russian LU's. The main idea is to *regroup* entries of the same LU into a new entry whose headword is the denotation of the LU. The structure of the LU entry is defined so that information in the source dictionary is not only kept all but also reorganized efficiently and rationally. The body of the LU entry is a set of lemmas. A lemma substructure contains, in brief, a Russian lemma, its senses and their French translations, its relevant locutions, and its cross-references (Fig. 2).

The regrouping is implemented as a top-down sequence of *integration* operations. Each entry of the source dictionary will be integrated into the new LU dictionary as follows: if in the new dictionary there is no entry of the same LU as the source entry, a new LU entry will be created from the source entry. Otherwise, the source entry will be integrated into the LU entry corresponding to its LU. To integrate a source entry into a LU entry is to create a new lemma element in the lemma set of the LU entry if the source entry's lemma has not been yet in that set, or to integrate the source entry into the element corresponding to its lemma otherwise. And so on. Intuitively, integrating an object into a unit can be understood as modifying the contents of (some components of) the unit by adding to it some information supplied by the object.

Entries of the source dictionary which cause errors discovered in the regrouping process are recorded and then re-imported to the LU dictionary after correction. As result, a R-F LU LDB has been obtained with more than 5700 entries (Fig. 3).

```
unit-lex
    ULm : denotation of the LU
    lemme
        lem : source language lemma
        cont
            n-plsem : sense number
            contp
                code : (grammatical) code of lem
                trad
                    n-trad : translation number
                    lemc : translation of lem
                    code : code of lemc
                    ULc : LU of lemc
                    compos (a component of lemc
                                  when lemc is a syntagme)
                        n-ord : position of the component
                                   in the dependency tree
                        ULc : LU of the component
                        code : code of ULc
                    compos ...
                    ...
                trad ...
                ...
            renvoi
                n-plsem-rv : sense number of rv
                rv : source language cross-reference
                code : code of rv
            renvoi ...
            ...
            loc
                loclem locution whose kernel is lem
                code code of loclem
                trad ...
                trad ...
                ...
            loc ...
            ...
        cont ...
        ...
    lemme ...
    ...
```

Fig. 2. Structure of the R-F UL.

```
unit-lex
    ULm : garmonika
    lemme
        lem : garmoµnika
        cont
            n-plsem : 1
            contp
                code $f
                trad
                    n-trad : NIL
                    lemc : harmonique
                    code: $f
                    ULc : harmonique
        cont
            n-plsem : 2
            contp
                code $f
                trad
                    n-trad : NIL
                    lemc : accordéon
                    code : $m
                    ULc : accordéon
            loc
                loclem : garmoµnika (gubnaµya --)
                code : $f
                trad
                    n-trad : NIL
                    lemc : harmonica
                    code : $m
                    ULc : harmonica
    lemme
        lem : garmoniµst
        cont
            n-plsem : NIL
            contp
                code $f2
                trad
                    n-trad : 1
                    lemc : harmoniste
                    code : $f2
                    ULc : harmonie
                trad
                    n-trad : 2
                    lemc : accordéoniste
                    code : $f2
                    ULc: accordéon
```

Fig. 3. The LU entry "garmonika"

2.4. Transformation: Producing a F-R LDB of French LU's.

Obtaining an inverse dictionary, ie. a F-R LU LDB, seems to be an interesting idea, because it could help demonstrate persuasively the usefulness of transformation in lexical accumulation. Although not all target language translations in a dictionary could constitute an entry in the inverse dictionary, one might hope a majority of them could. Moreover, translations which cannot stand alone as an entry could be retained as idiomatic expressions or as examples. At last, a lexicographer might do an ultimate revision in which he will exclude unappropriate entries. An assumption is thus made here that each French translation in the source dictionary will constitute a lemma or an idiomatic expression in a F-R LU entry; the elimination of unsuitable ones will be the work of a lexicographer.

At first, it seems that the production of the F-R LU LDB could be implemented in a similar way to the production of the R-F LU LDB, except that the input is the original dictionary inverted - that line dictionary looks symmetrical for both languages. Deeper analyses, however, show that this strategy is not efficient, since the regrouping of (a) Russian cross-references and of (b) information concerning French syntagmatic translations, are very costly, while they have been done in the course of the production of the R-F LU LDB. (Note also that both kinds of data (a) and (b) occur frequently in our source dictionary by its science-technology oriented nature.)

The R-F LU LDB instead of the original dictionary is thus chosen as the input for the inversion process. The *inversion* consists of *splitting* and integration operations. Splitting a tree structure means to generate a set of items of a new structure containing information collected on a path of the tree (Fig.

4). Each R-F LU entry - viewed as a tree because it contains several lemmas, each lemma contains several senses, and so on - is first split into a set of intermediate structures (Fig. 5). That structure can be considered as an entry having a French translation of a Russian lemma/locution as the headword, the Russian lemma/locution itself and other relevant information (including regrouped data kinds (a) and (b) said above) as the contents. The intermediate entries are then integrated into the F-R LU LDB which is being constructed gradually. The resulting F-R LDB has more than 5700 French LU's (Fig. 6).

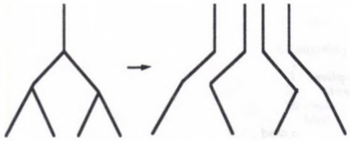


Fig. 4. Splitting a tree structure.

```
lem-art2
    lem: accordéon
    code: $m
    ULm: accordéon
    trad
        lemc: garmoµnika
        code: $f
        n-plsem-s: 2
        ULc: garmonika
```

```
lem-art2
    lem: accordéoniste
    code: $f2
    ULm: accordéon
    trad
        lemc: garmoniµst
        code: $f2
        n-trad-s: 2
        ULc: garmonika
```

Fig. 5. Intermediate items split from the LU "garmonika"

```
<unit-lex>
  <ULm> accordéon
    <lemme>
      <lem> accordéon
      <code> $m
      <trad>
        <lemc> garmoµnika
        <code> $f
        <n-plsem-s> 2
        <ULc> garmonika
    <lemme>
      <lem> accordéoniste
      <code> $f2
      <trad>
        <lemc> garmoniµst
        <code> $f2
        <n-trad-s> 2
        <ULc> garmonika
```

```
<unit-lex>
    <ULm> atelier
    <lemme>
      <lem> atelier flexible
      <trad>
        <lemc> GAP
        <code> $n
        <ULc> GAP
        <renvoi>
          <rv> Giµbkoe
          <code> $S1
        <renvoi>
          <rv> Avtomatiziµrovannoe
          <code> $S2
        <renvoi>
          <rv> Proizvoµdstvo
          <code> $S3
      <trad>
        <lemc> GPS
        <code> $f
        <n-plsem-s> 2
        <ULc> GPS
        <renvoi>
          <rv> Giµbkaya
          <code> $S1
        <renvoi>
          <rv> Proizvoµdstvennaya
          <code> $S1
        <renvoi>
          <rv> Sisteµma
          <code> $S1
      <comp>
        <n-ord> 1
        <compos>
          <n-ord> 1
          <ULc> atelier
          <code> $m
        <compos>
          <n-ord> 2
          <ULc> fléchir
          <code> $v3
```

Fig. 6. Examples of F-R LU's in SGML format.

3. TOWARDS A GENERIC APPROACH

We are now trying to come to a generic approach to the problem of accumulation of dictionary resources; in this approach, many ideas have been suggested from the work described above.

3.1. A framework for the lexical accumulation process

A principal idea proposed in this article is that lexical accumulation consists not only of acquisition but also of transformation work:

ACCUMULATION = ACQUISITION + TRANSFORMATION

Each work can be divided into smaller tasks as follows:

3.1.1. Acquisition

(i) Analyzing the source dictionary (and defining its structure) in linguistic aspect.

This clarifies the nature of the dictionary (mono/bi/multiligual? popular/specialized? human-oriented/machine-oriented? etc), its structures (global structure, entry structure, etc), its contents (signification of components of an entry, the exceptions, the quantity of words, etc), etc. A difficulty often encountered in this task is that the documentation of the dictionary is not always complete and exact.

(ii) Acquiring the dictionary in computational aspect.

This task transfers the source dictionary into a "suitable" form. It consists of: transferring the dictionary from a device to another, doing some "cleaning", defining the structures in the dictionary, and most important, executing an analysis process on the dictionary. In general, that analysis process tries to identify the elements in the dictionary till the "atomic" ones. Since the "database approach" has proved so fruitful for lexicon manipulation [Calzolari 86], [Byrd & al 87], the resulting dictionary is usually contained in a database system.

(iii) Verifying and correcting the acquired dictionary.

The errors detected in the dictionary acquisition may be real errors of the dictionary or some "exceptions" that the analysis specification have not mentioned. Correction, as the state of the art, is usually done by hand. Verification can be automated somehow (eg. verification of the numeration of items in an entry).

3.1.2. Transformation

(i) Defining the linguistic structures of the target dictionary and specifying the transformation.

To define the structures of the target dictionary, one should specify the signification of every component, that means the correspondence between target and source structures. We will call the set of all correspondence specifications the transformation specification. A formalism which permits to specify (and even to execute) the transformation is obviously useful, and this will be discussed in section 3.3.

(ii) Executing the transformation.

One may think of three levels in accomplishing this task (and the precedent one, too):

Level 1: The transformation is executed by direct programming from the lexicographer's informal specification. The programs written are thus specific for the current case. This approach seems acceptable when, in fact, lexicography (computational or not) is done only in a few specialized centers, where cooperation between lexicographers and computer scientists is well established. (Moreover, there appears more and more specialists competent in both domains.) However, it cannot serve efficiently the quickly growing needs of lexicons of all kinds for various language engineering applications.

Level 2: Direct programming but with formal specification. Ideas for the transformation could be expressed more exactly and more quickly. Programming would be easier while the cooperation between the lexicographer and the programmer could be looser.

Level 3: Formal specification followed by automatic execution. This implies the use of a specialized language and an execution environment, which will be the theme of section 3.3.

(iii) Verifying and correcting the resulting dictionary.

3.2. Transformation operations

Some operations which seems most useful in dictionary resources manipulation are introduced here. There are roughly two great transformation categories: synthesis and extraction. An operation could be a composition of several other ones.

(1) *Creation* of a unit from a unit.

(2) *Integrating* a unit into another unit of "greater" structure. Ex: Integrating a new sense into an entry.

(3) *Regrouping* (or *Union*) a set of units to form another set or a "great" unit. Intuitively, the original units are of the same kind, of the same rank, etc, and their common information will be hold as unique (ie., not repeated) in the target structures.

(4) *Combining* sets to make a new set. Ex: [Byrd & al 87] suggests as example that one might think of relating grammatical information from the *Longman Dictionary Of Contemporary English* with definitions from the *Webster's Seventh Collegiate Dictionary* .

(5) *Synthesis* of given data to form new data. Operations 1 to 4 are special cases of synthesis. The significance of the synthesis is that generally it gives a synergy result.

(6) *Splitting* a unit into "smaller" ones.

(7) *Inversion*, eg. inversing a bilingual dictionary.

(8) *Filtering* a set to select elements satisfying some condition.

(9) *Projection* to take only some components of a unit.

(10) *Extraction*. There are two sorts of extraction: syntactic and semantic. Projection is a syntactic extraction. The extraction of the "genus" and the "differentia" in a word definition [Amsler 80] is a semantic one.

3.3. A transformation formalism (= a specialized language)

A formalism for the lexical accumulation process will be appreciated because it will help lexicographers and computer scientists form and carry out quickly their ideas. Such a formalism should include possibilities of lexical structure description and dictionary transformation specification. [Boitet & Nédobejkine 86] proposes a lexicon organization which can integrate in it both human- and machine-oriented lexical knowledge. [Sérasset 94] defined languages for general lexical structure description (LINGARD and LEXARD). In what follows we suppose that lexical structure are implemented in an object-oriented paradigm, the concepts *class*, *object*, *slot* understood as usual.

Ex: The *unit-lex* class implemented here has two slots: *ULm* contains the denotation of the LU, *lemme** contains the set of all lemmas of the LU (Fig. 7).

unit-lex
ULm: *string* (Russian LU)
lemme*: *set of lemme's* (all lemmas of the Russian LU)

lemme
lem: *string* (Russian lemma)
cont*: *set of cont's* (a *cont* contains information related to a sense of the Russian lemma)

cont
n-plsem: *number*
contp: *contp* (contains all French translations for this sense of the Russian lemma above and their related information)
renvoi* : *set of renvoi 's* (cross-references for this sense)
loc*: *set of loc's* (all Russian idioms of this sense and their related information)

etc.

Fig. 7. Structures for the R-F LU entry.

We concentrate on transformation specification with the TRANSARD language whose operations can be described in 4 groups as follows.

1. *Low-level* operations include assignments, conditionals, and iterations. The design intention is that iteration operations are replaced by more descriptive operations of the "higher" levels, and thus can be avoided by user as much as possible; we keep them, however, for professional programmers to use if they like to, and for the sake of the Turing machine equivalence.

2. A library of *data manipulation functions*, such as string functions (comparison, substring, taking the first letter of a word, etc).

3. The *intermediate level* is made up of set and first-order predicate calculus operations. Some important set operations:

(subset *set pred*) takes the subset of all elements of *set* that satisfy *pred*.
(f-set *set pred func*) gives the set {*func*(x) | x in *set* and *pred*(x)}.
(partition *set* **by** *func*) partitions *set* into subsets whose elements have the same value of applying *func*.

Some principal predicate calculus operations:

.(**exists** *var* **in** *set* **suchthat** *cond*) (existential quantification. *cond* denotes a predicative expression)
(**all** *var* **in** *set* **satisfy** *cond*) (universal quantification)
(**if-exists** *var* **suchthat** *cond* **then** *work1* [**else** *work2*])
(**for-all** *var* **in** *set* [**suchthat** *cond*] **do** *work*) (parallel work on the elements of *set*)
(**for-all*** *var* **in** *set* **suchthat** *cond* **do** *work*) (sequential work)

Ex: Select all the R-F LU's which are a verb. From the design of the original dictionary, a LU is a verb if in its lemma set, there is a (Russian) lemma having a verb sense. (A sense is a verb when its code is an element of the set ("$vi" "$vi.r" "$vp" "$vp.r" "$vd" "$vd.r")). This can be specified as follows:

```
(define-function verb-LU (LU)
    (exists lemme in LU.lemme*                              ; See Figs. 2 and 7
        suchthat (exists cont in lemme.cont*
                    suchthat (is-element cont.contp.code
                                  '("$vi" "$vi.r" "$vp" "$vp.r" "$vd" "$vd.r")))))
```

So the subset containing all the verb LU's of the R-F LU LDB (named by RF-LULDB) could be obtained by:

```
(subset RF-LULDB 'verb-LU)
```

4. In the *high level* are transformation operations. Some operations mentioned in section 3.2 are general concepts and it is not obvious how to implement them (4, 5, 10). The language, however, is designed to be *open*, that is one can always add new operations to it.

The transformation operations implemented at present are:

(1) *Creation* of a unit *x* from a unit *a*. User can do this by one of two ways: executing direct assignments to the slots of a newly created unit x or executing the generic operation:

```
(create-from a new-class class-of-x assign-list ((slot1 f1) (slot2 f2) ...))
```

This gives a new unit whose slots *slot1*, *slot2*, ... are assigned the values of applying the functions *f1*, *f2*, ... on *a*, respectively.

(2a) *Integrating* a unit *a* into a unit *x*. One can use the following operation:

```
(integrate a into x assign-list ((slot1 f1) (slot2 f2) ...)
```

slot1, *slot2*, ... of *x* are assigned the values *(f1 a x)*, *(f2 a x)*, ... , respectively.

(2b) A special case of integration is to integrate a unit *a* into an element of a set *x*. The element is chosen by some condition (*pred*), *a* being integrated into it by some function (*func1*). If no such element found, a new element will be created from *a* by *func2* and inserted into *x*.

```
(integrate a into-set x find-pred pred integrate-func func1 create-func func2)
```

(3a) *Regrouping* a set of units to form a unit *x*.

```
(unite set new-class class-of-x assign-list ((slot1 f1) (slot2 f2) ...)
```

slot1, *slot2*, ... of *x* are assigned the values *(f1 set)*, *(f2 set)*, ... , respectively.

(3b) *Regrouping* a set of units to form another set of units. This can be done by one of two ways:

1- Partition the source set, then unite each subset to form an element of the target set:

```
(regroup-by-partition set partition-by partition-func unite-into-new-class new-class with-assign-list assign-list)
```

Ex: To regroup a R-F line entry dictionary (LINEDIC) into a R-F LU entry dictionary (see section 2.3 and Fig. 2):

```
(regroup-by-partition LINEDIC
   partition-by 'source-LU   ; source-LU takes the source language (Russian) LU of the line entry
   unite-into-new-class 'unit-lex
   with-assign-list '((ULm same-source-LU) (lemme* regroup-lemme*))
```

(Thing from a semi-colon (;) to the end of the line is a comment. The apostrophe has the same meaning as the "quote" of LISP.)

```
(define-function same-source-LU (line-subset)
   ; the elements of line-subset has a common source LU
   (source-LU (any-of line-subset)))              ; any-of gives an arbitrary element of a set

(define-function regroup-lemme* (line-subset)
   (regroup-by-partition line-subset
        partition-by 'source-lem
        unite-into-new-class 'lemme
        with-assign-list '((lem same-source-lem) (cont* regroup-cont*))))
etc.
```

So in a regroup operation, to compute the value of a slot one might have to provoke another regroup; this example thus also illustrates the top-down style of TRANSARD in transformation specification.

2- Integrate each element of the source set into the target set (see (2b)):

```
(regroup-by-integration source-set target-set find-pred pred intergrate-func func1 create-func func2)
```

(6) *Splitting* a unit into "smaller" ones.

```
(split unit by func assign-list assign-list)
```

In fact, it is the function *func* that "splits" the unit, ie., returns some set of objects. Some slots of these objects not having yet value will be assigned by *assign-list*. An example will help make clear the use of split:

Ex: To split a R-F *LUentry* into a set of objects of the *lem-art2* structure (Fig. 5):

```
(split LUentry by 'from-lemme* assign-list '((trad.ULc ULm-of)))
```

The assign list says that for all lem-art2 entries, returned by applying *from-lemme** on *LUentry*, the slot *ULc* of the slot *trad* will be given a common value equal to the *ULm* (Fig. 2) of *LUentry* (computed by *ULm-of*). The Pascal-like notation *object.slot* signifies an access to a slot of an object. *from-lemme** is defined as follows:

```
(define-function from-lemme* (LUentry)
   (union2 (f-set LUentry.lemme* true 'from-lemme)))
; from-lemme splits a lemme and returns a set of lem-art2's. union2 takes the union of its argument which is a
;set of sets.
(define-function from-lemme (lemme)
   (split lemme by 'from-cont* assign-list '((trad.lemc lem-of))))
etc.
```

(8) *Filtering* is equivalent to taking a subset.

(9) *Projection* can be carried out by a create operation (see (1)) with an appropriate assign list.

The first prototype of TRANSARD has been implemented as an embedded language in CLOS (so, there are some unimportant changes in the syntax of the language), that permits the specifications to be executed at hand. Sets are implemented as LISP lists and DOP databases. We have re-written the transformations to produce the R-F and the F-R LU LDB's using the generic operations. For the first LDB, two specifications have been implemented using the two styles of regroup (by partition and by integration). For the F-R LDB, the *inversion* has been implemented by splitting each R-F LU entry into lem-art2's (Fig. 5) and then regrouping the lem-art2 set (using regroup-by-partition) to obtain the F-R LU set. The specifications written are shorter than the programs directly implemented described in section 2 but with an equivalent execution performance. To sum up, the design intention of TRANSARD is that users can specify the transformations in a top-down style, using simple low-level control structures (assignments and conditionals), set and predicate calculus operations (the intermediate level), which make the language more logic, more descriptive, and specialized operations (the high level), which augment the power and convenience of the language.

4. CONCLUSION

Accumulation is the first step for dictionary resources reutilization. In this perspective, transformation can help produce new valuable lexical sets. Transformation is in fact complicated, so a formalism with well-defined and possibly implemented primitive operations would be favourable for quickly specifying and executing the transformation. A generic environment with acquisition and transformation main subsystems could be very helpful to huge lexical accumulation work and various reutilization requirements.

ACKNOWLEDGEMENT

The author is grateful to the French Government for her scholarship, to Christian Boitet and Gilles Sérasset for the suggestion of the theme and their help. Thanks to all other GETA's members for all their cooperation.

REFERENCES

AMSLER R. 80, The Structure of the Merriam-Webster Pocket Dictionary. PhD Thesis, U. of Texas at Austin.

BOGURAEV B. 86, Machine-Readable Dictionaries and Research in Computational Linguistics. Workshop on Automating the Lexicon, Marina di Grosseto, Italy 1986, 33 p.

BOGURAEV B. & BRISCOE T. 89, Chapter 1 (Introduction) of "Computational Lexicography for NLP", Boguraev B. & Briscoe T. ed., Longman 1989, pp. 1-40.

BYRD R., CALZOLARI N., CHODOROW M., KLAVANS J., NEFF M., RIZK O. 87, Tools and Methods for Computational Lexicology. Computational Linguistics Vol 13, N° 3-4, 1987, pp. 219-240.

BOITET C., NEDOBEJKINE N. 86, *Towards Integrated Dictionaries for M(a)T: Motivations and Linguistic Organisation.* COLING 86, pp.423-428.

CALZOLARI N. 86, Structure and Acess in an Automated Lexicon and Related Issues. Workshop on Automating the Lexicon, Marina di Grosseto, Italy 1986, 18 p.

CALZOLARI N. 88, The dictionary and the thesaurus can be combined. Dans "Relational models of the lexicon", Evens M. ed., Cambridge U. Press 1988, pp. 75-96.

CALZOLARI N., BRISCOE T. 93, ACQUILEX-I and -II. Dans Calzolari N., Course Computational Lexicons, Fifth European Summer School in Logic, Language and Information, U. de Lisboa, 1993, 17 p.

DYMETMAN M. 86, Logiciel de bases de données et expérimentation. Récupération du contenu lexical d'un système de TAO pour constituer un jeu d'essai important: spécification. RI DGT N° 7 et 10 - 1986. GETA.

EVENS M. 88. "1. Introduction" dans "Relational models of the lexicon", Evens M. ed., Cambridge U. Press 1988, pp 1-38.

GASCHLER J., LAFOURCADE M. 94, Manipulating human-oriented dictionaries with very simple tools. COLING 94, pp. 283-286.

HARIE S.89, Analyse automatique d'un dictionnaire en vue de la constitution d'une BDL. Mémoire DEA 1989, U. Aix-Marseille III, 68 p.

KAY M. 84, The Dictionary Server. Panel Session "MRDs", COLING 84, p. 461.

KLAVANS J. 88, COMPLEX: A Computational Lexicon for Natural Language Systems. COLING 88, pp. 815-823.

KNOWLES F. 86, Computational Lexicography and Lexical Databases. Workshop on Automating the Lexicon, Marina di Grosseto, Italy 1986, 47 p.

LAFOURCADE M., SÉRASSET G. 95, DOP (Dictionary Object Protocol). cambridge.apple.com. 1995.

MADSEN B. 86, Danish Projects within the Field of Computational Lexicography. Workshop on Automating the Lexicon, Marina di Grosseto, Italy 1986, 15 p.

MICHIELS A., MULLENDERS J., NOEL J. 80, Exploiting a Large Data Base by Longman. COLING 80.

NEDEAU N. 94, Dictionnaire naturel russe-français issu des fichiers codés ARIANE. Document interne du GETA, Grenoble, 8 p.

RIVEPIBOON W. 95, Dictionnaires Personnels Génériques: Etude et Prototypage. Thèse doctorat, GETA, IMAG, U. J. Fourier, Grenoble, 1995, 121 p.

SANFILIPPO A. 94, Word Knowledge Acquisition, Lexicon Construction and Dictionary Compilation. COLING 94, pp. 273-277.

SERASSET G. 94a, SUBLIM: Un système universel de bases lexicales multilingues et NADIA: Sa spécialisation aux bases lexicales interlingues par acception. Thèse doctorat, GETA, IMAG, U. J. Fourier, Grenoble, 1994, 194 p.

ST CLAIR B. 95, WOOD (William's Object Oriented DataBase). bill@cambridge.apple.com, 1995.

WILKS Y., FAȘS D., GUO C., MCDONALD J., PLATE T., SLATOR B. 88, Machine Tractable Dictionaries as Tools and Resources for NLP. COLING 88, pp 750-755.

ZAMPOLLI A. 91, Linguistic Tools for Multifunctional Applications in Natural Language Processing. ISCIPA'91 (International Symposium for Chinese Information Processing Application 1991, Beijing), pp. 4-21.

Extracting raw material for a German subcategorization lexicon from newspaper text

JUDITH ECKLE – ULRICH HEID

Abstract

This paper is about extracting evidence for syntactic subcategorization phenomena from German newspaper text. The purpose of this work is to support and partly automatize the construction of a subcategorization lexicon for NLP, similar, for example, to COMLEX.
We here report on the extraction of verb lists and sample sentences illustrating syntactic construction possibilities. The lists are ordered by subcategorization types; they are manually screened to remove noise, and then used to automatically produce proto-entries of the lexicon.
Since no phrasal parsing is yet available for German, we use part-of-speech shapes (a regular grammar over categorially and morphosyntactically annotated word forms) and lemma information; to reduce the noise produced by general part-of-speech shapes, we have defined "constraining contexts" and use a context-dependent modeling.

The retrieval results contain less than 5% of noise. Moreover, we can retrieve specific types of syntactic information which cannot be found in any traditional dictionary: we can, for example, identify verbs with "obligatory coherent" infinitives (cf. [Haider 1993]).

We explain the principles and procedures of our extraction work, discuss the case of infinitive-taking verbs and assess the results obtained on the first 3.000 readings extracted.

1 Introduction

1.1 Motivation and Approach

There is no freely available electronic subcategorization dictionary of German yet[1]. A typical example of this kind of dictionary is COMLEX, for English. It contains detailed syntactic information in a format which supports in principle the reformatting towards different specific representations, as they are used in computational linguistic grammar formalisms (cf. [Grishman/MacLeod/Meyers 1994], [Grishman/MacLeod 1994]). It has mostly been created manually, from machine-readable dictionaries and NLP lexicons.

The goal of the work we report on here is, however, to provide as much raw material for a subcategorization dictionary of German verbs (and later adjectives and nouns) as possible by automatic means. The dictionary construction itself still necessitates human intervention, but a large part of the preparatory work is done automatically.

For such a task, one could make use of low-level parsing (phrase types); for English and French, robust broad coverage grammars for the identification of phrasal categories are available (for example the *Fiddich* parser, cf. [Hindle 1991], or the English Constraint Grammar, cf. [Voutilainen et al. 1992]). But no similar tools are yet available for German, and parsing results must (in part) be simulated through the use of part-of-speech shapes, i.e. sequences of categorially and morphosyntactically annotated word forms. The part-of-speech shapes must be specific, to avoid too much noise in the extraction result. Thus we search for "constraining contexts", i.e. sentences where certain phrases or sequences of annotated word forms can only and unambigously be interpreted as illustrating exactly one given subcategorization pattern of a verb.

To prepare candidate verb lists, a set of queries is applied to a given text corpus; the lexicographer may select a query corresponding to a given construction[2]. The intermediate output consists of lemma frequency lists and sample sentences, both sorted by syntactic constructions (see section 3.1). The frequency lists indicate the absolute number of contexts unambigously illustrating a given syntactic construction of a given verb.
The lexicographer checks the lemma lists (by assessing the sample sentences) and decides for which lemmata a dictionary entry should be created. The resulting candidate list is input to a program which constructs proto-enties for each syntactic reading (i.e. pair of lemma and subcategorization frame).

1.2 Infrastructure

Requirements for retrieval. The extraction of linguistic and lexicographic evidence from text corpora typically leads to large quantities of material. As in Information Retrieval, one has to ensure acceptable precision (the retrieved material must be relevant for the task expressed in the query, and there should be little noise) and an acceptable recall (as much as possible of

[1] The PAROLE project, a development project financed partly by the European Commission, DG XIII E, Luxemburg, under the Language Engineering programme, will create such lexicons (of some 10.000 entries) for the major European languages.

[2] By running all available queries, evidence for the whole fragment (see below, section 4.1) can be found. Similarly, subcategorization in specialized language could be analysed, if a large enough corpus of specialized texts were available.

the relevant material contained in the corpus should indeed be found, and there should be little silence).

We opted for high precision at the price of lower recall, accepting the fact that we may exclude some relevant candidates. The reasons for this procedure are that the elimination of noise must be done manually, which is time-consuming and expensive, and that the silence produced contains a large portion of material just corroborating the facts obtained with the restrictive approach.

The information available in corpora. Raw text material usable for the acquisition of linguistic knowledge is available for many languages[3]. But the degree to which the texts can be automatically annotated differs considerably between languages: other than for English, there is no robust phrase-level parsing yet for German[4]. Thus, for our extraction work, we can only use categorial information, morphosyntactic tags[5] and lemmatization results, introduced into the texts by means of the appropriate tools.

To extract evidence for syntactic constructions, we must rely on part-of-speech shapes and on lemma information. The extraction routines basically encode a regular grammar.

Tool infrastructure. Our extraction scenario comprises two main phases. The first one is linguistic pre-processing and automatic annotation of texts (see above), the second one is corpus query.

We use the following corpus query tools[6]:

- CQP, a general corpus query processor, for complex queries with any number and combination (including negation) of annotated information types, such as word forms, part-of-speech tags, lemmas, as well as possibly sentence or phrase boundaries.

- A macro processor for the CQP query language allowing to execute the same query on elements from lists. Moreover, query expressions can be named, stored and reused.

- XKWIC, an X Windows/MOTIF-based graphical user interface for the CQP corpus query language (cf. [Christ 1994b]) which provides keyword in context concordances, and allows to automatically sort the extracted material according to user-defined context parameters; lists of absolute and relative frequency of search items can be compiled.

[3] We make use, among others, of the following German newspapers: *Stuttgarter Zeitung* (special contract), *die tageszeitung* (CD-ROM), *Frankfurter Rundschau* (from the ECI CD-ROM). The material adds up to around 200 million tokens.

[4] The SPARKLE project (an Linguistic Engineering project financed partly by the European Commission, DG XIII, Luxembourg) will produce a chunk parser for German (cf.[Rooth/Carroll 1996]) in the medium term.

[5] Provided by the STTS tagset, an EAGLES-conformant morphosyntactic annotation scheme with 54 different tags; see [Schiller/Teufel/Thielen 1995] on STTS; the tagset (Stuttgart-Tübingen Tag Set) is trivially mappable onto the EAGLES specifications for the morphosyntactic description of German, as described in [Teufel 1995]

[6] The corpus tools have been implemented in Perl, C, C++ and to some extent in UNIX-tools. TheXKWIC user interface is also based on C and C++ and integrated into an X/MOTIF environment. The corpus queries are written in the CQP corpus query language which uses the standard posix-egrep regular expression notation. For details see [Schulze 1996]

2 Principles and Method

2.1 Motivation

In figure 1[7], we give an example of a simplistic extraction scheme for transitive verb candidates, along with, in the three rightmost columns, examples of both expected results (col. 3) and noise (cols. 4 and 5).

	Fact (1)	Encoding (2)	Examples		
			(3)	(4)	(5)
a	Subord.conj.	[pos = "KOUS"]	daß	daß	daß
b	Article	[pos = "ART"]	der	die	die
c	Noun	[pos = "NN"]	Hund	Regierung	Zahl
d	Article	[pos = "ART"]	das	der	der
e	Noun	[pos = "NN"	Rennen	Forderung	Neuzulassungen
f	Verb candid.	[pos = "VVFIN"]	verlor	zustimmte	sinkt
g	within a sent.	within s	</s>	</s>	</s>

Figure 1: Search for verbs with two NPs by means of part-of-speech shapes only

The examples show that mere pos-shapes (as given in column (2)) are not apt to capture transitive verb constructions, because they are not constrained enough: (4) is an example of a verb with an indirect object (*der* in box (4d) is a dative) and (5) is an example illustrating an intransitive verb (*sinken*; the NP in (5d/e) is a genitive attribute to the subject NP (5b/c)).

The queries must be more constrained. This is achieved through two types of devices, namely the search for constraining contexts, and a context dependent modeling of phrasal constructs.

2.2 Constraining contexts

To avoid noise in the extraction results, the queries used should only match contexts which unambiguously illustrate a given subcategorization frame of a verb. This implies searching for noun phrases or components thereof which have unambiguous morphosyntactic case marks. Not all noun forms have clearly identifiable case endings (cf. *Frauen* in figure 2), but many pronouns and determiners do have such forms (example: *einigen* in figure 2).

The table 1 in appendix A contains more examples of pronouns and determiners. These alone show that there will be some silence in the query results. Mostly, we have to rely on sentences with NPs whose head nouns are masculine, because many feminine and neuter forms are ambiguous.

However, some ambiguities do not cause problems in the extraction; for example, the ambiguity between accusative and nominative is not a major problem in the extraction of transitive verb evidence (see section 3.1); similarly, when extracting two-place constructions with other complements than direct (accusative) objects, it is sufficient to describe this complement unambiguously.

[7] In this figure and in the subsequent analogous ones, we display the sentences from top to bottom; we usually give a paraphrase of the phenomenon searched (column 1), the encoding used (col. 2) and examples.

2.3 Context-dependent modeling

A number of parameters have to be kept track of to achieve a significant coverage. These include morphosyntactic properties of the verb under analysis (separable verb prefix: *die Entscheidung hängt von ihm ab* (*abhängen*); reflexive verbs: *er sorgt sich um seine Familie* (*sich sorgen*)), and more crucially, syntactic variation, such as the three different models of word order in German (see below, section 3.2). This leads to slight differences in the searchable pos-shape models for, e.g., NPs, depending on the word order model in question: an independent encoding, as in a normal analysis grammar and its reuse in all possible contexts would be more modular in design, but would either lead to much more silence (if the most restricted definitions were used) or to much more noise and ambiguous corpus samples (if more liberal definitions were used).

The sentences in figure 2 provide some illustration of this problem. The noun form *Frauen* is ambiguous with respect to case: it can have any of the four cases. The ambiguity does not cause problems when a subject NP with an intransitive verb is considered (see sentence (1)), because the context forces a nominative interpretation. It does lead to noise, however, in the extraction of two-place verbs, as illustrated by the retrieval of sentences (2) and (3), which are examples of a direct and an indirect object, respectively. Since NPs without determiner can lead to ambiguities of the kind of (2) vs. (3) with one and the same query, the queries for verbs with direct and indirect objects have been modified to include an obligatory determiner; unambiguous examples of this type are found in (4) and (5).

No.	conj	subject	complement	verb	case of compl	ambig.
(1)	weil	Frauen		kommen	nominative	(Y)
(2)	weil	er	Frauen	sieht	accusative	Y
(3)	weil	er	Frauen	vertraut	dative	Y
(4)	weil	er	einige Frauen	sieht	accusative	N
(5)	weil	er	einigen Frauen	vertraut	dative	N

Figure 2: Interaction between constraining contexts and context dependent modeling

3 The semi-automatic construction of lexicon entries – Examples

3.1 A simple example: transitive verbs

Queries. To find evidence for two-place transitive verbs, sentences containing one nominative NP and one accusative NP (both identified by the appropriate determiners) are retrieved. To reduce the amount of silence, the order of the NPs is left open by additionally allowing determiners which are ambiguous between nominative and accusative. So constituent order variation is captured, but two place predicative verbs (two nominatives: *sein*, *werden*, *heißen*, *bleiben*) must be explicitely removed from the result set. Depending on the word order type, a few additional constructions can be allowed in the NPs without introducing ambiguity (e.g. postnominal PPs in a noun phrase in the "Vorfeld" of a verb-second sentence).

Raw material: frequency tables. Figure 6 in annex B contains an extract from a frequency list of candidate verbs taking a nominative NP and an accusative NP[8]. To get a full picture of the distribution of a subcategorization scheme across a corpus, the frequency tables obtained from the analysis of different contexts (e.g. verb-first, verb-second, verb-final) need to be merged.

The frequency tables are relevant for the lexicographer, because in many cases, low frequency items include some noise. An example are verbs with subcategorized prepositional objects: usually, the most frequent preposition candidates for a given verb tend to be the prepositions governing a prepositional object ("semantically empty" complement prepositions), but the low frequencies contain adjunct prepositions[9]. For example, the frequency table for verbs with *an*-objects has *liegen an*, *erinnern an*, *glauben an*, *denken an* at the high frequency end, which all have semantically empty subcategorized prepositional objects; low frequency items include as well adjuncts such as *zerfallen (<u>an</u> der Luft)*, but also examples of prepositional objects, such as *gewöhnen an*. In such cases, the lexicographer should consult the examples and decide on the inclusion in the dictionary.

Sample sentences. Examples of sample sentences are given in figure 7, in annex B, for verbs taking a nominative and an accusative NP[10]. Repetitions in the set of samples are automatically detected: instead of displaying large amounts of analogous keyword in context (= kwic) concordances, we use a simple "condensation tool" which calculates the number of identical kwic matches and displays only one of them along with a frequency count.

Proto-entries. Once the lexicographer has decided which lemmas are admitted to the lexicon, proto-entries for the identified subcategorization readings are automatically produced. An example, illustrating the use of *fordern* with a nominative and an accusative NP, is displayed in figure 8, in annex B. Each record consists of the verb lemma ("`<verb>...</verb>`"), a subcategorization pattern corresponding to the query executed, and a set of randomly chosen example sentences[11].

3.2 Case Study: Verbs taking infinitives with *zu*

The Problem. In German, verbs taking infinitives with *zu* can occur in verb-last sentences in two different constructions:

(1) *..., weil Hans das Buch zu lesen versucht*: no extraposition [*zuinf fin*]

(2) *..., weil Hans versucht, das Buch zu lesen*: extraposition [*fin zuinf*]

[8] The frequency list refers to verb-second sentences in present tense or imperfect (without separable prefix), in 200 million words of German newspaper text.

[9] The same problem comes up here as in any descriptive linguistic work; there are no corpus-reproducible facts from where to derive any clear argument ↔ adjunct distinction. As a rule of thumb, we assume that highly frequent combinations tend to have argument status.

[10] They have been taken from the subset of a 200 million word newspaper corpus which contains subclauses in present or past tense (verb-final word order) introduced by conjunctions.

[11] If the "condensation tool" has provided frequency counts for contexts, the most frequent (most typical?) ones are selected.

In (1), the zu-infinitive comes before the finite verb, whereas in (2) extraposition of the zu-infinitive has taken place. For a number of verbs however, extraposition of the zu-infinitive is not possible[12], for example the verb *scheinen*:

(3) ..., *weil Hans das Buch zu lesen scheint*: no extraposition [*zuinf fin*]

(4) ..., *⋆weil Hans scheint, das Buch zu lesen*: extraposition [*fin zuinf*]

As the possibility to take one or the other construction, or both, seems to be a lexical property of the respective verb, an NLP lexicon has to provide information about the possible constructions for each verb taking zu-infinitives. Since in traditional dictionaries such information is not available, it is worth while to extract it from text corpora.

Extraction procedure. We look for verbs taking zu-infinitives which do not allow extraposition of the zu-infinitive, i.e. of the type [*zuinf fin*] as in (3); we expect that these do not occur in constructions of type [*fin zuinf*] (see (2) and (4)). Text corpora however do not provide negative evidence: when a certain construction does not occur in the corpus, it can not be concluded that it is not possible. What we can extract therefore from corpora are lists of candidates with the behaviour of the lexical class of obligatorily coherent verbs.

Our approach to identify verb candidates is to extract two sets of verbs together with their frequency distributions and to compare them: firstly, a set of verbs in verb-last sentences where the zu-infinitive comes before the finite verb (set *zuinf-fin* and frequency distribution *freq-zuinf-fin*, see (1) and (3)) and secondly, a set of verbs in verb-last-sentences with extraposed zu-infinitive (set *fin-zuinf* and frequency distribution *freq-fin-zuinf*, see (2)).

Then the verbs we are looking for should occur only in set *zuinf-fin* and not in set *fin-zuinf*. A very simple method to get these verbs would be to compute the set of verbs which are only in *zuinf-fin* and not in *fin-zuinf*. But this does not take into account that the sets of extracted verbs could contain noise resulting for instance from tagging errors. So this simple method fails, for example when set *fin-zuinf* contains the verb *scheinen* because of a single occurrence of this verb in a misclassified context.

Therefore we decided to compare the frequency figures of the two sets rather than the verb sets themselves: a verb in *zuinf-fin* is a successful candidate, when its frequency in *freq-zuinf-fin* is high compared with its frequency in *freq-fin-zuinf*. This reflects the idea that a big difference between the two frequencies of a given verb indicates a tendency of this verb to prefer one context to the other. Provided that there is little noise in set *fin-zuinf*, the method of comparing the frequency distributions should work very well.

One way of implementing the comparison is to check for each verb in set *zuinf-fin* whether the quotient $\frac{\text{freq. in set } \mathit{freq\text{-}fin\text{-}zuinf}+1}{\text{freq. in set } \mathit{freq\text{-}zuinf\text{-}fin}+1}$ is suffciently small; as bias we experimentally chose 0,02. A lower value of the bias leads to more silence, whereas a higher value results in less silence, but possibly more noise: then we might also get verbs which actually do allow extraposition of the zu-infinitive, and which are simply not frequently used in the text corpus.

Linguistic queries. For the extraction of the two verb sets, two query templates have been designed, which are illustrated in Figures 3 and 4, respectively. The purpose of the query templates is to find verbs which subcategorize only for a subject and an infinitival complement with *zu*.

Fact	Encoding	Examples	
subord. conj.	[pos = “KOUS”]	als	weil
item sequence: no verb no punctuation no “es”; up to 12 items	 [POS ≠ “V.*” & POS ≠ “IP.*” & word ≠ “es”] {1,12}	 Maria gestern Birnen	 Hans heute Äpfel
“zu”	[word = “zu”]	zu	zu
infinitive	[pos = “V.INF”]	kaufen	verkaufen
finite verb	[pos = “VVFIN”]	**versuchte**	**scheint**
within a sentence	within s		

Figure 3: Query for set *zuinf-fin*: find verbs taking subject and infinitival complement with *zu* in contexts with the finite verb following the zu-infinitive.

The query template for set *zuinf-fin* is designed to collect as many verbs as possible. Therefore verb complements are modelled indirectly by excluding certain categories like verbs and punctuation marks. In this context (zu-infinitive precedes the finite verb), complements of the zu-infinitive can not be distinguished from complements of the finite verb by means of POS-shapes. Hence, the resulting verb set will possibly contain misclassified verbs, which take not only a subject and a zu-infinitive, but also a direct or indirect object.

The query template for set *fin-zuinf* on the other hand, is designed to avoid as much noise as possible. Here we try to identify only verbs taking subject and zu-infinitive but no other complements. We do this by explicitely modelling the subject noun phrase in the matrix clause with a complex NP-POS-shape[13]. Constraing the context in this way minimizes noise in *freq-fin-zuinf*, which is important for the subsequent comparison of the frequency distributions.

Results and Evaluation. We applied the query templates to a corpus of German newspaper text of about 200 million tokens. One important result of the extraction experiment is that there are only very few obligatorily coherent verbs, i.e. which do not allow extraposition of the zu-infinitive. After automatically comparing the frequency distributions of the two verb sets, a list of 11 verbs remained.

The list has been manually checked to identify and remove misclassified verbs and to test for the other verbs, whether extraposition of the zu-infinitive is possible[14]. From the remaining 8 verbs, only one, namely *verstehen*, actually does allow extraposition of the zu-infinitive. Figure 5 in annex A shows the tested verbs together with evidence phrases and (manually made-up) test phrases.

[12] These verbs are often called ‘obligatorily coherent verbs’, see [Haider 1993].

[13] Furthermore, as finite verbs we exclude verbs of existence like *bestehen, bleiben.* These verbs tend to occur with nouns taking zu-infinitives, such as *weil die* Möglichkeit *besteht, ein Buch zu lesen.*

[14] By looking at the automatically collected evidence phrases from set *zuinf-fin*, we found 3 misclassified verbs, which subcategorize not only for subject and zu-infinitive, but also for a direct or indirect an object.

Fact	Encoding	Examples	
subord. conj.	[pos = "KOUS"]	als	weil
NP	complex NP-POS-shape	Maria	ein kleines Mädchen
adverbs	exclusion of non-adverb categories	gestern	heute
finite verb, but not: v. of ex.	[pos = "VVFIN" & !file(verbs-of-existence)]	versuchte	versucht
comma	","		
no verb no punctuation no "es"	[POS ≠ "V.*" & POS ≠ "IP.*" & word ≠ "es"]	Birnen	Äpfel
"zu"	[word = "zu"]	zu	zu
infinitive	[pos = "V.INF"]	kaufen	verkaufen
within a sentence	within s		

Figure 4: Query for set *fin-zuinf*: find verbs taking subject and infinitival complement with *zu* in contexts with the zu-infinitive following the finite verb.

4 Assessment

4.1 Fragment

The set of queries for subcategorization extraction is still under construction. As of early July 1996, the following types of constructions and their combinations can be extracted from German texts:[15]

- verbs with subject only (intransitive);
- verbs with subject and accusative and/or dative NP complement;
- verbs with subject (optional complement) and correlate construction (pointing to a prepositional object);
- verbs with subject (optional complement) and *zu*-infinitive, or complement clause with complementizer *daß* or *wh*-words.

Currently, some 3.000 verb readings have been extracted and validated. The noise rate is relatively low: on 1325 candidate verbs for the pattern "verb<[NP-NOM][NP-ACC]>", 57 items (= less than 5%) have been identified which do not qualify as dictionary-relevant.

4.2 Linguistic problems – possible solutions

The approach has a number of limitations, some of which are inherent to the use of a regular grammar. Moreover, the automatically tagged material contains the usual percentage of errors.

[15] As stated above, we have to keep track of word order variation, active/passive, complex tense, and morphosyntactic properties of the verbs (reflexive, separable prefix, etc.), as well as of combinations of these.

The procedures only allow to find the constructions we search for; the approach is dependent on the model of subcategorization classes used, and on the presence, for each class, of a discovery procedure.

A major limitation of the extraction devices is due to the use of a regular grammar: only sequences of phrase structural constructs can be identified, and no inference about grammatical functions is possible whenever the relationship between the pair of <phrasetype, case> and the grammatical function is not 1:1. Thus transitive (passivizable) verbs like *kaufen* and verbs taking a circumstantial complement (duration: *dauern - die Sitzung dauert eine Stunde*; weight: *wiegen - er wiegt 100 Kilogramm*; etc.) or an adverbial (*er arbeitet jeden Tag, er kauft eine Menge Waren*) are extracted by the same routine and need to be separated out manually[16].

Similar problems, well known from any theoretical work on valency dictionaries, concern the distinction between indirect objects and free datives, and, between complement and adjunct prepositional phrases[17].

5 Future work

Current work is aimed at completing the fragment coverage. In addition, work on noun and adjective subcategorization has started. For example, material for prepositional attributes of nouns has been extracted (*Freude auf...*, *Interesse an...*, etc.).

In a parallel strand, the corpus exploration tools will be used to validate data from machine-readable dictionaries in text corpora: the subcategorization information contained in an electronic dictionary will be used to parameterize queries for individual verbs. For each subcategorization indication from the dictionary, corpus evidence will be sought. Dictionary indications not documentable with corpus data will be manually assessed.

Another important dimension to follow is some sort of semantic clustering of the results. The entry in figure 8 clearly shows the need for this, since it contains at least two readings of the verb, one as a speech act, and one as an abstract collocate (*das Feuer fordert ein Todesopfer.*) A combination of our approach with one that allows for statistical clustering of heads of verb subjects and complements seems most promising.

References

[Abney 1991] Steven Abney: "Parsing By Chunks", in: Robert Berwick, Steven Abney and Carol Tenny (Eds.): *Principle-Based Parsing*, (Dordrecht: Kluwer Academic Publishers), 1991

[Christ 1994a] Oliver Christ: "A Modular and Flexible Architecture for an Integrated Corpus Query System", in: Ferenc Kiefer, Gábor Kiss, Júlia Pajzs (eds.): *Papers in Computational Lexicography, COMPLEX '94*, Budapest, 1994, pp. 23-32.

[Christ 1994b] Oliver Christ: "The XKwic User Manual", internal report, Stuttgart: IMS, 1994.

[16] A subset of the passivizable verbs could be identified automatically: those actually occurring in the passive in the corpus. Verbs taking a "theme" and an "experiencer" (*Die Frage interessiert ihn*) are also in the noise set.

[17] See above, section 3.1. Frequency can help somewhat with PPs: not only the relative frequency of a single <[NP-NOM][PP]>-construction is relevant, but also the range of prepositions found along with a given verb, and the relative importance of the prepositions of each verb.

[Grishman/MacLeod/Meyers 1994] Ralph Grishman, Catherine MacLeod, Adam Meyers: *Comlex Syntax: Building a Computational Lexicon*, (New York: New York University), 1994.

[Grishman/MacLeod 1994] Ralph Grishman, Catherine MacLeod: *COMLEX Syntax Reference Manual Version 1.1*, Draft prepared for the Linguistic Data Consortium, University of Pennsylvania, 1994.

[Haider 1993] Hubert Haider: *Deutsche Syntax - generativ: Vorstudien zur Theorie einer projektiven Grammatik.* Gunter Narr Verlag, Tübingen, 1993.

[Hindle 1991] Donald Hindle: "Structural Ambiguity and Lexical Relations", in: *Proceedings of the 29th Annual Meeting of the ACL*, 1991: 229-236

[Rooth/Carroll 1996] Mats Rooth, Glenn Carroll: "Valence Induction with a Head-Lexicalized CFG", (Stuttgart: IMS), ms. 1996

[Schiller/Teufel/Thielen 1995] Anne Schiller, Simone Teufel, Christine Stöckert, Christine Thielen: "Vorläufige Guidelines für das Tagging deutscher Textcorpora mit STTS", Stuttgart/Tübingen, 1995.

[Schulze 1996] Bruno Maximilian Schulze: *MP user manual*, Stuttgart: IMS, 1996.

[Teufel 1995] Simone Teufel: *ELM-DE: A typed incarnation for German of the EAGLES Standard Proposal for Morphosyntactic Annotation – Lexical Specification and Classification Guidelines*, (Stuttgart/Pisa: IMS/EAGLES) 1995, ms. 172 pp. See also the electronic version, on the URL of the EAGLES project: `http://www.ilc.pi.cnr.it/EAGLES/home.html`

[Voutilainen et al. 1992] Atro Voutilainen, Juha Heikkilä, Atro Anttila: "Constraint Grammar of English: A Performance-Oriented Evaluation", Publication No. 21, University of Helsinki, Department of General Linguistics, 1992.

A Appendix: Data

Det/Pron	Case	Examples
Det: article	nom/acc	*das, die. ein*
Det: demonstr.	nom/acc	*diese, jene, derselbe, derjenige, dasjenige ...*
Det: indef.	nom/acc	*irgendein, irgendeine, alle, jede, kein, manche, ...*
Det: article	dat	*dem, einem.*
Det: demonstr.	dat	*diesem, jenem, ...*
Det: article	gen	*des, eines.*
Det: demonstr.	gen	*desselben, desjenigen, ...*
Pron: pers.	nom	*ich, du, wir.*
Pron: demonstr.	nom	*derselbe, derjenige,*
Pron: indef.	nom	*man, jemand, niemand, irgendwer.*
Pron: pers.	acc	*ihn*
Pron: demonstr.	acc	*den,*
Pron: indef.	acc	*irgendwen, jeden, niemanden, ...*

Table 1: Determiners and Pronouns with unambiguous morphosyntactic case forms

Verb	Evidence phrase	Test phrase
brauchen	daß er nichts ernstzunehmen braucht	⋆ daß er braucht, nichts ernstzunehmen
pflegen	daß sie ihre Kritiker zu überleben pflegt	? daß sie pflegt, ihre Kritiker zu überleben
scheinen	obwohl er nach hinten zu kippen scheint	⋆ obwohl er scheint, nach hinten zu kippen
suchen	als ein Lkw eine Straßensperre zu durchbrechen suchte	? als ein Lkw suchte, eine Straßensperre zu durchbrechen
trachten	obwohl die Regierung dies zu verhindern trachtete	? obwohl die Regierung trachtete, dies zu verhindern
vermögen	daß man etwas zu leisten vermag	? daß man vermag, etwas zu leisten
verstehen	der mit Sprache umzugehen versteht	der versteht, mit Sprache umzugehen
wissen	daß sie Risiken abzuschätzen weiß	? daß sie weiß, Risiken abzuschätzen

Figure 5: The tested verb candidates from the resulting verb list.

B Examples of results

sehen	463
kennen	380
wissen	344
machen	269
brauchen	244
tun	187
finden	147
verstehen	145

Figure 6: Part of the frequency distribution of verb lemmas in 200 million words, with the subcategorization pattern *verb*<[NP-NOM][NP-ACC]>

```
Die überschätzen ihre eigene Kraft sehr .
die Bahn übersehe hier die topographische Lage .
Die Sensoren übersetzen die Bewegungen .
Ein Dolmetscher übersetzte die Vernehmung .
Das Frisierstübchen übersieht man fast .
Die Bilanzsumme übersprang die Acht-Milliarden-Mark-Grenze .
Er überstand die Vertrauensabstimmung unbeschadet .
Die Inszenierung übersteht gerade mal die Premiere .
Die gemessene Radioaktivität übersteige nicht die zulässige Norm .
```

Figure 7: Sentences illustrating verbs prefixed with "*über*"
which subcategorize for a nominative and an accusative NP

```
< record >   < verb > fordern < /verb >
< subcat > subj(NP_nom) obj(NP_akk) < /subcat >
< typical >
Aber niemand fordert ihre Legalisierung .
Auch die Seeleutegewerkschaft fordert ein umfassendes Waffentransportverbot .
Auch die afghanische Nachbarregierung fordert ihre Freilassung .
Das Büro fordert nun die Rekonstruktion des Kunstwerks .
Das Feuer fordert ein Todesopfer :
Das fordere auch nicht das Sozialstaatsprinzip .
Das fordern die niedersächsischen Christdemokraten .
< /typical >
< /record >
```

Figure 8: A sample proto-entry for *fordern*<[NP-NOM][NP-ACC]>

B Examples of results

schon	463
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]

Figure B.[illegible] [illegible]

[illegible] diese Argumente [illegible]

[illegible]

Die [illegible] Korrektion [illegible]

[illegible]

Das [illegible]

[illegible]

[illegible]

[illegible]

[illegible]

Figure B.[illegible] [illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

COMPASS - An Intelligent Dictionary System for Reading Text in a Foreign Language

HELMUT FELDWEG - ELISABETH BREIDT

Abstract

This paper presents a system which assists in the comprehension of on-line texts in foreign languages by utilising a context-sensitive dictionary look-up system. The system employs state-of-the-art finite state technology for the context analysis and uses converted bilingual print dictionaries as its primary resource of lexical information. A comprehenision-specific view of the conventional print dictionaries was made available for the system by analysing the dictionary structure and parsing the typesetting tapes. The lexical information was validated and augmented by corpus-based lexicographic revision, and was finally formalised to meet the requirements of an NLP system. An estimation of the efforts required to prepare full dictionaries for the system is given.

1 Introduction

Electronic dictionaries have substantially simplified the time-consuming task of looking up words. This is particularly true when the text to be read is in electronic form, a reading situation that is becoming more and more significant with the increasing spread of computer networks and electronic books and documents. However, at present neither electronic dictionaries themselves, nor the look-up techniques, are well suited to what is possible within an electronic medium. Most of the electronic dictionaries are simply an electronic copy of the conventional paper dictionary they are based on. The look-up process is mostly limited to match a given word-form in the text with the set of headwords contained in the dictionary and presenting the full dictionary entry to the user when a match could be found. In such a configuration, most of the intellectual tasks of the look-up process are still to be performed by the human user: the user is left with relating inflected forms to their base, identifying part of speech, and picking out the appropriate sense somewhere in an extended dictionary entry.

Furthermore, the possibility of offering the user a task–specific view on a standard multi–functional dictionary is not exploited by existing systems.

The COMPASS project[1] has demonstrated that these restrictions on conventional electronic dictionaries can be overcome by the application of existing NLP techniques. To this end, a prototype of a computer program was developed, which accesses enhanced and structurally elaborated dictionaries with an intelligent, context–sensitive look–up procedure and presents the information to the user by means of an attractive graphical interface.

The prototype's performance was evaluated through a series of user tests. These have given rise to quite ringing endorsements of the system by the test users. The results show that reading foreign–language texts is substantially easier with a system such as COMPASS, and a better understanding of the text can be gained (see Ostler 1996). In fact, we believe that in many cases where the reader already has a basic knowledge of the foreign language usage of such a system can obviate the need for translation.

2 Mono–functional view of a multi–functional dictionary

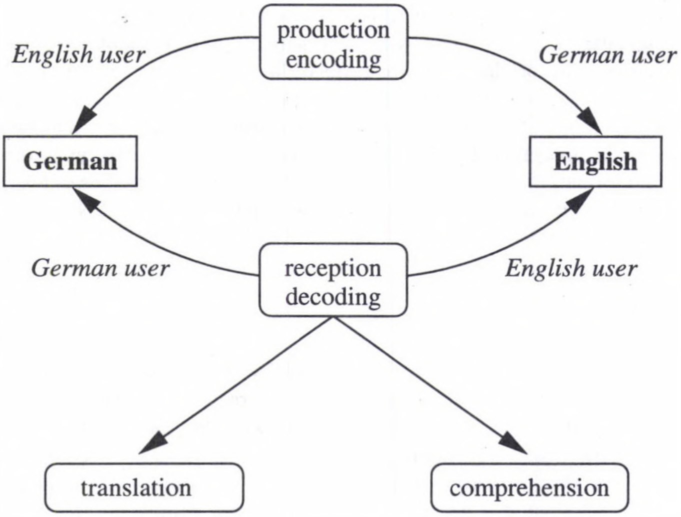


Figure 1: Multi–functionality of a conventional bilingual dictionary

[1] The project was funded by the European Communities within the *Linguistic Research and Engineering* Programme under LRE 62–080 from April 1994 to April 1996. The five partners of the project consortium were Rank Xerox Research Centre, Grenoble, (co–ordinator); Seminar für Sprachwissenschaft, Universität Tübingen, Germany; Fraunhofer Institut für Arbeitswirtschaft und Organisation, Stuttgart, Germany; Langues étrangères appliquées, Université Lumière, Lyon, France; Dept. of Marketing, Advertising and Public Relations, Bournemouth University, UK.

task	sub–task	source–target language		
encoding		German	→	English
encoding		English	→	German
decoding	translation	German	→	English
decoding	comprehension	German	→	English
decoding	translation	English	→	German
decoding	comprehension	English	→	German

Table 1: Functions of conventional bilingual dictionary

Most conventional bilingual dictionaries are multi–functional in the way they offer information for the encoding and decoding tasks to native– and non–native speakers of the two languages involved. Figure 1 illustrates the various functions of a standard bilingual dictionary, exemplified for the language pair German–English. Combinatorics of the two variables *task* and *source–target language pair* render four uses of a standard bilingual dictionary, and, if the decoding task is subdivided further into a translation and a comprehension task, the dictionary yields the six functions shown in table 1. While this multi–functionality has the advantage that one and the same dictionary offers information to support a number of different tasks, a dictionary user has to select those items from the dictionary entries that are relevant to the task she is performing.

The COMPASS system is an attempt to implement one task–specific view of a full standard bilingual dictionary as a prototype system, namely reading comprehension of German texts for English speakers, and of English texts for French speakers. While the full text of the dictionaries is held in the background (and available to the user on request), the system first presents of all only those lexicographic items to the user that have been selected as relevant for the comprehension task, suppressing irrelevant information and thereby distracting the user as little as possible from the main task being performed.

3 Lexicographic backbone

The lexicographic basis for the project is supplied by the Collins German Dictionary (German–English) and the Oxford–Hachette French Dictionary (English–French). Machine–readable versions of these dictionaries were licensed to the partners in the project for research purposes. With these two dictionaries the prototype is able to cover the English–French and German–English language pairs. In keeping with the terms of the licence for the dictionaries, and in order to make effective use of limited staff time, only excerpts from these dictionaries were used for the prototype system. At the same time, an assessment of the effort for the conversion of full dictionaries was made by applying the technical adaptions to the full text of one language direction per dictionary and projecting the costs for intellectual lexicographic enhancements from the excerpt data prepared for the prototype system.

3.1 Technical Adaptation of the Dictionaries

The machine–readable versions of the dictionaries provided by the publishers were a rudimentary tagged version of the typesetting tape for the Collins German Dictionary and a SGML–marked text for the Oxford–Hachette French Dictionary. Although the tagged version of the German dictionary had most of the lexicographic items marked as such, the entries needed to undergo a thorough structural analysis in order to enable selective access to the information in the dictionary entries. For this the dictionary parser LEXPARSE (cf. Hauser & Storrer 1994) was used, which can recognise, and explicitly represent, the hierarchical micro–structure of dictionary entries using a grammar defined by the user.

The French dictionary was made available in a SGML–tagged format, though without a corresponding DTD (document type definition). However, the type and amount of tagging of the French dictionary turned out to be not sufficient for our purposes and the dictionary had to be re–tagged much the same way as the German dictionary.

The LEXPARSE grammars developed for the two dictionaries cover as comprehensively as possible all the structures of the dictionary entries, excluding inconsistent and faulty entries: these make up a considerable part of the dictionary. The faulty entries were corrected manually and parsed a second time. The resulting SGML–annotated dictionaries together with the DTD generated by LEXPARSE could then be lexicographically adapted in an SGML editor.

Partly during the parsing, partly during the subsequent processing, some unpacking of, and corrections to, the mark–up were introduced. To create an index for the look–up system it was necessary to spell out lemma–variants and expand sub–entries. For the most part these tasks were performed automatically. Finally, the two resulting *lexical databases* derived from each dictionary were converted into a common data structure used by the LOCOLEX look–up system.

3.2 Extensions to the Dictionaries

To make true *comprehension dictionaries* from the parsed dictionaries, various lexicographical adjustments were necessary. All information in an entry that is unnecessary for the understanding of the word has to be marked explicitly for suppression in the COMPASS system. For example:

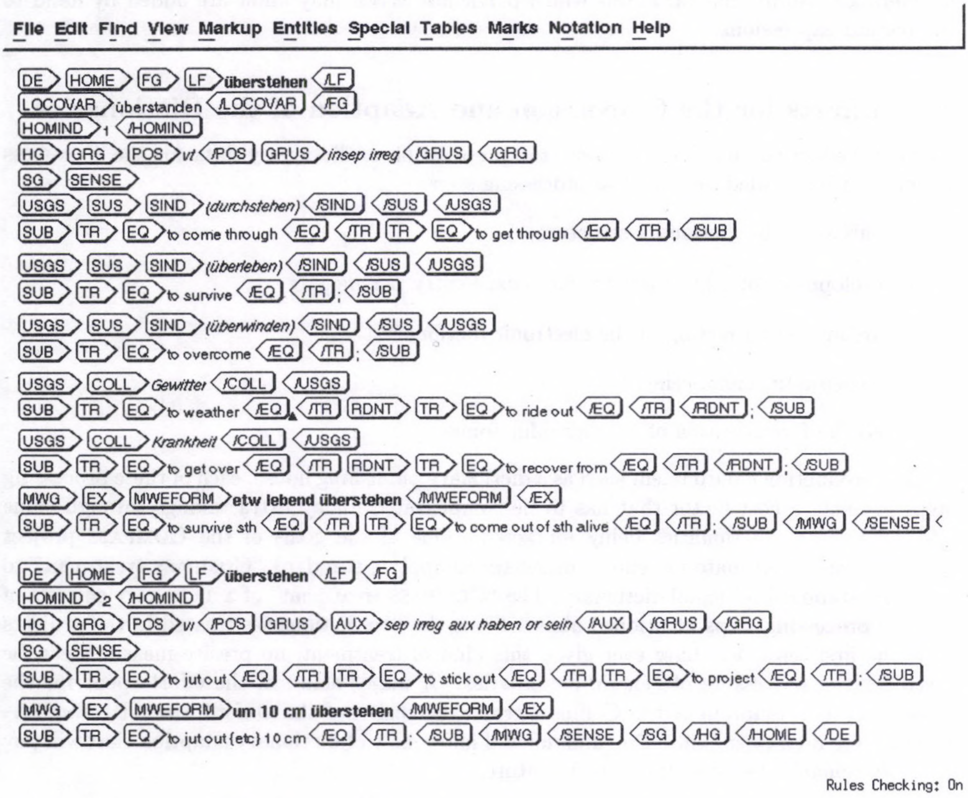
- Explicit marking of alternative, almost synonymous translations; e.g. the complex translation equivalent *to switch or turn or put on* for *einschalten* is transformed into three simple translations and marked up as such allowing COMPASS to hide the second and third translation variants.

- Using different tags to distinguish usage examples, which are only important for language production, from semantically complex multi–word lexemes, which can only be understood as a whole. (Only the latter should appear in a comprehension dictionary.)

- Separate marking of prepositional complements when they appear within the translation equivalent.

Further unpacking was sometimes necessary, e.g. to supply explicit translations where for reasons of space only implicit example phrases are given. Of course we also needed to supply missing variant forms, missing senses, completely absent headwords and multi–word expressions (MWE), the latter discovered from corpus excerpts and the automatic extractions of possible MWE from textual corpora.

An example of an SGML–tagged dictionary entry adapted for the COMPASS system is given in figure 2.


File Edit Find View Markup Entities Special Tables Marks Notation Help

```
<DE> <HOME> <FG> <LF>überstehen </LF>
<LOCOVAR>überstanden </LOCOVAR> </FG>
<HOMIND>1 </HOMIND>
<HG> <GRG> <POS>vt </POS> <GRUS>insep irreg </GRUS> </GRG>
<SG> <SENSE>
<USGS> <SUS> <SIND>(durchstehen) </SIND> </SUS> </USGS>
<SUB> <TR> <EQ>to come through </EQ> </TR> <TR> <EQ>to get through </EQ> </TR>; </SUB>
<USGS> <SUS> <SIND>(überleben) </SIND> </SUS> </USGS>
<SUB> <TR> <EQ>to survive </EQ> </TR>; </SUB>
<USGS> <SUS> <SIND>(überwinden) </SIND> </SUS> </USGS>
<SUB> <TR> <EQ>to overcome </EQ> </TR>; </SUB>
<USGS> <COLL>Gewitter </COLL> </USGS>
<SUB> <TR> <EQ>to weather </EQ> </TR> <RDNT> <TR> <EQ>to ride out </EQ> </TR> </RDNT>; </SUB>
<USGS> <COLL>Krankheit </COLL> </USGS>
<SUB> <TR> <EQ>to get over </EQ> </TR> <RDNT> <TR> <EQ>to recover from </EQ> </TR> </RDNT>; </SUB>
<MWG> <EX> <MWEFORM>etw lebend überstehen </MWEFORM> </EX>
<SUB> <TR> <EQ>to survive sth </EQ> </TR> <TR> <EQ>to come out of sth alive </EQ> </TR>; </SUB> </MWG> </SENSE> <

<DE> <HOME> <FG> <LF>überstehen </LF> </FG>
<HOMIND>2 </HOMIND>
<HG> <GRG> <POS>vi </POS> <GRUS> <AUX>sep irreg aux haben or sein </AUX> </GRUS> </GRG>
<SG> <SENSE>
<SUB> <TR> <EQ>to jut out </EQ> </TR> <TR> <EQ>to stick out </EQ> </TR> <TR> <EQ>to project </EQ> </TR>; </SUB>
<MWG> <EX> <MWEFORM>um 10 cm überstehen </MWEFORM> </EX>
<SUB> <TR> <EQ>to jut out {etc} 10 cm </EQ> </TR>; </SUB> </MWG> </SENSE> </SG> </HG> </HOME> </DE>
```

Rules Checking: On

Figure 2: A (shortened) sample dictionary entry after COMPASS treatment

3.3 Formalisation of Context Patterns

The COMPASS system should recognise whether a queried word occurs in a specific context where a special translation is appropriate, and in that case select it. To make this possible, corresponding context patterns must be supplied in the COMPASS dictionary. For this purpose Rank Xerox has developed a finite state formalism and a compiler which allows coding of such context patterns as regular expressions. The context formalisation is restricted initially to the recognition of MWE and grammatical collocations.[2]

[2] See Breidt et al. 1996 for a more detailed description of this formalism and how to use it to encode MWEs.

The formalisation is achieved through a number of steps. First the decision is made as to which contexts overall should be formalised. MWE and grammatical collocations are then transformed into a so-called *canonical* form, which also includes lexical variants. Morphologically variable elements are marked as such. On the basis of these canonical forms a regular expression is generated, which encompasses, for example, the standard word-order variations for German. Additional variations which particular MWE may allow are added by hand to the regular expressions.

3.4 Efforts for the Conversion and Adaption of the Dictionaries

As described in the previous sections, the preparation of dictionary data for the COMPASS system can be divided into the five processing steps:

- analysis of the dictionary structure,
- development of a grammar for dictionary entry parsing,
- parsing and correcting of the electronic dictionary,
- lexicographic enhancement,
- and the formalisation of lexicographic items.

In a production environment such as a dictionary publishing house, each of these processing steps presents a cost factor that has to be compensated for by extra takings through value additions to the dictionaries being marketed. One of the goals of the COMPASS project therefore was to estimate the efforts necessary to apply a standard "COMPASS treatment" to a generic standard bilingual dictionary. The "COMPASS treatment" of a dictionary consists of the five processing steps mentioned above. Since the two dictionaries treated in the projects were the first ones that have ever given this kind of treatment, no precise measures can be given for the processing of generic dictionaries. A major share of the effort spent for the conversion and adaption of the Collins German–English and the Oxford–Hachette English–French were investments into the fundamental know-how of dictionary adaption that will pay off in dictionaries being adapted in the future.

Despite missing data for exact calculations for the efforts of a COMPASS treatment, the project has shown that:

Dictionary analysis is a matter of weeks for an experienced lexicographer. Efforts spent for this task can almost be minimized if a detailed structural description of the dictionary entries is available from the dictionary editor, or if the person analysing the dictionary and the dictionary editor are one and the same.

Development of a dictionary grammar is also a matter of weeks, given a fully elaborated analysis of the dictionary. The focus of this task lies in the formalisation of the preceding dictionary analysis task. Therefore, the effort for this task is a function of the degree of formalisation of the preceding step and depends on the available skills to write formal context-free grammars. The implementation of the grammar implies an analysis of the typesetting tape for the consideration of structural markers (e.g. font types, punctuation) in the formal dictionary analysis to allow for segmentation of dictionary entries into unambiguous lexicographic items.

Parsing and correction is a linear function of the size of the dictionaries and the portion of dictionary entries not conforming to the dictionary entry structure (ill-formed entries). State-of-the-art dictionary entry parsers such as the LEXPARSE system used in the COMPASS project achieve a throughput of approx. 10,000 entries per hour, parsing a dictionary of the size of the full German-English dictionary in about one day's time. More time-consuming is the analysis of parsing errors to detect inconsistencies in the typesetting tapes and shortcomings of the dictionary grammar and the following correction of the tape or grammar. About 900 of the approx. 84,500 entries of the Collins German-English dictionary were found to be ill-formed and had to be corrected this way. If we allow only 5 Minutes per error for its detection in a possibly large and deeply structured entry and correction in a huge typesetting file, this amounts to 75 working hours for all 900 errors to be corrected, re-parsing not included (cf. Thielen & Breidt, 1996).

Lexicographic enhancement is the part of the dictionary adaption process most difficult to measure. The efforts here are comparable to those necessary to update and improve the quality of a dictionary in general, esp. if electronic text corpora are evaluated. Attempts were made to use time-stamps in the parsed dictionary files to measure the time spent on post-editing the dictionaries, but this device was to cumbersome to record each minor change made to an entry. Besides, some of the overall time spent on editing was necessary to develop the editing methods for the lexicographic enhancements and for corpus consultations.

Formalisation as the last step involved in the adaption process can be supported by means of computer programs to some extent, especially if standardisation of lexicographic items has been a main target of the preceding lexicographic enhancement process. With standardised lexicographic items (esp. with respect to variant forms, scope markers, and canonical representation of multi-word expressions), the formalisation process can be supported substantially by the development of software tools to convert standardised items into the format required by the COMPASS system, though a certain amount of manual and intellectual support is unavoidable. With such tools, the formalisation of contextual patterns for a big, one-volume bilingual dictionary shouldn't take more than six to ten person-monhts.

To summarise, the effort for the COMPASS treatment of an existing dictionary is highly dependent on the level of structuring and the structural integrity of the source material supplied by the dictionary publishers. As has been shown for the case of the Oxford-Hachette French dictionary, the fact that a dictionary text is available as an SGML-tagged document is of little help if the level of SGML mark-up does not match the requirements of the COMPASS system. But even in this case, a minimal COMPASS treatment can be given to a full bilingual dictionary within a couple of months, if no extended lexicographic enhancements are required and if a small amount of errors in the mark-up and formalisation is acceptable.

4 The Locolex Look–up System

The basis of the look–up system is the LOCOLEX system, developed and patented by Rank Xerox. On the basis of a linguistic analysis of the word's environment LOCOLEX controls the actual look–up and indicates the relevant parts of a dictionary entry to be presented by the interface. To speed access to individual dictionary entries it uses an index of headwords and their variants. The LOCOLEX software is largely system–independent. It can be developed on, and ported onto, a variety of computer architectures.

The components for linguistic analysis of the source language (the so–called *language model*) are not a direct part of the LOCOLEX kernel. Language models are developed separately for each language required and attached to the LOCOLEX kernel as finite automata. Among the most important components of a language model are algorithms for morphological analysis and identification of parts of speech. Over and above these, the language model includes definitions of the macros and variables for finite automata which are used to recognise multi–word expressions, rules to deal with regular spelling variations, and a table to translate the labels used for part of speech categories by the morphological analyser to those used in the dictionary.

The morphological analysis reduces inflected words to their base–form and thus allows to access dictionary entries from inflected words (e.g. of *gesungen* to the headword *singen*). It also provides morpho–syntactic information (part of speech, case, number and gender) which is used in subsequent steps of the analysis to select the correct meaning.

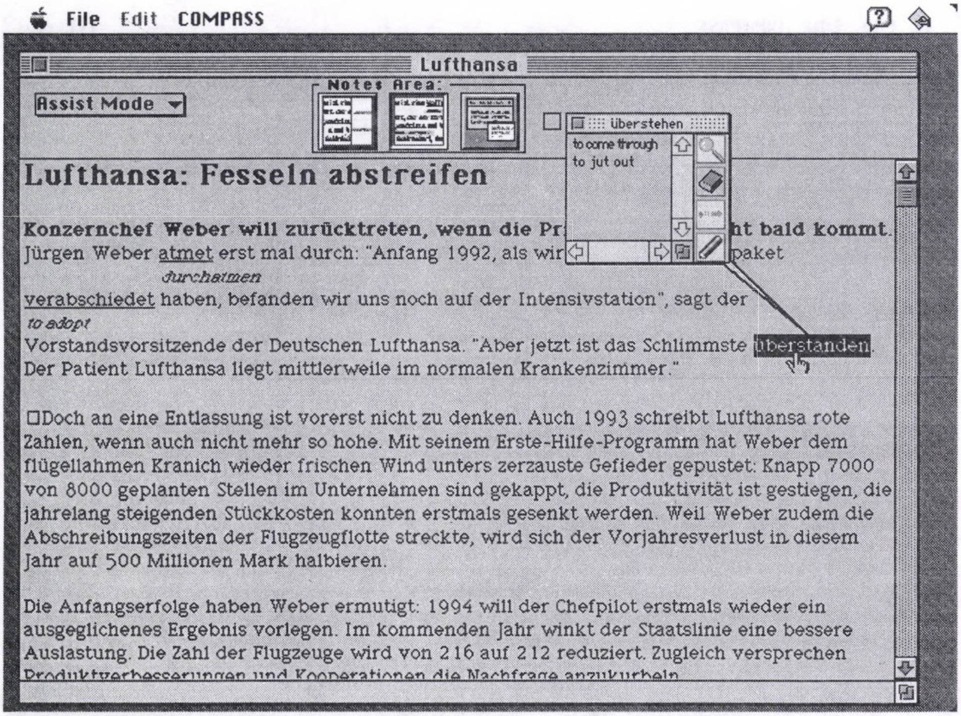
If morphological analysis results in ambiguous syntactic information (e.g. article, pronoun or verb for *einen* in German, noun or verb for *plan* in English) this ambiguity is resolved by a part of speech disambiguation component. This uses a probabilistic procedure known as a Hidden Markov Model. These components are especially important for English or French where many content words are ambiguous as to their part of speech.

The output of morphological analysis and part of speech disambiguation is used to select the parts of a dictionary entry relevant to a given context. The complete dictionary entry is loaded into main memory via an index. This procedure converts the given SGML structure of the dictionary entry into a largely dictionary–independent, system–internal data structure, and the part selected by the disambiguation is specially marked.

If the selected word is part of a multi–word expression and coded as such in the dictionary entry, the system returns the translation of the whole MWE. This is a further step towards selecting the information relevant to the context from the dictionary entry. For this, the MWEs coded as regular expressions in the selected dictionary entry are compared with the input text. If a regular expression matches the sentence context, the translation of the corresponding MWE is marked specially and displayed first to the user as an answer to her query.

5 Graphical User Interface

For the representation of texts and dictionary entries a special graphical user interface has been developed for Apple Macintosh computers (Wetzel 1996). The kernel of this user interface is a so–called *reader*, a simple editor program that permits the formatted display of HTML–marked texts, and annotation of individual words with translations, but also changes to the text itself. A look–up and analysis process can be activated by simple selection of a word with



überstehen

to come through
to jut out

Lufthansa: Fesseln abstreifen

Konzernchef Weber will zurücktreten, wenn die Pr ht bald kommt.
Jürgen Weber atmet erst mal durch: "Anfang 1992, als wir paket
durchatmen
verabschiedet haben, befanden wir uns noch auf der Intensivstation", sagt der
to adopt
Vorstandsvorsitzende der Deutschen Lufthansa. "Aber jetzt ist das Schlimmste überstanden.
Der Patient Lufthansa liegt mittlerweile im normalen Krankenzimmer."

□Doch an eine Entlassung ist vorerst nicht zu denken. Auch 1993 schreibt Lufthansa rote Zahlen, wenn auch nicht mehr so hohe. Mit seinem Erste-Hilfe-Programm hat Weber dem flügellahmen Kranich wieder frischen Wind unters zerzauste Gefieder gepustet: Knapp 7000 von 8000 geplanten Stellen im Unternehmen sind gekappt, die Produktivität ist gestiegen, die jahrelang steigenden Stückkosten konnten erstmals gesenkt werden. Weil Weber zudem die Abschreibungszeiten der Flugzeugflotte streckte, wird sich der Vorjahresverlust in diesem Jahr auf 500 Millionen Mark halbieren.

Die Anfangserfolge haben Weber ermutigt: 1994 will der Chefpilot erstmals wieder ein ausgeglichenes Ergebnis vorlegen. Im kommenden Jahr winkt der Staatslinie eine bessere Auslastung. Die Zahl der Flugzeuge wird von 216 auf 212 reduziert. Zugleich versprechen Produktverbesserungen und Kooperationen die Nachfrage anzukurbeln

Figure 3: The COMPASS user interface with reduced dictionary entry

the mouse. A small help window appears, placed close to the selected word so as to cover as little as possible of the context. The window displays a list of the translations that appear relevant in the light of the analysis of the context (cf. figure 3).

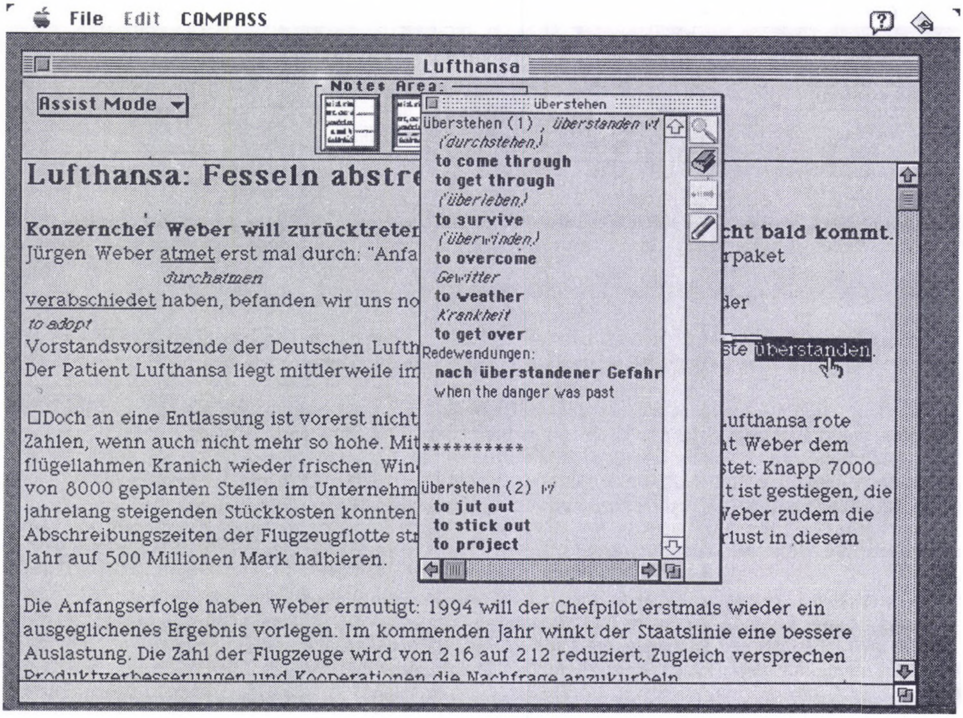
The user is offered various options in the help window: with the pencil icon the user can annotate the word in the text with one of the translations. If the user desires further information on an individual sense it can be displayed by selecting this sense and the magnifying lass icon. Finally, the whole dictionary entry can be displayed through clicking on the book icon (cf. figure 4). If the user makes no selection, the help window disappears after a pre-set time.

References

Breidt, E., F. Segond & G. Valetto: *Local Grammars for the Description of Multi-Word-Lexemes and Their Recognition in Texts.* This volume, 1996.

Hauser, R. & A. Storrer: *Dictionary Entry Parsing Using the LexParse System*, In: Lexicographica, Vol. 9 (1993), pp. 174–219, 1994.

Ostler, N.: *User Test Report 2.* COMPASS Deliverable 21, Rank Xerox Research Centre,



überstehen (1) , *überstanden* vt
(durchstehen)
to come through
to get through
(überleben)
to survive
(überwinden)
to overcome
Gewitter
to weather
Krankheit
to get over
Redewendungen:
nach überstandener Gefahr
when the danger was past

****

überstehen (2) vi
to jut out
to stick out
to project

Lufthansa: Fesseln abstre

Konzernchef Weber will zurücktreter … **cht bald kommt.**
Jürgen Weber atmet erst mal durch: "Anfa … rpaket
durchatmen
verabschiedet haben, befanden wir uns no … der
to adopt
Vorstandsvorsitzende der Deutschen Lufth … ste überstanden.
Der Patient Lufthansa liegt mittlerweile im

□Doch an eine Entlassung ist vorerst nicht … Lufthansa rote
Zahlen, wenn auch nicht mehr so hohe. Mit … at Weber dem
flügellahmen Kranich wieder frischen Win … tet: Knapp 7000
von 8000 geplanten Stellen im Unternehm … t ist gestiegen, die
jahrelang steigenden Stückkosten konnten … Weber zudem die
Abschreibungszeiten der Flugzeugflotte st … rlust in diesem
Jahr auf 500 Millionen Mark halbieren.

Die Anfangserfolge haben Weber ermutigt: 1994 will der Chefpilot erstmals wieder ein
ausgeglichenes Ergebnis vorlegen. Im kommenden Jahr winkt der Staatslinie eine bessere
Auslastung. Die Zahl der Flugzeuge wird von 216 auf 212 reduziert. Zugleich versprechen
Produktverbesserungen und Kooperationen die Nachfrage anzukurbeln

Figure 4: The COMPASS user interface with full dictionary entry

Grenoble, 1996.

Thielen, C. & E. Breidt: *Conversion of Bilingual Dictionaries: From Type–Setting Tape to Dictionary Database.* COMPASS Deliverable 23, Universität Tübingen, Seminar für Sprachwissenschaft, 1996

Wetzel, R. P.: *Human–Computer–Interface of the* COMPASS *System.* COMPASS Deliverable 22, Fraunhofer Gesellschaft, Institut für Arbeitswirtschaft und Organisation, Stuttgart, 1996

LEX4 - Yet Another Lexicon Formalism

GUNTER GEBHARDI

Abstract

LeX4 is an alternative lexicon formalism embedded in a complete system for lexicographic work, called the LeX4–SYSTEM, which sets the process of building up and maintaining lexicons into the system's focus. The paper gives an introduction to the formalism and discusses aspects of the system's architecture.

1 Introduction and Motivation

LeX4 and the LeX4–SYSTEM are devices to build up and to maintain lexicons, especially in the field of computational linguistics (CL).

During the last years, a lot of effort was made in CL to develop extensions of processing formalisms intended to support the representation of lexical knowledge or lexicon formalisms designed especially for these purposes. Formalisms with extensions are for example ALE (Carpenter, 1992a), CUF (Dörre and Eisele, 1991), ELU (Russell et al., 1993), TDL (Krieger and Schäfer, 1994) or TFS (Emele and Zajac, 1990), (Emele and Heid, 1993). The classical PATR–II (Shieber, 1986) has some special extensions aimed for this means, too. Known lexicon formalisms are for example ARIES (Goñi and González, 1995), COOL (Gates and Shell, 1993), DATR (Evans and Gazdar, 1989), IBL (Hartrumpf, 1994) or LKB (Copestake, Sanfilippo, and Briscoe, 1993).

These extensions of processing formalisms serve to have more comfortable means to encode lexical knowledge. Lexicon formalisms attack the task of representing linguistic knowledge in the lexicon directly, but partially losing sight of interface aspects. DATR for example is the most compact formalism and from this point of view the most elegant one, but (Kilbury, Naerger, and Renz, 1991), (Andry et al., 1992) or (Duda and Gebhardi, 1994) discuss the problem of how to connect DATR to processing formalisms.

Beside this stream of formal representational devices to support building-up lexicons there are a lot of tools in a wider sense, for example acquisition tools or database systems. In another direction we can find descriptional logics, which aim to represent knowledge in a very general manner.

LeX4 and partially COOL (Gates and Shell, 1993), set *the process of building and maintaining lexicons* into the system's focus. To be precise, the LeX4–SYSTEM is the complete system for this end and LeX4 is the representational device which controls the main functionality of the LeX4–SYSTEM. The motivation to use LeX4 as a general tool for lexicographic work is the idea to control the whole

*This work was funded by the German Federal Ministry of Education, Science, Research and Technology (BMBF) in the framework of the Verbmobil Project under Grant 01 IV 101 G. The responsibility for the contents of this study lies with the author.

computer side of the lexicographical process with only *one* device. Since lexicographers are not programmers or computer scientists, one tool should be enough.

Additionally, the $\mathcal{L}e\mathcal{X}4$–SYSTEM also supports the *maintenance* of a lexicon and the *cooperative work* of a group of lexicographers. These are aspects, which sometimes are of more practical evidence than questions of adequate linguistic representation, whatever this could be.

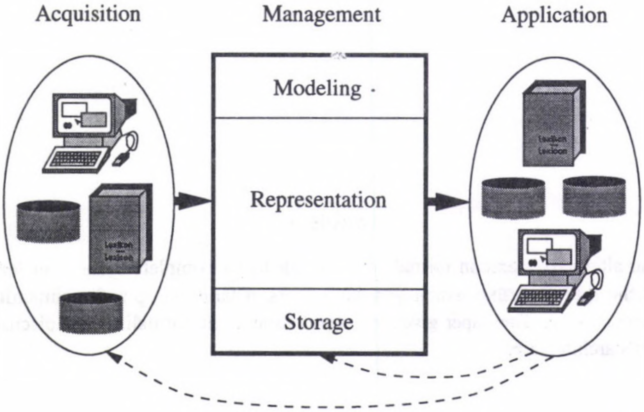


Figure 1: Subtasks of building a lexicon

The architecture of $\mathcal{L}e\mathcal{X}4$–SYSTEM is derived from a very general view of lexicographic processing shown in figure 1, influenced by proposals in (Heid and McNaught, 1991) ((Steffens, 1995a)). We distinguish the complexes of *acquisition*, *management* and *application*. The complex of management has three parts. The first one is *modeling* to describe the structure of all data and to define all relations between the data objects using $\mathcal{L}e\mathcal{X}4$. The second one is *representation*, which means in this lexicographic process dealing with data in the user defined $\mathcal{L}e\mathcal{X}4$–format. And the third one is *storage* as a service to keep the data.

A major decision in the design of $\mathcal{L}e\mathcal{X}4$ and $\mathcal{L}e\mathcal{X}4$–SYSTEM was to introduce the level of modeling as a center for controlling the structure of all data the system has to deal with. All data means also that kind of data which will be fed into the system in the course of acquisition as well as the data which will be specified for a certain application.

Being a formal descriptional language, $\mathcal{L}e\mathcal{X}4$ is not suitable as a graphical user interface for data acquisition and browsing, for data storage or as a special interface component for low level data transformation (to read in random data streams or to write it out). For these purposes, special means like a window system, a database system and a formal rewriting system are appropriate. Components specialized for these purposes are part of the $\mathcal{L}e\mathcal{X}4$–SYSTEM. Designed in correspondence to each other, to the formal representational device and to the system philosophy in general these components are efficient and optimized in use, but above all other they are controllable by one device: $\mathcal{L}e\mathcal{X}4$. So $\mathcal{L}e\mathcal{X}4$–SYSTEM contains the

- $\mathcal{L}e\mathcal{X}4$–BLAH (**B**erliner **L**exikon**A**kquisitions**H**ilfe)(Heinecke, 1996) — a graphical data acquisition and inspection tool; a compiler translates $\mathcal{L}e\mathcal{X}4$ definitions into window descriptions

- $\mathcal{L}e\mathcal{X}4$–DBS — the **D**ata**B**ase **S**ystem (Kruschwitz and Gebhardi, 1996) with some special functions to store ℒℰ𝒳4–structures, to have fast access to such structures, to support versions and cooperative work

- ℒℰ𝒳4–TRAFO — a kind of pretty printer for translating ℒℰ𝒳4–structures into structures of grammar formalisms

2 The Lexicon Formalism ℒℰ𝒳4

The lexicon formalism ℒℰ𝒳4 consists of two basic components: a feature term inferencing machine and a generation system.

The elementary data structures of ℒℰ𝒳4 are *feature structures* ((Kay, 1979), (Shieber, 1986)). Other lexicon formalisms also make successful use of feature structures ((Copestake, Sanfilippo, and Briscoe, 1993), (Goñi and González, 1995), (Hartrumpf, 1994)), and it is an important fact that these structures seem to be appropriate for lexicons in general, what (Veronis and Ide, 1992), (Ide, Maitre, and Véronis, 1994) propose and demonstrate.

An essential characteristic of the ℒℰ𝒳4 language is the concept of bases, classes and instances, which is deeply influenced by ideas of object oriented design (Coad and Yourdon, 1991). Partially, this concept takes the role of types (Carpenter, 1992b).

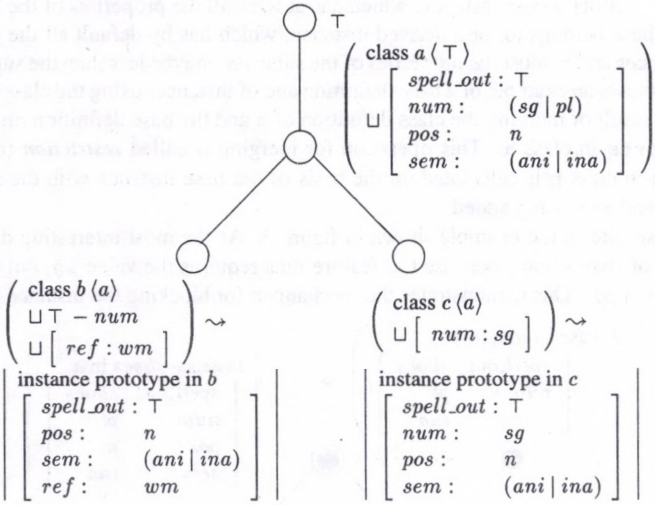
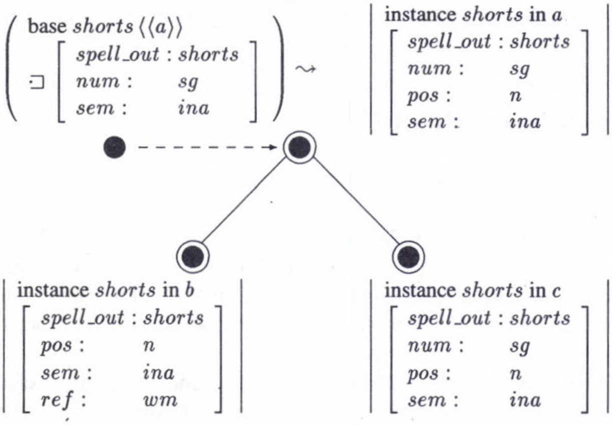

⊤
class a ⟨ ⊤ ⟩
⊔ [spell_out : ⊤ ; num : (sg | pl) ; pos : n ; sem : (ani | ina)]
class b ⟨a⟩
⊔ ⊤ − num
⊔ [ref : wm] ⇝
instance prototype in b
[spell_out : ⊤ ; pos : n ; sem : (ani | ina) ; ref : wm]
class c ⟨a⟩
⊔ [num : sg] ⇝
instance prototype in c
[spell_out : ⊤ ; num : sg ; pos : n ; sem : (ani | ina)]

Figure 2: A simple class hierarchy

A class denotes a set of entities. Each entity can be represented by a feature structure, but the number of entities can be infinite. Top (⊤) denotes the set of entities without specified properties. A simplified view of the subclass relation considers this as a relation over the set of all classes and has two arguments: a subclass and a non-empty set of superclasses. A subclass inherits all properties of the superclass, may *extract* (operator −) or introduce (using *unification*) additional properties. It is important to note that the subclass relation is *not* defined on the basis of feature term subsumption. Figure 2 gives an example of a small hierarchy. The classes b and c take the information of their superclass a and modify it. In class b the property num is extracted and ref: wm inserted. Class c is restricted to num: sg.

A base defines a subset of all possible entities introduced by the class or class expression (conjunction or disjunction of classes) the base belongs to.

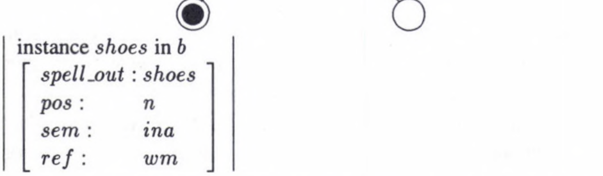


base $shorts \langle\langle a \rangle\rangle$
$\sqsupset \begin{bmatrix} spell\_out : & shorts \\ num : & sg \\ sem : & ina \end{bmatrix}$
$\rightsquigarrow$
instance $shorts$ in a
$\begin{bmatrix} spell\_out : & shorts \\ num : & sg \\ pos : & n \\ sem : & ina \end{bmatrix}$
instance $shorts$ in b
$\begin{bmatrix} spell\_out : & shorts \\ pos : & n \\ sem : & ina \\ ref : & wm \end{bmatrix}$
instance $shorts$ in c
$\begin{bmatrix} spell\_out : & shorts \\ num : & sg \\ pos : & n \\ sem : & ina \end{bmatrix}$

Figure 3: Instance propagation after inserting the base `shorts`

An instance is either a base instance, which has at least all the properties of the base and at most of the class the base belongs to, or a derived instance, which has by default all the properties of the superclass instances and at most the properties of the subclass, maybe less than the superclass instance bears. Figure 3 shows an example of a base definition and of instances using the class hierarchy shown in figure 2. The result of merging the class definition of `a` and the base definition `shorts` is the base instance for `shorts` in class `a`. This operation for merging is called *restriction* (operator $\sqsupset$) The derived instance in class `b` is calculated on the basis of the base instance with the `num` information being extracted and `ref` being added.

Figure 4 is similar to the example shown in figure 3. As the most interesting difference class `c` has no instance of base `shoes` because the feature `num` requires the value `sg`, but the value of this feature in `shoes` is `pl`. This demonstrates the mechanism for blocking the instance propagation.



base $shoes \langle\langle a \rangle\rangle$
$\sqsupset \begin{bmatrix} spell\_out : & shoes \\ num : & pl \\ sem : & ina \end{bmatrix}$
$\rightsquigarrow$
instance $shoes$ in a
$\begin{bmatrix} spell\_out : & shoes \\ num : & pl \\ pos : & n \\ sem : & ina \end{bmatrix}$
instance $shoes$ in b
$\begin{bmatrix} spell\_out : & shoes \\ pos : & n \\ sem : & ina \\ ref : & wm \end{bmatrix}$

Figure 4: Instance propagation after inserting the base `shoes`

All the operations like unification and extraction are part of the feature term inferencing machine. This machine deals with disjunctions and negation as special feature descriptions, which is essential for representations in a lexicon, as well as with the standard representational method of co-references. An extension of this machine is a device for dealing with sort descriptions. Sorts in LeX4 are defined with an open world semantics in contrast to classes.

Extraction introduces non-monotonicity and is an alternative to default unification ((Carpenter, 1993), (Lascarides et al., 1996), (Russell et al., 1993), (Bouma, 1990)).

The mechanism of instance propagation is part of the generation system. This produces, guided by the class relations, all user specified instances. The system introduced in (Gates and Shell, 1993) uses very similarly a production system (Shell and Carbonell, 1986). But compared to a production system, which has to search for rules to fire, the generation system is guided by rules. One supplement of the generation system is a method of co-instance references. This allows to use a part of one description as part of another one. A second supplement serves to define constraints over instance sets. For example these constraints restrict the use of a certain value to one instance only. So it can be controlled that homonyms get a unique identifier.

3 The LeX4–SYSTEM

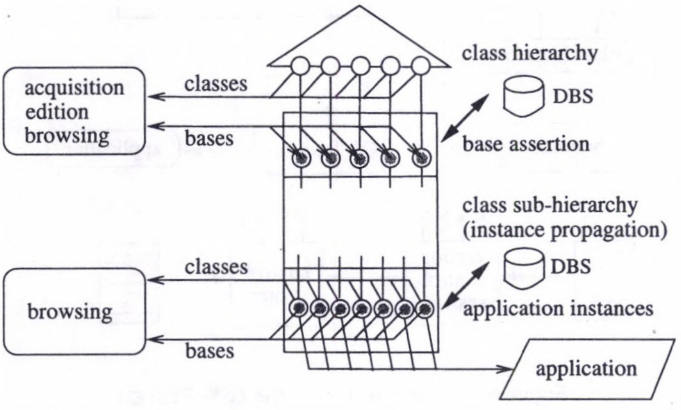


Figure 5: LeX4 embedded in the LeX4–SYSTEM

The lexicon formalism LeX4 is embedded in the LeX4–SYSTEM. Figure 5 sketches the basic principles of interaction between the formalism on the one hand side and system components on the other hand side.

The hierarchy on the top of the figure dominates the data structures to be accumulated in the lexicon. This hierarchy is the first part of the lexicon model. Classes definitions which take bases serve during acquisition to guide the process, as control information of a browser to structure the representations or to check the input from another machine readable lexicon. Bases asserted to the classes can be stored in the LeX4–DBS or by default in a simple file.

The second part of the lexicon model determines the mapping of accumulated data onto application specific data. Controlled by this part, the derived instances will be generated. The results can be inspected by a browser, again controlled by class definitions of this second part of the lexicon model, can be stored in the LeX4–DBS or can be transformed into application data.

Figure 6 gives a more detailed overview of the LeX4–SYSTEM. The schematic drawn pieces of paper in the upper part of the figure indicate input prepared by editors. Beside both parts of the

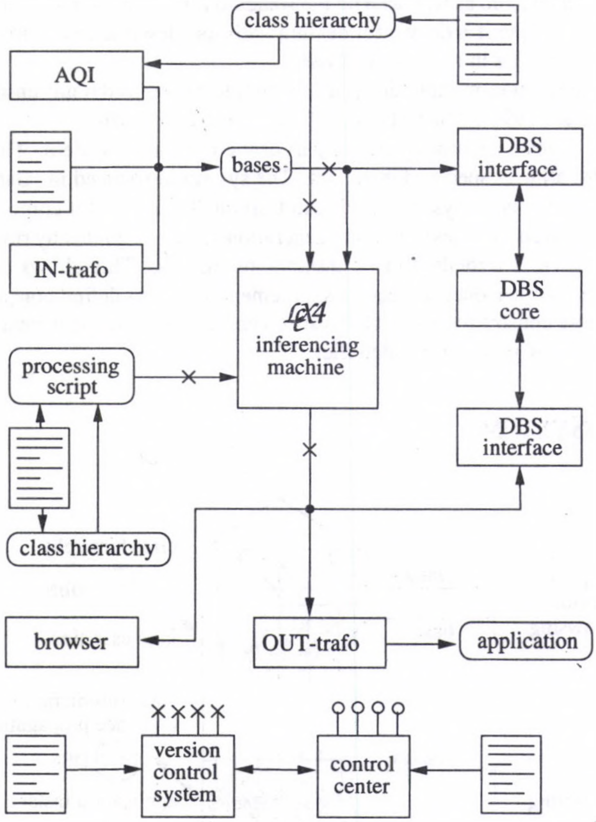
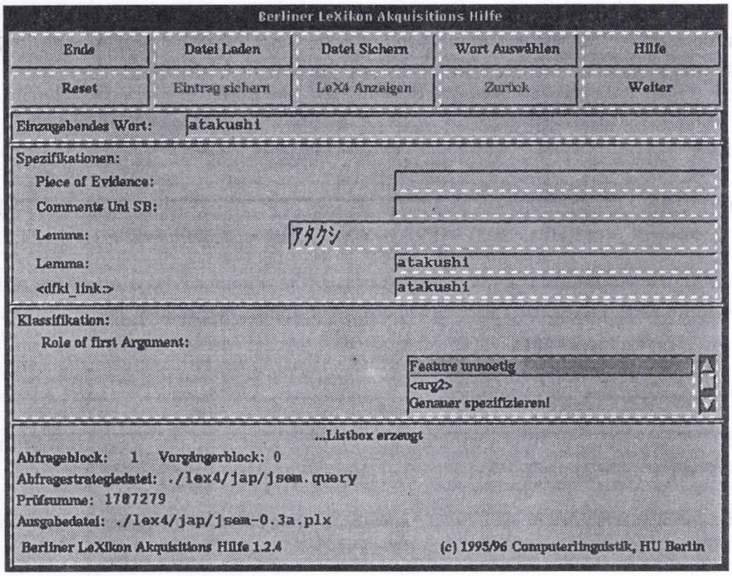


Figure 6: The architecture of the ℒ𝔢𝔵4–SYSTEM

class system base definitions are data which can be created by hand using an editor rather than the acquisition interface (labeled AQI in the figure) or an input transformation system (IN–trafo). The ℒ𝔢𝔵4 inferencing machine comes into play if application lexicons should be generated or if the input data should be checked. Using the same class descriptions the user can decide between these and other alternatives by adding a processing script.

In the lower side of the figure, there are two boxes labeled version control system and control center (ℒ𝔢𝔵4–BLAT). The control center serves with its graphical interface as a friendly guide through all steps of compiling lexicons. The version control system is a very sophisticated tool for compiling application lexicons as fast as possible. The system keeps book over all modifications of data. On request to compile a new lexicon version this control tool checks which data really have to be compiled and which can be read in the ℒ𝔢𝔵4–DBS as the results of former calculations. This process of generating only the modified pieces of information is called *incremental lexicon compilation*. The small X's in the figure shows the control points of this system. Small scripts configure the version control system and the control center.

4 Implementation and Application



Einzugebendes Wort: atakushi

Spezifikationen:

Piece of Evidence:

Comments Uni SB:

Lemma: アタクシ

Lemma: atakushi

<dfki_link> atakushi

Klassifikation:

Role of first Argument:

Feature unnoetig
<arg2>
Genauer spezifizieren!

...Listbox erzeugt

```
Abfrageblock:    1   Vorgängerblock: 0
Abfragestrategiedatei: ./lex4/jap/jsem.query
Prüfsumme: 1787279
Ausgabedatei: ./lex4/jap/jsem-0.3a.plx
```

Berliner LeXikon Akquisitions Hilfe 1.2.4 (c) 1995/96 Computerlinguistik, HU Berlin

Figure 7: A screen shoot of BLAH

The lexicon formalism LeX4 (Gebhardi and Heinecke, 1995) is implemented in SICSTUS- and QUINTUS-PROLOG. It runs on various machines, for example on PCs, MACINTOSHs and UNIX–workstations. The components of the LeX4–SYSTEM are implemented in C, TCL/TK and also PROLOG. Figure 7 shows a screen shot of the acquisition interface LeX4–blah (Heinecke, 1996) in an application with Japanese data.

LeX4 and the LeX4–SYSTEM are used in a large CL Project (Verb*mobil*) to build up and to maintain different types of syntactic lexicons and the semantic database of the project, which is a reference for all semantic components as well as for the translation component (Heinecke and Worm, 1996).

The system was used to maintain small as well as large lexicons for different languages (German, Japanese, Chechen, English) and to generate lexicons with more than 100 000 entries.

A spectacular application was to mix different syntax lexicons with one semantic lexicon and to maintain the lexicons while extending and modifying them. A lexicon with morpho-syntactic information and the semantic lexicon were used permanently. The two different syntax lexicons were a lexicon for an HPSG like grammar implemented in a constraint solver language and a lexicon for a GB influenced grammar written in PATR-II like style.

5 Conclusion

The lexicon formalism LeX4 was designed to support all the different steps of building and maintaining lexicons. These demands in mind lead to a formalism with some differences in respect to other known systems. Even processing aspects come in the main focus.

It was an unsurprising observation that a lexicon formalism is not able to support all the different special tasks during building up and maintaining a lexicon. As a consequence additional devices were developed and form together with $\mathcal{LEX}4$ the complete $\mathcal{LEX}4$–SYSTEM.

As a special feature $\mathcal{LEX}4$ and the $\mathcal{LEX}4$–SYSTEM also support the different aspects of lexicon maintenance. This property, not discussed in detail here, is conspicuously with respect to other systems.

The complete $\mathcal{LEX}4$–SYSTEM is a *homogeneous framework* intended to deal with lexicons. Using only *one* system a linguist can tackle a lot of different problems in this field.

References

Andry, F., N.M. Fraser, S. McGlashan, S. Thornton, and N.J. Youd. 1992. Making DATR Work for Speech: Lexcion Compilation in SUNDIAL. *Computational Linguistics*, 18(3):245 – 267.

Bouma, G. 1990. Defaults in Unification Grammar. In *28th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pages 173 – 179, Pittsburgh, PA.

Briscoe, T., A. Copestake, and V. de Paiva, editors. 1993. *Inheritance, Defaults, and the Lexicon.* Cambridge: Cambridge University Press.

Carpenter, B. 1992a. ALE — the attribute logic engine. user's guide. Technical report, Carnegie Mellon University, Pittsburgh, PA.

Carpenter, B. 1992b. *The Logic of Typed Feature Structures.* Cambridge: Cambridge University Press.

Carpenter, B. 1993. Skeptical and Credulous Default Unification with Applications to Templates and Inheritance. In *(Briscoe, Copestake, and de Paiva, 1993)*, pages 13 – 37.

Coad, P. and E. Yourdon. 1991. *Object–Oriented Analysis.* Englewood Cliffs, NJ: Yourdon Press.

Copestake, A., A. Sanfilippo, and T. Briscoe. 1993. The ACQUILEX LKB: An Introduction. In *(Briscoe, Copestake, and de Paiva, 1993)*, pages 148 – 163.

Dörre, J. and A. Eisele. 1991. A comprehensive unification–based grammar formalism. DYANA Deliverable R3.1.B, Universität Stuttgart, Stuttgart.

Duda, M. and G. Gebhardi. 1994. DUTR — A DATR–PATR interface formalism. In H. Trost, editor, *Konvens '94. Verarbeitung natürlicher Sprache*, Wien. Österreichische Gesellschaft für Artifical Intelligence.

Emele, M.C. and U. Heid. 1993. Formal Specification of a Typed Feature Logic Based Lexical Representation Language. Technical report, DELIS-Deliverable D-V-2, Universität Stuttgart.

Emele, M.C. and R. Zajac. 1990. Typed unification grammars. In *Proceedings of the 13th International Conference on Computational Linguistics, COLING-90*, Helsinki.

Evans, R. and G. Gazdar. 1989. Inference in DATR. In *Proceedings of the 4th Meeting of the European Chapter of the Association for Computational Linguistics*, pages 66 – 71, Manchester.

Gates, D.M. and P. Shell. 1993. Rule-based Acquisition and Maintenance of Lexical and Semantic Knowledge. In *Sixth Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, EACL-93*, pages 149 – 157, Utrecht.

Gebhardi, G. and J. Heinecke. 1995. Lexikonformalismus $\mathcal{L}e\mathcal{X}4$. Technical report, Verbmobil Technisches Dokument 19, Humboldt-Universität zu Berlin.

Goñi, J.M. and J.C. González. 1995. A framework for lexical represenation. In *AI95 – 15th International Confernece Language Engineering*, pages 243 – 252, Montpellier.

Hartrumpf, Sven. 1994. IBL: An inheritance–based lexicon formalism. Technical Report 1994-05, University of Georgia, Athens, Georgia. URL ftp://ai.uga.edu/pub/ai.reports/ai199405.ps.

Heid, U. and J. McNaught, editors. 1991. *Eurotra-7 Study: Feasibility and Project Definition Study on the Reusability of Lexical and Terminological Resources in Computerised·Applications.* Eurotra-7 Final Report, Stuttgart.

Heinecke, J. 1996. Lexikonakquisitionstools für den Lexikonformalismus $\mathcal{L}e\mathcal{X}4$. Technical report, Verbmobil Technisches Dokument 42, Humboldt-Universität zu Berlin.

Heinecke, J. and K.L. Worm. 1996. A lexical semantic database for verbmobil. Budapest. In this Volume.

Ide, N., J. Le Maitre, and J. Véronis. 1994. Outline of a Model for Lexical Databases. In *(Zampolli, Calzolari, and Palmer, 1994)*, pages 283 – 320.

Kay, M. 1979. Functional Grammar. In *Proceedings of the Fifth Annual Meeting of the Berkley Linguistic Society*, pages 142 – 158, Berkley, CA.

Kilbury, J., P. Naerger, and I. Renz. 1991. DATR as a lexical component for PATR. In *Fifth Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, EACL-91*, pages 137 – 142, Berlin.

Krieger, H.-U. and U. Schäfer. 1994. TDL — a type description language for constraint–based grammars. In *Proceedings of the 15th International Conference on Computational Linguistics, COLING-94*, Kyoto.

Kruschwitz, U. and G. Gebhardi. 1996. The $\mathcal{L}e\mathcal{X}4$–database system. Budapest. In this Volume.

Lascarides, A., E.J. Briscoe, N. Asher, and A. Copestake. 1996. Order Independent and Persistent Typed Default Unification. *Linguistics and Philosophy*, 19(1):1 – 90.

Russell, G., A. Ballim, J. Carroll, and S. Warwick-Armstrong. 1993. A Practical Approach to Multiple Default Inheritance for Unification–Based Lexicons. In *(Briscoe, Copestake, and de Paiva, 1993)*, pages 137 – 147.

Shell, P. and J. Carbonell. 1986. Frulekit: A Frame-Based Production System. Technical report, Center for Machine Translation, Pittsburgh, PA.

Shieber, S.M. 1986. *An Introduction to Unification-Based Approaches to Grammar*. CSLI Lecture Notes, number 4. Stanford, CA: Center for the Study of Language and Information.

Steffens, P. 1995a. Introduction. In *(Steffens, 1995b)*, pages 1 – 15.

Steffens, P., editor. 1995b. *Machine Translation and the Lexicon. Third International EAMT Workshop*, Lecture Notes in Artifical Intelligence 898, Berlin. Springer–Verlag.

Veronis, J. and N. Ide. 1992. A Feature-Based Model for Lexical Databases. In *Proceedings of the 15th International Conference on Computational Linguistics, COLING-92*, pages 588 – 594, Nantes.

Zampolli, A., N. Calzolari, and M. Palmer, editors. 1994. *Current Issues in Computational Linguistics: In Honour of Don Walker*. Giardini editori e stampatori in Pisa, Kluwer Academic Publishers, Norwell, MA.

Matching Corpus Translations with Dictionary Senses: Two Case Studies

ALEXANDER GEYKEN

Abstract

This paper addresses the question to what extent translations in bilingual parallel corpora match with dictionary senses. Automatic matching of corpus translations with dictionary senses depends on the quality of the lexicographic knowledge used, the quality of corpus processing, the impact of statistics to filter relevant entries from the corpora, and the quality of the translations in the multilingual corpora. We focus on the influence that the latter variable has on the performance of the automatic matching. Two case studies with two different corpora were conducted. Similarly to the BICORD system (Klavans 1996), we relied on MRDs, a Part-of-Speech tagger, and bilingual aligned corpora. Additionally, we used a shallow sentence parser for syntactic matching (Aleth® 1996). Our test set was the intersection of 500 French communication verbs with the corpora.

1. Introduction

Extraction of multilingual information from corpora plays a growing role for real-world NLP-applications. While some approaches argued for exclusively statistical non-linguistic methods to acquire translations (Brown et al. 1988; Church and Fung 1994), recent developments reveal the advantages of combining statistical methods with lexicographic knowledge as it is contained in Machine Readable Dictionaries (MRDs) (Catizone et al. 1989; Klavans 1996). In the latter approach, extraction of multilingual information corresponds to the matching of complete but unstructured information that is available from multilingual corpora to the structured but incomplete information that is stored in MRDs. Automatic matching of corpus translation with dictionary entries depends on the quality of the lexicographic knowledge used, the quality of corpus processing, the impact of statistics to filter relevant entries from the corpora, and, finally, the quality of the translations in the multilingual corpora. The first three variables have been subject to considerable research effort. Recently, corpus processing has become an important research area since it is expected that syntactic and semantic parsing of the corpus entry considerably improves the results of the matching process between multilingual corpora and dictionary senses (Klavans 1996, Briscoe 1994, EC-project SPARKLE-LE2111). Obviously, the performance of these methods depends on the translation quality of the multilingual corpus at hand.

This paper aims to contribute to this discussion by presenting two case studies with two parallel French-German corpora. One is a corpus of bilingual readings in literature of the 20th century (*Folio Bilingue*) of about 400 000 words; the other one is a running bilingual French-German newspaper corpus (translation started May 1995) of the monthly *newspaper Le Monde Diplomatique* of about 1 million words.

Similarly to Klavans (1996), our goal is bilingual lexicon construction, i.e., semi-automatic matching of corpus translations with dictionary sense and the semi-automatic incrementation of the lexicon where the data do not match. Our test set is a syntactically homogeneous subset of communication verbs. In the following sections, we describe the knowledge sources we used, and the method we used in order to automatize matching. We also report two experiments with the two above-mentioned corpora. Finally, we present and discuss the results.

2. Knowledge Sources

The knowledge sources used are a bilingual French-German lexicon of verbs, lexicons of frozen expressions, and parallel corpora.

2.1 Verb Lexicon and the test set

The contrastive French-German lexicon is based on the monolingual description of French verbs (Dubois, 1992). Dubois describes a lexicon of verbs where each verb meaning is related to a particular verb complement structure. His description contains 12 174 graphically different verbs that give rise to 25 086 verb meanings. The verb meaning splitting is due to the fact that different structures of the same verb may lead to different verb meanings. Each verb-sense contains (we consider only the information that is relevant for our purposes here) morpho-syntactic information, syntactic information such as the number and syntactic nature of the verb complements, and simple selectional restrictions like human, animate, and inanimate. Figure 1 below shows some of lexical entries of the contrastive verb lexicon for the verb *demander* (Fig. 1).

The test set is a large subset of French communication verbs that was determined with the help of previous work carried out at the L.A.D.L. (Gross, 1994). Here, we investigate the translations of the largest of these verb classes, i.e., all the French verbs sharing the following basic structure:

N0 V (N1 I Que S) à N2
(1) Max dit (ceci I qu'il fait beau) à Luc1
(1') (Max says (this I that the weather is fine) to Luc

There are 396 French verbs entering into this structure. It is important, however, that „a variable number of objects can be omitted, that is, verbs may enter into shorter substructures ... Thus, the structure has three possible substructures N0 V N1, N0 V à N2, and N0 V" (Gross, 1994). These 396 graphically different verbs give rise to 1900 verb meanings in the lexicon by Dubois. To date, the bilingual French-German lexicon

[1] *N0, N1, N2* denote the subject, direct object, and indirect object respectively; *que S* stands for sentential complement.

contains translations for 1850 of these word-senses producing approximatively 3500 translation pairs (Geyken 1996).

French Verb	hom_nb	Frame	German Translation	hom_nb
demander (to ask)	01	TR 13A	**fragen** (to ask)	01
demander **(to ask for)**	02	TR 14A	**bitten**	01
demander (to ask s.o.	03	TR 15A	**bitten**	02
demander to do sth)			erbitten	01
demander			verlangen	01
...				
demander	12	TR 13L	klagen	01

Figure 1[2]

2.2 Dictionary of frozen expressions

For French, a considerable number of frozen expressions has been collected and classified at the L.A.D.L.[3] (Gross 1994). By frozen expressions, we understand non-free, idiomatic, or technical usages of verbs (Gross 1994). Such expressions are non compositional in the sense that their meaning cannot be inferred from the meaning of the individual words composing them. They differ from „free“ expressions in that frozen positions do not allow substitution of phrases. For example, the expression *crier sur le toit („to shout on the roof top“; English sense: it´s common knowledge)* belongs to the structure N0 V N1 Prep C: this means that the subject and direct object are free positions whereas the preposition (*sur*) and the noun (*le toit*) are constants. These structures can be semi-automatically translated into local grammars (Silberztein 1993), thus, providing patterns which can be used to automatically match frozen expressions in corpora. For example, *crier sur le toit* is translated into <crier:V> sur <Det> <toit:N>[4], a simple local grammar that matches with phrases like *Ils le crient sur les toits* („They shout it from the roof“) *or n´importe qui pourrait le crier sur le toit.* („Everybody could shout it from the roof tops“). To date, more than 20 000

[2] The different values of the Frame determie the nature of the complements: TR stands for transitive, the first two numbers specify the subject/direct object this is: 1=human, 2=animated, 3=non animated, 4=sentential complement, 5=sentential complement with obligatory infinitive, 7=human plural, 8= non animated plural, 9=human or non animated. For the prepositional complement (the third number), the letters correspond to the following prepositions: A=*à;* B=*de;* C=*avec;* D=*contre;* E=*par;* G=*sur;* I=*de(partie);* J=*en, dans;* K=*pour(contre*); L=*auprès;* M=*devant;* M=*vers,à;* Q=*pour*. For example, *demander 02* has a transtive direct (TR) construction with a human subject, a sentential direct object, and a prepositional complement introduced by the complement à.

[3] Laboratoire d´Automatique Documentaire et Linguistique

[4] In this notation, <crier:V> means that the verb *crier* (to cry) can be replaced by any inflected form of the verb *crier;* similarly, <Det> by a determiner with plural or singular; and *toit* (roof) in <toit:N> can be replaced by a singular or plural form of the noun *toit*.

sentences have been classified in structures corresponding to minimal sentences of these expressions.

3. Using Knowledge to Match Translations with Dictionary Senses

Our goal is to filter out the role that translation quality plays for the automatic matching problem of corpora entries with dictionary senses. For comparison with previous work in this field, we decided to employ a method (Geyken & Tourovski 1996) that is similar to previous knowledge-based works in this area (e.g., Klavans 1996, McRoy 1992). Our input are bilingual corpora, a bilingual dictionary of verbs, and a dictionary of frozen expressions. Automatic processing occurs in the following steps: preprocessing (i.e., sentence alignment and corpus annotation), filtering of frozen expressions, and syntactic and semantic interpretation. The output is a classification according to the classes in Fig. 2. In the following, we describe the processing of the corpus entry more in detail.

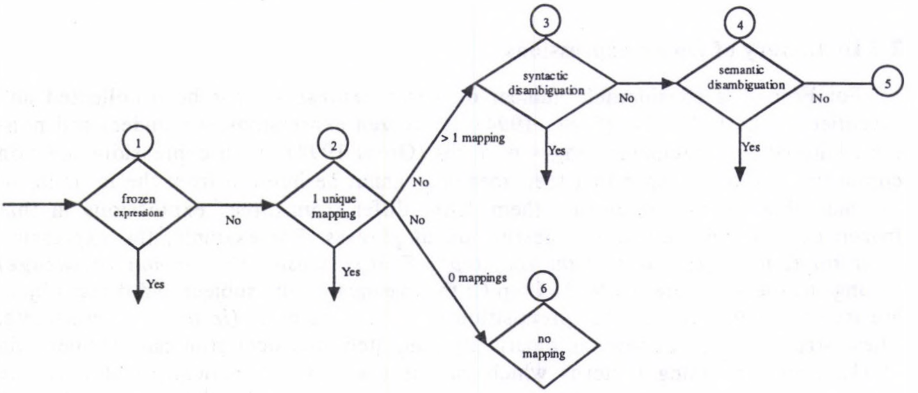


Figure 2: Steps of processing

The input of preprocessing is the parallel corpus and the bilingual verb lexicon. The output of preprocessing is the intersection of all translation pairs of the bilingual corpus within our test set. In order to realize this, we first have to determine which sentences in the parallel corpora are translations of one another; and second, we have to detect all the communication verbs in the source language of the parallel corpora. The first step corresponds to sentence alignment (Gayle and Church 1993) while the latter was realized with available corpus annotation tools. In particular, PoS tagging was realized with the corpus processing tool InteX for French (Silberztein 1993) and the CisLex for German (Maier 1995).

In the main processing of each sentence pair containing a verb from our test set, we determine the membership to one of the classes described above in the following way:

It is first tested if the corpus translation corresponds to a frozen expression (**Class 1**). This is done by matching the French sentence with local grammars extracted from the L.A.D.L. dictionaries of frozen expressions. If the look-up in the frozen expression dictionary fails, the translation pairs in the corpus are compared with the dictionary senses. There are several possible cases: a one-to-one correspondence between sentence pairs of the parallel texts and the translation pairs of the lexicon, a many-to-many correspondence, or no correspondence.

In the case of a one-to-one correspondence of sentence pairs of the parallel texts and the translation pairs of the lexicon, the matching problem is solved (**Class 2**).

In the case of a many-to-many correspondence, the syntactic parsing tree of the corpus sentence is matched with the dictionary senses. Different word senses of a verb can require different syntactic organization of the sentence: this is the underlying principle of the contrastive lexicon by Dubois. Sentence parsing may specify these differences, thus, allowing the algorithm to match the corpus translation to a unique dictionary sense (**Class 3**). In our experiment, we used a shallow parser (ALETH®), i.e., a parser that segments a text into phrasal groupings without always making the attempt to construct a full parse tree, to resolve prepositional phrase attachment, or the scope of conjunctions, etc.

If syntactic parsing still yields more than one translation, then further disambiguation is tried by comparing the selectional restrictions of the verb complements in the corpus translation with the frame of the dictionary sense. Selectional restrictions on the complements of the verb like human, animate, or inanimate can be used to match the translation with the dictionary sense. If the exploitation of selectional constraints reduces the match between the corpus translation and the dictionary sense to one correspondance, we are in the case of **Class 4**. If, despite of the exploitation of selectional constraints, the analysis yields still more than one correspondance, it is mapped to **Class 5**. Obviously, automatization of this process requires that dictionary senses be annotated with selectional restrictions and that the parsing procedures detect the lexical heads.

The only remaining case is that there is no correspondance between the corpus translation and the dictionary (**Class 6**). This case may be considered as a general failure for automatization. Even though all of these cases will have to be processed manually, they might reveal many interesting translation equivalents hitherto new for the bilingual lexicon.

4. Examples

Example 1 (Class 2): Consider the French verb *demander* (to ask) in the following sentence pair (*„Qu´est-ce qu´elle a?“ demanda l´homme ; „Was hat sie?“ fragte der Mann*; *--„What is the matter with her?“ asked the man*). Part of speech tagging yields that *demander* is the only verb in the French sentence and *fragen* in the German sentence. Since *demander* is part of our test set, we test if *demander* in the French sentence and *fragen* in the German sentence belong to a frozen expression, i.e. matches a local grammar constructed from a frozen expression. In the above sentence, the matching would fail since <demanda:V> <det> <homme:N> does not correspond to any local grammar from the dictionary. As this matching fails, we consult our contrastive verb lexicon where we find (*demander*$_{01}$,*fragen*) as the only translation pair in the lexicon (Fig. 1 above). Hence, the sentence pair in the parallel text can be mapped to a unique dictionary sense *demander*$_{01}$.

Example 2 (Class 3): Consider again the French verb *demander* in the following sentence pair: *Elle demanda de faire une halte*; *Sie bat darum, Rast zu machen*; *she asked to take a rest*. PoS tagging in this case produces two different verbs in the

French sentence (*demander, faire*) as well as in the German sentence (*bitten, machen*). Only the verbs *demander* respectively *bitten* belong to the test set. A look-up in the frozen expression lexicon fails since neither of the sentences is idiomatic. A look-up in the verb lexicon yields that the sentence would match two different dictionary senses, (*demander*$_{02}$, *bitten*), (*demander*$_{03}$, *bitten*). Thus, we are in the case of a many to many correspondences after PoS-tagging. In order to disambiguate the sentence we take advantage of the syntactic parse tree as it is produced by the partial parser (Aleth®, cf. Fig. 3).

```
Èlle demanda de faire une halte &period;

|gSentence
|    |gF_SUJ
|    |    |gNP
|    |    |    |frPro ELLE
|    |gVP
|    |    |gVtenseS
|    |    |    |frVerbe DEMANDER
|    |    |gF_OBJ2
|    |    |    |gClauseInf
|    |    |    |    |frPrep DE
|    |    |    |    |gVP
|    |    |    |    |    |gVtense0
|    |    |    |    |    |    |frVerbe FAIRE
|    |    |    |    |    |gF_OBJ1
|    |    |    |    |    |    |gNP
|    |    |    |    |    |    |    |gDet
|    |    |    |    |    |    |    |    |frArt UNE
|    |    |    |    |    |    |    |frNom HALTE
```

Figure 3

The parse tree contains an infinitive clause in the object position. From the lexicon, we know that *demander*$_{03}$ takes a sentential direct object in the infinitive (TR 15A) whereas *demander*$_{02}$ has a sentential but no infinitive possibility (TR 14A). Hence, the sentence matches to the unique dictionary sense *demander*$_{03}$..

Example 3 (Class 4): There are cases where syntactic information is not sufficient and simple selectional constraints are needed. An example for this case is provided by the verb *sonner* (to ring) in the following sentence pair: *(...lorsqu´on sonna.; ... als es an der Tür klingelte.; --... when the door bell rang.)*. The lexicon contains three translation pairs (*sonner*$_{04}$, *klingeln*), (*sonner*$_{05}$, *klingeln*), and (*sonner*$_{09}$, *klingeln)*. We can exclude a match with *sonner*$_{05}$ since we know from the lexicon that *sonner*$_{05}$ is transitive whereas sonner in our example is used intransitively. Furthermore, the subject of *sonner*$_{09}$ is coded „human" in the lexicon whereas the subject of *sonner*$_{04}$ is inanimate. Hence, the sentence pair can be mapped to word-sense 09 of the verb *sonner.*[5]

Example 4 (Class 5): Another example for class 5 is provided by the following sentence pair with the verb *dire*: (*„Tu ne peux t´imaginer", dit elle ...; „Du glaubst es nicht", sagte sie ... „You don´t believe it", she said ...*). After processing syntactic and semantic disambiguation, a sentence pair with the verb *dire* could still be mapped on the two translation pairs (*dire*$_{06}$, *sagen; : On dit que Paul est bon; One says that Paul is a good fellow), (dire*$_{12}$, *sagen; On dit que Paul est l´assassin*; *One says that Paul is the murderer*). However, even for a human „disambiguator", it would be difficult to map the sentence pair above to one of the two senses *dire*$_{06}$ or *dire*$_{13}$. In

[5] This example cannot be matched automatically since resolution presupposes semantic analysis.

this case, we would prefer to merge the two dictionary senses *dire*$_{06}$ and *dire*$_{13}$ to a single sense. Hence, in this case, the algorithm would contribute to an improvement of the dictionary.

Example 5 (Class 6): In the sentences *(„M. Bob Dunlop (...) dénonce le tabou persistant qu'est le suicide ..." Bob Dunlop (...) ist* ***empört*** *darüber, daß hier Selbstmord immer noch als Tabu behandelt wird-- Context: „Mr. Bob Dunlop is upset about the fact that suicide is still regarded as a taboo"*), the verb *dénoncer* is translated by the copula + adjective *empört sein*, an admittedly free translation. PoS-tagging detects the verb *dénoncer* in French but only a copula in the German sentence. Since the French sentence is not a frozen expression, i.e., no local grammar matches with the sentence and also a look-up of the pair *(dénoncer, sein)* in the verb lexicon fails, we are in another case of failure of the lexicon. Here, it has to be manually decided if the dictionary has to be updated with the translation or if the translation has to be rejected (which would certainly be the case in this example).

5. Experiment

An experiment with the different components was run on the two corpora described above. Two subsets of the corpora have been manually evaluated (Ebert 1996).

1. We manually evaluated a subset of the first corpus (*Folio Bilingue*), i.e., a book by *Gabriele Eckart: Stories and experiences in the G.D.R.* The French text consists of 998 sentences and 11466 words, which included 2501 different word forms. The intersection of verbal forms in the text with verbal forms in the contrastive verb lexicon yielded 48 different tokens with 168 occurrences. The parallel German text consisted also of 998 sentences, 9888 words with 2520 different word forms.

2. Ten articles in the February issue (1996) of the bilingual French-German edition of the French newspaper *Le Monde Diplomatique* were evaluated. The French articles consist of 871 sentences, 19 358 tokens with 5070 different word forms, and 62 different lemmas in the intersection with our test set with 191 occurrences. The German articles yielded 875 sentences, 18 792 tokens including 5825 forms.

6. Results and Discussion

The results according to the algorithm described above are displayed in Fig. 4. The classes correspond to the steps in the the algorithm.

Class \ Corpus	1. Frozen express-ions	2. 1:1 correspon-dence	3. n:n correspon-dences +(syntax)	4. n:n correspon-dences +(simple selectional restrictions)	5. n:n correspon-dences (else)	6.1 1:0 correspon-dence: Translation <> verb	6.2 1:0 correspon-dence: Translation = verb	Sum	Recall
Bilingual	21	64	14	14	24	12	21	168	67,26%
Literature	12,50%	38,10%	8,33%	8,33%	14,29%	5,95%	12,50%	100%	
Le Monde	25	12	26	5	16	46	61	191	35,60%
Diplomatique	13,09%	6,28%	13,61%	2,62%	8,38%	24,08%	31,94%	100%	

Figure 4

Before discussing the statistics, it is useful to make some remarks about the classes 5 and 6, i.e. the classes that cannot be handled automatically. Class 5 reveals two different cases which have to be manually categorized. On the one hand, these are cases where deeper semantic knowledge than knowledge about simple selectional restrictions like human, animate, or inanimate is required to match the translations to dictionary; on the other hand, translations of this class might reveal shortcomings of the dictionary, i.e., where the dictionary senses are too finely grained for the purposes of NLP.

Recall: Assuming that we use our verb lexicon, excellent lexicons containing frozen sentences, and perfect sentence parsers that detect even lexical heads of verb complements, we would be able to match not more than 67.26%, respectively, 35.60% of the sentences with dictionary senses (recall). Of course, this recall decreases if one realistically assumes less perfect detection of frozen expressions and sentence parsers (see discussion of failures below).

The case study also shows that syntactic parsing solves only 8.33%, respectively, 13.61 % of the cases (Class 3). Supposing that this parser would even be able to detect lexical heads and be able to categorize these heads as human, animate, or inanimate, the matching rate of the parsers would increase to 16.66%, respectively, 16.23% (Class 3+4). In other words, more than 80% remain either unsolvable with syntax or simple selectional restrictions (Classes 5, 6.1, and 6.2) or has already been solved with lexicons of frozen expressions (Class 1) or the simple translation in the target language (Class 2).

Precision: We stated above that *recall* decreases parallel to a decrease in quality of sources used. Also *precision* depends on the quality of sources. In particular, our method requires excellent dictionaries of frozen expressions, otherwise translations might be mapped to incorrect dictionary senses. The following example, even though it was the only one in our test set, illustrates the drawback an unrecognized frozen expression might have: *siffler un verre* (English sense: *have a drink*) in sentence (5) is translated in the corpus by sentence (5') *einen zischen.* If the expression *siffler un verre* was not detected as a frozen expression, one would incorrectly identify the extracted translation pair (siffler, zischen) with one of the verb senses of the dictionary, namely sense 02. (*Le serpent siffle - The snake hisses*).

(5) *On en sifflera une et on causera avec les nanas*
(5') *Dann zischen wir einen und dann quatschen wir mit den Bienen*

Impact of corpora: The differences in recall and precision between both corpora are obvious. In particular, the results show that „freer" translations in the newspaper corpus are reflected by a considerable increase of contextual translations that are not contained in a lexicon and that should not be used to augment the lexicon. Indeed, we considered 18 from 167 translations (10.71%) in the literature corpus as not being interesting while this number increased to 66 from 191 translations (34.55%) in the newspaper corpus (cf. Fig. 9). Another finding is that the matching of sentences to dictionaries by the translation alone (Class 2) decreases significantly from 41.03% in the literature corpus to 6.28% in the newspaper corpus.

Acknowledgements:

The author wishes to thank Bruno Menon for providing syntactic parses with ALETH® for a subset of the corpora (ALETH® is a registered trademark of GSI-Erli), and Vladimir Tourovski for many valuable discussions.

A longer version of this paper will appear in IJCL (3) - International Journal of Corpus Linguistics

References:

Briscoe, E.; Carroll, J (1994). *Towards automatic extraction of argument structure from corpora.* Rank Xerox Research Centre, MLTT-TR-06

Brown, P., Cocke, J., Della Pietra, S., Della Pietra, V., Jelinek, F., Mercer, R., and Roossin, P. (1988). *A statistical approach to machine translation.* In Proceedings of the Twelfth International Conference on Computational Linguistics., Budapest, Hungary.

Catizone, Roberta; Russel, Graham; Warwick, Susan (1989). *Deriving Translation data from Bilingual Text.* Proceedings of the First Lexical Acquisition Workshop, Detroit.

Dubois, Jean.; Dubois-Charlier, Francoise (1992). *Dictionnaire des verbes.* Rapport Technique no. 37. L.A.D.L University Paris 7.

Ebert, Ulrich (1996). *Evaluation of a French-German lexical database of communication·verbs.* unpublished masters thesis, University of Munich.

Gale, William A.; Church, Kenneth W. (1993). *A Program for Aligning Sentences in Bilingual Corpora.* Computational Linguistics, 19(1):121-142.

Geyken, Alexander (1996). *Constructing a contrastive French-German Lexicon-Grammar: the case of communication verbs.* Technical Report, CIS - University of Munich.

Geyken, Alexander; Tourovski, Vladimir (1996). *The use of parallel corpora for verb sense disambiguation.* Proceedings of ECAI-MULSAIC 1996, Budapest.

Gross, Maurice (1994). *Constructing Lexicon Grammars.* In Atkins, B.T.S, Zampolli, A. (Ed). Computational Approaches to the lexiçon, pp. 213-265. Clarendon Press Oxford.

Klavans, Judith; Tzoukermann, Evelyne (1996). *Combining Corpus and Machine-Readable Dictionary Data for Building Bilingual Lexicons.* Machine Translation, 10:3-4, 1-34.

Maier, Petra (1995). *Lexicon and Automatic Lemmatization.* Ph. D. thesis , University of Munich (in German).

McRoy, Susan (1992). *Using Multiple Knowledge Sources for Word Sense Disambiguation.* Computational Linguistics, Vol. 18 (1):1-30.

Silberztein, Max (1993). *Dictionnaires électroniques et analyse automatique de textes.* Masson: Paris.

Approximate Linguistics

GREGORY GREFENSTETTE

Abstract

One of the purposes of lexicography is describing the ordinary uses of words. Capturing these uses means abstracting away unimportant surface differences of how they are used in text. While awaiting the arrival of perfect and robust parsers which strip away surface differences in a principled way and reveal the complete structure of text, different levels of surface abstraction can be approximated by available text processing tools. The more evolved the tools, and the more linguistic information they use, the cleaner the results.

1 Introduction

Lexicography is concerned with describing how words are used in a given language. Underlying this question of *how* is a secondary question of *how often*, particularly when one is interested in the common rather than extraordinary uses of a word. *How often* implies some sort of relative frequency, whose capture implies, ultimately, counting. The computer is very good at counting, but, unfortunately, only good at counting things which can be made to look exactly alike.

Parsing is concerned with describing the syntactic structure of text. When a text corpus is parsed, one can accurately ignore some aspects of surface text variation, and use the abstract respresentation that the parser provides to extract and count the common usages of a word. For example, a parser can describe the adverbial modifications of a certain verb while abstracting away surface variations caused by word order, tense, voice, mood, number, etc.

The use of computer parsers in lexicography is limited by the unavailability of swift, robust parsers able to process accurately large corpora of naturally occurring text. Until the day such parsers become available, the abstraction process provided by parsers can be approximated by lesser text processing means. The successfulness of this approximation depends, of course, of the amount of linguistic information embodied in the process. In the following sections, we present how a typical lexicographical problem can be attacked by successively refined tools. This example shows, we believe, that the linguistic processing of text for lexicography need not be considered an all-or-nothing choice ("Either we solve the parsing problem, or we stay with KWIC lists"), but that natural language processing (NLP) can be seen as a continuum of successively cleaner approximations of true parsing.

2 Abstraction Levels

In order to have a computer count two things as the same, these two things must be made to look exactly similar. If we wish to use a computer to help cluster lexical phenomena in textual corpora, this means producing a substitute representation of the original text in which similar phenomena are made to look alike.

Certain surface variations often disappear when a text is put into electronic form: font differences, type size, etc. By reducing differently printed characters to the same bit sequence, this is the first level of abstraction. Other levels are provided by

Tokenisation: Deciding where to find the boundaries of the objects to be compared.

Lemmatisation: Collapsing inflections to normalised lemmata.

Part-of-Speech tagging: Abstracting away from individual words to grammatical classes of words.

Low-Level Parsing: Abstracting from part-of-speech to syntactic function.

Semantic Tagging: Abstracting away from individual words to semantic classes.

Machine implementation of these levels of abstraction can reduce different surface forms of lexicographical phenomena to equivalent representations that can be matched, and counted, by a computer. Each abstraction level requires different elements of lexical knowledge, but levels with less knowledge can approximate those with more, at the cost of noise that the human lexicographer must eliminate.

3 A Lexicographic Problem

To illustrate what different levels of abstraction can provide to the lexicographer, let us consider a typical lexicographic problem. Here, we suppose that we have a large corpus of American English text and that we want to find to find the typical arguments of the verb *appeal* in this text. This argument-searching example is additionally interesting since the verbal and nominal forms of the word are written the same: to make an *appeal*, to *appeal*. We suppose that we are looking for the common direct objects (what can be *appealed*) and the indirect objects (what can be *appealed* to) of the verb *appeal*.

3.1 Tokenisation

Tokenisation defines the units that will be counted. The techniques of tokenisation involve using regular expressions to describe contextually where the input string need be separated into units[Grefenstette & Tapanainen 1994]. Tokenisation is a well known problem in computer science, being an inherent part of computer language compiler construction[Aho *et al* 1986], and many computer-based tools have been developed that can be used for tokenising, e.g. *lex*[1], *awk*[2], *perl*[3]. A tokeniser for a language contains simple information about how words are formed in that language: what characters are letters, what characters are numbers, what characters are punctuation, what elements of punctuation can appear within a word (like the apostrophe or the dash in English.)

Using one of these computer-based tools integrating this word-structure knowledge, we can divide the input text into separate tokens. As a first approach to our lexicographical problem, we can process our corpus in the following way: Whenever the token *appeal* appears we can store the next three tokens that occur in the input. This simple heuristic is based on the rudimentary linguistic knowledge that English verbal arguments and adjuncts generally occur shortly after the verb. This same heuristic is similar to that used by practicing lexicographers visually scanning right-sorted KWIC files for regularities. Figure 1 shows the most common tokens extracted using this windowing technique for the tokens *appeal* and *appeal* over a 250 MB corpus of Wall Street Journal text.

This Figure 1 shows that nominal and verbal forms of *appeal* are mixed. We find data coming from verbal uses such as *appeal a decision* as well as from nominal cases such as *an appeal was filed*. The results seem rather noisy, and require a certain patience and habit in order to extract any information. One can derive even from this noisy data that *rulings* and *decisions* seem to be common arguments of *appeal*. But the confusion between nominal forms and verbal forms renders speculative any conjectures about common prepositions associated with the verb *appeal* other than *to*.

One simple linguistic refinement of this windowing techniques is to borrow the idea of a stoplist from the information retrieval community. Information retrieval has used such lists of common words in order to reduce the automatic indexing of text to content-bearing words. These lists[4] are comprised of personal pronouns, articles, prepositions, conjunctions, etc., the closed-class words of the language. For English such a list runs to about one hundred words. Of course, such a list can be counted as linguistic knowledge beyond the word-structure knowledge already used by the tokeniser. This simple process of filtering out such words provides a much cleaner list using this simple three-word window technique, as seen in Figure 2.

[1] See http://www.cs.columbia.edu/~royr/tools.html

[2] See http://w4.lns.cornell.edu/public/compdoc/info/gawk/gawk_1

[3] See http://www-cgi.cs.cmu.edu/cgi-bin/perl-man

[4] Available via ftp in the directory /pub/med/smart/ at ftp.cs.cornell.edu.

REFORMATTER, TOKENISER

	appeal _ _ _		*appeals* _ _ _
570	to	1350	**court**
558	the	348	said
348	,	292	.
237	by	279	the
233	.	267	to
194	a	179	,
178	for	147	that
166	of	121	in
130	in	105	**ruled**
124	on	97	**ruling**
110	and	81	for
101	was	67	**panel**
85	's	67	a
78	that	62	**upheld**
77	said	56	and
73	**today**	54	by
68	his	50	from
61	is	49	**courts**
59	**ruling**	47	has
59	**acted**	45	also
58	**decision**	44	of
50	from	43	**judge**
44	**filed**	39	**rejected**
43	"	32	have

Figure 1: The most common strings following the strings "appeal" and "appeals" in a 3-word window. The numbers given are the frequency with which the words appear in the 250MB corpus. Non-function words are shown in bold.

REFORMATTER, TOKENISER, STOPLIST

	appeal _ _ _		*appeals* _ _ _
73	today	1350	court
59	ruling	105	**ruled**
59	**acted**	97	ruling
58	decision	67	panel
44	**filed**	62	**upheld**
35	court	49	courts
33	case	43	judge
28	conviction	39	**rejected**
24	verdict	27	process
24	judge	26	**ordered**
22	supreme-court	26	friday
22	federal	24	wednesday
21	tuesday	24	**refused**
18	state	24	**overturned**
17	monday	24	**noted**
16	**made**	23	decision
15	u.s.-supreme-court	17	today
15	order	17	judges
14	wednesday	15	tuesday
14	sentence	14	restudy
14	**aimed**	14	board
13	people	13	thursday
13	lower	12	monday
11	**contended**	12	**made**

Figure 2: The most common non-stopword list strings following the strings "appeal" and "appeals" in a 3-word window. Obvious verbal forms are shown in bold.

Figure 2 shows the semantically rich words appearing after *appeal* and *appeals.* In our initial quest for arguments of the verb *appeal,* we now see many more nominal candidates. In addition to *ruling* and *decision,* we see *case, conviction, verdict, order* and *sentence* as well as adverbial complements and nominal candidates for what can be *appealed to* in American English: *court, judge, Supreme-Court, people.* The only change from Figure 1 to Figure 2 is the elimination of function words (a small increment in language-specific knowledge), but the effect is a clearer solution to the original lexicographic problem.

3.2 Lemmatisation and Tagging

A next level of linguistic sophistication, beyond tokenisers and lists of function words, is the ability to morphologically analyse and to lemmatise surface forms of words into some canonical form, for example, masculine singular for nouns, or an infinitive form for verbs. This requires the linguistic resources of a lexicon and an analyser, but thanks to the efforts of computational linguists and lexicographers over recent years, these basic resources are becoming available in more and more languages[Karttunen 1983][Chanod 1994].

With a lemmatiser all surface forms can be reduced to one or more lemmata, words can be recognised as possible adverbs, conjunctions, articles, etc. Recognising the possible grammatical functions of the words allows us to do away with a stop-word list, which can be replaced by a list of grammatical categories that we wish to exclude from consideration. Figure 3 shows what happens when we use the three-word window heuristic over lemmatised forms. That is, each time we come across a token that can lemmatise to *appeal*, we tabulate the lemmatised forms of the non-function words appearing up to three words after *appeal*. One effect of this lemmatisation is to collapse the two columns appearing in Figure 2, as well as to add in data from previously ignored word forms, *appealing* and *appealed*, and to normalise tokens such as *courts* to *court*.

The resulting list in the first column of Figure 3 is not all that very different at first glance from the first column from the first column of Figure 2, yet we have included a relatively expensive resource, a morphological analyser and lemmatiser in the process. In fact, the greatest difference comes in the numbers. Thus, in the same corpus, after lemmatisation, the number of recognised instances of *appeal ... conviction* grows from 28 to 97. This growth comes from the fact that the lemmatiser allows the computer to match variants such as *appealed all previous convictions* and *appeals today their conviction* to a single form *appeal conviction*, abstracting away morphological variation and intervening words within the window. The number of recognised instances of the desired phenomena grows as more linguistic information is added.

Since the number of recognised instances grows, we can consider rarer phenomena. The second column in Figure 3 shows the results of extracting and tabulating all lemmata appearing in the three words after a *to* appearing itself within three words of a form of *appeal*. This is an approximation of the verbal structure *appeal to*, and allows us to suppose that the things that can be appealed to are *courts, governments, states, people, nations* and *workers*. Since the structure is more complex, involving two words, it is rarer. The morphological analysis and lemmatisation, by abstracting away differences, improves the counts of the data.

Still, Figure 3 shows a confusion between verbal and nominal uses of *appeal*. In order to distinguish these uses when the surface form is *appeal* used as a noun or *appeal* used as verb, we can supplement the tokeniser, morphological analyser and lemmatiser, with one additional linguistic tool: a part-of-speech disambiguator[5]. In recent years, statistical techniques[Cutting *et al* 1992] have been developed for creating part-of-speech disambiguators with success rates between 95% and 99% of words correctly tagged. Using such a tool is important for our problem since the identical nominal and verbal forms are both frequent.

Figure 4 shows the results of applying the window-based technique to uses of *appeal* tagged as a verb by a part-of-speech disambiguator. The part-of-speech tagging allows us to consider in addition, if we wish, only nominal arguments to the verb. Here the solution to our original problem become even clearer. The second column in the Figure 4 shows the lemmata appearing in a window of three words after any preposition (not just *to*) itself appearing within three words of a verbal use of *appeal*. From these two columns, the lexicographer can be led to discover common uses involving prepositions: *appeal to governments/states/people* also *appeal for calm/help*, *appeal to stop/help/release*.

In addition, the part of speech tagging provides quantitative information about just what prepositions most often follow the verbal uses of "appeal" in this corpus, shown in Figure 5. Seeing that *to* is by far the most common preposition, we can extract the lemmata most often following the *appeal ... to* construction. Figure 6 provides these results after part-of-speech

[5] The Common Lisp source code for version 1.2 of the Xerox part-of-speech tagger is available for anonymous FTP from parcftp.xerox.com in the file pub/tagger/tagger-1-2.tar.Z. Another freely available English tagger, developed by Eric Brill, uses rules based on surface strings and tags. This can be found via anonymous ftp to ftp.cs.jhu.edu in pub/brill/Programs and pub/brill/Papers.

	appeal _ _ _		*appeal _ _ _ to _ _ _*
1470	court	68	supreme-court
302	rule	45	court
170	uphold	43	allow
144	ruling	36	government
142	decision	32	state
113	judge	29	help
104	today	28	us-supreme-court
97	conviction	26	people
81	order	24	us-circuit-court
77	panel	23	high
73	reject	22	end
64	act	21	let
59	file	20	reconsider
59	case	19	stop
56	overturn	19	nation
54	state	18	worker
52	wednesday	16	public
51	tuesday	16	give
50	friday	16	force
47	supreme-court	16	american
46	district	15	restudy
45	monday	15	federal
42	sentence	14	strike
39	refuse	14	release

Figure 3: The most common lemmata, excluding function words and punctuation, following the lemma "appeal" or the pattern "appeal ... to" in a 3 word window.

REFORMATTER, TOKENISER, TAGGER, LEMMATISER

	appeal/Verb _ _ _		*appeal/Verb _ _ _ PREP _ _ _*
135	**court** /NOUN	55	**Supreme-Court** /NOUN
96	**decision** /NOUN	40	**court** /NOUN
89	**ruling** /NOUN	33	**government** /NOUN
74	**conviction** /NOUN	31	**state** /NOUN
37	**judge** /NOUN	24	**calm** /NOUN
33	**sentence** /NOUN	24	**help** /NOUN
32	**verdict** /NOUN	22	**people** /NOUN
31	**case** /NOUN	22	federal /ADJ
28	**order** /NOUN	20	**release** /NOUN
28	**Supreme-Court** /NOUN	20	**US-Supreme-Court** /NOUN
21	**calm** /NOUN	19	**US-Circuit-Court** /NOUN
18	**release** /NOUN	17	high /ADJ
17	**government** /NOUN	17	**appeal** /NOUN
16	Wednesday /ADV	17	allow /INF
14	**state** /NOUN	16	stop /INF
14	**people** /NOUN	16	help /INF
14	federal /ADJ	15	public /ADJ
14	**board** /NOUN	14	**end /noun**
13	public /ADJ	13	**unity** /NOUN
13	**help** /NOUN	13	**ground** /NOUN
13	directly /ADV	12	release /INF
13	**US-Supreme-Court** /NOUN	12	much /ADJ
12	uphold /ACTVERB	12	let /INF

Figure 4: Non stopword lemmata appearing in 3 word window after a form of "appeal" tagged as a verb. In the second column, nouns, adjectives and verbs appearing after a preposition appearing after a verbal use of "appeal" are extracted from the corpus. Nouns are listed in bold.

frequency	*appeal/Verb PREP*
773	to
322	for
112	in
110	of
54	on
36	by
14	with
12	against
10	from

Figure 5: Prepositions found in the three words following "appeal" part-of-speech tagged as a verb.

REFORMATTER, TOKENISER, TAGGER, LEMMATISER

	appeal/Verb _ _ _ to _ _ _
32	Supreme-Court /NOUN
21	government /NOUN
14	court /NOUN
13	state /NOUN
13	people /NOUN
13	US-Supreme-Court /NOUN
12	public /ADJ
11	high /ADJ
9	US-Circuit-Court /NOUN
8	worker /NOUN
7	member /NOUN
7	congress /NOUN
7	authority /NOUN
7	allow /INF
7	Iran /NOUN
6	young /ADJ
6	student /NOUN
6	nation /NOUN
6	decide /INF
6	community /NOUN
6	citizen /NOUN
5	world /NOUN
5	woman /NOUN
5	war /INGVERB

Figure 6: Most frequent lemma-tag-pairs in a three word window following "to" following "appeal" as a verb. Compare to the second column in Figure 3 in which nominal and verbal readings cases of "appeal to" were not distinguished.

disambiguation and should be compared to the second column of Figure 3 in which only tokenisation and lemmatisation was applied. Though the most common things *appealable to* overlap, the list extracted after tagging is cleaner, and less frequent arguments appear higher in the list of Figure 6, such as *congress, authority, students, citizens, etc.* Again adding more linguistic knowledge, here tag-sequence probabilities for part-of-speech dismabiguation, improves and focuses the results.

3.3 Low-Level Parsing

As the previous sections have shown, some aspects of syntax can be approximated by simple position information, i.e. the window appearing fter the word being examined. We have been supposing that words appearing in this window probably play some role as an argument. Other words appearing in the window are abstracted away so that different surface configurations can be made to look equal for the computer.

A further linguistic refinement that can be applied is use regular patterns of the tags

REFORMATTER, TOKENISER, TAGGER, LEMMATISER, PARSER

	appeal — direct objects		*to-PPs following appeal*
100	decision	28	Supreme-Court
87	court	20	court
74	ruling	13	US-Supreme-Court
73	conviction	11	government
39	case	10	people
36	sentence	8	US-Circuit-Court
31	verdict	7	worker
27	order	7	leader
11	rule	7	Iran
11	official	6	party
9	government	6	congress
8	judgment	6	authority
6	refusal	5	nation
6	process	5	member
6	fine	5	honor
6	board	5	citizen
6	United-States	5	US-Court
5	injunction	4	student
5	dismissal	4	republican
5	award	4	judge
5	action	4	President-Reagan
4	sale	4	President-Bush
4	plan	3	youth
4	loss	3	world

Figure 7: Nouns found as direct objects of verbal uses of "appeal", and nouns heading 'to'-prepositional phrases appearing after "appeal".

provided by the part-of-speech disambiguator to partially recreate the syntactic structure of the sentence. Recognising nominal chains and verbal chains as sequences of part-of-speech tags allows us to recognise certain syntactic relations, such as government of a noun by a preposition, or the voice of a verbal chain. This classification allows us to further pinpoint possible objects and prepositional arguments while eliminating others. Much work is being done on creating such low-level parsers in a number of languages. Using such a low-level syntactic pattern extractor, built using finite-state regular expressions and transducers, we can analyse the corpus at a different level of abstraction.

Figure 7 shows the lemmata identified as direct objects and objects of the structure *appeal to* that are extracted using a low-level parser[Grefenstette 1996] over the same 250MB corpus used throughout this example. Though noise persists, as is the case with all the previous approximations to full linguistic parsing, the lists produced are more precise. Figure 8 provides a comparison. In both lists shown in Figure 8 , there is much overlap with the most frequent cases, but the focusing power of adding more linguistic knowledge appears nore clearly at the end of the list where one discovers only with the low-level parser that one can *appeal fines, injunctions, dismissals, awards, sales, plans* and *losses.* These lemmata would also appear

COMPARING CONFIGURATIONS USING TAGGERS BUT WITH AND WITHOUT PARSERS

appeal – direct objects		*nouns within 3 words of appeal*	
100	decision	135	court
87	court	96	decision
74	ruling	89	ruling
73	conviction	74	conviction
39	case	37	judge
36	sentence	33	sentence
31	verdict	32	verdict
27	order	31	case
11	rule	28	order
11	official	28	Supreme-Court
9	government	21	calm
8	judgment	18	release
6	refusal	17	government
6	process	14	state
6	fine	14	people
6	board	14	board
6	United-States	13	help
5	injunction	13	US-Supreme-Court
5	dismissal	12	end
5	award	10	unity
5	action	8	student
4	sale	8	clemency
4	plan	7	nation
4	loss	7	iran

Figure 8: Parsing vs. Window. Looking for direct object collocates of "appeal." The first column shows nouns extracted as direct objects by a low-level parser, the second columns shows nouns within a window of three words after a verbal use of "appeal." Though many of the same words are found among common patterns, noise from prepositional adjuncts appears rapidly in the second list.

further down the list of the second column of Figure 8, but they would be swamped in the noise present there and harder to discern.

The initial problem of finding argument for the verb *appeal* has been treated with a sequence of more and more sophisticated linguistic tools: from tokenisers, to list of stopwords, to morphological analysers and lemmatisers, to part-of-speech disambiguators, to low-level parsers. The results obtained (Figures 1 to 7) show a gradual focusing in which more noise is eliminated and in which more infrequent phenomena are brought to light as more linguistic information is incorporated into the process.

3.4 Semantic Tags

One further linguistic refinement, this time using the WordNet[Miller *et al* 1990] thesaurus as a linguistic resource, is shown in Figure 9. This figure shows the semantic tags associated with each of the most frequent direct objects recognised for *appeal*. Semantic tags are not as well defined as grammatical tags where the classes are more constrained and better understood[Bolinger 1965]. Ideally, a semantic part-of-speech tagger would use context, as a grammatical part-of-speech tagger does, in order to choose the most likely tag, but research in this area has not been as successful due to a lack of semantic dictionaries and due to the fact that the problem is no longer one of structure but of meaning.

The only conclusion to be drawn from Figure 9 is that the direct object of *appeal* is likely to be something classified as an *act* or as a *communication*, but the meanings of these semantic tags are not very clear in themselves.

4 Conclusion

According the Adam Kilgarriff [1992] the ideal lexicographer must (i) gather corpus of citations for a given word, (ii) divide the citations into clusters, (iii) decide why the cluster member belong together, and (iv) code their conclusions into a dictionary definition. A computer can be used to help cluster, but a computer can only match things that are exactly the same. Thus some representation of the original citations must be found that abstracts away surface differences so that exact matches can be within that representation.

We have seen in this paper that the abstraction process can be seen as a series of successively more informed linguistic approximations to full parsing. Simpler linguistic tools can approximate more advanced ones with more or less success, and adding more linguistic information improves the results obtained. In order to answer the question of what are the common clusters of arguments for a given verb, we have shown we can obtained answers using (a) simple tokenisation using only knowledge of word boundaries, (b) morphological analysers and and lemmatisers containing information about inflectional variation and grammatical parts-of-speech, (c) part-of-speech taggers using knowledge about probabilities of sequences of parts-of-speech, and (d) low-level parsers encompassing information about the local syntactic structures within verbal chains and nominal chains, and across chains. Each successive refinement produces answers with less noise and improved recall of the phenomena sought.

	appeal DOBJ	*WORDNET semantic tags*
100	decision	act, event
87	court	artifact, group
74	ruling	(not present)
73	conviction	act, cognition, state
39	case	artifact, cognition, communication, event,person, quantity
36	sentence	communication
31	verdict	act
27	order	act, attribute, communication, group, state
11	rule	artifact, cognition, communication
11	official	person
9	government	act, group, state
8	judgment	act, cognition, communication
6	refusal	communication
6	process	body, cognition, process
6	fine	act, possession
6	board	artifact, food, group, substance
6	United-States	location
5	injunction	communication
5	dismissal	act
5	award	communication
5	action	act, artifact, attribute, state
4	sale	act
4	plan	artifact, cognition
4	loss	attribute, event, possession, relation

Figure 9: The WordNet semantic tags associated with the most common direct objects of a verbal form of "appeal". The semantic tag "act" (10 of 24 words) or "communication" (7 of 24) appear most frequently. Choosing between the semantic tags using context is a problem akin to part-of-speech disambiguation, but on a much larger scale, as the ambiguity of words is greater than with grammatical tags, and because there are many more potential semantic tags than grammatical ones.

References

[Aho *et al* 1986] Afred V. Aho, Ravi Seth, and Jeffrey D. Ullman. *Compilers: Principles, Techniques, and Tools.* Addison Wesley, Reading, Massachussetts, 1986.

[Bolinger 1965] D. Bolinger. The atomization of meaning. *Language*, 41(4):555–573, 1965.

[Chanod 1994] Jean-Pierre Chanod. Finite state composition of french verb morphology. Technical Report MLTT-005. Rank Xerox Research
Centre, Meylan, France, April 1994.
http://www.xerox.fr/grenoble/mltt/reports/mltt-005.ps.

[Cutting *et al* 1992] Doug Cutting, Julian Kupiec, Jan Pedersen, and Penelope Sibun. A practical part-of-speech tagger. *Proceedings of the Third Conference on Applied Natural Language Processing*, April 1992.

[Grefenstette 1996] G. Grefenstette. Light parsing as finite state filtering. In *Workshop on Extended finite state models of language*, Budapest, Hungary, Aug 11–12 1996. ECAI'96.

[Grefenstette & Tapanainen 1994] G. Grefenstette and P. Tapanainen. What is a word, what is a sentence? Problems of tokenization. In *3rd Conference on Computational Lexicography and Text Research*, Budapest, Hungary, 7–10 July 1994. COMPLEX'94. http://www.xerox.fr/grenoble/mltt/reports/mltt-004.ps.

[Kilgarriff 1992] Adam Kilgarriff. *Polysemy.* PhD thesis, University of Sussex, 1992. ftp://ftp.cogs.susx.ac.uk/pub/reports/csrp/csrp261.ps.Z.

[Karttunen 1983] Karttunen L. KIMMO: a general morphological processor. In *Texas linguistics forum*, 1983.

[Miller *et al* 1990] George A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. Introduction to WordNet: An on-line lexical database. *Journal of Lexicography*, 3(4):235–244, 1990.

La formation des noms en -JIL en coreen

HAN, SUN-HAE

RESUME

Notre travail porte sur la formation des noms suffixés en *-jil* en coréen. L'observation d'un ensemble des noms dérivés par cette suffixation nous amène à distinguer deux suffixes *-jil* homonymes : suffixe ***prédicatif*** *-jil* et suffixe ***péjoratif*** *-jil*. Les noms formés par l'adjonction de chaque suffixe *-jil* n'ont pas le même comportement des points de vue sémantique et syntaxique. En ce qui concerne la productivité, on trouve aussi une grande différence entre ces deux suffixations. Tandis que le cas du suffixe *prédicatif -jil* affecte une liste relativement restreinte, l'autre suffixe *-jil* a une forte productivité : il s'applique à une classe de noms humains définie syntaxiquement et à un bon nombre de noms prédicatifs. En conclusion, nous proposons que ces deux suffixations en *-jil* soient traités différemment dans le lexique-grammaire des noms coréens et lors de la construction des dictionnaires électroniques coréens.

1. Introduction

Dans la description du lexique coréen, la catégorie des noms occupe une place cruciale. D'abord, la majeure partie des verbes aussi bien que des adjectifs se forme sur la base de noms, par l'adjonction de suffixes verbaux ou adjectivaux. Les études sur ce type de verbes ou d'adjectifs dérivés devront être menées en parallèle avec celles sur les noms reliés. De plus, en coréen, les relations dérivationnelles entre noms sont très fréquentes : un suffixe ou un préfixe s'ajoute à des noms pour produire d'autres noms. D'après le travail de J.S. Nam (Nam 1994), le nombre d'affixes qui concernent ce type de dérivations est vraiment étonnant : 889 préfixes et 836 suffixes ont été recensés. Les noms dérivés par ces affixations constituent la grande majorité de la couverture lexicale du coréen. Dans l'état actuel, pourtant, le traitement des noms dérivés dans les dictionnaires usuels est loin d'être systématique et les études sur ce domaine en sont à leurs débuts. Une tâche importante de la linguistique coréenne est d'établir la liste exhaustive autant que possible de cette énorme catégorie de noms dérivés et de leur attribuer systématiquement des informations sémantiques et syntaxiques. En particulier, du point de vue de la construction des dictionnaires électroniques, ce travail sera indispensable. Dans cette perspective, la productivité d'un affixe constitue un des problèmes qui se posent naturellement. En effet, les nombreux affixes coréens dont nous avons parlé plus haut ont une productivité très inégale : il y a des cas qui ne concernent qu'un seul nom et d'autres qui en affectent plusieurs centaines. Un affixe qui a une forte productivité mérite d'autant plus d'être étudié en détail.

Nous parlerons dans cet article de la formation de noms dérivés en prenant un cas précis, celui des noms suffixés en *-jil*. Notre étude part d'une liste d'environ 300 noms dérivés en *-jil*, obtenue à partir de deux grands dictionnaires d'usage coréens. En examinant les propriétés des noms en *-jil* des points de vue sémantique et syntaxique, nous distinguerons deux emplois du suffixe *-jil*, qui seront appelés par commodité «*suffixe prédicatif -jil*» et «*suffixe péjoratif -jil*». Nous pourrons vérifier en même temps que notre liste de départ n'est pas suffisante pour refléter exhaustivement cette suffixation. Comme résultat, nous proposerons des traitements différents pour ces deux types de suffixation en *-jil* dans les dictionnaires.

2. La suffixation en *-jil* et le verbe *hada* (*faire*)

2.1. Du point de vue morphologique, le suffixe *-jil* s'adjoint à une base nominale pour produire un autre nom. Prenons quelques exemples :

(1)	noms de base	suffixe *-jil*	nom dérivé
	mangchi (marteau)	*-jil*	*mangchijil* (action de marteler)
	binu (savon)	*-jil*	*binujil* (action de savonner)
	jumeg (poing)	*-jil*	*jumegjil* (action de donner un coup de poing)
	datum (dispute)	*-jil*	*datumjil* (action de se disputer)
	mogsu (menuisier)	*-jil*	*mogsujil* (activité professionnelle de menuiserie)[1]

Sémantiquement et syntaxiquement, les noms dérivés en *-jil* ont en commun les points suivants :

--- Ce sont des noms *abstraits* qui expriment une «*action*» ou une «*activité*».

[1]. Pour la transcription phonétique du coréen, nous adoptons le système qui est utilisé actuellement au Laboratoire d'Automatique Documentaire et Linguistique (LADL) pour les études sur le coréen.

--- Ils se combinent avec le verbe *hada* (*faire*), en jouant le rôle du prédicat dans la phrase dont le sujet est humain.

Autrement dit, ils font partie des noms prédicatifs (*Npréd*) du coréen qui peuvent se caractériser par la construction suivante[2] :

(2) *[Nhum]*$_0$ *W Npréd-**Acc** hada* (*[Nhum]*$_0$ *fait Dét Npréd W*)

Commençons par noter que ce verbe support (*Vsup*) occupe une place très importante dans la grammaire du coréen, en affectant la grand majorité des *Npréd* : environ 4500 *Npréd* simples, c'est-à-dire sans y compter les *Npréd* dérivés et composés, acceptent ce *Vsup*. Or, il nous faut rappeler ici une caractéristique morpho-syntaxique des *Npréd* coréens, supportés par le verbe *hada*. Avec ce type de *Npréd*, on observe une relation très systématique entre les deux phrases :

(3) a. N_0 *W Npréd-**Acc** hada* (N_0 *fait Dét Npréd W*)
= b. N_0 *W V[Npréd-hada]* (N_0 *V[Npréd-hada] W*)[3]

(4) a. *Max-ga* *sanchaig-eul* *ha-nda*
Max-**nmtf** promenade-**Acc** faire-**St**
(Max fait une promenade)
= b. *Max-ga* *sanchaigha-nda*
Max-**nmtf** se promener-**St**
(Max se promène)[4]

Il est de tradition dans la grammaire du coréen d'analyser différemment ces deux expressions *Npréd-Acc hada* et *Npréd-hada*. Dans la première où le *Npréd* est accompagné par la postposition de l'accusatif (*Acc*), *hada* est considéré comme un verbe : il s'agit d'une construction syntaxique avec verbe et complément d'objet. Dans l'autre où *hada* apparaît directement derrière le *Npréd*, le statut d'un suffixe verbal est donné à cette même forme : il s'agit ici d'un seul verbe. En fait, la description de cette relation régulière a toujours été au centre de discussions dans la linguistique coréenne de différents points de vue. Nous n'irons pas plus loin dans ce problème qui peut être terminologique. Pour nous, la forme *hada* du coréen assume deux rôles en se combinant avec les *Npréd* : suffixe verbal et *Vsup*. La relation entre les deux constructions dans (3) correspondra donc à une nominalisation entre la phrase verbale (4b) et phrase nominale à *Vsup* =: *hada* (4a). Un autre problème pourra se poser ici : comment peut-on représenter systématiquement et exhaustivement cette relation d'équivalence ? En tout cas, cette relation confirme l'existence d'un ensemble de *Npréd* caractérisés par le *Vsup* =: *hada* dans le lexique du coréen.

Etant donné que les noms dérivés en *-jil* dont nous allons parler sélectionnent en commun le verbe *hada* comme *Vsup*, la situation est la même. On a toujours une paire de phrases construites autour d'un *Npréd* dérivé en *-jil* comme dans l'exemple suivant :

(5) a. *Max-ga* *Luc-eigei* *mongdungijil-eul* *ha-ess-da*

[2] . Les notations que nous avons utilisés ici désignent :

Nhum	Nom humain dont l'indice indique la numérotation de cet actant
W	Suite de compléments éventuels
Dét	Déterminant
Acc	Postposition du complément accusatif (*leul* ou *eul*)

[3] . *V[Npréd-hada]* désigne un verbe simple construit par l'adjonction du suffixe verbal *-hada* à un *Npréd*.

[4] . ***nmtf*** et ***St*** désignent chacun la postposition du nominatif et le suffixe terminal du mode déclaratif.

Max-**nmtf** Luc-**à** bastonnade-**Acc** faire-**Pas-St**
(Max a donné (une bastonnade + un coup de bâton) à Luc)
= b. *Max-ga* *Luc-eul* *mongdungijilha-ess-da*
Max-**nmtf** Luc-**Acc** bastonner-**Pas-St**
(Max a bastonné Luc)[5]

Ainsi, une même famille de mots se forme par une suite de dérivations comme :

(6) *mongdungi* (bâton)--*mongdungi**jil*** (bastonnade)---*mondungi**jilhada*** (bastonner)
(nom simple) (nom dérivé en -*jil*) (verbe dérivé en -*hada*)

Les *Npréd* en *-jil* comme *mondungijil* (bastonnade) se forment sur la base d'un nom comme *mongdungi* (bâton), puis ces *Npréd* servent de base de dérivation, à leur tour, pour produire un verbe comme *mongdungijilhada* (bastonner) à l'aide d'un suffixe verbal *-hada*.

2.2. Quels sont les noms qui peuvent subir cette suite de dérivations ? Quand on considère les exemples donnés dans (1), on constate qu'il ne s'agit pas de noms sémantiquement homogènes : il y a des noms concrets (*Nconc*) comme *mangchi* (marteau) et *binu* (savon), un nom partie du corps (*Npc*) comme *jumeg* (poing), un *Nhum* comme *mogsu* (menuisier) et même un *Npréd* «*abstrait*» comme *datum* (dispute). On peut alors se demander si l'adjonction du suffixe *-jil* est un phénomène aléatoire ou un processus motivé sémantiquement et syntaxiquement.

Nous essayons d'apporter une réponse à cette question. D'abord, il nous semble nécessaire de distinguer deux emplois du suffixe *-jil* : il y a donc deux suffixes, dont l'adjonction n'amène pas le même effet sémantique aussi bien que syntaxique à des noms de base. On a un suffixe *-jil* qui s'applique à des noms de base pour produire des *Npréd.* La plupart des noms quī subissent cette suffixation se caractérisent naturellement par le trait «*concret*». Par contre, l'autre suffixe *-jil* s'ajoute à des *Npréd* qui, de plus, sélectionnent aussi le *Vsup* =: *hada*. Dans ce cas-là, l'adjonction du suffixe *-jil* n'apporte aucun changement syntaxique aux noms de base : le nom source et le nom dérivé en *-jil* ont exactement la même distribution. La différence entre les deux relève plutôt d'un effet sémantique : la suffixation en *-jil* ajoute aux noms sources une nuance péjorative. Considérons maintenant ces deux suffixations en *-jil* plus en détail.

3. Le suffixe *prédicatif -jil*

3.1. La plupart des noms qui peuvent subir l'adjonction du suffixe *prédicatif -jil* constituent une classe sémantiquement assez homogène. Ce sont des *Nconc* caractérisables par le classifieur «*outil*» : nous incluons aussi sous ce classifieur des objets servant à donner des coups. Une autre petite classe comprend quelques *Npc* d'êtres animés (humains ou non). Les noms prédicatifs, dérivés par suffixation en *-jil*, expriment des gestes, des coups ou des actions qui sont appropriés à chaque objet ou *Npc* concerné. Prenons des exemples :

(7)	a.	*tob* (scie)	*tobjil* (un coup de scie)
		sol (brosse)	*soljil* (un coup de brosse)
		hoicholi (fouet)	*hoicholijil* (un coup de fouet)
	b.	*bal* (pied)	*blagiljil* (un coup de pied)
		songalag (doigt)	*songalagjil* (montrer un objet du doigt)
		judungi (*bec*)	*judungijil* (un coup de bec)

[5] ***Pas*** désigne le suffixe de passé.

Notre liste comprend 105 noms en *-jil* dérivés de *Nconc* «*outil*» et 8 entrées construites sur la base d'un *Npc*. Il faut noter qu'en ce qui concerne ce dernier type, les noms en *-jil* sont parfois sémantiquement lexicalisés : par exemple le nom *sonjil* dérivé du nom *son* (*main*) désigne l'action *de réparer quelque chose*. On trouve aussi des cas qui ont une ambiguïté sémantique : une expression comme *songalgjil-il hada* peut signifier soit *montrer un objet du doigt* soit *se moquer de quelqu'un*.

On peut remarquer ici un phénomène intéressant qui relève de la grammaire comparée entre le coréen et le français, deux langues très éloignées. Dans le cadre des études sur le *Vsup* =: *donner* en français, G. Gross a fait remarquer l'existence d'un parallélisme entre l'élément *un coup de* et le suffixe italien *-ata*. (G. Gross 1989). Maintenant, nous pouvons y ajouter le cas du suffixe coréen *-jil*. Comme on pourrait déjà le remarquer dans les exemples ci-dessus, il existe une relation assez régulière entre l'élément *un coup de N* en français et *N-jil* en coréen. Selon notre observation sur 106 entrées de noms qui figurent dans la table *DRC* de G. Gross (G. Gross 1989), au moins 78 noms peuvent trouver leurs expressions équivalentes dans des *N-jil* en coréen. Quant à la sélection du *Vsup*, il y a une différence. Les noms précédés par l'élément *un coup de* en français sélectionnent souvent le *Vsup* =: *donner*, tandis qu'en coréen, les noms en *-jil* se combinent régulièrement avec le *Vsup* =: *hada* au lieu du *Vsup* =: *juda* (*donner*).

Nous observons une autre différence morpho-syntaxique entre les deux langues. La table *DRC* de G. Gross compte un certain nombre de noms déverbaux. Ici, la relation morphologique intervient directement entre nom et verbe associés :

(8) *balai* *balayer*
bêche *bêcher*
scie *scier*

Dans ce cas, l'élément *un coup de* aide à former une phrase nominale en précédant le nom :

(9) *Luc a balayé sa chambre*
= **Luc a donné un (balai + balayage) à sa chambre*
= *Luc a donné un coup de balai à sa chambre*

Selon G. Gross, l'élément *un coup de* dans (9) peut être considéré comme un opérateur de nominalisation d'événement. Or, comme le note cet auteur, les phrases reliés par la nominalisation ne sont pas toujours en relation de synonymie. Ainsi, il donne l'exemple suivant :

(10) *Luc a scié la planche*
Luc a donné un coup de scie à la planche

" Ces deux phrases ne sont pas totalement synonymes : seule la première implique que la planche a été découpée en deux morceaux. On peut supposer que l'élément *un coup de* a ici le rôle d'exprimer un aspect spécifique : l'aspect qui désigne une action rapide et qui ne se prolonge pas."[6]

Considérons maintenant le cas du coréen. En coréen, comme nous l'avons déjà dit plus haut, la relation dérivationnelle entre les noms [*N-jil*] et les verbes *[N-jil-hada*] est systématique (cf. l'exemple (6) ci-dessus). Ainsi, on a régulièrement une paire de phrases comme :

[6] . G. Gross, 1987, p. 157.

(11) a. *Luc-i* *nelpanji-leul* *[tobjilha]-ess-da*
Luc-**nmtf** planche-**Acc** scier-**Pas-St**
(Luc a scié la planche)
b. *Luc-i* *nelpanji-leul* *[tobjil]-eul* *ha-ess-da*
Luc-**nmtf** planche-**Acc** un coup de scie-**Acc** faire-**Pas-St**
(Luc a scié la planche)

Notons que ces deux phrases sont parfaitement synonymes : on ne peut y percevoir aucune différence sémantique, à l'inverse des exemples (10) du français. Or, en coréen, à côté de ces deux types de phrases, on observe encore une autre phrase comme (11c) :

(11) c. *Luc-i* *nelpanji-ei* *[tobjil]-eul* *ha-ess-da*
Luc-**nmtf** planche-**Loc**[7] un coup de scie-**Acc** faire-**Pas-St**
(Luc a donné (un + des) coups de scie à la planche)

Quand on compare cette phrase avec celle de (11b), on ne trouve qu'une différence : le nom *tobjil* dans ces deux phrases n'est pas accompagné par la même postposition, la postposition de l'accusatif dans (11b) et celle du locatif *e* (à) dans (11c). Entre ces deux phrases (11b) et (11c), l'on observe la même différence sémantique qu'entre les deux phrases françaises dans (10). Soulignons qu'il s'agit d'un phénomène qu'on retrouve avec la plupart des noms en *-jil* ici concernés. Ainsi, on peut dire que les *Npréd* en *-jil* se caractérisent par les trois constructions parallèles suivantes :

(12) a. N_0 N_1-**Acc** [*N-jil-hada*]
b. N_0 N_1-**Acc** [*N-jil*]-**Acc** *hada*
c. N_0 N_1-**Loc** [*N-jil*]-**Acc** *hada*

3.2. Nous avons dit plus haut que la suffixation en *-jil* s'applique à des noms. Mais il faut ici nuancer cette remarque. Nous avons trouvé environ 30 noms en *-jil* qui sont formés sur des verbes simples. Or, pour ces noms déverbaux, le suffixe *-jil* ne peut pas être directement ajouté derrière la racine verbale. Comme dans les exemples ci-dessous, un suffixe de nominalisation comme *-(eu)m*, *-i* ou *-gai* [8] sert d'intermédiaire pour ce type de dérivation :

(13) ***Rv-Mnom-jil*** [9]
dali-m-jil (*repasser / repassage*)
gal-i-jil (*labourer / labourage*)
ddeu-gai-jil (*tricoter / tricotage*)

Une racine verbale obtient d'abord un statut nominal à l'aide d'un suffixe de nominalisation, puis le suffixe *-jil* s'y ajoute. Une forme comme *tali-m* (*Rv-Mnom*) n'a pas d'autonomie dans le lexique du coréen : ce n'est pas un nom, mais seulement une forme nominale qui subit d'adjonction du suffixe *-jil*. Les phrases qui ont ce type de nom comme prédicat sont en relation d'équivalence avec les phrases verbales correspondantes :

(14) a. *Ida-ga* *chima-leul* *dali-ess-da*
Ida-**nmtf** jupe-**Acc** repasser-**Pas-St**

[7] . ***Loc*** désigne la postposition du locatif ***e*** (***à***).

[8] . Ces suffixes servent en général à produire des noms déverbaux en coréen.

[9] . ***Rv*** racine verbale, ***Mnom*** suffixe de nominalisation.

(Ida a repassé sa jupe)

b. *Ida-ga* *chima-(leul + ei)* *dalimjil-eul* *ha-ess-da*
Ida-**nmtf** jupe-(**Acc** + **Loc**) repassage-**Acc** faire-**Pas-St**
(Ida a donné un coup de repassage à sa jupe)

Syntaxiquement, cette petite classe des noms déverbaux en *-jil* a les mêmes comportements que ceux dérivés de *Nconc* dont nous avons parlé.

4. Le suffixe péjoratif *-jil*

Il s'agit ici d'une suffixation en *-jil* tout à fait différente que celle dont nous avons parlé jusqu'à maintenant. Cette suffixation opère sur des *Npréd* qui sélectionnent eux-mêmes le *Vsup* =: *hada*. En plus, du point de vue syntaxique, les noms de base et les noms dérivés ont les mêmes distributions. La différence entre eux est sémantique. L'adjonction du suffixe *-jil* apporte un effet péjoratif aux noms de base et aussi parfois un sens aspectuel itératif. Les noms qui subissent cette suffixation peuvent se classer en deux types suivants selon leur trait sémantique.

4.1. Un ensemble de noms qui désignent des personnes exerçant un métier, une profession ou jouissant d'un statut social particulier accepte cette suffixation. Nous utilisons ici un critère formel pour définir un type de noms que nous appellons «*noms de métier*» (*Nmétier*) les noms qui apparaissent en même temps dans les deux constructions suivantes :

(15) a. *N_0 Nmétier ida* (*N_0 est Nmétier*)
= b. *N_0 Nmétier-***Acc** *hada* (*N_0 fait Nmétier*)

Prenons un exemple :

(16) a. *Max-neun* *mogsu-ida*
Max-**nmtf** menuisier-**être**
(Max est menuisier)

b. *Max-neun* *mogsu-leul* *ha-nda*
Max-**nmtf** menuisier-**Acc** faire-**St**
(Max est menuisier)[10]

Sémantiquement, ces deux phrases sont en relation d'équivalence. On pourra dire que dans ce cas le *Vsup* =: *hada* correspond à une variante stylistique de la copule *ida* (*être*) supportant les *Nmétier*. Cependant ces deux types de phrases n'ont pas les mêmes propriétés syntaxiques. Dans les phrases supportés par *hada* comme (16b), les *Nmétier* n'acceptent aucun modifieur :

(17) a. *Max-neun* *hullyunghan* *mogsu-ida*
Max-**nmtf** bon menuisier-**être**
(Max est un bon menuisier)

b. **Max-neun* *hullyunghan* *mogsu-leul* *ha-nda*
Max-**nmtf** bon menuisier-**Acc** faire-**St**

[10]. La traduction française mot-à-mot de cette phrase sera *Max fait le menuisier*. Mais cette traduction ne correspond pas à la phrase coréenne concernée.

A côté d'une paire de phrases comme dans (16), on observe régulièrement une phrase construite autour d'un nom dérivé en *-jil* :

(18) a. *Max-neun* *mogsujil-eul* *ha-nda*
Max-**nmtf** menuiserie-**Acc** faire-**St**
(Max fait de la menuiserie)

Cependant, les noms dérivés comme *mogsujil* qui désignent une activité professionnelle - souvent avec un sens péjoratif - ne peuvent être supportés par la copule *ida* . Ainsi, on n'a pas :

(18) b. **Max-neun* *mogsujil-ida*
Max-**nmtf** menuiserie-**être**
(Max est menuiserie)

On ne trouve qu'une différence sémantique ou plutôt stylistique entre les deux phrases (16b) et (18a) où le *Nmétier mogsu* (menuisier) et le nom dérivé *mogsujil* sont supportés par *hada* : ces deux phrases comprennent toutes les deux le sens que Max exerce professionnellement le métier de menuisier. L'utilisation du nom dérivé *mogsujil* attribue à la phrase une nuance péjorative.

Il faut une remarque ici. Selon notre test sur environ 900 *Nmétier* enregistrés dans les deux dictionnaires que nous avons consultés, la suffixation en *-jil* s'applique quasi-systématiquement. Pourtant, nous n'avons trouvé dans les mêmes dictionnaires que 7 noms dérivés en *-jil* associés à ce type de *Nmétier*. D'un côté, on peut faire la supposition suivante : le fait que ces noms en *-jil* appartiennent aux mots familiers a fait prendre aux auteurs la décision de les enlever de leurs dictionnaires. D'un autre côté, ce phénomène vient à l'appui de notre observation que la suffixation en *-jil* des *Nmétier* est productive et régulière. En tout cas, la forte productivité de cette suffixation en *-jil* mérite d'être remarquée dans la description du coréen.

4.2. Considérons maintenant les paires de phrases suivantes :

(19) a. *Max-ga* *Luc-eigei* *jenhwa-leul* *ha-ess-da*
Max-**nmtf** Luc-**à** téléphone-**Acc** faire-**Pas-St**
(Max a (téléphoné + donné un coup de téléphone à Luc)
b. *Max-ga* *Luc-eigei* *jenhwajil-eul* *ha-ess-da*
Max-**nmtf** Luc-**à** des coups de fil-**Acc** faire-**Pas-St**
(Max a donné des coups de fil a Luc)
(20) a. *Max-ga* *Luc-wa* *maldatum-eul* *ha-ess-da*
Max-**nmtf** Luc-**avec** dispute-**Acc** faire-**Pas-St**
(Max a eu une dispute avec Luc)
b. *Max-ga* *Luc-wa* *maldatumjil-eul* *ha-ess-da*
Max-**nmtf** Luc-**avec** disputaillerie-**Acc** faire-**Pas-St**
(Max a eu des disputailleries avec Luc)

Nous observons qu'un certain nombre de *Npréd* caractérisés par le *Vsup* =: *hada* acceptent aussi la suffixation en *-jil péjoratif.* Comme nous pouvons le voir dans les exemples (19) et (20), il ne se trouve pas de changement syntaxique entre la phrase comportant le *Npréd* de base et celle ayant le nom dérivé en *-jil*. La suffixation en *-jil* a pour premier effet de dévaloriser ou d'atténuer le sens du nom de base : par exemple *jenhwa* (*téléphone*) / *jenhwajil* (*coups de fil*) dans (19) ou *maldatum* (*dispute*) / *maldatumjil* (disputaillerie) dans (20). Un second effet de cette suffixation est aspectuel.

Le suffixe *-jil* semble comporter dans ce cas une signification d'aspect itérative : il s'agit d'une action répétée plusieurs fois. Or, les *Npréd* qui subissent cette suffixation sont syntaxiquement très hétérogènes, bien qu'ils sélectionnent en commun le *Vsup* =: *hada*. Cette suffixation semble constituer une propriété morpho-sémantique spécifique de certains *Npréd*. Nous proposons ainsi que ce type de suffixation fasse partie de la description du *Npréd* de base.

5. En guise de Conclusion

Dans ce travail, nous avons décrit les deux suffixation en *-jil* en coréen : le suffixe *prédicatif -jil* et le suffixe *péjoratif -jil*. Du point de vue de la productivité aussi bien que des points de vue sémantique et syntaxique, ces deux suffixations ont des aspects différents. Ainsi, nous proposons en conclusion que les deux types de noms, dérivés par chaque suffixation en *-jil*, soient traités différemment dans les dictionnaires, et notamment dans les dictionnaires électroniques. Les noms dérivés par l'adjonction du suffixe prédicatif *-jil*, qui forment une liste relativement restreinte, devront figurer indépendamment de leurs noms de base dans les dictionnaires, avec un ensemble d'informations grammaticales qui les caractérisent. Par contre, en ce qui concerne des noms dérivés par le suffixe péjoratif *-jil*, ils peuvent être représentés, dans le lexique-grammaire du coréen, sous forme d'une colonne dans les tables de leurs noms de base. Autrement dit, la formation de ce type de noms dérivés pourra correspondre à une information morpho-sémantique des noms de base concernés.

REFERENCES

Dictionnaires

Kim, Min-Soo ; Ko, Young-Geun ; Lim Hong-Pin ; Lee, Seung-Jae, 1991, *geumseungpan Guge Dai Sajen* (Geumseung Dictionnaire du Coréen), Séoul : GeumSeung Presse.

Sin, Gi-Chel ; Sin, Yong-Chel, 1990, *Sai Ulimal Keun Sajen* (Nouveau Dictionnaire de la Langue Coréenne), Séoul : Samseung Presse.

Giry-Schneider, Jacqueline, 1978, *Les Nominalisations en français : l'opérateur faire dans le lexique*, Genève : Droz.

Giry-Schneider, Jacqueline, 1987, *Les prédicats nominaux en français : les phrases simples à verbe supports*, Genève : Droz.

Gaston, Gross, 1984, Etude syntaxique de deux emplois du mot coup, *Linguisticae Investigationes* VIII : 1, Amsterdam : John Benjamins B.V.

Gaston, Gross, 1989, *Les constructions converses du français*, Genève : Droz.

Gross, Gaston, 1994, Classes d'objets et description des verbes, *Langages* 115, Paris : Larousse.

Gross, Maurice, 1981, Les bases empiriques de la notion de prédicat sémantique, *Langages* 63, Paris : Larousse.

Nam, Jee-Sun, 1994, *Dictionnaire des noms simples du coréen*, Rapport Technique N° 46, Paris : LADL.

Nam, Jee-Sun, 1996, *Dictionnary of Korean Simple Verbs*, Rapport Technique N° 49, Paris : LADL.

Shin, Kwang-Soon, 1994, *Le verbe support hata en coréen contemporain* : Morpho-syntaxe et comparaison, Thèse de Doctorat, Université Paris 7.

A Lexical Semantic Database for Verbmobil

JOHANNES HEINECKE – KARSTEN L. WORM

Abstract

This paper describes the development and use of a lexical semantic database for the Verbmobil speech–to–speech machine translation system. The motivation is to provide a common information source for the distributed development of the semantics, transfer and semantic evaluation modules and to store lexical semantic information application–independently.

The database is organized around a set of abstract semantic classes and has been used to define the semantic contributions of the lemmata in the vocabulary of the system, to automatically create semantic lexica and to check the correctness of the semantic representations built up. The semantic classes are modelled using an inheritance hierarchy. The database is implemented using the lexicon formalism $\mathcal{L}e\mathcal{X}4$ developed during the project.

1 Introduction

The distributed development of the modules of a large natural language processing system at different sites makes interface definitions a vital issue. It becomes even more urgent when several modules with the same intended functionality are developed in parallel and should be indistinguishable with respect to their input–output–behaviour.

Another important issue is the acquisition and maintenance of lexical information which should be stored independently of an application to make it (re)usable for different purposes.

This paper describes the design and use of the Verbmobil Semantic Database which we developed in order to deal with these issues in the area of lexical semantics in Verbmobil.

[*] The research reported in this paper was supported by the German Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie under contract 01 IV 101 R and 01 IV 101 G6. We wish to thank our colleagues in the Verbmobil project, especially Ronald Bieber, Johan Bos, Michael Dorna, Markus Egg, Martin Emele, Björn Gambäck, Gunter Gebhardi, Manfred Gehrke, Julia Heine, Udo Kruschwitz, Daniela Kurz, Kai Lebeth, Christian Lieske, Yoshiki Mori, Rita Nübel, Joachim Quantz, Sabine Reinhard, Stefanie Schachtl, Michael Schiehlen, Feiyu Xu.

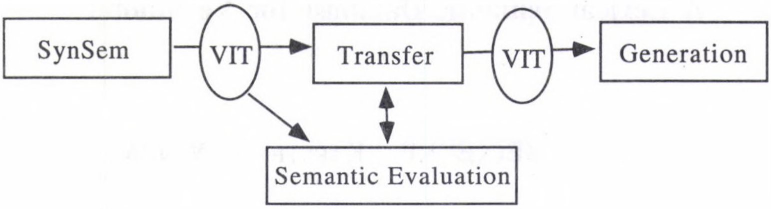


Figure 1: The relevant part of the Verbmobil architecture (simplified)

2 The Verbmobil Project

The Verbmobil project[1] (Wahlster 1993; Bos et al. 1996) aims at the development of a speech–to–speech machine translation system for face–to–face appointment scheduling dialogues.

The application scenario of Verbmobil is that a speaker of German and a speaker of Japanese try to schedule an appointment. They communicate mostly in English, which they understand better than they speak it. If they they want to say something they cannot express in English, they can have the Verbmobil system translate from both their native languages to English.

The system is being developed by about 30 partners from academia and industry in Germany, the United States and Japan. A first version, the *Demonstrator*, was completed in early 1994; for autumn 1996 the release of the *Research Prototype* is scheduled, which marks the end of the first project phase. A second phase is expected to start in 1997.

Verbmobil employs a semantic transfer approach to translation (Dorna and Emele 1996), i. e. an input utterance is syntactically analyzed, a semantic representation of the content is built up,[2] and this source language semantic representation is mapped to a target language semantic representation by the transfer module. This representation is the input for the target language generation. Additionally, a dialogue processing module and a semantic evaluation module keep track of the discourse and answer disambiguation queries. (The relevant part of the system architecture is shown in figure 1.)

3 Motivation and Goals for the Semantic Database

The architecture of Verbmobil makes it necessary for the semantics, transfer, semantic evaluation and generation modules to agree on the format and contents of the semantic representations they exchange. E. g. the developers of the transfer module need to know how the semantics of the different lemmata in the vocabulary is represented in the structures produced by the syntax–semantics module (*SynSem* for short), i. e. which predicates and structures they have to map to the target language. On the other hand, semantics need to know which readings have to be distinguished by transfer in order to arrive at correct translations.

This need for information becomes even more urgent when, like in Verbmobil, there are several SynSem modules (two for German, one for Japanese), which have to produce compatible output, and the different modules are developed independently and in parallel by several partners at different sites.[3]

[1] Information about Verbmobil, such as available reports, can be retrieved via the World Wide Web: `http://www.dfki.uni-sb.de/verbmobil/`.

[2] Syntactic and semantic analysis proceed in parallel in the *Research Prototype*, while they were two consequent processing steps in the *Demonstrator*.

[3] In the following, we concentrate on the Semantic Database for German. The Japanese version follows the same principles.

```
vit( segment_description(ttestr4u1, yes,
                                     'wir machen einen termin aus'),
     [termin(l6,i2),                         % Semantics
      ausmachen(l4,i1),
      decl(l5,h1),
      arg1(l4,i1,i3),
      arg3(l4,i1,i2),
      ein_card_qua(l3,i2,l1,h2,1),
      pron(l9,i3)],
     l5,                                     % Main Label
     [s_sort(i1,ment_communicat_poly),       % Sorts
      s_sort(i2,&(space_time,time_sit_poly)),
      s_sort(i3,&(human,person))],
     [prontype(i3,sp_he,std)],               % Discourse
     [num(i3,pl),                            % Syntax
      pers(i3,1),
      gend(i2,masc),
      num(i2,sg),
      pers(i2,3),
      cas(i2,acc),
      cas(i3,nom)]
     [ta_mood(i1,ind),                       % Tense and Aspect
      ta_tense(i1,pres)],
     [ccom_plug(h2,l2),                      % Scope
      ccom_plug(h1,l3),
      leq(l2,h2),
      leq(l2,h1),
      leq(l3,h1)]
     [pros_mood(l5,decl)],                   % Prosody
     [sem_group(l2,[l4]),                    % Groupings
      sem_group(l1,[l6])])
```

Figure 2: A VIT for *Wir machen einen Termin aus* ("We arrange an appointment").

As a frame for the exchange of semantic representations a common format, the *Verbmobil Interface Term*, `VIT` for short, has been defined (Bos, Egg, and Schiehlen 1996). The `VIT` is the central data structure used at the interfaces between the language modules of Verbmobil. A `VIT` is a ten–place term with slots for an utterance identifier, a list of labelled semantic predicates, a pointer to the most prominent predicate, sortal, anaphoric and syntactic information, temporal and aspectual properties, scope relations and prosodic features. Figure 2 shows a `VIT` for the sentence *Wir machen einen Termin aus* (We arrange an appointment).

A `VIT` is an underspecified representation for a set of discourse representation structures (Kamp and Reyle 1993) in which the scope of operators is not fixed yet. In the example shown in figure 2 both the scope of the declarative sentence mood operator, `decl/2`, and of the quantifier/indefinite, `ein_card_qua/5`, are left unspecified. They introduce *holes*, written as `h1` and `h2`, as their scope, which can be *plugged* by structures subordinated to them by means of less or equal constraints, written as `leq/2`. Different ways of plugging the holes result in different readings. In addition to the `leq/2` constraints determining all possible readings, we supply a default scoping based on syntactic structure in the predicates `ccom_plug/2`.[4]

All semantic predicates in the `VIT` are labelled (their first argument is the label). This allows us to group several predicates together (using the `sem_group/2` predicate) and form complex substructures which can occur in the scope of operators.

Apart from the purely semantic information mentioned so far, a `VIT` contains sortal constraints associated with discourse markers, discourse information about anaphoric elements, syntactic agreement and tense information. Since Verbmobil deals with spoken input, we also represent prosodic information in the `VIT`.[5]

What is needed then in addition to the `VIT` data structure definition is a definition of the `VIT`'s contents, for each lemma in the vocabulary of the system a definition of the semantic predicates and other types of information, e g. sortal restrictions, it introduces in the slots of the `VIT`. E. g. for the verb *ausmachen* in the example above, we need to specify that it introduces a predicate `ausmachen(L1,I1)` together with argument roles `arg1(L1,I1,I2)` and `arg3(L1,I1,I3)` in the semantics slot and `sort(I1,ment_communicat_poly)` in the sorts slot.

If a source providing this kind of information to the developers of the separate modules is available, the modules which deliver (the two SynSem modules) or process (especially the transfer module) `VIT`s conforming to this definition can be developed in parallel. It would also be desirable to use this information source directly in the construction of the linguistic knowledge bases to guarantee consistency between the output and the specifications.

To meet these goals, we have developed the *Verbmobil Semantic Database*, which we will describe in the remainder of this paper.

4 Design and Implementation of the Database

The semantic database is organized around a set of abstract semantic classes (Bos, Egg, and Schiehlen 1996), which are used to classify the lemmata in the vocabulary. It is implemented using the lexicon formalism ℒℯ𝒳4.

4.1 Semantic Classes

The semantic classes in use are originally based on a morpho–syntactic classification of the words in the vocabulary of the system which has been refined to account for the semantic properties. This has

[4] For more details on this underspecified approach to semantics, the reader might consult (Bos 1995; Bos et al. 1996).

[5] The `VIT` in figure 2 has been generated from typed input and thus contains no real prosodic information.

Class	PredScheme	Example
transitive_verb	R(L,I), argX(L,I,I1), argY(L,I,I2)	*treffen*
common_noun	R(L,I)	*Termin*
det_quant	R(L,I,H)	*jeder*
demonstrative	demonstrative(L,I,L1)	*dieser*
wh_question	whq(L,I,H), tloc(L2,I2,I1), time(L1,I1)	*wann*

Table 1: A few examples of semantic classes

been decided upon, because words of a certain word–class usually have the same semantic properties. In the example given below, it is shown that transitive verbs all need an instance and two arguments with their semantic/thematic roles.

For each semantic class a representation scheme, called the *predscheme*, has been defined, which specifies the predicates together with their arity and arguments appearing in a VIT for instances of the class.

As an example consider the class transitive_verb. A transitive verb is represented as R(L,I), argX(L,I,I1), argY(L,I,I2).[6] I. e., it introduces some relation R and two thematic roles (I is the event variable, L a label used to refer to the verb's semantic contribution, and I1 and I2 are the instances filling the roles). The verb's relation and the thematic roles it assigns have to be defined for each verb in the database. Cf. table 1 for further examples of semantic classes together with their predschemes.

4.2 The Lexicon Formalism $\mathcal{L}e\mathcal{X}4$

The semantic database makes use of the lexicon formalism $\mathcal{L}e\mathcal{X}4$ developed in the course of the Verbmobil project (Gebhardi and Heinecke 1995a; Gebhardi 1996).

The **Lexicon Formalism** $\mathcal{L}e\mathcal{X}4$ has been used since summer 1994 within Verbmobil's lexicon group. It is based on feature-structures (permitting disjunction and negation) embedded in an inheritance hierarchy of classes.

In $\mathcal{L}e\mathcal{X}4$ the task of constructing a lexicon is split up into four parts:

1. Modelling the lexicon (i.e. its linguistic classes),

2. data-acquisition (can be done at the same time by different contributors),

3. definition of the application-interface (data can be compiled into every format needed after being processed by the $\mathcal{L}e\mathcal{X}4$-machine), and

4. efficient storage.

Modelling a lexicon involves defining classes, their appropriate features, and inheritance relations between classes. Examples for defining classes will be given below in section 4.3; appropriateness of features is dealt with in the remainder of this section. For data acquisition, a graphical acquisition tool has been implemented (Heinecke 1996). How the application interface is used in the context of the semantic database will be shown in section 5. Part of the application interface is the $\mathcal{L}e\mathcal{X}4$-TRAFO which outputs the stored information in any format required. A database system for efficient storage has been developed (Kruschwitz and Gebhardi 1996)

[6] X and Y stand for the values $\{1, 2, 3\}$, since arg1, arg2, arg3 are the thematic roles used in Verbmobil.

Among other formalism constructs, the possible values of a feature can be specified in two ways. If there is no restriction on the value of a feature, it is assigned the *most general value* keyword (`top`):

```
predname: top
```

Otherwise, the formalism allows to define the appropriateness conditions of a feature, using disjunctions to specify the appropriate values as in the following example (the underlined values are the appropriate ones which can be assigned to the feature `sort_of_inst`):

```
sort_of_inst: ( abstract \ anything \ communicat_result_poly \
                communicat_sit \ person )
```

For constructing morphological lexica, inflection or lexical rules can easily be implemented to generate multiple instances of a single entry (Gebhardi and Heinecke 1995b; Heinecke and Gebhardi 1995).

Database entries, called *bases*, are instances of a class. Consequently, they assign values to the features they inherit from their class which are not yet fully specified by the class definition. For a verb's base, e. g., one has to specify its predicate name, thematic roles, the sort of its instance, etc.

4.3 Semantic Classes and their Representation in LeX4

The abstract semantic classes of section 4.1 have been modelled in the lexicon formalism LeX4 along the following lines.

Firstly, a general superclass `semdb_c` is defined from which all classes inherit features for the lemma, the main predicate's name, the part of speech etc. The individual subclasses corresponding to the abstract semantic classes additionally introduce a specific predscheme for each predicate associated with words of this class and features for sortal information, thematic rôles etc.

```
class semdb_c :< top >:       % - Main class from which
                              %   all classes inherit.
    syntax_link: top &        % - Link to syntactic lexicon.
    predname: top &           % - Name of the semantic predicate.
    lemma: top &              % - Lemma of the entry.
    pos: top .                % - Part of Speech of the occurrences
                              %   in the corpora.
```

While the abstract semantic classes are not hierarchically organized, their modelling in LeX4 makes use of a hierarchy to capture generalizations. For instance, we integrate all properties the verb classes have in common and place them in an abstract verb class `verb_c` from which all verb classes, e. g. `transitive_c`, inherit, cf. figure 3 (classes corresponding to semantic classes are shown in boldface) and below.

```
class verb_c :< semdb_c >:         % - All verbal classes inherit this.
    sort_of_inst: top .            % - Sort of eventuality.


class transitive_c :< verb_c >:         % - Transitive verbs
    semclass: transitive_verb &         % - Semantic class.
    predscheme: 'L,I' &                 % - PredScheme for the PredName
                                        %   of all transitive verbs.
    predscheme_a1: 'L,I,I1' &           % - PredScheme for the first
```

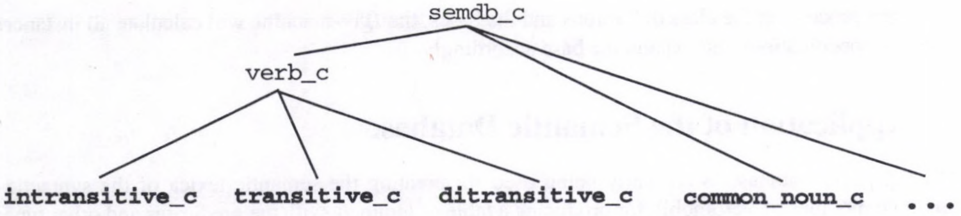


Figure 3: Part of the class hierarchy

```
    predscheme_a2: 'L,I,I2' &               %  and the second argument.
    role_a1: (arg1 \ arg2 \ arg3) &         %  - Thematic roles of the arguments
    role_a2: (arg1 \ arg2 \ arg3) .         %  of the verb (restricted
                                            %  to three valid values).
```

As a second example, consider the following definition for the $\mathcal{L}e\mathcal{X}^4$ equivalent of the abstract semantic class common_noun:

```
class common_noun_c :< semdb_c >:   % - Standard nouns
    predscheme: 'L,I' &             % - PredScheme for standard nouns.
    sort_of_inst: top &             % - Sort of instance.
    semclass: common_noun           % - Semantic class.
```

4.4 Representation of Lemmata

A base for a lemma consists of its classification together with its idiosyncratic properties in terms of feature values; it inherits the feature values which are specified in the definition of the class. Among the idiosyncratic information we have predicate names, sortal restrictions etc. Thus an entry inherits the predscheme from the class, while the concrete predicate name in the predscheme is defined in the entry itself.

```
base 'Termin' :<< common_noun_c >>:      % - The entry 'Termin'
                                         %   inherits its structure from
                                         %   from the class 'common_noun_c'.
    pos: 'NN' &                          % - Further individual
    lemma: 'Termin' &                    %   specification for
    syntax_link: 'termin' &              %   the current entry.
    predname: 'termin' &
    sort_of_inst: 'time_sit_poly'


base 'ausmachen' :<< transitive_c >>:    % - The entry 'ausmachen'
                                         %   inherits its structure from
                                         %   the class 'transitive_c'.
    pos: 'VVFIN;VVINF' &                 % - Further specifications.
    lemma: 'ausmachen' &
    syntax_link: 'ausmachen' &
    predname: 'ausmachen' &
    sort_of_inst: (communicat_sit \ mental_sit) &
    role_a1: 'arg1' &
    role_a2: 'arg3'
```

When processing the class definitions and the bases, the ℒℰ𝒳4-machine will calculate all instances from the specifications and expand the base accordingly.

5 Application of the Semantic Database

The Semantic Database is currently being used for creating the semantic lexica of the syntactic–semantic modules of Verbmobil, for producing a table of lemmata with the predicates and other types of information they introduce in a VIT and for checking the correctness of the generated interface terms automatically; it can also be accessed via the World Wide Web.

A similar procedure is used to generate the semantic lexicon etc. for the Japanese syntactic–semantic module of Verbmobil (Mori 1996).

5.1 Creation of the Semantic Lexicon

Consider the compilation of the semantic lexicon from the database for the German SynSem module SynSemS3.[7] To guarantee consistency between the output of the SynSem module and the specifications in the database, the semantic lexicon is generated out of the semantic database.

After the ℒℰ𝒳4–machine has processed the entries and expanded them according to the class definitions, the ℒℰ𝒳4–TRAFO compiles the ℒℰ𝒳4 output into the format required for the semantic lexicon.

```
sort1_trafo(Base, Class,               % - Default rule for entries
  [ predname:Predn,                    %   with one sort.
    syntax_link:Sl,
    sort_of_inst:Si,
    usb_macro:M
  ] ) =>
fmt("sem_lex(Cat, ~w) short_for~n       ~w(Cat, ~w, (~w)) .~n",
    [Sl, M, Predn, Si], []).


trans_trafo(Base, Class,               % - Rule for bivalent verbs.
  [ predname:Pn,
    syntax_link:Sl,
    sort_of_inst:Si,
    role_a1:R1,
    role_a2:R2,
    usb_macro:M
  ] ) =>
fmt("sem_lex(Cat, ~w) short_for~n      ~w(Cat, ~w, (~w), [~w,~w]) .~n",
    [Sl, M, Pn, Si, R1,R2], []).
```

The two examples above appear in the semantic lexicon as:

```
sem_lex(Cat, termin) short_for
     common_noun_sem(Cat, termin, (time_sit_poly)) .
sem_lex(Cat, ausmachen) short_for
```

[7] SynSemS3 is the syntactic–semantic module developed by Siemens AG (syntax), University of the Saarland and University of Stuttgart (semantics). The other SynSem module developed by IBM Germany makes use of the table output (cf. section 5.2) of the database to create a semantic lexicon.

```
trans_verb_sem(Cat, ausmachen, (communicat_sit;mental_sit),
                  [arg1,arg3]) .
```

The syntactic lexicon contains calls to the macro `sem_lex/2` which is expanded in the semantic lexicon as shown above. The mapping from syntactic to semantic lexical entries is achieved via the second argument of `sem_lex/2`, which originates from the feature `syntax_link` in the semantic database.[8]

5.2 Table–based Representation

Apart from compiling out semantic lexica, we generate a table of lemmata together with their semantic representations and additional information out of the database by using a different set of transformation rules for ℒℰ𝒳4-TRAFO. This table is used by the transfer developers as a basis for writing transfer rules and as an information source for the automatic correctness check on VIT representations.

```
 transitive_trafo(Base, Class,         % - Rule for bivalent verbs.
    [ lemma:Lm,
      pos:Pos,
      semclass:Semc,
      predname:Pn,
      predscheme:Ps,
      predscheme_a1:Ps1,
      predscheme_a2:Ps2,
      role_a1:Ra1,
      role_a2:Ra2,
      sort_of_inst:Si,
      inst_link:Il,
      sort_a1:Sa1, a1_link:Al1,
      sort_a2:Sa2, a2_link:Al2
    ] ) =>
  fmt("~w ~w ~w ~w,~w,~w ~w ~w(~w),~w(~w),~w(~w) ~w/~w,~w/~w,~w/~w - -~n",
     [ Base, Lm, Pos, Pn,Ra1,Ra2, Semc, Pn,Ps, Ra1,Ps1, Ra2,Ps2,
       Il,Si,Al1,Sa1,Al2,Sa2], []).

  default_ps1_inst1(Base, Class,                % - Default rule for entries with
    [ lemma:Lm,                                 %   one PredScheme and one Sort
      pos:Pos,                                  %   (used e.g. by 'common_noun').
      semclass:Semc,
      predname:Pn,
      predscheme:Ps,
      sort_of_inst:Si
    ] ) =>
  fmt("~w ~w ~w ~w ~w ~w(~w) ~w  - -~n",
    [ Base, Lm, Pos, Pn, Semc, Pn,Ps, Si], []).
```

In the table output the two examples above appear as:

```
Termin Termin NN termin common_noun termin(L,I) I/time_sit_poly - -
ausmachen ausmachen VVFIN;VVINF ausmachen,arg1,arg3 transitive_verb ...
 ausmachen(L,I),arg1(L,I,I1),arg3(L,I,I2) I1/communicat_sit;mental_sit - -
```

[8] The first argument of `sem_lex/2` ranges over entry nodes of the feature structures of the lexical entry.

In general the concept of TRAFO is trying to map the output of the LeX4-machine onto the first matching rule in the rule system. Thus only a few class specific rules are necessary, default rules will cover the entries of the majority of the classes to be transformed.

6 Summary

We have successfully used the semantic database to deal with about 2000 German and 150 Japanese lemmata for version 1.0 of the Research Prototype in the way described, especially to generate semantic lexica for the German syntax–semantics module SynSemS3, and the Japanese one developed by DFKI Saarbrücken and the University of the Saarland.

The use of the semantic database by both the semantics module and the transfer module guarantees consistency between the representations produced by the semantics module and the expectations of the transfer module, while both can be developed in parallel.

References

Bos, J. (1995). Predicate logic unplugged. In *Proceedings of the* 10*th Amsterdam Colloquium*, Amsterdam, The Netherlands, pp. 133–142. ILLC/Department of Philosophy, University of Amsterdam.

Bos, J., M. Egg, and M. Schiehlen (1996). Definition of the Abstract Semantic Classes for the Verbmobil Forschungsprototyp 1.0. Verbmobil–report, Universität des Saarlandes, Computerlinguistik, Saarbrücken.

Bos, J., B. Gambäck, C. Lieske, Y. Mori, M. Pinkal, and K. Worm (1996). Compositional semantics in Verbmobil. In *Proc. of the 16^{th} COLING*, Copenhagen, Denmark.

Dorna, M. and M. C. Emele (1996). Semantic–based transfer. In *Proc. of the 16^{th} COLING*, Copenhagen, Denmark.

Gebhardi, G. (1996). LeX4— yet another lexicon formalism. Budapest. In this Volume.

Gebhardi, G. and J. Heinecke (1995a). Lexikonformalismus LeX4. Verbmobil Technisches Dokument 19, Humboldt–Universität zu Berlin, Computerlinguistik, Berlin.

Gebhardi, G. and J. Heinecke (1995b). Substantivflexion in LeX4. Ein Applikationsbericht. Verbmobil–Memo 62, Humboldt–Universität, Computerlinguistik, Berlin.

Heinecke, J. (1996). Lexikonakquisitionstools für den Lexikonformalismus LeX. Verbmobil Technisches Dokument 42, Humboldt–Universität zu Berlin, Computerlinguistik, Berlin.

Heinecke, J. and G. Gebhardi (1995). Konjugation der Verben im Lexikonformalismus. Verbmobil–Memo 63, Humboldt–Universität, Computerlinguistik, Berlin.

Kamp, H. and U. Reyle (1993). *From Discourse to Logic*. Dordrecht: Kluwer Academic Publishers.

Kruschwitz, U. and G. Gebhardi (1996). The LeX4–database system. Budapest. In this Volume.

Mori, Y. (1996). Multiple discourse relations on the sentential level in Japanese. In *Proc. of the 16^{th} COLING*, Copenhagen, Denmark.

Wahlster, W. (1993). Verbmobil: Translation of face-to-face dialogues. In *Proceedings of the 3^{rd} European Conference on Speech Communication and Technology*, Berlin, Germany, pp. 29–38.

The Data is The Dictionary: Corpus at the Cutting Edge of Lexicography

RAMESH KRISHNAMURTHY

Abstract

The aim of this paper is to demonstrate that using a general corpus of natural language can provide better answers to questions about specific language usage than any published dictionary. Further, the paper will show that the actual process of consulting a corpus is in itself a language learning lesson of enormous value. Objections about the unavailability and the excessive cost of corpus access are no longer as valid as they once were, and it is imperative that all language institutions involved in teaching or research should acquire corpus access as a priority.

Introduction

The aim of this paper is to show that using a general corpus of natural language can provide better answers to specfic questions about language use than traditional methods, and that the process of consulting a corpus is in itself a language learning lesson. Until fairly recently, the objection could be fairly made that access to corpora was restricted to a few professionals and academics. However, as corpus access gradually comes within reach of most language teaching institutions, and corpora become available for many more languages, this objection will soon cease to be a valid one. Similarly, although the cost of access may still be an inhibiting factor, the cost is likely to decrease as opportunities for access increase.

Questions about specific language usage

When we come across an unfamiliar phrase or sentence for the first time, (in fact, what happens more commonly is that we come across familiar phrases or sentences in an unfamiliar usage or context), various questions will inevitably and necessarily arise, whether we are native speakers of a language or learners of the language. We may think that it happens less frequently to native speakers, but perhaps this is because native speakers are often uninterested, inattentive, or simply lazy, having understood the gist of what has been said or written from the general context and not being bothered about the linguistic details. Accessing a corpus on a regular basis helps to make us realize that we are all learners at some level, because the data almost always shows us something we didn't know before, or had never bothered to think about.

Let us look at a few example sentences that include the phrase '**cutting edge**' used in the title of this paper:

> (A) 'Waterman says that he is suffering because he is at the **cutting edge** of rail privatisation.' (*The Independent, London, 2nd July 1995*)
>
> (B) 'The area is at the **cutting edge** of residential subdivision.'
> (*Courier Mail, Brisbane, 2nd June 1995*)
>
> (C) 'Tracks at the **cutting edge** of dance music tend to be one-off collaborations'.
> (*Vogue magazine, London, September 1992*)
>
> (D) 'The hawker is at the **cutting edge** of a piracy industry.'
> (*The Guardian, London, 28th January 1995*)
>
> (E) 'They're at the **cutting edge** of all the things you need.'
> (*The Wall Street Journal, New York, 16th August 1989*)

Any of these could easily have been the first example we came across. Here are some of the questions that we may ask ourselves:

What does it mean, 'to be at the cutting edge of something'? Is it good or bad? It is difficult to tell from these examples: (1) mentions 'suffering' as a result of being at the cutting edge, (3) is rather dismissive with its 'one-off collaborations', and (4) refers to 'hawker' and 'piracy', so that doesn't sound too good; (2) and (5) are too vague to provide any clues.

(If we were listening to someone using the sentence in speech, we would probably know from their tone of voice whether the subject of the sentence is being positively or negatively evaluated, but it is often difficult to make this out from a written text.)

How common is the expression? Is it significant that all the examples are from the media? Do they represent a creative use of language by a handful of journalists or a typical/common phraseology?

If we think it is typical/common, further questions arise:

Is it always '**be** at the cutting edge', or are other verbs possible? Is it always '**at** the cutting edge' or are other prepositions possible? Is '**cutting** edge' a fixed collocation, or can we use other modifiers such as 'sharp' or 'leading'? Can 'cutting edge' be modified, for example by 'very' or 'extreme'? Can 'edge' be pluralized? Is the following preposition always '**of**', or are other prepositions (such as 'good/bad <u>for</u> something' or 'best/worst <u>in</u> something') also possible?

And finally, we may wonder, are there any other ways of expressing the same meaning, and if so, what are they?

Traditional solution 1: Consult a Native Speaker

One traditional method of trying to answer such questions is: ask a native speaker. The first problem with this method is that native speakers are not always available. And even if you happen to have access to native speakers, they are, of course, limited by their own experiences, interests, and idiolects. Sometimes, they may not be sure, but may still feel obliged to give an answer. Native speakers also tend to think of what is possible rather than what is common or usual, and are very adept at imagining possible contexts for almost any utterance. (I can adduce anecdotal evidence for this: in a Spanish-for-beginners lesson, we were taught the phrase "¿Quién eres?", 'Who are you?'. I objected that this sounded rather a rude question, and asked when a native Spanish speaker might ever use it. After a brief pause, 'At Halloween,' came the reply, 'when the person is wearing a mask and you want to know who it is.')

Traditional solution 2: Consult a Dictionary

The second traditional method also entails various problems. Which dictionary should we consult? How many are available? Or, more likely, which one is available? How big is it? Is it a recent edition? Can we trust it?

Pre-corpus dictionaries are necessarily not empirical in their inclusion and exclusion criteria, but it is surprising that even corpus-based dictionaries vary so much. While some variation can be justified by the vagaries of 'editorial judgment', the rest must rather depend on the size and composition of the corpora, the extent of the dictionary's fidelity to the data, and the skill of the lexicographers as data analysts.

All printed dictionaries are hampered, whether corpus-based or not, by the sheer necessity of having to condense their information onto a couple of thousand pages or so. As corpora are a fairly recent phenomenon, very large dictionaries are unlikely to be corpus-based, as adequately large corpora have not yet come into existence.

The different types of dictionary have different target audiences and different problems. As a lexicographer myself, it is not my aim to criticize any individual dictionaries or colleagues. Most of the problems are inherent and institutionalized, not idiosyncratic. I will therefore make my observations general, and quote dictionary entries without exact references (but there is a list of the dictionaries consulted at the end of the paper, if anyone wishes to check or follow up any of my points).

2.1 Bilingual Dictionaries

The main problem with bilingual dictionaries is that they have to try to deal with two languages. This means that they have to halve the space they afford to each. Also, most published dictionaries tend to be dominated by one language culture or the other, so there is often a disparity between the L1-L2 section and the L2-L1 section. Research into dictionary use suggests that very few users take the trouble to look in more than one place anyway, whether it is a cross-reference in a monolingual dictionary, or the other section of a bilingual dictionary.

Two of the dictionaries I consulted had entries for 'cutting edge', but only in the literal sense, omitting the figurative use altogether. Note also that they are embedded entries, not headword entries:

> **cutting** ... **~ edge** *s. (of a blade, tool)* filo *m*, taglio *m*.
>
> **cutting** ... **the ~ edge** le tranchant;

(In the second case, some figurative uses are in fact given for 'le tranchant' in the French-English section, but 'cutting edge' is still not mentioned.)

Another dictionary (which says on its back cover that it is 'The first ever bilingual dictionary to be compiled entirely from large electronic text collections of French and English') actually has a head word entry for 'cutting edge' in the English-French section, with the translational equivalent 'avant-garde'; but makes no mention of 'cutting edge' at the entry for 'avant-garde' in the French-English section:

> **cutting edge I** n **1** (blade) tranchant *m*; **2** fig avant-garde *f*; **to be at the ~ of** être à l'avant-garde de [*technology, fashion*] **II** *modif* [*film, industry, technology*] d'avant-garde
>
> **avant-garde** ... avant-garde ... to be avant-garde ... in the vanguard ... in the vanguard of research

The imbalance between L1 and L2 sections is particularly evident in another dictionary, which provides a great deal of information in one section, but very little in the other. Note that 'cutting edge' is once again relegated to the status of an embedded entry:

> **cutting** ... **~ edge** filo *m*, *(fig)* vanguardia *f*;

> **vanguardia** *nf* vanguard, van (*t fig*); **estar en la ~ del progreso** to be in the van of progress; **ir a la ~, ir en ~** to be in the vanguard; to be foremost, be ahead, be in front; **un pintor de ~** an ultramodern painter, a painter with a revolutionary style.

And again, 'cutting edge' is mentioned in the English-Spanish section, but is absent at the corresponding entry in the Spanish-English section.

2.2 Native-speaker dictionaries

The problem with native-speaker dictionaries, whether they are American or British in origin, is the assumptions that they have to make about the knowledge that their users bring to the consultation. Also, perhaps for fear of being over-restrictive or prescriptive in their treatment, they tend to opt for the safer stance of being over-general in their descriptions. Yet again, the problem is the resultant lack of detail in phraseology.

One dictionary has the following headword entry for 'cutting edge':

> **cutting edge** *n.* **1.** An effective quality or element. **2.** The position of greatest advancement or importance; the forefront: *"California is on the cutting edge of trends that spread nationwide"* (Carl Ingram)

Interestingly, only '**on** the cutting edge' is shown (in the example) for the prepositional phrase, with no mention of the possibility of '*at* the cutting edge'. The example is in fact rather vague: 'on the cutting edge **of trends**' (which rather belies the statement on its cover flap: "Thousands of quotations from the finest writers illuminate and add depth to the meanings") .

Another dictionary says:

> **cutting edge** *n.* the leading position in any field; forefront: *on the cutting edge of space technology*

This too only shows '**on** the cutting edge' and omits '*at* the cutting edge'.

Both native-speaker dictionaries have the item as a headword, and both mention the near-synonym 'forefront'.in their definitions, but both omit '*at* the cutting edge'.

2.3 Learner's dictionaries

Learner's dictionaries have gained a reputation for innovation and lexicographical skill in recent years. Last year, five new dictionaries (or new editions) were published in Britain, all claiming to have some basis in a corpus. I therefore feel that I can be a bit more critical about these.

One has no entry for 'cutting edge' at all, but it is considerably smaller than the others.

Another has the following headword entries:

> **cutting**[1] *n*

> **cutting**^{2} *adj*
> **cutting board**

and we find 'cutting edge' - not in its appropriate position in the alphabetical order - but within the second (adjective) homograph entry for 'cutting':

> **cutting**^{2} *adj* ... **3 be (at) the cutting edge of sth** to be the most advanced form of an activity, in which the newest methods, systems, equipment etc are developed and used: *The information highway is the cutting edge of the electronic revolution.*

The preposition '(at)' is given in brackets, but does not appear either in the definition or in the example, so we are left wondering exactly how or when to use it. In exact opposition to the native-speaker dictionaries, no mention is made of '**on** the cutting edge'.

The same dictionary has a very precise cross-reference at the end of the entry for **edge**^{1}:

> **edge**^{1} *n* **1** ... **2**... **3** ... **4** ... **5** ... **6** ... –see also **the cutting edge of** (CUTTING^2 (3))

This ironically makes 'cutting edge' a bit easier to find if you start your search at 'edge' rather than at 'cutting'. However, note that the precision in specifying the location of the entry is not matched by the precision of the wording: the difference in exact phraseology between the cross-reference ('the cutting edge of') and the entry itself ('be (at) the cutting edge of sth') might cause some learners a problem.

The third learner's dictionary also has a cross-reference at the entry for 'edge':

> **edge**^{1} *n* **1** ... **2** ... **3** ... the point or state immediately before something unpleasant, dangerous or exciting occurs: *species on the edge of extinction* ○ *The country was brought to the edge of war.* See also CUTTING EDGE.

However, the placement of the cross-reference in this meaning of 'edge', and the nature of the examples, suggests that 'cutting edge' might indeed also refer to something that is in an 'unpleasant or dangerous position' (as there is no example for 'something exciting'). Also, the cross-reference is given no homograph or meaning numbers, suggesting that we will find a headword entry for 'cutting edge', which is not in fact the case.

The headword entries are in fact as follows:

> **cutting**^{1} *n*
> **cutting**^{2} *adj*
> **cuttlefish**

Once again, 'cutting edge' is not in alphabetical order, but whereas the previous dictionary placed it in the adjective homograph, this dictionary embeds it in the entry **cutting**^{1}, the noun homograph:

> **cutting**^{1} *n* **1** ... **2** ... **3** ...
> ■ **cutting edge** *n* (usu *sing*) **1** ~ **(of sth)** the latest, most advanced stage in the development of sth: *working at the cutting edge of computer technology.* **2** an advantage over sb: *We're*

relying on him to give the team a cutting edge.

This dictionary does not mention 'at' in the definition, but introduces it in the example. As in the previous learner's dictionary (but unlike the native-speaker dictionaries), '**on** the cutting edge' is not mentioned at all. Instead of the slightly more helpful 'form of an activity' in the previous dictionary's definition, here we just have the vague 'sth'. However, we do see some information about grammar: (usu *sing*) at least suggests that we should be wary about using it in the plural.

The fourth learner's dictionary has numerous headword entries for 'cut':

cut (*obj*) [KNIFE]
cut *obj* [REDUCE]
cut *obj* [REMOVE]
cut *obj* [MISS]
cut *obj* [MAKE SAD]
cut (*obj*) [STOP SUDDENLY]
cut *obj* [IGNORE]
cut *obj* [GROW TEETH]
cut [CROSS]
cut (*obj*) [CARDS]
cut *obj* [RECORD]
cut (*obj*) [BEHAVE]
cut *obj* [DEAL WITH]

Then some phrasal verb headwords:

cut across
cut off
cut up

Then some compound headwords:

cutback
cute
...
cutlet
cutoff
cutout
cutthroat
cuttlefish
cutup

Altogether, these entries cover four columns of dictionary text, but there is no headword entry for 'cutting edge'. So where are we to start looking for it? If we begin at the first entry for 'cut', i.e. **cut** (*obj*) [KNIFE], and read through 42 lines of the entry, with no numbers to distinguish between one use and the next, we eventually come to it:

- If you are at the **cutting edge** of something, you are involved in its most recent stage of development: *The company is* **at the** *cutting edge of television miniaturization.*

(Note the full-sentence definition style, beginning 'If you ...'. This style was in fact introduced by Cobuild in its 1987 Dictionary, and it is interesting to see that most other learner's dictionaries now use it, either throughout, or at least for some types of entry, as here.)

The definition is again rather vague: 'the cutting edge of <u>something</u>', 'the development of <u>something</u>'. There is no grammar. Somewhat curiously, 'cutting edge' is in bold type in the definition, and 'at the' is in bold in the example. As in the previous learner's dictionaries, no mention is made of '**on** the cutting edge'.

The same dictionary has five headword entries for 'edge':

edge [OUTER POINT]
edge [BLADE]
edge [MOVE]
edge [ANGER/NERVOUSNESS]
edge [ADVANTAGE]

But none of them contain a cross-reference to 'cutting edge'. Of course, I should not have started my search for 'cutting edge' in the dictionary text at all, but in the Phrase Index in the backmatter. However, not only is the Phrase Index in extremely small print, but the ordering of items is rather complicated. Beginning at the first entry for 'cut':

a □**cut** a■bove 339L15
□classic ■**cut** 238L75
□clean ■**cut** 241L61/65/66
□clear-■**cut** 242L32
■crew □**cut** 323R10
■crinkle-□**cut** 324R78
□**cut** a ■corner 339R61
□**cut** a fine ■figure 339R77
□**cut** a ■swath through 338R75
□**cut** a■cross 340L3

... (53 more items omitted here)

□**cutoff** ■jeans 340R8
■cold □**cuts** 256R21
■glass-□cutter 598R11
□**cutthroat** ■razor 340R28
□**cutting** ■edge 339L5
■press □**cutting** 1118R75
■**cycle** □clips 340R66

By the way, 339L5 means 'see page 339, left column, 5th line', and that at least is correct.

The fifth learner's dictionary also has a cross-reference at the entry for 'edge':

edge
1 ... **2** ... **3** ... **4** ... **5** ... **6** ... **7** ...
8 See also **cutting edge, knife edge, leading edge.**
9 ... **10** ... **11** ... **12** ... **13** ...

However, it it is the only learner's dictionary to give 'cutting edge' as a headword, so it is easy to find.

It is also the only dictionary to tell you the frequency of 'cutting edge': ♦◊◊◊◊ (which is explained in the front matter: the phrase is in the fifth band of items, between 6600 and 14,600 in the list of the most common items in English). This makes us realize that 'cutting edge' indeed represents a frequent and important institutionalized phraseology, considering that the dictionary covers 75,000 references.

This dictionary alone supplies very specific grammar information about the phrase: 'N SING: usu at/on the N of n', and thus shows that both 'on' and 'at' are common, unlike all the other dictionaries consulted.

The full entry is as follows:

cutting edge	♦◊◊◊◊
1 If you are at the **cutting edge** of a particular field of activity, you are involved in its most important or most exciting developments. *This shipyard is at the cutting edge of world shipbuilding technology.*	N SING: usu at/on the N *of n* = forefront
2 If someone or something gives you a **cutting edge**, they give you motivation and energy, and an advantage over your competitors. *If Pearce had been fit, we would have won. We missed the cutting edge he would have given us.*	N SING

The definition refers to 'a particular field of activity', which is rather more specific than 'something' in two of the other dictionaries. In addition, this dictionary alone gives a synonym: 'forefront'.

My only criticism of this entry is that it lacks a reinforcing pragmatics note, to make sure that the positive evaluation intended by the phrase was explicitly stated, rather than implicitly by 'most important or most exciting'.

The real solution: Consult a Corpus

However, once we start looking at the corpus evidence, we will see how much more information we can obtain than even the best dictionary can supply. Even rigorously corpus-based (or we might call them ***corpus-driven***) dictionaries can only give an impressionistic account of the language, because of lack of space in a printed dictionary, and different lexicographers' judgments on which features are important and which are not.

However, it is important to use the data properly: not to just dip into the corpus to find examples to back up your intuitions. If you do that, you will only ever find what you wanted to find, and will never discover anything that you didn't know before. The point is not that it is wrong to use your intuition, but that it is counter-productive to use it too soon in the investigation process. Let the data guide you gradually towards the information you seek. This will enable you to put the information into a better perspective. At the very least, in most cases the data will refine and enhance your intuitions. At the most, in a few cases the data will surprise you or even astonish you, as it reveals aspects of the language that you had not even considered.

Cobuild has just increased the size of its corpus, called the Bank of English, to over 320 million words. Subscribers to the CobuildDirect service can access about 50 million words of this. Other English language corpora are available, such as the British National Corpus (for British English), and various smaller corpora, for example those supplied with the MicroConcord software. For the purposes of this paper, I will restrict myself to using the 50 million word CobuildDirect corpus, as it is publicly available. When you first access the corpus, various options are open to you: corpus access, collocations listings, etc. Then you are informed about the corpus contents, and you can select particular corpora or access them all. [see Appendix 1]

3.1 Frequencies

The first type of information that most corpus retrieval systems can supply is about the frequency of the items in the corpus. Following the advice I gave above, instead of limiting my search by looking immediately for information about 'cutting edge', I will start with 'cut*' and 'edg*' (i.e. all words beginning with the letters 'cut' and 'edg').

Frequency displays are available from CobuildDirect separately, showing the number of occurrences for each form and further subdivided according to its grammatical category or wordclass. [see Appendix 2 and 3]

I will merely note in passing that we can now see that in general, the forms of 'cut' are much more frequent than forms of 'edge'. These are displays obtainable from CobuildDirect without looking at the corpus itself. If we now proceed to use the corpus retrieval software, we are given frequency information of a slightly kind.

3.2 Subcorpus Distribution

As mentioned above, the CobuildDirect corpus is subdivided into smaller corpora according to the major varieties and modes. American, Australian, and British English are kept separate, as are newspapers, books, and spoken data. Further subdivisions would enable a finer-grained analysis, such as dividing books into fiction and non-fiction, or classifying newspaper articles into sports, politics, business, etc, however these are not yet available.

Now we are at 'Query' level, having selected all the corpora. Again we have a whole array of options for continuing our search, such as wildcards, wordclass, alternative lexical items, and positional relationship of items. [see Appendix 4]

If we ask for all the forms of the lemma 'cut' (i.e. cut, cuts, cutting), we are shown another potentially useful display: subcorpus distribution. We see, for example, that the forms of 'cut' occur most frequently in the 'npr' corpus (National Public Radio, Washington, USA), and least frequently in the 'ukspok' corpus (General UK Spoken data). [see Appendix 5] A similar search on the lemma 'edge' shows that it occurs most frequently in 'ukmags' (British magazines) and least in 'bbc' (BBC World Service). [see Appendix 6]

3.3 Collocation

If we ask for all the lines for 'cut', then type 'c', we are shown a list of the top collocates of the lemma 'cut'. We notice, after 'off', 'down', and 'out' (all of which are probably, from my intuition, particles associated with 'cut' in the phrasal verb combinations 'cut off', 'cut down', and 'cut out'), and after 'tax', 'interest', 'spending' and other financial/political items, that 'edge' also collocates with all forms of 'cut'. [see Appendix 5] When we do the same for 'edge', the collocate list is particularly surprising: the top collocates of 'edge' are precisely the elements in our phrase: 'the', 'of'; 'on', 'at', and 'cutting'! Who could have predicted that from intuition? [see Appendix 6]

Now we can focus on 'cutting' and 'edge|edges' in more detail. [see Appendix 7 and 8]

Subtle shifts in subcorpus distribution can be seen, but none seem particularly significant. In contrast, major changes in collocation are evident: 'edge' is a much stronger collocate of 'cutting' than of the lemma 'cut', as is 'cost'; 'deficit' and 'room' did not show up as collocates of the lemma 'cut' at all, but are significant for 'cutting'. [compare Appendix 7 and 5]

By eliminating the forms 'edging' and 'edged' from our consideration, we have lost the collocation with 'gilt', but gained the collocates 'competitive', 'box', and 'cliff'. [compare Appendix 8 and 6]

Finally, we can look at the co-occurrences of 'cutting+edge|edges'. [see Appendix 9] There are 145 of these in the 50 million word CobuildDirect corpus. They occur most frequently in 'usephem' (American ephemera) and 'ukmags' (British magazines), suggesting a prominent use of 'cutting edge' in advertising copy, and least frequently in the spoken data ('ukspok', 'npr', and 'bbc') and books ('usbooks' and 'ukbooks', and less in British than American books). Interesting new collocates appear: 'modern' and 'contemporary', emphasizing innovation, and 'fashion' and 'design', representing typical fields.

3.4 Concordances

The next level of detail provided by the software is the facility to look at the concordance lines themselves. [see Apendix 10] Even in their raw form, we can see several occurrences of 'cutting edge of' and 'on the cutting edge' and 'at the cutting edge'. Sorting the concordances to right or left immediately groups these together. [see Appendix 11]

3.5 'Picture': Positional Collocation

One last display is all I have space for in this paper. It is called 'picture' in the Cobuild software, and shows which collocates occur in which positions in relation to the 'nodeword' (in this case,'cutting' has been selected by the computer, because its frequency is lower than 'edge|edges'). The 'picture' display is available for up to 6 words before and after the NODE. It can be ordered by raw frequency [see Appendix 12] , by T-score [see Appendix 13], or by MI-score. Whichever is used (compare Appendix 12 and 13), the significance of 'at' and 'on' before and 'of' after 'cutting edge' is indisputable. Summarizing the figures, the facts are as follows: of the 145 lines,

20 are for 'at the cutting edge', of which 12 are for 'at the cutting edge of' (i.e. 8 are for '... at the cutting edge.' not followed by 'of', which was not specifically mentioned in any of the dictionaries). 14 are for 'on the cutting edge' (of which 2 are not followed by 'of'). [see Appendix 14] There is one line for 'cutting edge in' (but no lines for 'cutting edge for'):

Queensland and Australia were close to the cutting edge in this smart pacemaker technology.

There are 5 lines for 'cutting edges' [see Appendix 14] but all of them are for the literal meaning, and none are for 'at the cutting edges of'.

Most of the lines are for '**be** at/on the cutting edge (of)', but there are also few for '**be** the cutting edge'.

There are no examples of 'at the sharp edge of', but there are 9 examples of 'at the leading edge of' [see Appendix 15].

If I were to point out one pattern that has not been mentioned before, but is very evident in the data, it would be for 'cutting-edge' as an adjective. [see Appendix 15]

Finally, the questions about semantics and pragmatics (what does 'be at the cutting edge' mean, and is it good or bad), are answered in the concordance lines: example 25 of Appendix 15 describes a school as '*a very progressive school, very cutting-edge secondary school*', and other lines include words and phrases like 'excitement', 'original', 'modern', 'thoroughly contemporary', 'exciting', 'new', 'the shape of things to come', and 'prided themselves on being at the cutting edge'.

The current implementation of the Cobuild software does not enable us to easily answer the question about alternatives: 'are there other ways of expressing the same meaning', but synonymic and thesaural investigations will no doubt be available in the future.

Conclusion

I think it is obvious now, after seeing the data first-hand, how much richer and deeper our linguistic knowledge is after this exposure to the corpus data, both in terms of our knowledge about the phrase 'be at/on the cutting edge (of)', but also in terms of how many aspects there can be to investigate when researching such a linguistic pattern. As more people gain access to corpora, I am sure we will see more discussion of the composition and arrangement of corpora, the software tools available, the techniques and practices that are most effective, and the range of information (linguistic and non-linguistic) that can be obtained from them.

Cobuild has long been promoting the wider use of corpus data by teachers, students, and researchers. It has published concordance lines in its Concordance Sampler series, providing suggestions about how to use them in the classroom. It has recently published two CD-ROMs with corpus data on them, and initiated an Email service and the online access CobuildDirect service. I am sure that electronic dictionaries and other new products and facilities will be developed over the next few years. Specialist corpora (such as Cobuild's Business Corpus, Corpus of Academic Writing, and Children's Corpus) will also become available in the future. But in order to benefit from these developments in the future, it is vitally important that people start accessing corpora now, and build up the new skills and methodologies they will need.

<u>Dictionaries consulted</u>:

Collins Spanish (1992)

Collins Sansoni (1988)
Collins Robert (1987)
Oxford Hachette (1994)
American Heritage (1992)
Collins English Dictionary (1992)
Harrap's Essential English Dictionary (1995)
Longman Dictionary of Contemporary English (1995)
Oxford Advanced Learner's Dictionary (1995)
Cambridge International Dictionary of English (1995)
Collins Cobuild English Dictionary (1995)

APPENDIX 1

```
IBM AIX Version 3 for RISC System/6000
(C) Copyrights by IBM and by others 1982, 1990.
login:

*                       Welcome to COBUILD
******************************************************************************

ATTENTION: The Cobuild Direct corpus is now 50 MILLION words, made up
of 11 sub-components: 9 of these are new updated data.
                               -------**O**-------
Please mail your queries, comments, complaints, etc. to:
      direct@cobuild.collins.co.uk
------------------------------------------------------------------------------
Cobuild Direct subscribers:

  Remember to delete your files after you have ftp'd them!  (You can
  usually do this with the command "del" in ftp).  List your files every
  so often, just in case there are some old ones you no longer need.
  (This is usually the "ls" command in ftp).


1. Interactive corpus access tool

2. Collocations listings

3. This week's Wordwatch page

6. If the information on your screen seems to be garbled...

7. Register for WorldWide Web restricted services [***NEW!***]

8. Change password

9. Quit and logout

(Enter a number for the required option): 1


Enter the names of corpora you would like to use or type 'd' or RETURN
to use all corpora.  To quit from lookup, type 'q'. To get a usage message
type anything else.

Type the names of any number of corpora, taken
from the first column of the following list:

oznews  5m      01      Australian newspapers
ukephem 3m      02      UK ephemera
ukmags  5m      03      Popular magazines (UK)
ukspok  10m     04      Informal speech (UK)
usephem 1m      05      US ephemera
bbc     3m      06      BBC World Service radio
npr     3m      07      National Public Radio (US)
ukbooks 5m      08      Books: miscellaneous (UK)
usbooks 5m      09      Books: miscellaneous (US)
times   5m      10      Times newspaper (UK)
today   5m      11      Today newspaper (UK)
If you just hit RETURN, all corpora are used.

Corpora ('q' to quit):
```

Matched Wordlist Items

```
cut V 10347     VB cut 3184     VBD cut 964     VBG cutting 1924        VBN cut 3808
cut N 4397      NN cut 2252     NNS cuts 2145
cute JJ 344
cutting N 272   NN cutting 65   NNS cuttings 207
cutter N 218    NN cutter 143   NNS cutters 75
cutback N 152   NN cutback 34   NNS cutbacks 118
cutlery N 115   NN cutlery 115
cutler N 54     NNS cutlers 5   NP cutler 49
cutlet N 40     NN cutlet 15    NNS cutlets 25
cuticle N 37    NN cuticle 27   NNS cuticles 10
cuthbert N 27   NP cuthbert 27
cutaway N 26    NN cutaway 25   NNS cutaways 1
cutout N 23     NN cutout 9     NNS cutouts 14
cutt N 21       NP cutts 21
cuttlefish JJ 19
cutoff N 19     NN cutoff 14    NNS cutoffs 5
cutlass N 17    NNS cutlasses 3 NP cutlass 14
cutilla N 16    NP cutillas 16
cutmore N 16    NP cutmore 16
cutthroat N 16  NN cutthroat 9  NNS cutthroats 3        NP cutthroat 4
cutesy N 15     NN cutesy 15
cutoff JJ 13
cutie N 12      NN cutie 12
cuttle N 12     NN cuttle 2     NP cuttle 9     NP cuttles 1
cutest JJ 10
cutwork N 9     NN cutwork 9
cuteness N 8    NN cuteness 8
cuthbertson N 7 NP cuthbertson 7
cuttack N 7     NN cuttack 1    NP cuttack 6
cutcliffe N 6   NP cutcliffe 6
cuttitta N 6    NP cuttitta 6
cutely RB 6
cutaneous JJ 5
cutout JJ 5
cuticura N 4    NP cuticura 4
cutivate N 4    NN cutivate 1   NP cutivate 3
cuttin N 4      NN cuttin 4
cutty N 4       NP cutty 4
cuty N 4        NNS cuties 4
cuter N 3       NN cuter 3
cutex N 3       NN cutex 3
cutner N 3      NP cutner 3
cutpurse N 3    NP cutpurse 3
cutrona N 3     NP cutrona 3
cuttable N 3    NN cuttable 2   NP cuttable 1
cutten N 3      NN cutten 1     NP cutten 2
cutworm N 3     NN cutworm 1    NNS cutworms 2
cuter JJ 2
cutbush N 2     NP cutbush 2
```

Return to Wordlist Search page

Matched Wordlist Items

```
edge N 3709       NN edge 3091      NNS edges 596     NP edge 22
edge V 684        VB edge 59        VBD edged 221     VBG edging 172    VBN edged 232
edgbaston N 278 NN edgbaston 10 NP edgbaston 268
edgar N 209       NP edgar 209
edged JJ 150
edgy JJ 74
edgware N 30      NP edgware 30
edger N 16        NN edger 5        NP edger 4        NP edgers 7
edgley N 9        NP edgley 9
edgeways RB 9
edginess N 8      NN edginess 8
edgington N 8     NP edgington 8
edgerton N 7      NP edgerton 7
edgell N 6        NP edgell 6
edgeware N 6      NP edgeware 6
edging N 6        NNS edgings 6
edg N 5 NN edg 2          NP edg 3
edgers N 5        NN edgers 5
edgard N 4        NP edgard 4
edgardo N 4       NP edgardo 4
edgehill N 4      NP edgehill 4
edgewood N 4      NP edgewood 4
edghill N 4       NP edghill 4
edgebrook N 3     NP edgebrook 3
edgecombe N 3     NP edgecombe 3
edgeley N 3       NP edgeley 3
edgcumbe N 2      NP edgcumbe 2
edgier N 2        NN edgier 2
edginton N 2      NP edginton 2
edgily RB 2
```

Return to Wordlist Search page

APPENDIX 4

```
Query (or RETURN to exit):

     Examples of responses to 'Query:' prompt -
          juncture    (matches the word 'juncture')
          brief*      (matches 'brief', 'briefly', 'briefcase' etc)
          jam/VB      (matches the word 'jam' used as a verb)
          free+hand   (matches the string 'free hand')
          bring@      (matches any form of the verb 'bring')
          orient|orientate
                      (matches 'orient' or 'orientate')
          missle|mistle+thrush
                      (matches 'missle thrush'or 'mistle thrush')
          left+1,5behind
                      ('left' followed by 'behind', with between one
                      and five words in between)
          foul+0,1up ('foul' followed by 'up', with either no words
                      or one word in between)
          pay+1,1respects
                      ('1,1' specifies exactly one intervening word;
                      'pay his respects', 'pay their respects' etc)
          catch+\22   (matches 'catch 22')

     For more information, see the Reference document.
```

APPENDIX 5

```
Corpus            Total Number of          Average Number per
                  Occurrences              Million Words

npr                     1461                 466.9/million
today                   2192                 417.7/million
times                   2398                 416.0/million
bbc                      867                 332.2/million
ukmags                  1579                 322.1/million
usephem                  378                 308.6/million
oznews                  1607                 301.1/million
ukephem                  915                 292.9/million
ukbooks                 1098                 205.1/million
usbooks                 1064                 189.1/million
ukspok                  1250                 134.8/million

Hit any key to continue ...
Query is "cut@"
Term 1 in your query has been selected as the node

14809 matching lines
How many required (hit RETURN to get them all):
```

```
[Type 'c' to get collocates in T-score order]

  off                       1166    31.302580
  down                       698    22.424359
  tax                        520    21.947974
  out                        803    19.732259
  rates                      399    19.352143
  spending                   356    18.450522
  short                      369    17.990428
  to                        3956    17.869214
  rate                       342    17.338102
  back                       483    16.499042
  by                         999    16.434364
  interest                   316    16.233900
  into                       506    15.080528
  budget                     245    14.999790
  through                    341    13.696596
  price                      245    13.281007
  costs                      202    13.268319
  hair                       179    12.369377
  half                       230    12.303152
  cost                       184    11.948844
  edge                       157    11.948654
  and                       3561    11.665972
  clear                      172    11.292719
  from                       765    10.500855
```

APPENDIX 6

```
Corpus          Total Number of       Average Number per
                Occurrences           Million Words

ukmags                837             170.7/million
ukbooks               758             141.6/million
usephem               154             125.7/million
usbooks               623             110.7/million
ukephem               291              93.1/million
times                 516              89.5/million
oznews                443              83.0/million
today                 407              77.5/million
npr                   158              50.5/million
ukspok                284              30.6/million
bbc                    72              27.6/million

Hit any key to continue ...
Query is "edge@"
Term 1 in your query has been selected as the node

4543 matching lines
How many required (hit RETURN to get them all):
```

```
[Type 'c' to get collocates in T-score order]

  the                       4488   39.545381
  of                        2014   25.638424
  on                         946   22.802784
  at                         462   13.088476
  cutting                    148   12.050353
  over                       183   10.153130
  around                     131    9.851335
  gilt                        86    9.252197
  along                       97    9.065162
  leading                     86    8.815040
  knife                       77    8.667874
  area                        76    7.552518
  off                        108    7.525430
  town                        68    7.457136
  sharp                       54    7.141408
  sat                         56    7.009066
  an                         209    6.953866
  bed                         56    6.916344
  outer                       48    6.855606
  water                       65    6.754872
  rough                       46    6.613967
  double                      51    6.558856
  hard                        60    6.487688
  with                       342    6.087163
```

APPENDIX 7

```
Corpus          Total Number of        Average Number per
                Occurrences            Million Words

npr                  183                   58.5/million
times                318                   55.2/million
ukephem              172                   55.1/million
today                256                   48.8/million
ukmags               225                   45.9/million
usephem               53                   43.3/million
bbc                  101                   38.7/million
oznews               190                   35.6/million
usbooks              164                   29.1/million
ukbooks              131                   24.5/million
ukspok               196                   21.1/million

Hit any key to continue ...

Query is "cutting"
Term 1 in your query has been selected as the node

1989 matching lines
How many required (hit RETURN to get them all):
```

```
[Type 'c' to get collocates in T-score order]

  edge                          142   11.834279
  cost                          122   10.778841
  by                            208   9.956191
  off                           115   9.507441
  down                          111   9.190017
  back                          109   8.891459
  out                           104   6.986592
  the                          1022   6.786726
  rates                          42   6.222902
  budget                         41   6.188833
  on                            193   6.182300
  through                        56   5.902528
  costs                          38   5.871981
  cutting                        34   5.725749
  deficit                        30   5.394340
  and                           504   5.310581
  tax                            33   5.288430
  room                           33   4.947296
  spending                       24   4.683045
  up                             76   4.642346
  taxes                          21   4.475230
  its                            50   4.352158
  price                          27   4.236691
  of                            469   4.201318
```

APPENDIX 8

```
Corpus          Total Number of       Average Number per
                Occurrences           Million Words

ukmags                722               147.3/million
ukbooks               649               121.2/million
usephem               132               107.8/million
usbooks               525                93.3/million
ukephem               231                73.9/million
times                 390                67.7/million
oznews                350                65.6/million
today                 318                60.6/million
npr                   115                36.8/million
ukspok                273                29.4/million
bbc                    63                24.1/million

Hit any key to continue ...

Query is "edge|edges"
Term 1 in your query has been selected as the node

3768 matching lines
How many required (hit RETURN to get them all):
```

```
[Type 'c' to get collocates in T-score order]

  the                         4194    41.211815
  of                          1901    27.175906
  on                           912    23.480102
  at                           415    13.015587
  cutting                      148    12.070001
  over                         175    10.366560
  around                       127    9.926535
  along                         92    8.924230
  leading                       84    8.780301
  knife                         66    8.028100
  area                          75    7.687339
  off                          101    7.591048
  town                          67    7.526022
  an                           187    7.095891
  sat                           56    7.089969
  bed                           56    7.013065
  water                         64    6.907205
  outer                         48    6.867991
  sharp                         44    6.442994
  competitive                   38    6.010206
  rough                         36    5.842151
  box                           38    5.571753
  cliff                         30    5.391248
  edge                          32    5.329236
```

APPENDIX 9

```
Corpus          Total Number of        Average Number per
                Occurrences            Million Words

usephem               13                  10.6/million
ukmags                35                   7.1/million
times                 27                   4.7/million
oznews                20                   3.7/million
ukephem               11                   3.5/million
today                 12                   2.3/million
usbooks                9                   1.6/million
ukspok                13                   1.4/million
npr                    2                   0.6/million
bbc                    1                   0.4/million
ukbooks                2                   0.4/million

Hit any key to continue ...

Query is "cutting+edge|edges"
Term 1 in your query has been selected as the node

145 matching lines
How many required (hit RETURN to get them all):
```

```
[Type 'c' to get collocates in T-score order]

  edge                         140    11.826132
  the                          124    5.865194
  of                            61    4.281858
  at                            25    3.846678
  <h>                           12    2.744983
  </h>                          11    2.568377
  on                            19    2.567478
  modern                         6    2.396316
  lacked                         5    2.231282
  edges                          5    2.230075
  fashion                        4    1.963858
  international                  4    1.854969
  always                         4    1.736222
  comedy                         3    1.709269
  contemporary                   3    1.707205
  design                         3    1.673065
  new                            5    1.518342
  </c>                           4    1.464158
  decorex                        2    1.413943
  serrated                       2    1.413435
  documentary                    2    1.405152
  92                             2    1.400350
  collapsed                      2    1.397330
  excitement                     2    1.395915
```

APPENDIX 10

```
Query is "cutting+edge|edges"
Term 1 in your query has been selected as the node

145 matching lines
How many required (hit RETURN to get them all):

Television programmers were # on the cutting edge of history last night # the
<    programmed a documentary called Cutting Edge, about the dangers # of an
<<h> EDITORIAL </h> February 5, 1995 Cutting edge MOST people # are amazed at
<     the world. <p> The boat on the cutting edge of technological #
released by Interscope, label such # cutting-edge acts as Nine Inch Nails and
will argue that 'Australia is on the cutting edge of the international food
<  argued that Australia # is on the cutting edge of the international food
<     who really wanted to be on the cutting edge # of the international food
Idol and The Who's The Seeker plus # cutting edge originals like Ill Wind,
Singer, in # an argument grounded in cutting-edge human experiences # rather
120. <p> <b> BRETT B </b> Loggers at cutting edge Bill Brett is chairman # of
<       police who were to be at the cutting edge of Fitzgerald # Inquiry, the
<   <p> <h> FOOD </h> WEEKEND Page 6 Cutting edge LESS is # more. <p> I read
<         said FIME had been at the 'cutting edge" enterprise # bargaining in
<   <p> The # move was part of a new cutting-edge arts project called 'NObody'
and women's issues, Biloela. <p> <h> CUTTING EDGE </h> COFFEE lovers can hark
withdrawn from the account." <p> <h> CUTTING EDGE </h> 2) Gas Lighter AVOID
<     society to appreciate both the cutting edge of modern design # as well a
<    and Australia were close to the cutting edge # in this smart pacemaker
<      pace and Queensland is at the cutting edge. <p> The Keating Government,
< 5.00pm <p> <c> PHOTOS </c> <h> THE CUTTING EDGE OF HISTORY </h> Jacobean Fai
<  molecular bonding to give tougher cutting edges marks the introduction of a
<the London Comedy Store innovative 'Cutting Edge" team. <p> A refreshing
Line 1 of 145.  Corpus oznews.  Text <tref id=N5000950118>.  '?' for help.
```

APPENDIX 11

```
<  over a decade. <p> Closest to the cutting edge of comedy" Melbourne Age <p>
<classic humour of yesteryear to the cutting edge of alternative comedy, there
<       for those interested in the cutting edge of international theatre"
<  and full-bodied, often with a dry cutting edge of ripe tannins, which
<    that finds her, as ever, at the cutting edge of contemporary music. Secre
<      is however concerned with the cutting edge of club culture. It is
< who've felt annoyed at the way the cutting edge of British dance music has
day. <p> So, naturally, staff at the cutting edge of the retail trade are
<   leaders, these shops display the cutting edge of New York fashion,
<  The Dragonaires were never at the cutting edge of ska, always lacking the
only to lose their lives at the very cutting edge of aeronautical research and
<  the South African struggle is the cutting edge of the global struggle for
those who think Don Henley is at the cutting edge of the American music scene.
<   puts it, the disease is # at the cutting edge of evolution # There are fou
<  prided themselves on being at the cutting edge of Cool. Their black and
<  believe that this band are on the cutting edge of rock, when in fact
Nothing to excite you in the modern, cutting-edge of music? Then herald the
< alacrity by the Americans. But the cutting edges of Matisse's collaged blue
<   Costeau. She's never been at the cutting edge of what's going on: just now
Line 67 of 145.  Corpus oznews.  Text <tref id=N5000950531>.  '?' for help.
```

```
<  I do try to remain at astronomy's cutting edge. Recently, for example, I
<       police who were to be at the cutting edge of Fitzgerald # Inquiry, the
<         said FIME had been at the 'cutting edge" enterprise # bargaining in
<      pace and Queensland is at the cutting edge. <p> The Keating Government,
<in 1988, Richard Feynman was at the cutting edge of modern science. Part
<    that finds her, as ever, at the cutting edge of contemporary music. Secre
day. <p> So, naturally, staff at the cutting edge of the retail trade are
<   puts it, the disease is # at the cutting edge of evolution # There are fou
<  The Dragonaires were never at the cutting edge of ska, always lacking the
<  prided themselves on being at the cutting edge of Cool. Their black and
<   Costeau. She's never been at the cutting edge of what's going on: just now
those who think Don Henley is at the cutting edge of the American music scene.
shoe shops, their designs are at the cutting-edge and stylish, too. <p> Le
<     name <ZZ0> which erm is at the cutting edge. You know # it's tiny it's
<   of painful contradictions at the cutting edge <ZF1> the <ZF0> the barriste
<year-old Talk is meant to be at the cutting # edge the shape of things to
<This, remember, is a company at the cutting edge of # modern retailing.  <p>
<  Brentford have always been at the cutting edge.  <p> Cadbury's Chomp bars
<  came # along.  <p> Located at the cutting edge of cheesy, so to speak, the
for-Spain movements # <p> <h> At the cutting edge of the beauty business;Body
<     safe.  <p> We had items at the cutting edge," he said. 'Like when we
Line 23 of 145.  Corpus times.  Text <tref id=N2000960203>.  '?' for help.
```

```
<  argued that Australia # is on the cutting edge of the international food
<     who really wanted to be on the cutting edge # of the international food
<     the world. <p> The boat on the cutting edge of technological #
will argue that 'Australia is on the cutting edge of the international food
Television programmers were # on the cutting edge of history last night # the
<  believe that this band are on the cutting edge of rock, when in fact
<       club remix and always on the cutting edge of the New York dance scene.
<Alzheimer's Association stay on the cutting edge of research to help find new
<     pasta for chefs who are on the cutting edge and for home cooks who are
Scissors Fun Trio </h> <p> Be on the cutting edge! Set of colorful scissors
<    garde that was no longer on the cutting edge of aesthetic and artistic
<   young blacks wanted to be on the cutting edge of the resistance. Out of
<        will # absolutely be on the cutting edge of contemporary musicals."
< The doctors' association is on the cutting edge of the reforms because # it
Line 91 of 145.  Corpus usephem.  Text <tref id=E9000000708>.  '?' for help.
```

APPENDIX 12

```
the        the        to         is         at         the        NODE
of         that       the        be         on         a          NODE
<p>        <p>        <p>        the        the        <h>        NODE
to         a          of         are        to         and        NODE
<h>        and        who        </c>       with       modern     NODE
</h>       is         </h>       a          <p>        of         NODE
a          of         and        to         a          new        NODE
for        have       for        and        lacked     very       NODE
all        s          right      of         is         life       NODE
that       <c>        australia  been       s          is         NODE
they       <h>        were       s          as         for        NODE
each       for        had        <p>        <h>        it         NODE
exciting   was        you        in         </h>       in         NODE
and        but        associatio at         and        at         NODE
is         their      picture    but        for        with       NODE
it         along      decorex    that       was        s          NODE
lot        did        life       they       from       are        NODE
an         wanted     this       so         t          its        NODE
two        food       a          92         but        1995       NODE
in         </h>       way        <h>        lack       plus       NODE
have       this       it         this       way        more       NODE
full       to         by         it         of         four       NODE
with       by         queensland longer     in         some       NODE
"the". Tot freq:2610249. Freq as coll:8. t-sc:0.2348. MI:0.1250. '?' for help
-------------------------------------------------------------------------
NODE       edge       of         the        the        s          of
NODE       edges      </h>       <p>        s          the        the
NODE                  and        <h>        what       and        a
NODE                  <p>        for        internatio in         scene
NODE                  the        modern     a          food       in
NODE                  to         a          and        <p>        on
NODE                  he         and        it         it         new
NODE                  it         is         in         of         <p>
NODE                  in         in         are        as         have
NODE                  was        on         when       to         s
NODE                  you        but        new        black      are
NODE                  excitement people     but        they       that
NODE                  a          said       then       who        only
NODE                  is         history    design     i          well
NODE                  less       contempora </h>       music      we
NODE                  at         this       to         out        way
NODE                  with       to         is         is         to
NODE                  about      it         of         than       and
NODE                  up         by         an         an         is
NODE                  again      of         right      right      for
NODE                  fashion    have       two        under      it
NODE                  set        has        on         on         an
NODE                  that       s          with       level      at
"of". Tot freq:1225673. Freq as coll:8. t-sc:1.6105. MI:1.2157. '?' for help
```

APPENDIX 13

```
of         <c>        who        is         at         the        NODE
<p>        that       decorex    be         on         a          NODE
</h>       along      associatio </c>       lacked     <h>        NODE
<h>        wanted     picture    are        with       modern     NODE
all        is         australia  been       lack       very       NODE
renaults   have       to         92         as         new        NODE
flanger    did        right      so         </h>       serrated   NODE
rah        <h>        </h>       interscope <h>        gloriously NODE
ornitholog their      were       biloela    t          curved     NODE
cosm       scabrous   had        npg        from       combines   NODE
fluted     fime       feynman    yesteryear loggers    militant   NODE
prided     chrysalis  slickers   collectibl serrated   tougher    NODE
alacrity   remix      prongs     wah        seeker     innovative NODE
choruses   excite     bayonets   flanks     grounded   heavyweigh NODE
breakout   empress    sturridge  synonymous astronomy  harsh      NODE
sbs        molecular  programmer contradict pedal      wire       NODE
brentford  bursts     josephine  stainless  retains    technical  NODE
scathing   percussion bonding    lacked     is         contempora NODE
hub        chefs      programmed blade      boasts     musical    NODE
infinitely bodied     monstrous  manhattan  virtual    sharp      NODE
alzheimer  renault    henley     confined   diagram    reality    NODE
knives     calf       enthusiast frightenin documentar dry        NODE
sandwiches blur       unlimited  classics   thoroughly tonight    NODE
"of". Tot freq:1225673. Freq as coll:7. t-sc:1.3438. MI:1.0231. ‘?’ for help
```

```
NODE       edge       of         the        internatio s          scene
NODE       edges      </h>       <h>        what       food       new
NODE                  excitement modern     design     music      of
NODE                  he         contempora new        black      only
NODE                  and        for        then       as         well
NODE                  chanteuses history    s          who        on
NODE                  yohji      people     when       in         we
NODE                  originals  said       lippman    collaged   saskatchew
NODE                  barbed     yamamoto   jeroen     chomp      hark
NODE                  enthusiast aeronautic roguish    jacobean   diminutive
NODE                  compilatio skater     soldiered  pacemaker  erase
NODE                  sampling   ska        tannins    teetering  experiment
NODE                  techno     cheesy     frightenin modernism  mellow
NODE                  experiment rappers    musicals   separates  symptom
NODE                  cuisine    exemplifie retailing  beardsley  barton
NODE                  documentar matisse    liners     culinary   barrister
NODE                  <p>        mckenna    colorful   stab       astonished
NODE                  secondary  cadbury    readings   cooks      explosions
NODE                  shaped     depicted   9pm        examines   titled
NODE                  developmen guru       refuge     scissors   nails
NODE                  enterprise aesthetic  lung       helmut     nineties
NODE                  guitar     bargaining imperialis appropriat abstract
NODE                  acts       ripe       stripped   refreshing lang
"scene". Tot freq:4129. Freq as coll:4. t-sc:1.9942. MI:8.4300. ‘?’ for help
```

APPENDIX 14

```
<     safe.  <p> We had items at the cutting edge," he said. ‘Like when we
for-Spain movements # <p> <h> At the cutting edge of the beauty business;Body
<  came # along.  <p> Located at the cutting edge of cheesy, so to speak, the
<  Brentford have always been at the cutting edge.  <p> Cadbury's Chomp bars
<This, remember, is a company at the cutting edge of # modern retailing.  <p>
<year-old Talk is meant to be at the cutting # edge the shape of things to
<   of painful contradictions at the cutting edge <ZF1> the <ZF0> the barriste
<     name <ZZ0> which erm is at the cutting edge. You know # it's tiny it's
shoe shops, their designs are at the cutting-edge and stylish, too. <p> Le
those who think Don Henley is at the cutting edge of the American music scene.
<   Costeau. She's never been at the cutting edge of what's going on: just now
<  prided themselves on being at the cutting edge of Cool. Their black and
<  The Dragonaires were never at the cutting edge of ska, always lacking the
<   puts it, the disease is # at the cutting edge of evolution # There are fou
day. <p> So, naturally, staff at the cutting edge of the retail trade are
<    that finds her, as ever, at the cutting edge of contemporary music. Secre
<in 1988, Richard Feynman was at the cutting edge of modern science. Part
<      pace and Queensland is at the cutting edge. <p> The Keating Government,
<          said FIME had been at the ‘cutting edge" enterprise # bargaining in
<       police who were to be at the cutting edge of Fitzgerald # Inquiry, the
```

```
< The doctors' association is on the cutting edge of the reforms because # it
<        will # absolutely be on the cutting edge of contemporary musicals."
<   young blacks wanted to be on the cutting edge of the resistance. Out of
<    garde that was no longer on the cutting edge of aesthetic and artistic
Scissors Fun Trio </h> <p> Be on the cutting edge! Set of colorful scissors
<     pasta for chefs who are on the cutting edge and for home cooks who are
<Alzheimer's Association stay on the cutting edge of research to help find new
<       club remix and always on the cutting edge of the New York dance scene.
<  believe that this band are on the cutting edge of rock, when in fact
Television programmers were # on the cutting edge of history last night # the
will argue that ‘Australia is on the cutting edge of the international food
<     the world. <p> The boat on the cutting edge of technological #
<     who really wanted to be on the cutting edge # of the international food
<  argued that Australia # is on the cutting edge of the international food
```

```
<have sharp stainless steel serrated cutting edges, perfect for cutting cakes,
< a monstrous frightening shape with cutting edges; it bored up into her head,
<    like bayonets, their points and cutting edges shaped by an unusual
<  molecular bonding to give tougher cutting edges marks the introduction of a
< alacrity by the Americans. But the cutting edges of Matisse's collaged blue
```

APPENDIX 15

```
ozn N5000950730    them at the leading edge of Nineties technology. <p> For a
ozn N5000951127   right at the leading edge of # meeting the challenges imposed
uke E0000001508Trading at the leading edge of natural floorcovering
uke E0000002356      is at the leading edge of facecare. Launched in Europe in
use E9000000318   to be at the leading edge of an unproven, controversial and
use E9000000396         at the leading edge of their specialities and fulfill
npr S2000921109ve been at the leading edge of a one and a half billion dollar
tim N2000960120 simply at the leading edge of technology, setting trends that
tod N6000951101    were at the leading edge of # the market # <p> Ex-singer
```

```
released by Interscope, label such # cutting-edge acts as Nine Inch Nails and
<   <p> The # move was part of a new cutting-edge arts project called 'NObody'
<    exciting and that NPG are # the cutting edge # band. <p> then they have t
< the hub of Edinburgh's night life. Cutting edge chanteuses, musicians,
<make for a thoroughly contemporary, cutting-edge compilation; and 2) the whol
<  civil servant <p> Virtual reality Cutting-edge computer science project in
<  fashion business, so 'they expect cutting edge cuisine," says Lippman. 'And
<        monitor over 200 sources for cutting-edge developments in the field.
<    flanger and even wah pedal. New cutting-edge effects have also been
<          said FIME had been at the 'cutting edge" enterprise # bargaining in
<  ever live the life of the modern, cutting-edge enthusiast? <p> I did, yes #
<       against In The City's lack of cutting-edge excitement. In effect, it
<  to ensure the events retains some cutting edge excitement. <p> One figure
<     English pop values and modern, cutting-edge, experimental electronics.
< gowns indicate that you won't find cutting-edge fashion, but elegant
new COSM technology with 20 years of cutting-edge guitar effects processing
Singer, in # an argument grounded in cutting-edge human experiences # rather
<         capital which would allow a cutting edge industry to develop in
<  have something collectible.  As a cutting-edge media guru on constant alert
Idol and The Who's The Seeker plus # cutting edge originals like Ill Wind,
<    are perfectly all right. It's a cutting edge product and it's only for a
<         <F01> Right. Yeah. <M01> and cutting edge production. Er but
which features CREDIT TO THE NATION; cutting-edge rock-rappers COLLAPSED LUNG;
<percussion enthusiasts and, for the cutting edge sampling crowd, the
<  as a very progressive school very cutting edge secondary school but who are
<     and of course, traditional and cutting-edge sounds. <p> The demands on
<p> Reilly: All of this, I think, is cutting-edge stuff.  I think it's a very
<    like 2 Unlimited aren't exactly cutting-edge Techno-but the work of peopl
< new breed of preamp which combines cutting-edge technology for the preamp
< part of a global design team whose cutting edge work products are sold in
```

The LEX4-Database System

UDO KRUSCHWITZ - GUNTER GEBHARDI

Abstract

The aim of the paper is to present the usage of the LeX4–database system (LeX4–DBS) as a tool for lexicographic work, especially in the field of computational linguistics (CL). The system inherits ideas from relational databases for maintenance and fast access. But on the other hand it has to deal with entries of arbitrary length and feature term unification as a matching method. The paper focusses mainly on the presentation of the different methods of multi–user functionality within the LeX4–DBS to support cooperative work in lexicographic projects.

1 Introduction

"In our day, lexicographic projects carried out by one man can result, apart from exceptional cases, only in smaller dictionaries. The usual situation is that there is a staff of workers, the most important members of which are the editors, or the sub–editors grouped around one or two (chief) editor(s)." (Zgusta, 1971). We consider this observation as a major demand for our lexicographic system: support of cooperative work.

The whole system for building up and maintaining lexicons considered here consists of the lexicon formalism LeX4 (Gebhardi and Heinecke, 1995) and the LeX4–SYSTEM, a system of additional tools for special tasks concerning data acquisition, storage and data adaption for applications. LeX4–DBS is the storage component of the system which is a sophisticated system for handling lexicographic data. All necessary data control information for lexicon access and incremental lexicon generation is located in the storage component.

In section 2 we give a brief description of the overall structure of the LeX4–DBS. The different aspects of data access, data ownership, locking, selection and modification control will be discussed in section 3. We discuss these aspects representing some of the important problems arising from data modelling and data storage. These mechanisms become more important the larger the lexicon grows. Finally a conclusion is given.

2 The Basic Structure of the LeX4–DBS

The most important data structures in CL during the last decade have been (typed) *feature structures* (for example (Shieber, 1986), (Carpenter, 1992)). As a consequence we decided to use feature

***This work was supported by the German Federal Ministry of Education, Science, Research and Technology (BMBF) under contract 01 IV 101 G.**

structures also in the lexicon. Moreover (Veronis and Ide, 1992), (Ide, Maitre, and Véronis, 1994) demonstrate the use of feature structures as a data structure to encode lexical information in general. Hence, the decision of using feature structures seems to be a successful base for prospective lexicons in general.

2.1 The Storage Component

A known problem is that none of the established database architectures fits best for storing and handling feature structures. For instance, relational database systems, which are the most important in the field, cannot handle feature structures efficiently, but on the other hand the relational systems support fast data access. Object-oriented database systems are appropriate for storing feature structures, but then it is expensive to encode efficient access methods and, more important, exorbitantly expensive if a modification of the access methods is necessary. A series of experiments using several database systems confirmed this.

As a consequence we decided to use a *heterogeneous* database configuration as the back end of the ℒℯ𝒳𝒜–DBS: one component (general storage component GC) being responsible for storing all information in terms of *random size*, where each stored item can be of arbitrary size, the basis for storing feature structures, and a second component (relational storage component RC) manages information of *restricted size*, where each item can have only a specified maximal size. The actual lexical entries are stored in the GC independent of the fact that certain attributes (features) may have a restricted size and could be stored in the RC. The reason is, keeping all information in one block results in less data organizing effort and faster database access, because we can avoid merging data from the different components.

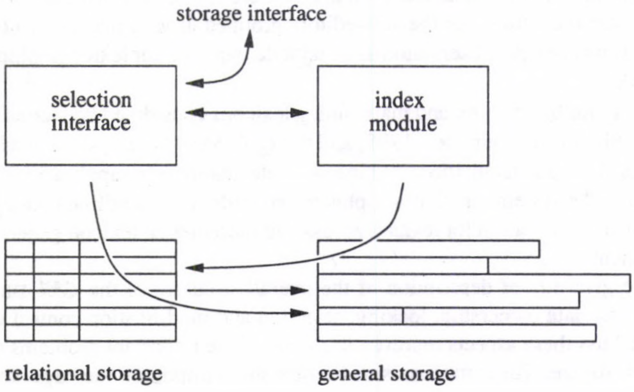


Figure 1: The basic structur of the storage component

The RC is used for indexing the content of the GC. Each entry in the GC is labeled with a unique identification mark and by default the RC contains simple pointers to the entries of the GC using this identification mark. This default linkage is indicated in the sketch of the storage component of the ℒℯ𝒳𝒜–DBS in figure 1 by an arrow from the RC to the GC.

Usually the RC is filled with different indexes. The user of the storage component defines the number of indexes as well as the index function, which calculates the index values on the base of the content of the GC. Besides standard pattern matching indexes which are appropriate for strings and numbers there are functions for feature term indexing (see also section 3.3). The arrow starting at the index block in figure 1 shows the way of building indexes in the RC evaluating the content of the entry in the GC.

The access is guided using the content of the RC. The selection component, see figure 1, analyses a query against the storage component mapping it onto a number of index sets, calculates the intersection of these sets (which is necesseray for indexes on feature structures) and executes an optimized access step using the resulting set. For each user defined index the appropriate mapping function has to be stated.

The advantage of the architecture of our storage component is, that we can make use of standard database technology and only have to add our application specific components, i.e. feature structure dependent index and mapping functions. On this base it is possible to combine relational database technology and technology of database systems that are able to deal with entries of nearly arbitrary length and structure.

2.2 The Storage System Interface

The plain storage component does not provide any lexicographically specific functionality. The storage system interface serves to add this. Figure 2 gives an overview of this component.

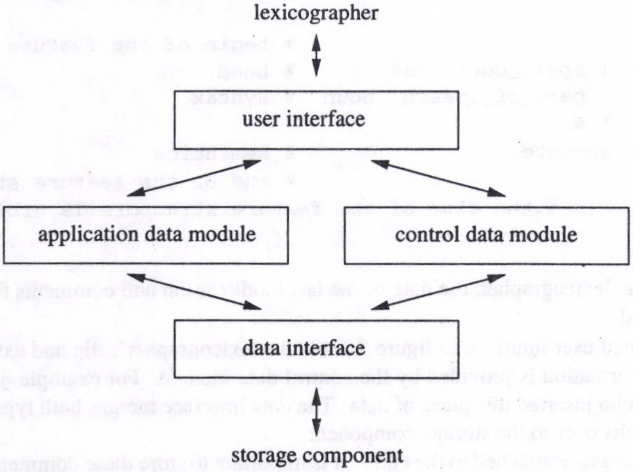


Figure 2: The storage system interface

In order to explain the functionality of the interface component we consider the (toy) feature structure

$$(1)\quad \begin{bmatrix} syn : \begin{bmatrix} spell\_out : & cat \\ part\_of\_speech : & noun \end{bmatrix} \\ sem : animate \end{bmatrix}$$

which could be represented in a pretty printed format of a lexicon as

cat n <*animate*>

The entry is encoded as

```
% this comment should be bound to the following entry
                                  % begin of the feature structure
     syn: ( spell_out: 'cat' &    % head
            part_of_speech: noun  % syntax
          ) &
     sem: animate.                % semantics
                                  % end of the feature structure
```

and stored in a file the lexicographer is editing, using for example the acquisition tool of the ℒℯ𝒳𝒜–SYSTEM (Heinecke, 1996).

Additional information has to be attached to the source code to get the functionality we need for management purposes. In

```
entry(john,                       % lexicographer
      20/3/96,                    % date of last modification
% the number of arguments is arbitrary, but
% has to be of fixed length for all entries in a database

% this comment should be bound to the following entry

                                  % begin of the feature structure
     syn: ( spell_out: 'cat' &    % head
            part_of_speech: noun  % syntax
          ) &
     sem: animate,                % semantics
                                  % end of the feature structure
              % the size of the feature structure is arbitrary
     ).
```

the name of the lexicographer, the date of the last modification and comments for demonstration purposes are added.

The block named user interface in figure 2 reads the lexicographer's file and extracts the entries. The additional information is provided by the control data module. For example `john` is the login name of the user who inserted this piece of data. The data interface merges both types of information and hands the results over to the storage component.

A lot of comments are attached to the entry. It is important to store these comments, because these notes are fairly useful for the person who is responsible for this entry as well as for other lexicographers to compare and to control their work.

3 Data Management

The section before described the basic architecture of the system. In the remainder of the paper we discuss aspects of data management with special attention to the application as a lexicographic database.

3.1 Access restrictions and data ownership

To support cooperative work a major decision was to implement a way of *data ownership* and *restricted access*. In 𝓛𝓔𝓧4–DBS each piece of data belongs to an owner. By default the owner is the user's login name. The access to the data is controlled and we distinguish two levels:

- user access

  A user is permitted to read, insert and modify information he/she owns. A user is *not* permitted to modify information of other users.

  With the reading permission for all data each user can coordinate his/her own work with respect to the existing data in order to get a consistent lexicon as a result. However, the responsibility for each entry is restricted to the editor and, for all entries, to the chief editor.

- super user access

  The super user can take away the access rights of a user and declare the data of that user to be someone else's. This is only required if urgent modifications of some user data are necessary and the user is unable to do it. The rights of a super user are restricted to one user, usually to the person who is the chief editor.

Super user access should only be necessary in emergency cases. A user may hand over his/her data to another user by checking out the data for modification without locking (see below) from the database and give the data to the other user.

This simple method of ownership and different rights of access is sufficient: Each lexicographer in the group is responsible for his/her own data and can control his/her own work with respect to the other members of the group.

The functionality of access restriction and data ownership is located in the storage system interface (2.2).

3.2 Locking

Locking is an important method of avoiding some kind of inconsistencies in the database, i.e. in the lexicon.

When a user is retrieving data it may be desirable that the data is not just selected or deleted but that the selected data should be marked as long as this user does not insert the modified data. So anyone else who wants to read these entries will get the information that they are locked (they can be read but it is expected that they will be changed soon). The process of this kind of selection is called *checking out*.

After checking out entries there are threee different ways of inserting into the database – called *checking in*:

- checking in modified data

  If this happens, the system really deletes the locked data and inserts the new.

- checking in new data

  The system inserts the additional data (and keeps a possible lock).

- checking in no data (explicitly)

  If the user checks in an empty set of data, then the system deletes the locked data.

  Alternatively the user may check out the data without setting a lock. The system will delete the data without checking in an empty set.

The user can unlock the locked data. But only the super user owns the right to do that for any user. Consider for example the situation that a user checks out data and does not check in new entries. Just in this period the data of this user are necessary for delivering the next version of the lexicon. The super user has to unlock the entries of that user.

Some basic ideas of locking in ℒℯ𝒳4–DBS are similar to the concepts integrated in the source code control system SCCS (Rochkind, 1975) or the revision control system RCS (Tichy, 1982). The main difference is that these systems lock complete files rather than single entries.

The functionality of this component is also part of the storage system interface (2.2).

3.3 Selection

Selection is an essential operation in database systems. A query in ℒℯ𝒳4–DBS consists of two parts:

- an entry specific part (the lexicographic information)
- a management data specific part (for example the specified name of the owner)

Of course, the system supports also queries with only one specified part.

The selection algorithm uses two different principles to match the query term against the lexicon entries:

- pattern matching for all non feature structure information
- feature structure unification (including class (type) resolving) for all feature structure information

For example, in order to select the entry (1) discussed in section 2.2 the user could retrieve all entries that `john` owns. The system uses pattern matching.

More interesting are entry specific queries. Consider for example an entry like

$$(2) \quad \begin{bmatrix} spell\_out : bar \\ \left\{ \begin{matrix} \begin{bmatrix} pos : noun \\ syn : \ldots \end{bmatrix} \\ \begin{bmatrix} pos : verb \\ syn : sub : \neg sg3 \end{bmatrix} \end{matrix} \right\} \end{bmatrix}$$

with two alternative readings ($pos : noun$ or $pos : verb$) and the query

$$(3) \quad \begin{bmatrix} spell\_out : & bar \\ pos : & verb \\ syn : & prep : up \end{bmatrix}$$

Possible results are

1. none – the query does not match the entry exactly because the query contains additional information
2. the entry, because the query and the entry unify
3. the entry with only one reading, i.e.

$$(4)\quad \begin{bmatrix} spell\_out: & bar \\ pos: & verb \\ syn: & sub: \neg sg3 \end{bmatrix}$$

because the query restricts the reading to verb as the part of speech

4. the result of the unification of entry and query, i.e.

$$(5)\quad \begin{bmatrix} spell\_out: & bar \\ pos: & verb \\ syn: & \begin{bmatrix} prep: up \\ sub: \ \neg sg3 \end{bmatrix} \end{bmatrix}$$

The user can configure the LeX4–DBS to determine which of the results should be the appropriate one. All different readings are provable equations using the feature logic inferencing machine of LeX4.

The access using feature structure unification is time expensive. Therefore an efficient index mechanism is necessary. The method used in LeX4–DBS was presented in (Gebhardi, 1994).

3.4 Modification Control

Modification Control is a very useful device to observe changes in the database. When checking in entries they have to be compared with the set of entries that were checked out. When an entry is checked in unchanged then only the lock has to be removed. Data that are not checked in again have to be deleted and everything that is new in the set of entries to be checked in has to be physically inserted.

The advantages of this modification control device are:

- the lexicographer may check out as many entries he/she wants to have, but the database system has to deal only with really modified entries

- efficient incremental generation of lexicons

  Compiling a complete lexicon is time expensive. Consider a lexicon with some thousand entries. Usually a lexicographer does not change all entries in the same time, but changing only one entry leads to an new version of the lexicon. Without incremental generation it will be necessary to compile the complete lexicon, only because of one modification. Incremental generation avoids this.

- efficient database garbage collection

  The general storage component contains entries of different length. A maximal or minimal blocksize can hardly be determined. Therefore deleted entries in these files are marked with a flag rather than physically deleted which would be far too expensive. As a result a garbage collection is necessary from time to time. The modification control reduces the number of deleting and inserting actions to a minimum by setting and unsetting flags for single entries. The effect is that the garbage collection has to be called less frequently. However, garbage collection affects also the index files (the relational storage component), which are of much smaller size than the files with the lexical entries. These files usually contain pointers to entries that were deleted. But the index files are sorted (for effective file search) and it has proven to be more effective applying a garbage collection on the index files rather than reordering the index files after every transaction. This is particularly true with large files.

3.5 Further Steps

Currently we make experiments with two *additional locking methods*:

- multi-locking

  A user may check out different data successively and check in the data in any order. Also with additional devices it is complicated to control which data outside the database correspond to which locked data in the database.

- pre-locking

  The idea is that a user may indicate which data he/she wants to lock in the near future.

A second field of ongoing work is to support *versioning* more efficiently. Currently the user can decide whether the system should delete old data or move these to a store. The problem is, that the size of this store grows rapidly. Source code or revision control systems compress the data and reconstruct it on the basis of only one complete data set. In a database system we cannot do so, because fast access on compressed data is impossible. The dilemma is that either we can keep all data, but we have to compress it and cannot have fast access, which is even necessary in the situation of having much data, or we can keep only some data.

4 Implementation

The system is basically implemented in Prolog (e.g. © Quintus-Prolog). In order to allow and to control a restricted user access by different users and to ensure the whole functionality described before it is designed for use in a © UNIX environment. However, it can be run as a single user component as well by changing the user defined settings.

Because ℒℯ𝒳4–DBS is intended for large databases (> 10.000 entries), it is based on stream handling and keeps only a limited amount of data in the working memory. The system setup file can also be adapted to the available memory.

5 Conclusion

The presented system is a means of managing large amounts of relatively unstructured data. This is realized by a combination of relational database methods and extended matching facilities that provide not only pattern matching but also feature term unification. A main point in the design of ℒℯ𝒳4–DBS is the concept of many lexicographers working on the same project. The database functions support this by keeping the collected data consistent and allowing the users to work on their own entries while having the possibility to check the current state of the rest of the database.

References

Carpenter, B. 1992. *The Logic of Typed Feature Structures.* Cambridge: Cambridge University Press.

Gebhardi, G. 1994. Lexical Access in an Integrated Speech-Language System. In F. Kiefer, G. Kiss, and J. Pajzs, editors, *Papers in Computational Lexicography COMPLEX '94*, pages 69 – 78, Budapest.

Gebhardi, G. and J. Heinecke. 1995. Lexikonformalismus $\mathcal{LEX}4$. Technical report, Verbmobil Technisches Dokument 19, Humboldt-Universität zu Berlin.

Heinecke, J. 1996. Lexikonakquisitionstools für den Lexikonformalismus $\mathcal{LEX}4$. Technical report, Verbmobil Technisches Dokument 42, Humboldt-Universität zu Berlin.

Ide, N., J. Le Maitre, and J. Véronis. 1994. Outline of a Model for Lexical Databases. In *(Zampolli, Calzolari, and Palmer, 1994)*, pages 283 – 320.

Rochkind, M.J. 1975. The Source Code Control System. *IEEE Transactions on Software Engineering*, 1(4):364 – 370.

Shieber, S.M. 1986. *An Introduction to Unification-Based Approaches to Grammar.* CSLI Lecture Notes, number 4. Stanford, CA: Center for the Study of Language and Information.

Tichy, W.F. 1982. Design, Implementation and Evaluation of a Revision Control System. In *Proceedings of the 6th International Conference on Software Engineering*, pages 58 – 67.

Veronis, J. and N. Ide. 1992. A Feature-Based Model for Lexical Databases. In *Proceedings of the 15th International Conference on Computational Linguistics, COLING-92*, pages 588 – 594, Nantes.

Zampolli, A., N. Calzolari, and M. Palmer, editors. 1994. *Current Issues in Computational Linguistics: In Honour of Don Walker.* Giardini editori e stampatori in Pisa, Kluwer Academic Publishers, Norwell, MA.

Zgusta, L. 1971. *Manual of Lexicography.* Praha: Academia, Publishing House of the Czechoslovak Academy of Science.

CISLEX – An Electronic Dictionary for German: Its Structure and a Lexicographic Application

STEFAN LANGER – PETRA MAIER – JÜRGEN OESTERLE

Abstract

This paper presents the German electronic dictionary CISLEX and some of its applications. First, it gives a short outline over general requirements for electronic dictionaries, namely completeness, correctness and portability, and shows how the CISLEX meets these requirements. We then describe the size of the lexicon and the kinds of information included. In the next paragraph we describe how the lexicon of full forms can be used by a sentence-oriented lemmatizer for efficient lemmatization of German texts, including segmentation of compounds. Our approach to the semantic classification for nouns in CISLEX is then discussed in more detail and compared to some other approaches, such as, for example, WordNet. We show that the encoding is suitable for a variety of applications. The NP-grammar outlined in the last paragraph finally demonstrates the application of all information included in the lexicon to achieve a semi-automatic extraction of subcategorization frames.

1 Introduction

Electronic dictionaries including semantic information and robust parsers for phrases are important tools for a lot of different NLP tasks like intelligent spell and grammar checking, intelligent text retrieval, automatic classification of texts etc. Besides these well known applications, there is another important application to the lexicon building process itself. The information about the subcategorization frames of verbs is the bridge between parsing phrases like NPs or PPs and parsing complete sentences. In this paper we will show how a semi-automatic extraction of this kind of lexical information from large corpora can be done by using an electronic dictionary with semantic information in combination with an NP-grammar.

2 CISLEX

Electronic dictionaries are quite different from traditional dictionaries on the one hand and from the toy lexicons used in usual NLP systems on the other hand. An electronic dictionary has to fulfil the following demands:

1. completeness
2. correctness
3. portability and reusability (which means that the lexicon is independent from a certain linguistic theory and independent from special applications).

On the basis of an electronic dictionary it must be possible to identify every unit of a text correctly. This identification is called lemmatization or tagging.

Electronic dictionaries that fulfil these demands already exist for several languages. For German the CISLEX seems to be the most extensive one. It contains not only a dictionary of the standard German vocabulary (the so called German Core Lexicon) but also proper names, foreign words, terminology, abbreviations, morphemes etc., all of them contained in separate sublexicons. Naturally the most important part is the German Core Lexicon, that contains all the simple forms of German. Where a simple form is a word that can't be further segmented into subparts that are themselves German words. These simple forms are all encoded with their part of speech and morphological subclass. The morphological subclass corresponds to the inflectional paradigm of the word. In total there are 11 different parts of speech and over 6,000 morphological subclasses in the lexicon of simple forms. The number of entries in the simple forms lexicon is about 95,000 (40,000 nouns, 19,000 verbs, 28,000 adjectives and 8,000 function words). These 95,000 base forms correspond to over one million inflected forms (the inflected forms of the simple forms, prefixed simple forms and regular derivations). This lexicon of the inflected forms is the actual database for the lemmatization and other applications such as spell checking, indexing and grammatical analysis. In addition to this morpho-syntactic classification there exists a semantic lexicon for all the nouns in the simple forms lexicon that is described in more detail in section 3. A syntactic classification of the verbs is in progress as well as a phonological version of the CISLEX.

3 Lemmatization

For further processing, texts have to be lemmatized (or tagged, which means normally an unambiguous assignment of part of speech tags). Under lemmatization we understand the assignment of the following kinds of information to every wordform: <base form, morpho-syntactic category, morphological features>. In addition, syntactic and semantic information is added, whenever it is available. For simple forms the lemmatization is only the look up in the lexicon of the inflected simple forms. Compounds must be segmented according to the German word formation rules, and special forms (i.e. punctuation marks, numbers and forms that contain non alphabetic characters) are analysed in a separate way. This is done by a lexicon based sentence oriented lemmatizer (cf. Maier 1995). For the analysis of unrestricted text, it is very important that each element of the text is assigned a correct lemma information. For this purpose the lexicon of proper names plays an important role. In this subpart of the CISLEX there are more than 250,000 proper names encoded according to their type and morphological properties.[1]

[1] This means, for example, that in addition to the name of a city also the corresponding forms for the inhabitants and adjectives are derivable: to "München" also "Münchner" (someone living in Munich or coming form Munich),

This large amount of lexical information accesible during the lemmatization allows a nearly complete lexical analysis of unrestricted text. As an application we built a large annotated corpus of German newspaper texts (30 mio. running words) where only 0.5% of the tokens remained unknown (most of them misspellings).

Part of the lemmatization process is the lexical disambiguation. This disambiguation either takes place on the word level (as far as possible) or on the sentence level using local constraints on the categorial context that specify possible and impossible part of speech sequences. With this kind of rule based disambiguation, we achieve at the moment a disambiguation on the lexical and morpho-syntactic level of 84%. Instead of using probabilistic methods for further disambiguation, our strategy is to leave the remaining ambiguities to the semantic lexicon and to the grammatical analysis.

With respect to the semi-automatic extraction of subcategorization frames for verbs, we have to pay special attention to one type of ambiguity that arises from the German separable verb prefixes. In German there are a lot of complex verbs of the type preposition+verb. In a verb second sentence the preposition fills the position of the so called right sentence bracket, whereas in verb final sentences the preposition is not separated from the base verb:

(1) Er steht langsam auf
'He stands slowly up'
He stands up slowly

(2) daß er langsam aufsteht
'that he slowly up+stands'
that he stands up slowly

In both cases we have the same verb, i.e. "aufstehen" (to stand up). In the first sentence "auf" is ambiguous because it could be a preposition or it could be part of a complex verb. Therefore, we must specify for each verb the list of possible separable prefixes. In the case of the first example, the prefix list for "stehen" (to stand) contains the preposition "auf" (up or on) and from the fact, that the last position in a sentence is not a possible position for a preposition, we can infer the correct verb "aufstehen" (to stand up). As, in general, the subcategorization properities of the simple verb and its prefix derivations are different, the disambiguation is very important when we are going to extract the subcategorization frame out of a sentence.[2]

In a second step larger units that don't really require a grammatical analysis or that can be treated with local grammars like various forms of dates (10.10.96, 10. Oktober, 10/10/1996, etc) or complex names (Dr. Karl X. Napf, Daimler Benz AG, ...) are tagged. The idea behind this second step is that a grammar (in our case the NP-Grammar) can use these units as a whole regardless of their internal structure.

The lemmatizer produces SGML output, where sentence boundaries, word form, base form, part of speech, morphological features, the type of the form in the case of special forms and the head and other components in the case of compounds are marked.

An example for the output of the lemmatizer can be found in the appendix. A tagged sentence is defined as an element "satz":

```
<!ELEMENT satz         (token|mwl)+ >
<!ELEMENT mwl     - -  (token)+ >
<!ELEMENT token   - -  (form,lemma+) >
<!ELEMENT form    - -  (#PCDATA) >
<!ELEMENT lemma   - -  (#PCDATA) >
```

A "token" contains the analysis of single forms, "mwl" contains multi word units like complex names. The "token" element has at least two subelements: the "form" element, containing the original form and one ore more "lemma" elements, each of them corresponding to a possible analysis. The "lemma" element contains the base form. Furthermore, it has several attributes which can have a great number of different values which are not listed in the description above. The attributes have the

"Münchnerin" (the corresponding femal from) and "münchnerisch" (the corresponding adjective 'being form munich', 'typical for Munich') can be recognised.

[2] In general, the subcategorization of the complex verb can't be predicted form the subcategorization of the base verb.

following function: In the "typ"-attribute it is encoded whether the form is analysed as a simple form, a compound or a dashed word etc. The "kat"-attribute contains the part of speech and the possible morphological features of the word are defined by the "mor"-attribute. The possible feature bundles are enumerated and "mor=265", for example, stands for the disjunction of the following feature bundles: nom-sing-fem, acc-sing-fem, dat-sing-fem, gen-sing-fem. "syn" is an attribute for the syntatctic information, for example the type of pronouns (relative pronoun, possessive pronoun,...) or subcategorization. Subcategorization is only encoded for prepositions, in the case of verbs this information is not yet complete, therefore this attribute is not filled for verbs. In section 5 we present a method to extract this kind of information semi-automatically by using the lexical information so far encoded for the parsing of noun phrases and prepositional phrases.The semantic features described in further detail in section 4 are represented as the argument list of the attribute "sem". The "pref" attribute gives a preference ranking for a lemma.

4 Semantic Encoding of Nouns

In addition to morpho-syntactic information, used for efficient lemmatizing, there is a semantic encoding for all simple forms in CISLEX. This information is designed to serve a variety of purposes, such as description of selectional constraints, analysis of compound nouns and disambiguation of polysemic words. It can also serve as a basis for information retrieval purposes.
When conceiving the semantic encoding for CISLEX, several choices had to be made, as lexical semantics is an extremely difficult field. Much more than in morphological and syntactical encoding, there is little consensus about the way semantics should be included in a comprehensive dictionary (cf. Calzolari 1994). Generally speaking, there are two approaches for encoding lexical semantics for broad coverage:

- manual encoding by a lexicographer, eventually using printed and machine readable dictionaries as a resource. The disadvantage here is that this is a very time consuming task.
- statistical approaches, mainly based on word-coocurrence in large text corpora. This is much more efficient, but it isn't straightforward what to do with the output.

The best known electronic resource based on the first type of approach is certainly WordNet (Miller et al. 1991), a freely available and comprehensive semantical database for English. For every lemma, it contains links to synonyms, hyponyms, meronyms and otherwise related words.
Statistical approaches like the one described in Church (1994) have been shown to have an only partly useable outcome, and the application of methods adopted for English to languages with a richer morphology has turned out to be difficult.(cf. Breidt (1993) for German). Of cause there are also hybrid approaches, as the ones described for Italian in Velardi/Pazienza/Fasolo (1991).
After the evaluation of existing encodings, we have adopted the following approach for the CISLEX-dictionary (for details see Langer (forthcoming)): The simple forms (not prefixed and not complex forms) are being manually encoded. For the complex forms, especially compound nouns, we currently investigate the possibilities of statistical clustering, based on the code for the simple forms. This seems to be the only viable approach, as compounding is an extremely productive pattern in German word formation.The core of the semantic encoding for the nouns is the indication of one or several semantic classes for each lemma. A semantic class can be viewed as a node in a hyponymic structure of the vocabulary similar to WordNet, such as BIRD or PROFESSION. The difference between our approach and most approaches we know of is the inclusion of information about different features of semantical classes in our encoding. Whereas often classes of lexemes obeying common selectional restrictions on the one hand and taxonomic classes on the other are conflated, and selectional restrictions and combinatoric features are used to define taxonomic classes, we make a difference between the features 'taxonomic' and 'selectional'. A pure class of the first type would certainly be one like BIRDS, whereas pure selectional classes, due to their definition, can't be subsumed under a single expression, and have to be defined by means of typical contexts, e.g. following the approach described in Gross (1994). Certainly, the assumption of a certain uniformity between selection and taxonomic organisation can be shown to be supported in the CISLEX-classification, as most classes have both features. The distinction of this two features allows a quick evaluation of the suitability of classes for different types of applications - whereas for text generation, machine translation etc. selectional classes are more relevant, for purposes of information retrieval only taxonomic classes have to be considered. In addition to the hyponymic structure, the semantic encoding contains links to synonyms, meronyms and antonyms, and some other semantic relations. The number of compound lexemes occuring in corpora largely exceeds one million and new

compounds are always formed, which makes it impossible to encode all occuring compound nouns manually. Using the classes encoded for the simple nouns, we have carried out studies on disambiguation of compounds. These first tests have shown that semantic patterns can be detected in large compound noun corpora, which could be exploited for an automatic or semi-automatic analysis, making use of a large, one million words corpus of compounds consisting of two nouns. Sense disambiguation of compounds and their parts corresponding to these classes includes the choice of the appropriate class for a lexeme in a text, when it has several semantic categories in the lexicon entry. It also includes the selection of a relation between the parts of the nouns, based on the semantic category of the parts. The probabilistic analysis of the corpus showed that it is possible to identify clusters of compounds where parts have similar relations to each other. Also nouns having polysemic parts can be classified according to the different meanings of the parts.

In tests we have carried out on a sample of newpaper texts, it could be shown that the classes in CISLEX are also suitable as a basis for information retrieval. For this purpose, semantic classes were bundled to build thematic classes. Subsequently, the texts were tagged with the thematic classes and then statistically classified (for details see Langer (forthcoming)).

5 NP-Grammar: Its application to semi-automatic extraction of subcategorization frames

For the most frequent NP-structures in German a definite clause grammar (DCG) was developed and implemented. The basis of this grammar is the output of the lemmatizer described above. This means that the terminal symbols for our grammar consist of the entities which are recognised and marked up by the lemmatizer. The grammar then identifies chains of entities that can be analysed as NPs or PPs. Besides the combination of syntactic categories, the grammar also checks for agreement between determiner, adjectives and nouns. Furthermore, it is checked that the case of an NP which functions as the complement of a preposition is governed by the preposition. As a preposition can gover different cases depending on the semantic function it has, it is only checked that one of perhaps several possible cases is licensed by the preposition. Furthermore, the nominal head of an NP with its semantic class and its morphological features is given (cf. the appendix).

At the moment, we do not have any subcategorization information in the lexicon. Of course, one possibility to encode this information in the lexicon, is to fill in this information manually. This means that an encoder has to specify for each single lexical unit which can subcategorize other elements, i.e. verbs, nouns and adjectives, what kind of elements can be subcategorized. This task is very time consuming and, furthermore, it is often very difficult to think of all possible usages of a word without having a context in which it is used. Therefore, our approach is to use the output of the NP-analysis in the process of the encoding of subcategorization information. As can be seen from the example given in the appendix, a sentence gets transformed by the NP-grammar to a list of NP-, PP- and verb constituents. This list is then used to assign semi-automatically subcategorization frames to verbs, i.e. the program presents to the encoder a possible subcategorization frame for the verb in the sentence. If he or she answers affirmatively the entry gets added to the lexicon. This is a situation comparable to the one the "elves" were in, who encoded the lexical entries for the COMLEX-system (cf. Grishman/Macleod/Meyers (1994)). However, our system has the advantage that due to the preprocessing in form of the lemmatization and the NP-analysis, the encoder is confronted with examples that have a high probability to be correct entries. The information in a subcategorization frame consists of the syntactic category (NP or PP), the case or preposition and the semantic class of the head noun of the PP or NP.

There are, however, some problems. First, if there are more than two main verbs in a sentence, it can be difficult to decide which complements belong to which verb, as is demonstrated by the following example:

(3) Peter versuchte den Ball mit der Hand zu fangen.
Peter tried to catch the ball with his hand.

In order to determine, e.g., that "Peter" is the subject of "versuchen" (to try) and "den Ball" (the ball) the object of "fangen" (to catch), it would be neceessary to recognise imbedded sentences - which is, of course, not possible if there is no information about subcategorization frames included in the lexicon.

Another problem stems from the fact that we can, at the moment, not distinguish between modifiers and complements. For example, in both of the sentences

(4) Peter antwortete auf die Frage.
Peter answered to the question.
(5) Peter saß auf dem Tisch.
Peter was sitting on the table.

the PP would be assumed to be a complement of the verb, i.e. the encoder would be asked whether the PP is a complement. To get better-guesses whether a phrase functions as a modifier or as a complement, we want to use the semantic encoding of nouns. For example, it is unlikely that the PP in sentence (4) could function as a directional adverbial because the noun "Frage" is tagged with a feature saying that it is abstract and, probably, such a noun cannot function as the head of an NP which is inside a directional PP-adverbial. Furthermore, the NP-analysis together with the semantic encoding of nouns could be used for PP attachment disambiguation along the lines of Brill/Resnik (1994) who use semantic information in form of word classes from WordNet (Miller et al 1991).
A last point which has to be mentioned, is the problem of ambiguities, as far as the PP-attachment to nouns and verbs is concerned. So, e.g., in the following sentence it cannot be decided by our grammar whether the PP is part of an NP ("Dem Mann auf die Frage") or functions as a complement or adverbial of the verb.

(6) Peter antwortete dem Mann auf die Frage.
'Peter answered the man on the question'
Peter gave the man the answer to the question.

All the problems mentioned above can result in giving the encoder subcategorization frames which are not correct and have to be rejected, which is, of course, something one wants to avoid. Therefore, the first thing we do, is only to select sentences with one main verb from the tagged corpus [3] which get then parsed by the NP-parser. Thus, the problem of deciding which phrase corresponds to which verb is avoided. As far as the problem of PP-attachment ambiguities and the distinction of complements and modifiers is concerned, the position of the verb in the sentence is taken into account. In German, in sentences with only one main verb the finite verb (an auxiliary or the main verb) can be in sentence initial position which means that the sentence is a question or an imperative:

(7) Hast [auxiliary verb] du mir das Buch gegeben?
Have you given the book to me?
(8) Gib [imperative verb] mir das Buch!
Give the book to me.

Another possibility is that the verb is in verb-second position, which means that there is one non-verbal constituent at sentence initial position and then a finite verb (e.g. NP V NP, NP V, PP V NP), as e.g.:

(9) An dem seit einem Jahr geltenden Grenzwert für Alkohol am Steuer wird [finite, auxiliary verb] nichts geändert [main verb].
The alcohol limit for drivers which has been in force for one year will not be altered.

Because of the position of the finite verb, all constituents occuring before the verb have to make up one consituent. In the example above, therefore, all the PPs occuring before the verb have to be subconstituents of one big PP and are definitely no complements or modifiers of the verb. As sentences of this kind make up the greatest part of our sentences, the number of problematic examples can be reduced very much.

[3] Approxiamtely a third of the corpus consists of such simple sentences.

6 Appendix

In (11) an example is given for the SGML-markup which is produced by the lemmatizer for sentence (10):

(10) Der ehemalige Nationaltorhüter Uli Stein überreichte Dirk Schuster am Mittwoch die Trophäe.
The former international goal keeper Uli Stein gave on wednesday the trophy to Dirk Schuster.

(11)

```
<satz>
        <token>
                <form>Der</form>
                <lemma typ=EF kat=D mor=244 syn=D sem=[] pref=1>d</lemma>
        </token>
        <token>
                <form>ehemalige</form>
                <lemma typ=EF kat=A mor=274 syn=0 sem=[] pref=1>ehemalig</lemma>
        </token>
        <token>
                <form>Nationaltorh&uuml;ter</form>
                <lemma typ=KF kat=N mor=283 syn=0 sem=[mta,bla,ber,men,leb]
                          pref=1>national+tor@h&uuml;ter</lemma>
        </token>
        <mwl typ=PN mor=146>
                <token>
                        <form>Uli</form>
                        <lemma typ=EN kat=E mor=146 syn=0 sem=[] pref=1>uli</lemma>
                </token>
                <token>
                        <form>Stein</form>
                        <lemma typ=EN kat=E mor=146 syn=0 sem=[] pref=1>stein</lemma>
                        <lemma typ=EF kat=N mor=282 syn=0 sem=[kon,knk]
                                  pref=1>stein</lemma>
                </token>
        </mwl>
        <token>
                <form>&uuml;berreichte</form>
                <lemma typ=EF kat=V mor=14 syn=0 sem=[] pref=1>&uuml;berreichen</lemma>
        </token>
        <mwl typ=PN mor=146>
                <token>
                        <form>Dirk</form>
                        <lemma typ=EN kat=E mor=146 syn=0 sem=[] pref=1>dirk</lemma>
                </token>
                <token>
                        <form>Schuster</form>
                        <lemma typ=EN kat=E mor=146 syn=0 sem=[] pref=1>schuster</lemma>
                        <lemma typ=EF kat=N mor=283 syn=0 sem=[bha,ber,men,leb]
                                  pref=1>schuster</lemma>
                </token>
        </mwl>
        <token>
                <form>am</form>
                <lemma typ=EF kat=D mor=179 syn=PD sem=[] pref=1>an</lemma>
        </token>
        <token>
                <form>Mittwoch</form>
        <lemma typ=EF kat=N mor=282 syn=0 sem=[zwt,zei] pref=1>mittwoch</lemma>
    </token>
    <token>
        <form>die</form>
        <lemma typ=EF kat=D mor=337 syn=D sem=[] pref=1>d</lemma>
        <lemma typ=EF kat=D mor=362 syn=P3 sem=[] pref=1>d</lemma>
    </token>
    <token>
        <form>Troph&auml;e</form>
        <lemma typ=EF kat=N mor=265 syn=0 sem=[kon,knk]
                pref=1>troph&auml;e</lemma>
    </token>
    <token>
        <form>.</form>
        <lemma typ=SF kat=S mor=S1 syn=0 sem=[] pref=1></lemma>
    </token>
</satz>
```

The NP-grammar, then, produces the following output for this tagged sentence:

(12) NP(nom, [mta,bla,ber,men,leb], "Der ehemalige Nationaltorhüter Uli Stein") überreichte
NP(nom + acc + dat, [], "Dirk Schuster") PP(am, [zwt,zei], "am Mittwoch ")
NP(acc, "die Trophäe")
Subcategorization frame: NP(nom), NP(dat), NP(acc)

In this example, the PP is recognised as a temporal adverbial because the head noun of the imbedded NP is marked as a day of the week. Therefore, the PP is not analysed as a modifier of the proper noun "Dirk Schuster" and it is also not presented as a possible complement of the verb, i.e. an element of the subcategorization frame.

7 References

BREIDT, Elisabeth (1993): Extraction of V-N-Collocations from Text Corpora. A Feasibility Study for German. In: ACL: Workshop on Very Large Corpora.

BRILL, Eric; Philip RESNIK (1994): A Rule-Based Approach to Prepositional Phrase Attachment Disambiguation. cmp-lg 9410026

CALZOLARI, Nicoletta (1994): Issues for Lexicon Building. In: Zampolli/Calzolari/Palmer: Current Issues in Computational Linguistics. Pisa/ Dordrecht: Giardini/Kluwer, pp. 267-281.

COURTOIS, Blandine; Max SILBERZTEIN (1989): Les dictionnaires electroniques DEELAS et DELAC. Actes colloque sur les langues romanes, Universite Laval:Quebec.

CHURCH, Kenneth Ward et al. (1994): Lexical substitutability. In: Atkins/Zampolli: Computational Approaches to the Lexicon. Oxford: Oxford University Press, pp. 153-177.

CHURCH, Kenneth Ward (1988): A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In: Proc. of the Second ACL Conference on Applied Natural Language Processing.

GROSS, Gaston (1994): Classes d'objets et description des verbes. In: Langages 115. Paris: Larousse, pp. 15-30.

GROSS, Maurice (1989): The Use of Finite Automata in the Lexical Representation of Natural Languages. In: Gross, M. & D. Perrin (eds.) Electronic Dictionaries in Computational Linguistics. Springer.

GRISHMAN, Ralph; Catherine MACLEOD and Adam MEYERS (1994): Comlex Syntax: Building a Computational Lexicon. Computer Science Department, New York University. cmp-lg 9411017.

LANGER, Stefan (forthcoming): Selektionsklassen und Hyponymie im Lexikon. Doctoral Dissertation. München: CIS-Berichte.

MAIER, Petra (1995): Lexikon und automatische Lemmatisierung. CIS-Bericht 95-84. CIS, Universität München.

MILLER, George et al. (1990): Wordnet: An on-line lexical database. In: International Journal of Lexicography 3/4 (special issue), pp. 235-312.

VELARDI, Paola; Maria Teresa PAZIENZA; Michela FASOLO (1991): How to Encode Semantic Knowledge. A Method for Meaning Representation and Computer-Aided Acquisition. In: Computational Linguistics 17/2, pp. 153-170.

Linguistic Foundations for a Computerised Analysis of Latvian

BAIBA METUZĀLE-KANGERE

Latvian is not only highly inflected as a language, but it also has a rich structure of word-formation that is both regular and productive. The feature of the derivational system of particular interest for computer analysis is the system of suffixation through which change of word category is effected and which functions in close relation to the paradigms of word classes. This paper presents a model of computer analysis of running texts developed around lists of roots, suffixal patterns and sets of paradigms based on morphemic segmentation. Through the classification of root morphemes and the examination of morphemes to the right of the root and the left of the inflectional morpheme and matching these with sets of paradigms, parsing of texts by computer may be achieved.

Linguistic Foundations for a Computerised Analysis of Latvian

Baiba METUZĀLE-KANGERE

1.0 Introduction

Computer linguistics on Latvian is relatively new and started at the Artificial Intelligence Laboratory (AIL) at the Department of Mathematics and Computer Science at the University of Latvia in Riga under the direction of Dr. Andrejs Spektors in 1990. It soon became evident to Dr. Spektors that one of the keys to developing language programmes for Latvian is the morphology of the language. Our co-operation began when Dr. Spektors decided to enter into the computer a dictionary of Latvian roots plus the list of lexical items pertaining to each root and divided into morphemes, as represented in my Derivational dictionary of Latvian (DDL)[1]. This corpus was used in order to effect an automatised analysis of Latvian, i.e. the segmentation of running texts into morphemes. The project has been operating for two years and to date the rate of correctness in segmenting newspaper texts is around 90% which we aim to improve to a higher level. It is considered that current newspaper texts introduce a high level of difficulty since they contain large numbers of occasionalisms that need not necessarily become part of the language in a longer perspective. It is thought that more conservative texts would actually yield a much higher rate of correctness.

The next step is to achieve a full parsing of the language. It is considered that the models of parsing evolved by Scandinavian scholars such as Karttunen, Koskinniemi, Hellberg , Karlsson etc. can be applied to Latvian, but an element not covered by these models that is of importance to Latvian is derivational morphology. Now derivation is a much less perfect instrument than inflection, but, as noted by Bybee 1985[2], it is nott easily possible to draw a distinct line between inflection and derivation. There are several good reasons for using the morphemes right of the root morpheme and left of the inflectional morpheme in texts analysis by computer:

a) these are units discernable by computer

b) these units have two main functions: firstly, a semantic function; secondly, changing word category. The latter introduces new possibilities of inflection and thus of syntactic structure .

c) inflection in Latvian is fraught with syncretism and homography across word categories, e.g. the ending *-u* may be a marker of accusative singular, genitive plural, first. person singular marker for all tenses, the marker for the conditional; similarly, *-ām* may mark the dative plural feminine, first person plural present tense, first person plural past tense, present participle active. In order to cope with this, determining word category, noun declension and verb conjugation is of importance and this paper aims to show that this may be done by the computer to a large extent.

2.0 A model of computer analysis based on morphemic segmentation

2.1 The model presented here uses the segmentation into morphemes of words of a running text referring them to a lexicon of roots (R) and matching them with the paradigm of the base word according to the DDL. If this match results in finding the flection (F) registered in the running text , then it is assumed that we have identified the form of the word. If no match can be effected, then further lists and patterns are consulted until the form of the word is identified.

The procedure is henceforth presented in more detail, followed by an example of the operation. It is to be pointed out that this is a general framework to be further investigated and that a wealth of research is to be invested into the perfection of the method so as to cover every instance that can occur in texts and clear away the difficulties that are bound to arise.

2.11 The following tools are necessary:

a) Automatic morphemic segmentation

[1] Metuzäle-Kangere, Baiba. A Derivational Dictionary of Latvian, Hamburg, Buske, 1985

[2] Bybee, Joan Morphology Amsterdam, Philadelphia 1985

b) A set of roots listed according to part of speech: verb (V); noun (N); adjective (A); adverb (ADV); Numeral (NUM), etc. Further, the class of International Words (IW) is also to be marked[3]. It may seem a contradiction of terms to assign a word category to a root. In fact, the DDL postulates a base word from which the other words containing the root are said to be derived, the rationale being that words are derived from other words and not through a process of IA. In the majority of cases, the choice of the base word was to large degree formal: words of the type R+F were seen as the simplest forms and thus given precedence. There were, of course, cases where other than formal criteria needed to be invoked , e.g. in the case of *skāb s* (sour), *skāb e* (acid), *skāb t* (to turn sour), the adjective was chosen, since the noun form is a late development and the verb belongs to a series of change of state verbs i.e. they imply a quality as the point of departure.

c) Sets of paradigms for above classes V, N, A etc. according to cojugation or declension respectively.

d) A set of roots together with a paradigm according to the base word as registered in the DDL.

e) A list of R+F type words that are not base words together with their paradigms

f) Derivational hierarchies[4] (see appendix). It is to be noted that these need to be expanded and revised, but they are reproduced here for the sake of example.

g) Word class and paradigms for pre-inflection morphemes. In Latvian, the penultimate derivational suffix as a rule determines the declension of the noun. For this reason, in fact, traditional descriptions of word formation do not use the concept of morpheme, but work with the notion *izskaņa* (lit. "final sound or sounds") that is in fact a morpheme cluster of F and preceding suffix/es. In some cases, there is both a masculine and feminine declension for a given suffix that must also be taken into account.

2.12 The following procedure is suggested:

a) Segmentation of the text into morphemes.

b) Identification of the root morpheme/s and their word class with reference to the list of roots and their type.

c) Classification of the words of the text into : (i) R; (ii) R+F; (iii) R+Suffix+F

d) Matching of root morpheme paradigm for (i) and (ii) with the word form in the text. If the match may be effected, then the form of the word is identified, i.e. parsed. Note that this copes with zero endings in verb paradigms. If not, it is referred to the list as in e) above.

e) In the case of (iii), the category of the root morpheme is checked against the derivational hierarchy that should arrive at a penultimate pre-inflectional suffix. this in turn is checked against the list as in g) above. A single long vowel morpheme (*ā, ē, ī, o, ū*[5]) after R marks the word as a secondary verb or a form derived from a secondary verb.

2.13 Not all words will be parsed immediately through the above procedure and further solutions are to be sought in the syntactic relations in the text. However, the bulk of parsing of the mainstream corpus in Latvian should be able to be effected in this way, as demonstrated in 2.2.

3.00 The proof of the pudding...

The following sentence was chosen from the section on home and family in the main daily in Latvia, *Diena* (Day) early July, 1996:
Salīdzinot dažādu dārzeņu spējas palīdzēt cilvēkam cīnīties pret slimībām, liekas pārsteidzoši, bet saslimšanu ar vēzi var novērst, uzturā lietojot kāpostus.
(Comparing the potential of various vegetables for helping a person to fight against illness, it seems surprising, but contracting cancer may be avoided by using cabbages in nutrition.)

3.10 If we leave aside all prepositions and conjunctions, the text may be analysed into morphemes as follows:

sa līdz in o t	dաž ād u	dārz eņ u	spēj as
pa līdz ē t	cilvēk am	cīn ī t ie s	slim īb ām
liek as	pār steidz oš i	sa slim šan u	vēz i
var	no vērs t	uz tur ā	liet o j ot
kāpost us			

[3] For a discussion of the rationale for this , see DDL, pp. xiv-xxii and Metuzāle-Kangere "International Words (IW's) in Latvian and Lithuanian" in Slavic Themes from Two Hemispheres ,Selecta Slavica 12, eds. B. Christa, W. Gesemann,M. Pavlyshyn, H.W.Schaller, H-P. Stoffel, R. Sussex, Hieronymus, Neuried 1988

[4] As in the DDL, pp. xii-xiii

[5] *o* is in fact a diphthong written as *uo* in Lithuanian and so qualifies a long vowel

3.20 The root morphemes may be identified and classified as follows:

līdz	ADV	daž	A	dārz	N	spēj	V
līdz	ADV	cilvēk	N	cīn	V	slim	A
liek	V	steidz	V	slim	A	vēz	N,V
var	N,N	vērs	N,V	tur	N	liet	N
kāpost	N						

3.30 The words of the form R or R+F are the following:

spēj as
var
cilvēk am
no vērs t
vēz i
uz tur ā
liekas
kāpost us

On comparing these with the paradigm of the base word according to the root morpheme, we locate *cilvēkam, kāpostus* and thus may determine their type, in these cases declension which gives us automatically their gender, and the place of the inflectional ending in the paradigm, i.e. the number and case. Similarly, for *liekas,* we know the conjugation, tense, person. Since it is 3rd. person, we cannot determine number, since Latvian has a common 3rd. person verb ending for singular and plural throughout . The case of *novērst* is disambiguated, since only the verb paradigm gives us the item in the text. However, *vēzi* can be identified in both of the paradigms for V and N - syntactic means, e.g. preceding preposition or the existence of another declined verb between the commas can be used to determine the correct paradigm. In the case of *var,* neither of the noun paradigms fit, nor can a match be found for *spējas, uzturā.* These are then referred to the list as in 2.11 e) and from this list we may identify each of the above , i.e. fully parse them.

3.31 The next step would be to take the words that have a long vowel directly after the root morpheme since these are all secondary verbs and the long vowel to a large extent determines their conjugation, whereafter the paradigms for the conjugational subtypes may be matched to the verb in the text.. If this cannot be done, then one must refer to the derivational hierarchy for secondary verbs. In this way, the parsing for *palīdzēt, cīnīties, lietojot* may be effected. Note that in this way, we may locate zero endings.

3.40 The remaining words are of the form R+Suffix/es+F, viz.:

sa līdz in ot
sa slim šan u
daž ād u
dārz eņ u
slim īb ām
pār steidz oš i

Although we do not have as yet a derivational hierarchy for adverbs, we can forecast that the causative suffix *in* forms verbs of a given subtype in the third conjugation with no exceptions would be part of this and the participial form in this text may be found through matching of paradigms and thus fully parse *salīdzinot.* In the case of *slimībām, saslimšanu,* the penultimate suffixes provide all information necessary for matching with a paradigm, incidentally one and the same for both. For the former, this is sufficient for full parsing, but in the case of the latter, we need to make the choice between accusative singular or genitive plural and this must be done by syntactic means, e.g.. a gen. requires either a following noun without the intervention of a preposition or else that the verb *vērst* governs the accusative case and we can find an accusative in between the commas. In the case of *dārzeņu,* the paradigm gives us the genitive plural of the noun. Since *dažādu* is an adjective, it can be either masc. or fem.,and this is determined syntactically with reference to the nearest following noun. With regard to *pārsteidzoši,* the derivational hierarchy chart is of no benefit, since nothing from a verbal root is derived with the suffix *oš*. However, on matching with the verbal paradigm, we find that this is a participial form and from which the adverb may further derived .

3.50 The above is an outline, model for further research and therefore does not pretend to take into account numerous details that are likely to lead to difficulties in programming with which one would need to deal before we could boast of system of analysing Latvian by computer. In fact, I took care to select a text that glosses over rather than accentuates some of the problems with which we have been struggling to date. This was done in order to present a clearly discernable framework so as to avoid the situation of not seeing the forrest for the trees. However, the aim of our research is to achieve an analysis of all printed texts as they occur, not texts selected for their lack of difficulty. At his stage, it seems to me that the direction outlined above is worth pursuing and that the mini example demonstrates that there is substance for the claims made.

4.0 Concluding remarks

The method presented above is developed around lists of roots, suffixal patterns and sets of paradigms. The lexicon as such is not considered in this outline as such although the DDL may also be considered as the lexicon. This seems a natural way to analyse Latvian texts by computer, since many of the suffixal derivations are regular and productive: in the case of the suffix *šan,* the potential for productivity may be rated at 100%. The fact that this suffix, as many of the others, invariably forms nouns of one particular declension means that this information may actually be coded and used as well as the fact that this is a deverbal noun. In this lies further potential: compare the words *slim īb a* and *sa slim šan a* from the above text. Traditional computer analysis would place them in one and the same category: 4th. declension, i.e. feminine nouns. The system of working with roots and suffixal patterning tells us that although both nouns are formed from an adjective, the latter has undergone a process of deverbalising before becoming a noun. Furthermore, if we were to come across the noun *slimošana,* the long vowel after the adjectival root would tell us that a secondary verb may also be formed and a noun from this stem. This provides the potential to discover by computer the difference in meaning between the two deverbal nouns, viz. the act of becoming ill vs. the state of being ill. Whilst we are a long way from processing such information, it seems desirable nonetheless to capture it in the hope that some headway in this direction may be made in future.

Through the channelling of information with regard to word category into the formants making up words, it seems plausible that the writing of syntactic rules could be facilitated. This is desirable in a highly inflected language, since a high rate of inflection brings in its wake loose word order. It seems logical to assume that the elements that have a good rate of productivity are a dynamic part of the language. For this reason, they should be treated as such, not left as static, or even worse ignored, which is largely the case with affixation in systems of tagging the lexicon.

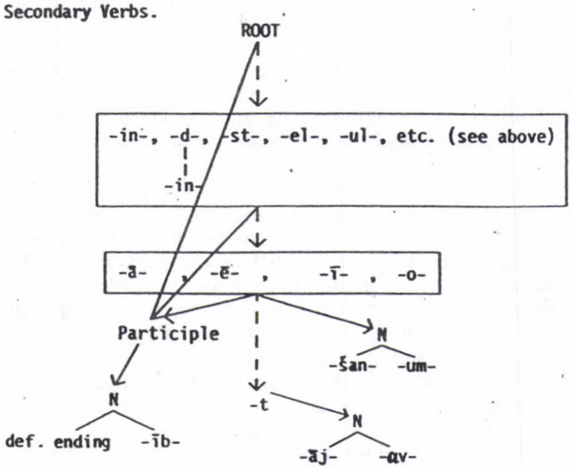
Another reason for working with word formational elements is that lists of these do not increase at the same rate as lexical lists with the advent of new words in the lexicon unless it is a question of exotic and esoteric loanwords. Similarly, the lists themselves are significantly shorter. This may not be a deciding factor for computer analysis, since the computer can cope with large numbers. However, if systems are sought then quantity must be seen as unwieldy.

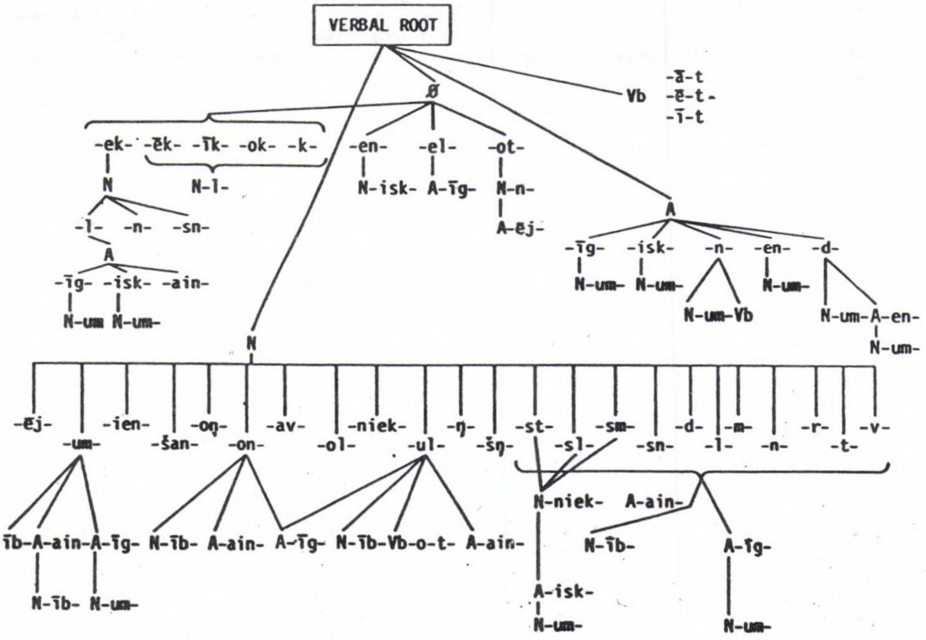
The obvious disadvantage of the method outlined above is that it only works for languages that are relatively easy to segment into morphemes. Of the IE languages with which I am acquainted, this is satisfactory only for the Baltic, Slavic and Classical languages. However, the fact that the findings of research cannot immediately be generalised should not be a deterrant for expanding fields of enquiry.

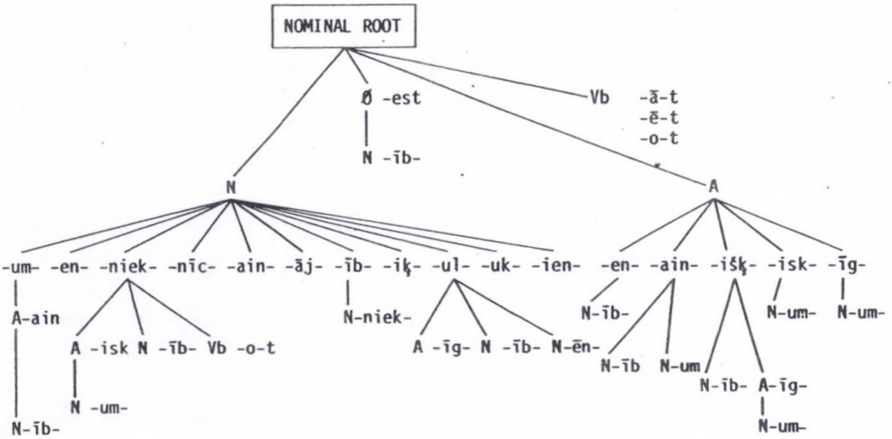
APPENDIX

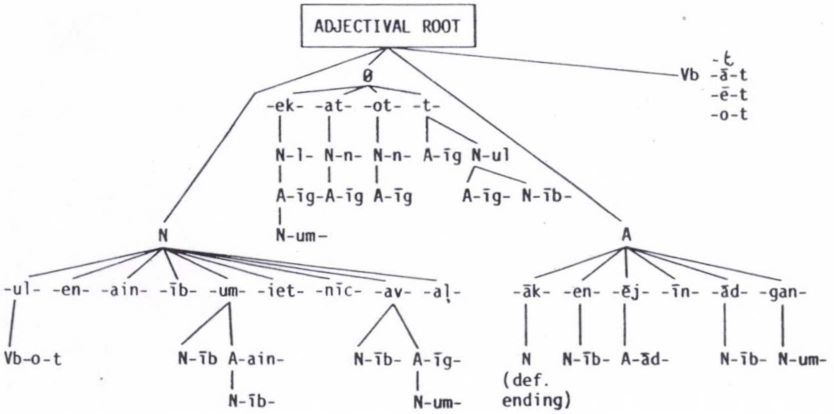
Derivational Hierarchies

1. Secondary Verbs.






NOMINAL ROOT
Ø -est
N -īb-
Vb -ā-t
-ē-t
-o-t
N
A
-um- -en- -niek- -nīc- -ain- -āj- -īb- -iķ- -ul- -uk- -ien-
A-ain
N-īb-
A -isk N -īb- Vb -o-t
N -um-
N-niek-
A -īg- N -īb- N-ēn-
-en- -ain- -išķ- -isk- -īg-
N-īb-
N-īb N-um
N-īb- A-īg-
N-um-
N-um- N-um-


ADJECTIVAL ROOT
Ø
-ek- -at- -ot- -t-
N-l- N-n- N-n- A-īg N-ul
A-īg-A-īg A-īg
N-um-
A-īg- N-īb-
Vb -ṯ
-ā-t
-ē-t
-o-t
N
A
-ul- -en- -ain- -īb- -um- -iet- -nīc- -av- -aļ-
Vb-o-t
N-īb A-ain-
N-īb-
N-īb- A-īg-
N-um-
-āk- -en- -ēj- -īn- -ād- -gan-
N
(def.
ending)
N-īb- A-ād-
N-īb- N-um-

Architecture and Functioning of a System for the Acquisition of Taxonomical Information from Dictionary Definitions

SIMONETTA MONTEMAGNI

Abstract. The paper describes a system for the automatic acquisition of taxonomical information from dictionary definitions. The system operates in two different stages: first, dictionary definitions are syntactically parsed by a general purpose Italian grammar which has been tailored to deal with dictionary input; the acquisition of taxonomical information is carried out during the second stage by a component mapping lexico-structural patterns onto the output of the syntactic analysis stage. Performance and results of the acquisition procedure applied to the whole sets of noun and verb definitions of a monolingual Italian dictionary are illustrated.

1. Introduction

In this paper, a system for the automatic acquisition of taxonomical information from dictionary definitions is illustrated: first, a general overview of the approach to lexical acquisition is provided together with the system architecture; second, the extraction procedure is described and exemplified; finally, performance and results of the acquisition procedure applied to the whole sets of noun and verb definitions of the Garzanti monolingual Italian dictionary (Garzanti 1984) are discussed.

2. The general approach

The approach we adopted for extracting taxonomical information from dictionary definitions follows a two-stage strategy: during the first step, a general purpose Italian grammar, which has been specialized to handle dictionary language, provides an organized structure corresponding to an initial syntactic analysis for each definition; during the second step, a pattern-matching procedure is in charge of mapping lexical and/or structural patterns onto the syntactic description computed at the previous stage, with the result of deriving and making explicit the taxonomical information implicitly stored in definitions. This approach

* The work illustrated in this paper was carried out in the framework of the ACQUILEX-II (Esprit BRA n. 7315) project. The results are now being analysed and used in the framework of the EuroWordNet project (LE 4003).

was originally developed by Jensen and Binot (1987) for acquiring the semantic information necessary for the resolution of prepositional phrase attachment ambiguities.

Generally speaking, there are two main advantages in basing the knowledge acquisition procedure on parsed syntactic structures (Montemagni and Vanderwende 1992). First, it is possible to abstract away from most of the variations in the surface realization of the same definition pattern. In fact, although recurring defining formulae are systematically used in the language of dictionary definitions to express conceptual categories as well as semantic relations, these formulae undergo variations which can be better captured by means of patterns operating on syntactic structures than by means of patterns (typical of a text retrieval system) operating on the raw sequence of strings within the definition text. Secondly, the information extracted is expected to be more reliable. For instance, the relevant terms of the semantic relations detected by means of the structural patterns can be safely identified. In fact, the real extraction process often consists in identifying the relevant complements of the defining formulae and so accessing structural information yields more reliable results. Obviously, this could not easily be achieved in typical text retrieval systems operating on the raw sequence of strings.

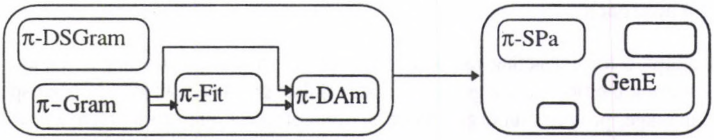
On the other hand, the main disadvantage usually attributed to approaches based on the syntactic analysis of definitions is that failures in the syntactic parsing process, which are unavoidable, affect somehow the final results of the extraction procedure (see, for instance, Ahlswede and Evens 1988).

It has often been argued that, for the acquisition of taxonomical information, pattern matching at the string level yields promising results (see Calzolari 1984, Chodorow et al. 1985). Nevertheless, genus extraction can benefit quite a lot from the two-stage approach: more detailed genus information can be extracted; in the case of complex coordinated genuses, it is possible to easily identify the heads of the coordinated terms; finally, "mixed" definitions (i.e. combining different kinds of semantic relations) can be recognised and appropriately dealt with.

With respect to the main drawback of operating on syntactic analyses rather than on raw text, we contend that this does not apply to our case, as we made our extraction procedure robust by specializing part of it for the treatment of unresolved incomplete syntactic parses. In this way, some information is extracted in any case regardless of parsing failures; in the worst case, the information is not very deep or detailed.

3. System Architecture

The acquisition of taxonomical information from dictionary definitions is carried out by AcqSys, the modular system sketched in the figure below:



The two-stage approach to lexical acquisition reflects itself in the general architecture of the system: there are components in charge of carrying out the first stage of the extraction procedure, i.e. computing a syntactic analysis for each dictionary definition, and others in charge of deriving the taxonomical information implicitly stored in it through pattern-matching procedures mapping lexico-structural patterns onto the output of the previous stage. π-Gram (a general purpose Italian Grammar, Montemagni 1995), π-Fit (a Fitting procedure, Jensen et al. 1983) and π-DAm (a Dictionary disAmbiguator, Montemagni 1992) are the components performing the first stage of analysis; taken together, they form the module in charge of the syntactic analysis of dictionary language, or π-DSGram, i.e. a Dictionary-Specific Grammar of Italian. GenE (i.e. the Genus Extractor) is instead the component in charge of carrying out the second stage of analysis, i.e. the extraction of taxonomical information. GenE is part of π-SPa, i.e. a Semantic

Parser, together with other components which operate on the output of the syntactic analysis stage to extract other kinds of lexico-semantic knowledge: for a description of other modules of π-SPa operating on the differentia part of the definition see Calzolari et al. (1993) and Montemagni (1994).

In the architecture above, there are components built specifically for the lexical acquisition task, as well as components which were initially designed independently of this specific task, but which appeared very suitable for it. This is the case of π-Gram which has been designed and developed as a general purpose grammar of Italian, and of π-Fit which is a general fitting procedure conceived to deal with parsing failures. In order to refine the syntactic analysis performed on the basis of general grammatical expertise, a small dictionary-specific component operating after the grammar, π-DAm, has been integrated into the system to tailor the output of the general purpose Italian grammar to the dictionary language peculiarities. With the same goal of tailoring the output of the general grammar to dictionary language, π-Fit has been specialized to deal with parsing failures typical of dictionary text. GenE has been developed for the acquisition task and thus represents a completely dictionary-specific component.

4. The extraction procedure

4.1 The syntactic analysis of dictionary definitions

This section focuses on the first stage of the extraction procedure, i.e. the syntactic analysis of dictionary definitions, which represents a crucial stage of the whole acquisition process, since it creates the data structures on which the further processing stage operates and thereby determines the quantity and the quality of the information that can be extracted. The analysis produced at this stage is provided by π-Gram and is subsequently reshaped and/or refined by π-Fit and π-DAm.

4.1.1 Parsing dictionary language with a general purpose grammar

The use of a general purpose grammar and parser to parse dictionary text represents a controversial choice, since dictionary-specific parsing tools are often preferred to general ones: for instance, a dictionary-specific pattern matching and parsing tool has been applied to the Spanish Vox dictionary (Ageno et al. 1991); special purpose grammars, developed by utilizing a general purpose parser, have been used with the Longman Dictionary of Contemporary English and the Dutch Van Dale dictionary (Vossen et al. 1989). In our opinion, there are several reasons for approaching Italian dictionaries such as Garzanti with a parser and a grammar which are both domain independent. First of all, unlike the Longman Dictionary of Contemporary English (Procter 1987), Italian dictionaries do not use a restricted vocabulary. Therefore, the scope of the vocabulary used in dictionary text is the same as that of unrestricted texts. The same happens at the syntactic level; the variety of phrasal constructions used within dictionary text is comparable to that of textual corpora. In fact, the regularity of the lexically and syntactically constrained language used within Italian dictionary definitions lies in the frequent occurrence of lexical and syntactic patterns conveying particular conceptual categories or semantic relations, rather than in a restricted vocabulary and limited range of syntactic constructions. Hence, the formulae used in dictionary language, however crucial to the extraction of semantic information, can be considered almost irrelevant from the point of view of parsing because the variety of lexical choices and syntactic constructions in which these formulae are manifested can be compared to that of textual corpora. This entails that parsing dictionary text poses the same range of problems a parser is faced with in analysing ordinary texts.

Several disadvantages are usually attributed to the use of a general purpose grammar for parsing dictionary text. First, at the end of the syntactic analysis performed with a general grammar, ambiguities still remain; yet, in the dictionary context, part of these ambiguities do not present themselves any longer as such. Second, a general grammar rejects constructions which are considered syntactically deviant when found in ordinary text but which typically occur within dictionaries. A further more general problem is concerned with the fact that definitions rarely form a complete sentence: depending on the part of speech of the word being defined, they are formulated as a noun phrase, a verbal phrase, an adjectival phrase or a relative clause etc.

Two different - somehow contrasting - conclusions can thus be drawn from these short remarks. On the one hand, the syntactic analysis of dictionary language requires the same kind of knowledge in terms of vocabulary and constructions to be covered as the analysis of unrestricted text. From this, it naturally follows that a general text parser and grammar are very well suited to parse dictionary definitions. On the other hand, there are aspects of dictionary language differing from the language of unrestricted text which may require ad hoc parsing strategies or simply additional knowledge. These dictionary-specific peculiarities would support the use of a dictionary-specific grammar and/or parser. We thus decided to use a general purpose Italian grammar and parser and to revise its output during a dictionary-specific post-processing stage.

Inherent features of the parser behind π-Gram as well as its parsing strategy already provide an answer to some of the specific parsing problems posed by dictionary language. Dictionary definitions rarely form complete sentences. Yet, their syntactic form is largely predictable from the part of speech of the word being defined: nouns are typically defined by a noun phrase, verbs by a verb phrase, adjectives by an adjectival phrase or a relative clause. The parsing system provides a definitive answer to this problem since, by setting a switch, it is possible to force a parse of any desired syntactic category (e.g. NP, VP); this switch indicates whether the input string should be parsed, say, as a nominal or as a verbal phrase. Thus, depending on the part of speech of the definiendum, the switch can be set accordingly.

The switch forcing the syntactic category of the top node of the parse tree to be produced by the grammar does not solve all problems concerned with dictionary parsing. In fact, as pointed out above, the dictionary text does not always form even a complete phrase but often only fragments of phrases. Consider the case of definitions formulated as condensed fragments of real texts, with elided elements which make the definition syntactically ill-formed from a general perspective and interpretable only by reference to a wider context. This occurs, for instance, when obligatory complements of verbs are omitted, therefore resulting in ellipsis: see *prodigare* 'be lavish' (sense 1b) whose definition reads *dare con larghezza* 'to give generously' where the object of *dare*, which in principle should be obligatorily specified, has been omitted. While a general grammar would normally reject these constructions as ill-formed, a dictionary specific grammar has to parse them as typical occurrences of dictionary language. This observation holds in general terms but does not apply to the case of π-Gram in which the relaxed approach to parsing (Jensen 1988) does not expect the text to always conform to the subcategorization requirements of verbs.

Yet, dictionary-specific constructions cannot always be handled by π-Gram; there are cases in which the analysis produced on the basis of general grammatical expertise needs to be revised, namely reshaped or disambiguated. In the following section the dictionary-specific revision task is illustrated.

4.1.2 Reshaping and disambiguating the syntactic analysis of dictionary language

π-Fit and π-DAm are two different post-processors operating on the output of π-Gram carrying out two different dictionary-specific revision tasks: π-Fit is in charge of handling incomplete parses, in particular those due to dictionary-specific constructions not occurring in free text; π-DAm operates on complete analyses to rule out ambiguities in modifier attachment and functional role assignment which are not applicable in the dictionary context.

Consider first the case of deviant contructions whose analysis by π-Gram results in a parsing failure. This is exemplified by verb definitions formulated as a verb phrase where instead of the prepositional phrase following the genus verb only the preposition is specified, i.e. where the preposition complement has been omitted being somehow recoverable from a wider context. The Garzanti definition for the verb *abbandonare* 'abandon' (sense 2) in its transitive reading, which reads *rinunziare a, desistere* *da un'impresa* 'to turn down, to give up doing something', is an instance of the case at hand. The analyses for this definition performed by the general grammar and then revised by π-Fit are represented below:

```
Before:                                            After:
XXXX1 |-VP1*----VERB1*-----------"rinunziare"      VP1 |-VP1-----VERB1*---------"rinunziare"
      |-PREP1---PREP2*-----------"a"                   .         PREP2----------"a"
      |-PUNC1--------------------","                   |-CONJ1*-----------------","
      |-NP1-----VERB2*-----------"desistere"           |-VP2-----VERB2*---------"desistere"
      |         PP1-----PREP3----"da"                  |         PP1---PREP3----"da"
      |                 DETE1----"un'"                 |               DETE1----"un'"
      |                 NOUN1*---"impresa"             |               NOUN1*---"impresa"
      |-PUNC2--------------------"."                   |-PUNC2------------------"."
```

As shown in the "before" parse, π-Gram is unable to produce a VP node covering the whole input string given that the sequence Verb-Prep, not followed by any preposition complement, does not freely occur within ordinary texts. The XXXX label at the top node of this parse tree indicates that the parse is incomplete. π-Fit reshapes the analysis by restoring it as regular input on the basis of specialized dictionary use. The result of this reshaping process is illustrated in the "after" parse, where the XXXX label has been replaced by the proper label VP and the constituents under it have been reorganised and restructured: the definition presents itself as a conjoined VP where the first conjunct is anomalous since it consists of a verb followed by a preposition only.

Similar problems arise, for instance, with noun definitions consisting of a PP-NP construction, as in the definition for *nettare* 'nectar' (sense 2) which reads *nella mitologia classica, la bevanda degli dei* 'within classical mythology, the drink of gods': this construction is not an acceptable nominal phrase in general Italian (i.e. in text corpora) since a PP can only post-modify a noun phrase. In the case of definition text, π-Fit has been instructed to reshape the analysis produced on the basis of general grammatical expertise inserting the prepositional phrase among the regular pre-modifiers of the NP, thus restoring a "legal" NP top node, although this legality is restricted to the dictionary language.

The examples above have been handled by π-Fit which has been tailored to deal properly with ill-formed but, in the context of dictionary language, commonly occurring constructions. In this case, knowledge of dictionary peculiarities has been used to resolve the initial partial parse and to convert it into a complete and successful analysis. The strategy we adopted to cope with dictionary-specific parsing failures highlights the fact that the distinction between ill- and well-formed input is not always clear, even in regular text; the fitting procedure allows what appears to be ill-formed, or just irregular, in general text to become the norm with respect to some specialized use. Note, however, that not all incomplete parses can be so easily restructured. Parsing failures due to more general reasons (e.g. to gaps in the lexical as well as phrasal construction knowledge of the system or to really ill-formed input) are still handled by π-Fit, but do not necessarily result into a complete and successful analysis as in the case of failures due to the peculiarities of the dictionary input. With general parsing failures, a reasonably approximate but incomplete structure is assigned to the input. Such a rough parse is still used as input for further processing stages and for the extraction procedure itself.

Complete syntactic analyses, either directly provided by π-Gram or successfully reshaped by π-Fit, are further refined by π-DAm with respect to ambiguities in modifier attachment or functional role assignment. As an example of the refinement to reduce attachment ambiguity, consider the Garzanti definition for the noun *caduta* 'fall' (sense 1), which reads *atto, effetto del cadere* 'act, effect of falling'. The general NP parse and the dictionary-specific NP parse for this definition are contrasted below:

```
Before:                                               After:
NP1 |----NP2-----NOUN1*-----------"atto"              NP1 |----NP2-----NOUN1*---"atto"
    |----CONJ1*-------------------","                     |----CONJ1*-----------","
    |----NP3-----NOUN2*-----------"effetto"               |----NP3-----NOUN2*---"effetto"
    |    ?       PP1-----PREP1----"del"                   |----PP1-----PREP1----"del"
    |                    VERB1*---"cadere"                |            VERB1*---"cadere"
    |----PUNC1--------------------"."                     |----PUNC1------------"."
```

In the "before" parse, the general grammar has applied the default attachment strategy. PP1 (*del cadere*) is attached to the closest available head, *effetto*, and the alternative attachment site (i.e. the coordinated genuses) is indicated by a question mark. The "after" parse shows the analysis after the revision performed by π-DAm: PP1 now modifies the coordinated nominal phrase covering the coordinated genus terms. This refinement is made when a prepositional phrase or an infinitival clause post-modifies coordinated head nouns belonging to a closed class that are the top nodes of the syntactic analysis. This is a typical pattern used in the definition of deverbal nouns where the prepositional phrase or infinitival clause indicate which verb the definiendum is derived from (*cadere* 'to fall' in the case at hand).

π-DAm, whenever possible, can also solve ambiguous assignments of functional roles. For instance, in Italian functional role ambiguity typically occurs in relative clauses for which agreement in number and person between subject and verb does not determine which NP is the subject because all NP candidates agree with the verb. Consider the Garzanti definition for *dignità* 'dignitary' (sense 3), which reads *persona che occupa un'alta carica* 'person who holds a high position'. In this definition, which conforms to the structure of definitions of "generic agents", the general grammar cannot unambiguously assign the functional roles of subject and object on the basis of morpho-syntactic information; yet, such an ambiguity can be resolved easily and with certainty if one considers that the definition conforms to the general pattern of "generic agents" definitions, for which the implicit convention holds that the antecedent of the relative clause (which is also the definition head), and thus the relative pronoun, is always the subject. On this basis, π-DAm re-assigns the subject and object roles to *persona* and *carica* respectively.

4.2 The semantic analysis of dictionary definitions

The typology of semantic relations which can be acquired from the genus part of dictionary definitions is wide. It ranges from hyperonymical (IS_A) and synonymical (SYN) relations to more specific relations which vary depending on the part of speech of the definiendum: TYPE_OF, SET_OF, ELEM_OF, PART_OF, or derivative relations such as VERB_TO_NOUN, ADJ_TO_NOUN, or AGENT_OF in the case of noun definitions; ANT(onym), CAUS(ative), INCH(oative) in the case of verbs. A complete typology of semantic relations which can be extracted from the genus part of definitions of Italian dictionaries can be found in Hagman (1991) and Montemagni (1995). Here we will focus on the general strategy adopted for carrying out the semantic analyis stage.

The recognition of these semantic relations and the subsequent extraction of their associated value are based on structural patterns, to be mapped onto the parsed definition text, which are very often combined with lexical conditions. Thus, two different kinds of information are to be taken into account for the extraction of taxonomical information from dictionary definitions: syntactic structures on the one hand, and individual lexical items on the other hand. Depending on the kind of semantic relation, the two can be variously combined.

There are semantic relations whose recognition and extraction are based only on the syntactic structure associated with the definition; this holds for relations such as IS_A and SYN whose value is directly the syntactic head of the top phrase covering the whole definition text. An IS_A relation holds when the definition head is restricted in its meaning by pre- and/or post-modifiers; when no modifiers are detected, then a SYN relation is identified. This is shown in the examples below:

```
ABBOCCARE$0_1 'bite'
    (POS VERB)
    (DEF "afferrare con la bocca.") 'to grasp
    with the mouth'
    ISA (base afferrare) 'grasp'

ACQUA$0_1 'water'
    (POS NOUN)
    (DEF "liquido trasparente, incoloro, inodoro
    e insaporo, costituito di ossigeno e
    idrogeno, indispensabile alla vita animale e
    vegetale.") 'trasparent, colourless,
    odourless, and tasteless liquid, composed of
    oxygen and hydrogen, indispensable for
    animal and plant life'
    ISA (base liquido) 'liquid'
```

```
ABBONDARE$0_2 'abound'
    (POS VERB)
    (DEF "eccedere.") 'exceed'
    SYN (base eccedere) 'exceed'

ABBAGLIO$0_0 'blunder'
    (POS NOUN)
    (DEF "errore, svista.") 'mistake, slip'
    SYN (base errore) 'mistake'
    SYN (base svista) 'slip'
```

On the other hand, there are semantic relations expressed through more complex constructions whose identification is based on combinations of both syntactic and lexical criteria. The recognition of the syntactic structure of a noun post-modified by a prepositional phrase headed by the preposition *di* 'of' at the top node level is the first step towards the detection of a wide class of noun relations such as TYPE_OF, SET_OF, ELEMENT_OF, PART_OF and VERB_TO_NOUN (i.e. deverbals). The kind of relation is afterwards identified on the basis of lexical tests at the head of the top noun phrase. For instance, when the head of the top noun phrase belongs to the list [*classe* 'class', *complesso* 'whole', *insieme* 'set', *gruppo* 'group', etc.] then the definiendum stands in a SET_OF relation with respect to the entity designated by the *di* complement, as shown in the examples below:

```
ARMATA$0_2 'fleet'
     (POS NOUN)
     (DEF "complesso delle navi da guerra di una
     nazione.") 'the whole of the warships of a
     nation'
     SETOF (base nave) 'ship'
```

```
ABBAZIA$0_3 'abbey'
     (POS NOUN)
     (DEF "l'insieme degli edifici di una
     comunità monastica.") 'the set of buildings
     of a monastic community'
     SETOF (base edificio) 'building'
```

With verbs, it may be the case that an opposition - or ANT(onymy) - relation is used to define a given verb. To recognize it, the structural pattern used to identify IS_A and SYN needs to be combined with a lexical condition, i.e. that the definition head must be preceded by the adverbial *non* 'not', as exemplified with the entry of *fallire* below. The periphrastic causative construction with *fare* followed by an infinitive verb (and possibly other complements) is another example of lexico-structural pattern occurring in verb definitions. This construction indicates that the verb being defined is the causative counterpart of the infinitive verb which follows *far(e)*. This is to say that, in the example below, *abbeverare* is related via causation to the action referred to by *bere* 'drink'.

```
FALLIRE$0_1 'fail'
     (POS VERB)
     (DEF "non riuscire.") 'to not succeed'
     ANT (base riuscire) 'succeed'
```

```
ABBEVERARE$0_0 'water'
     (POS VERB)
     (DEF "far bere il bestiame.") 'to cause
     animals to drink'
     ISACAUS (base fare) 'cause'
               CAUSAT (base bere) 'drink'
```

Finally, consider the case of "mixed" definitions, combining different kinds of semantic relations, where the same noun or verb is described through a combination of defining strategies: e.g. synonyms and hyperonyms (*affanno*), synonyms and antonyms (*abbandonarsi*).

```
AFFANNO$0_2 'anxiety'
     (POS NOUN)
     (DEF "sofferenza morale, angoscia.") 'moral
     suffering, anguish'
     ISA (base sofferenza) 'suffering'
     SYN (base angoscia) 'anguish'
```

```
ABBANDONARSI$0_3 'give oneself up'
     (POS VERB)
     (DEF "cedere, non resistere.") 'to
     surrender, to not resist'
     SYN (base cedere) 'surrender'
     ANT (base resistere) 'resist'
```

5. Performance of AcqSys

The performance of AcqSys is illustrated in three different steps: first, the performance of the π-DSGram component on the whole set of noun and verb definitions of the Garzanti dictionary is illustrated; second, the results of the semantic analysis process are discussed; third, the adequacy of the two-stage approach for the acquisition of taxonomical information from noun and verb definitions is assessed by comparing the results obtained through the different analysis stages.

AcqSys has been applied to 59,699 word senses (corresponding to 30,181 lemmata), of which 14,091 were verb senses and 45,608 were noun ones.

5.1 Performance of π-SDGram

The table below provides a brief account of the parsing performances of π-Gram before the intervention of the dictionary-specific modules. For verb and noun entries, the number of parsed definitions, the average number of words per definition and the actual parsing performance of π-Gram are specified.

GARZANTI DICTIONARY	n° of parsed sentences	average n° of words	n° of parses	%
Verb definitions	14,091	5	0	12
			1	81
			2	5
			>2	2
Noun definitions	45,608	9	0	20
			1	65
			2	11
			>2	4

The average number of words per sentence highlights the minor degree of complexity of verb definitions with respect to noun ones, the latter being almost twice longer than the former. This fact has consequences at the level of parsing performance where better results are observed in the case of verbs: whereas π-Gram failed to provide complete parses for a fifth (i.e. 20%) of the definitions in case of nouns, this percentage decreases significantly to 12% in the case of verb definitions. A unique parse has been provided in 65% of the cases with noun definitions, whereas in more than 80% of the cases with verb definitions. Multiple analyses resulted in a minority of cases, ranging from 15% in the case of noun definitions to 7% of the verb ones. Note that the percentage of sentences with a number of parses greater than two is rather low.

The results obtained with only general grammatical expertise have considerably improved during the reshaping and refinement stage. The table below refers to the improvement due to π-Fit: it compares the percentages of parsing failures obtained on the basis of π-Gram and of π-DSGram respectively. In other words, it gives a measure of dictionary-specific parsing failures and thus of the improvement which follows from removing them by restoring the dictionary-specific but generally ill-formed constructions to the status of legal constructions.

GARZANTI DICTIONARY	π-Gram failures	π-DSGram failures	improvement
Verb definitions	12%	8%	-4%
Noun definitions	20%	15%	-5%

An improvement of 4% and 5% in the parsing performance of verb and noun definitions was achieved respectively: the parsing failures were reduced to 8% and 15% respectively. With respect to π-DAm, it can only be pointed out that all (or at least most) unique parses containing ambiguous attachments and/or assignments which could be resolved on the basis of dictionary-language peculiarities have been successfully disambiguated, as it can be inferred from the results of the semantic parsing process.

The statistical data illustrated so far provide support to our decision of using, at the syntactic analysis level, a general purpose grammar whose output is then revised, disambiguated or reshaped, on the basis of peculiarities of dictionary language. The performances of π-Gram on dictionary data were already adequate. Nevertheless, we chose to add dictionary-specific modules that would modify and improve the parses based on the peculiarities of dictionary language. This minor addition to the general parsing system architecture led to a marked improvement of the parsing results.

5.2 Performance of GenE

The typology of semantic relations acquired by GenE from noun and verb definitions respectively is reported in the tables overleaf:

	IS_A	SYN	TYPE OF	SET OF	PART OF	ELEM OF	VERB TO NOUN	ADJ TO NOUN	AGENT OF	TOTAL
n.	29,385	4,330	502	1,570	1,061	446	3,284	972	2,136	43,686
%	67.26	9.91	1.15	3.59	2.43	1.02	7.52	2.23	4.89	100.00

	IS_A	SYN	ANTONYM	CAUSATIVE fare+inf.	CAUSATIVE rendere+adj.	INCHOATIVE	TOTAL
n.	9,717	3,658	117	415	462	187	14,556
%	66.76	25.13	0.80	2.85	3.18	1.28	100.00

In this context, however, the results of GenE are considered globally (for a detailed discussion of GenE results, see Montemagni 1995). Our main concern is whether the semantic parser produced some results: 43,686 semantic relations have been extracted from noun definitions, and 14,556 from verb ones. These data refer to the output of the semantic parser without any interactive revision; hence, final results of the semantic analysis stage, i.e. interactively revised and disambiguated, may slightly differ from those reported above. We do not expect, however, significant variations: in fact, from a validation experiment carried out manually on a sample of 300 (randomly selected) formalised noun entries, it emerged that the correct semantic analysis has been identified in 96.5% of the cases; a similar experiment, carried out on a sample of 200 verb formalised entries, stated that with verbs the correct interpretation has been achieved in 98% of the cases. Note that in both tests the samples also included semantic analyses derived from incomplete syntactic analyses or "fitted" parses.

5.3 Assessing the adequacy of the two-stage approach

Our main concern in adopting a two-stage approach was whether it would have somehow affected the overall performance of the system, i.e. whether parsing failures at the syntactic analysis stage would have prevented the semantic parser from getting some results. We are now in a position to answer this question. This kind of evaluation is possible since, in principle, all definitions contain some genus information to be extracted; thus, what should have been extracted can be compared with what has actually been extracted. This comparison is the basis for evaluating the success of the acquisition strategy which has been adopted.

The GenE procedure has been applied to all definitions, those successfully parsed by π-DSGram as well as those without a successful syntactic analysis. In the latter case, the information extracted has been marked as derived from an incomplete structure or "fitted" parse. In this way, the results obtained from incomplete structures can be easily identified and interactively revised. By allowing the genus extraction procedure to apply to incomplete syntactic analyses as well, the coverage of the semantic analysis stage is not subordinated to the coverage of the syntactic analysis. This was one of the main drawbacks opposed in the literature to the choice of operating on syntactic structures rather than on the raw text for extracting lexico-semantic information from on-line dictionaries. Such a robust procedure, overcoming the variability of the parsing performances at the syntactic level, gave good results.

Consider as the starting point the number of π-DSGram parsing failures. GenE provided some results in 96% of the cases with verbs and in 92% with nouns: i.e. only 4% (verbs) and 8% (nouns) of π-DSGram failures translated themselves in GenE failures. This number, when compared with the total number of verb and noun definitions which have been dealt with, is very small. In fact, it appeared that GenE could not provide a semantic interpretation in 0.33% of the cases with verbs and in 1.2% with nouns. Looked at from the point of view of the results which have been obtained, some results have been extracted in 99.67% of the verb definitions which have been taken into account, and in 98.8% in the case of nouns. These figures, summarised in the table below, show that the adoption of a two stage strategy for the acquisition of lexico-semantic information from dictionaries affects in no way the robustness of the extraction procedure which provides accurate and reliable analyses in about 99% of the cases.

	π-DSGram		GenE failures			Successful analyses
	failures		n°	% wrt π-DSGram	% wrt to definitions	
Verbs	1,140	8%	47	4%	0.33%	99.67%
Nouns	6,886	15%	550	8%	1.20%	98.80%

6. Concluding remarks

The main features of the lexical acquisition system illustrated in this paper can be summarised as follows:

a) two-stage approach to the acquisition of information from dictionary definitions: first, dictionary definitions are syntactically parsed by π-DSGram; the semantic analysis, i.e. the acquisition of taxonomical information, is then carried out by another component - GenE - operating on the output of the previous stage;

b) the syntactic parsing of definitions has been performed by a general purpose Italian grammar - π-Gram - which has been tailored to deal with dictionary language by means of *ad hoc* components in charge of i) handling and regularising otherwise ill-formed input (π-Fit), and ii) ruling out ambiguous constructions (π-Dam).

We showed that the two-stage approach guarantees the quality and reliability of the semantic information extracted. On the other hand, it does not significantly affect the final results of the semantic parser which have been extracted from both complete and incomplete (but "fitted") syntactic analyses.

References

Ageno A., Cardoze S., Castellon I., Marti M.A., G. Rigau, Rodriguez H., Taule M., Verdejo M.F., 1991, *An Environment for Management and Extraction of Taxonomies from On-Line Dictionaries*, ACQUILEX Esprit BRA-3030, WP n. 20.

Ahlswede T., Evens M., 1988, 'Parsing vs. Text Processing in the Analysis of Dictionary Definitions', in *Proceedings of the 26th Annual Meeting of the ACL*, pp. 217-224.

Calzolari N., 1984, 'Detecting Patterns in a Lexical Database', in *Proceedings of the 10th International Conference on Computational Linguistics*, Stanford, California, pp. 170-173.

Calzolari N., Hagman J., Marinai E., Montemagni S., Spanu A., Zampolli A., 1993, 'Encoding Lexicographic Definitions as Typed Feature Structures', in F. Beckmann, G. Heyer, (eds.), *Theorie und Praxis des Lexicons*, Walter de Gruyter, Berlin, pp. 274-315.

Chodorow M.S., Byrd R.J., Heidorn G.E., 1985, 'Extracting semantic hierarchies from a large on-line dictioanry', in *Proceedings of the 23rd Annual Meeting of the ACL*, pp. 299-304.

Garzanti, 1984, *Il Nuovo Dizionario Italiano Garzanti*, Garzanti, Milano.

Hagman J., 1991, *Common and Odd Relations in MRD Definitions and their Treatment in Taxonomy Building*, ACQUILEX Esprit BRA-3030, Working Paper n. 44.

Jensen K., 1988, 'Issues in Parsing', in A. Blaser (ed.), *Natural Language at the Computer*, Springer Verlag, Berlin, pp.65-83.

Jensen K., Binot J.L., 1987, 'Disambiguating Prepositional Phrase Attachments by Using On-Line Dictionary Definitions', *Computational Linguistics*, vol. 13, n. 3-4, pp. 251-260.

Jensen K., Heidorn G.E., Miller L.A., Ravin Y., 1983, 'Parse Fitting and Prose Fitting: Getting a hold on ill-formedness', *American Journal of Computational Linguistics*, 9, n. 3-4, pp. 123-136.

Montemagni S., 1992, 'Tailoring a Broad Coverage Grammar for the Analysis of Dictionary Definitions', in *Proceedings of the Fifth Euralex International Conference*, Tampere, Finland, pp. 265-275.

Montemagni S., Vanderwende L., 1992, 'Structural Patterns versus String Patterns for Extracting Semantic Information from Dictionaries', in *Proceedings of COLING-92*, Nantes, France, pp. 546-552.

Montemagni S., 1994, *Extracting Typical Subjects and Objects of Verbs from Mono- and Bi-lingual Dictionaries*, ACQUILEX-II Esprit BRA-7315, Working Paper n. 42.

Montemagni S., 1995, *Subject and Object in Italian Sentence Processing*, PhD Dissertation, University of Manchester, Institute of Science and Technology, UK.

Procter P., 1987, *Longman Dictionary of Contemporary English*, Longman, London.

Vossen P., Meijs W., den Broeder M., 1989, 'Meaning and structure in dictionary definitions', in B. Boguraev, T. Briscoe, (eds.), *Computational Lexicography for Natural Language Processing*, Longman, pp. 171-192.

Construction of the Korean electronic lexical system DECO

JEE-SUN NAM

ABSTRACT

We here discuss the method and the principles we have adopted in constructing the Korean electronic lexical system DECO. Given that existing editorial dictionaries are not reliable for this purpose, the use of a large corpus is required. However, even though the scale of the corpus is considerably extended, we can never ensure that all basic lexical items occur. Therefore, a combinatorial linguistic method based upon explicitly defined lexical categories is necessary to obtain all morphological sets related to a given basic form. The results of exhaustive description will be represented by finite state automata in our electronic lexicons.

I. EDITORIAL DICTIONARIES AND A LARGE CORPUS

Given that most printed texts are now available in electronic forms, accumulation of this type of information is considerable. Hence, the development and refinement of natural language processing (NLP) systems are incessantly required in order to archive these documents and offer requested information in a better way.

In the implementation of all kinds of NLP systems, the construction of electronic lexicons is elementary and indispensable: it is necessary to build up reliable electronic lexicons on the basis of coherent and explicit principles.

The methods that have been adopted so far for the construction of Korean on-line dictionaries can be summarized to the following two procedures:

1. Use of editorial dictionaries

One uses existing editorial dictionaries, that contain some morphological and grammatical information such as indication of parts of speech or derivational relations among lexical entries. However, existing dictionaries that, whether in printed forms or in electronic forms, are a priori conceived for human users, are hardly disposable for this purpose. Thus, there are some problems, especially such as the followings:

1.1. Assignment of parts of speech

The assignment of parts of speech to lexical entries is not done in an explicit and coherent way. For example, as no formal criterion is given to distinguish verbs from adjectives, some items are considered as verbs in one dictionary, and as adjectives in another. Likewise, a great number of adjective roots are treated as nouns, whereas they do not have any lexical autonomy. Unless we do examine these problems, there would be no meaning in applying detailed syntactic rules based on high level grammars to the sets named *verbs*, *adjectives* or *nouns*.

1.2. Information about derivational relations

Information about derivational relations among lexical entries is not integrated in a systematic and exhaustive way. Thus, lists of verbs derived from adjectives by means of some suffixes (i.e. *Adj-Sfx = Verb*) are far from being complete. Affixed nouns and compound nouns are also selected without any coherency. This aspect is much less problematic for human users than for machines, because the former can *guess* the lack of information by reasoning by analogy. In fact, derivational and compositional information should be either all dropped out from a basic lexicon to be completed in a systematic way or all presented.

1.3. Encyclopedic entries

For practical purposes, the editorial dictionaries of Korean are not reliable, since they contain not only lexical entries (language dictionaries), but also encyclopedic entries such as proper nouns. For example, people's names, geographic places, historical events, artistic works, etc. are integrated in dictionaries as well as lexical items. Moreover, the number of proper nouns is incessantly increasing and it is difficult to establish their repertory. It is necessary to separate these two types of dictionaries from each other, so that we can complete them gradually.

2. Use of a large corpus

One uses a large corpus to establish on-line dictionaries. We can measure the frequency of lexical items, and then we can handle apart a great number of entries registered in editorial dictionaries that do not (or rarely) appear in texts. This point is not without importance in the case of Korean, since the number of archaic expressions is considerable in existing editorial dictionaries: this advantage is not mere nothing.

This procedure also allows one to process derived and compound forms more easily than using editorial dictionaries. Let us consider an example. The formation of some types of compound nouns is very productive and it would be very long to construct their complete list. The following compound nouns are of '*NounNoun*' type, one of the most productive types:

빨강색	*BbalgangSaig*	[red color]
영어책	*YengeChaig*	[English book]
칼자루	*KalJalu*	[knife handle]

We can obtain an interesting repertory of this type of nouns by using a large corpus. The fact that they usually appear without any typographical blank in them and that they are usually followed by a grammatical **postposition** (i.e. a grammatical particle such as nominative, genitive, or accusative postposition, etc. The functions of postpositions correspond to those of **prepositions** in English such as *of, to, for, by,* etc.) makes their recognition easy: if we omit the right part of a string, which is identified as one of the postpositions, the left part might be a noun, whether a simple noun or a compound one.

Notice that the smallest units in automatic processing of texts can be words, morphemes or still something else. For example, in English or in French, this basic unit is usually a string cut up by two separators (e.g. blank, apostrophe or comma): it can be then named a **word**. Thus, when we observe sentences such as the following, we identify 5 units in each case:

John is in this cafe
Jean est dans ce café

In the case of Korean, the units delimited by separators are not on the same level as in the above cases: most grammatical markers are typographically attached to verbs (such as tense suffixes, modal suffixes, and so on..), and to nouns (such as nominative postpositions, genitive postpositions, and so on..). Thus, in the following sentence, we identify 4 units, but the nouns corresponding to *John* and *cafe* in English are suffixed with grammatical markers, **nominative** and **locative** respectively:

존은	그	카페에	있다	
Jon-eun	*geu*	*kapei-ei*	*iss-da*	
John-**nmtf**	this	cafe-**loc**	be-St	[= John is in this cafe]

Then, it does not make any sense to consider that strings cut up by two separators are the smallest units in Korean. If one imagines the number of *'Noun-Postp'* strings that can be obtained by the combination of more than ten thousand simple nouns and a thousand sets of postpositions (notice that several postpositions can be linked to a noun and that these combinations can be described in a local grammar), one will easily understand why these strings must not be taken as basic units. Therefore, the recognition of **nouns** in these strings is required priori to automatic analysis of texts.

Likewise, we can here use this procedure to list up "compound" nouns: recognition of postposition(s) will not be too complicated, since their list is much smaller than that of nouns; and then, elimination of these parts can provide a list of compound nouns.

However, the situation is not so simple. This method, i.e. constructing lexicons of nouns (not only simple nouns, but also compound ones) by means of recognizing postposition(s) and deleting them from strings requires considerable refinement, for the following reasons:

2.1. Absence of postposition

All nouns are not necessarily followed by (a) postposition(s). Here are two cases:

2.1.1. Dropping of postpositions

Postpositions can be dropped out in some contexts: if those in noun strings in the above example (i.e. '*John*-**nmtf**' et '*cafe*-**loc**') can hardly be omitted, they can easily disappear in the following sentence. Consider:

선생님	학교	가셨니 ?	
sensaingnim	*haggyo*	*gasyessni ?*	
teacher	school	went ?	[= Did the teacher go to school ?]

However, describing these conditions and predicting the dropping of postpositions are not easy. Moreover, in the above case, it should be difficult to distinguish **nouns** from **adverbs** without syntactic analyses, since adverbs are usually not suffixed with postpositions:

선생님	방금	학교	가셨니 ?	
sensaingnim	*banggeum*	*haggyo*	*gasyessni ?*	
teacher	**a while ago**	school	went?	[=Did teacher go to school a while ago?]

2.1.2.Compound nouns

Nouns that constitute compound forms can appear separate from each other (i.e. with blanks between them). Then, postpositions will only be found at the end of the last noun of the compound sequence. In the following sentence, the compound sequence is '*jayu seigei* [liberty world]':

그들은	자유	세계를	구현하였다	
geuteul-eun	*jayu*	*seigyei-leul*	*guhyenhayessda*	
they	**liberty**	**world-Acc**	achieved	[=They achieved liberty world]

Sometimes, spaces are obligatorily required inside compound nouns. The following example illustrates a compound sequence composed of 8 nouns:

자연	보호	운동	추진	위원회	결성	합의안	채택
jayen	*boho*	*undong*	*chujin*	*wiwenhoi*	*gyelseng*	*habeuian*	*chaitaig*
nature	protection	movement	driving	committee	organization	proposition	acceptance

[**Acceptance** of the **proposition** of the **organization** of the **driving committee** of **Nature protection movement**]

Notice that we can link some of them as in *'NN N NN N N N'* or *'NNN NN NN N'*, but we can not write *'NNNNNNNN'* (symbol * indicates 'unacceptable sequence'):

*자연보호운동추진위원회결성합의안채택
**jayenbohoundongchujinwiwenhoigyelsenghabeuianchaitaig*
NatureProtectionMovementDrivingCommitteeOrganisationPropositionAcceptance

Therefore, recognizing nouns by eliminating (a) postposition(s) is no more a reliable method in this case, because we only observe (a) postposition(s) at the end of the eighth noun of this compound sequence:

국회는 자연 보호 운동 추진 위원회 결성 합의안 채택을 서둘렀다
gughoi-neun ***jayen boho undong chujin wiwenhoi gyelseng habeuian chaitaig-eul[Acc]*** *sedulessda*
[The National Assembly hastened [Acceptance of the proposition of the organization of the driving committee of Nature protection movement]]

2.2. Homograph

There are many cases where postpositions and the final morphemes of nouns are homographs. Let us consider an example:

우리가	무허가	주택가	근처를	배회할때,...
uliga	*muhega*	*jutaigga*	*geuncheu-leul*	*baihoihalddai,...*
We-**nmtf**	no-per**mit**	house-**area**	around-Acc	loiter-when

[When we are loitering around no-permit housing area, ...]

In this sentence, the first occurrence of '*ga*' is a nominative *postpostion* (i.e. **Noun-*nmtf***), whereas the second and the third ones are not: the second one is a part of the *noun* '*hega*' which a prefix '*mu*' is attached to (i.e. **Pfx-*Noun***); the third '*ga*' is a *suffix* which is attached to a noun '*jutaig*' (i.e. **Noun-*Sfx***). In other terms, only the first noun is linked to a postposition '*ga*'. Therefore, it would not be correct to automatically consider every final '*ga*' as a nominative postposition.

Here, we come across an important problem. Obviously, in order to make an appropriate analysis of the sentence above, we need a lexicon of nouns containing all these items, that is, not only simple nouns, but also affixed and compound nouns. However, let us recall that the lists of affixed and compound nouns we can obtain from editorial dictionaries are far from being complete and made up without any explicit principles. Nevertheless, the use of a large corpus to build these lists is not suitable either: we never can enumerate all sets of affixed and compound forms by using this procedure. Then, more refined linguistic studies about the mechanism of derivational relations among lexical items, based upon formal and coherent principles are required to build up a reliable on-line dictionary. Let us emphasize that linguistic descriptions can not easily be reduced to powerful general **syntactic rules**. We here have mentioned only some problems concerning *noun* sequences, but it is certain that one will come across such problems in other cases.

In the next paragraphs, we will present the method we have adopted and the principles of construction of the Korean electronic lexical system DECO.

II. THE KOREAN ELECTRONIC LEXICAL SYSTEM *DECO*

1. Lexicons of simple items *DECOS*, affixed items *DECOA* and compound items *DECOC*

The lexical system DECO is constructed not only by using existing editorial dictionaries and a large Korean corpus, but also a combinatorial method based upon explicitly defined lexical categories.

First of all, all simple items are separated from complex forms on the basis of syntactic criteria. Thus, '여기자 *yegija* [woman journalist]' is a complex form, i.e. '**pfx**(*ye*)-**noun**(*gija*)', whereas '여자 *yeja* [woman]' is a simple noun, because, even though it also contains the initial morpheme *ye*, the other part *ja* is not an autonomous unit. Diachronic and semantic analogies are not considered, but syntactic properties are taken as classifying criteria.

We have classified all simple items in 5 types of parts of speech: *Nouns*, *Adjectives*, *Verbs*, *Adverbs*, and *Functional Units*. They are encoded as NS, ADJS, VS, ADVS, and FUS where S stands for *simple*. Some syntactic and morphological information is integrated in the form of codes such as **PRED1** that indicates 'nouns that can be accompanied by 하다 *Hada* to form a sequence equal to a transitive verb' or **SM** that means 'adjectives the ending form of which is 스럽다 *Seulebda*. Each category itself is divided into sub-categories. These simple items constitute the lexicon DECOS (Korean electronic dictionary of simple items). The number of entries in the current version [DECOS-V01] is shown in the following table <figure 1>:

Nouns	*Adjectives*	*Verbs*	*Adverbs*	*Functional Units*	Total
15 000	5 300	7 500	7 000	200	35 000

< figure 1- number of the entries of DECOS-V01 >

Here are the first entries of the lexicon of simple nouns [DECOS-NS / V01] <figure 2> and those of the lexicon of simple adjectives [DECOS-ADJS / V01] <figure 3>:

가 NS.
가간 NS.
가감 NS. /PRED1/PRED3
가객 NS. /HUM
가게 NS. /*PREDHA
가격 NS.
가결 NS. /PRED1/PRED3
가경 NS.
가계 NS.
가공 NS. /PRED1/PRED3
가관 NS.
가교 NS. /PRED2
가구 NS. /*PREDHA
가군 NS. /HUM
가금 NS. /ANM
가난 NS. /*ADJH
가내 NS.
가늠 NS. /PRED1
가다랭이 NS.
가닥 NS.
가대인 NS. /HUM
가도 NS.
가동 NS. /PRED2
가두 NS.
가락 NS.
가락지 NS.
가랑니 NS.
가랑이 NS.
가래 NS.
가랭이 NS.
가로 NS.
가뢰 NS.
가루 NS.
가르마 NS. /NVS/*PREDHA
가르매 NS. /NVS/*PREDHA
가리 NS.
가리개 NS. /NVS/*PREDHA
가리마 NS. /NVS/*PREDHA
가림자 NS. /*PREDHA
가마 NS.
가마귀 NS. /ANM
가마니 NS.
가맹 NS. /*PREDH
가면 NS. /*PREDHA
가명 NS.
가모 NS.
가묘 NS.
가무 NS. /*PREDHA
가문 NS.
가물 NS.
가물음 NS. /NVM
가물치 NS. /ANM
가뭄 NS. /NVM
가미 NS. /PRED1/PRED3
가발 NS. /*PREDHA
가방 NS.
가변성 NS.
가보 NS.
가부 NS.
가부장 NS. /HUM
가부좌 NS. /PRED2
가불 NS. /PRED1/PRED3
가빈 NS. /HUM
가사 NS.
가산 NS. /PRED1/PRED3
가살 NS.
가상 NS.; NS. /PRED1/PRED3
가설 NS. /PRED1/PRED3

< figure 2 - Extract of [DECOS-NS / V01] >

가공적이다 ADJS. /CM
가깝다 ADJS. /RM
가깝디가깝다 ADJS. /RM
가깝하다 ADJS. /HM
가난하다 ADJS. /HM
가냘프다 ADJS. /RM
가느다랗다 ADJS. /RM
가느스레하다 ADJS. /HM
가느스름하다 ADJS. /HM
가늘다 ADJS. /RM
가늘디가늘다 ADJS. /RM
가능하다 ADJS. /HM
가닥가닥하다 ADJS. /HM
가당찮다 ADJS. /RM
가동적이다 ADJS. /CM
가득가득하다 ADJS. /HM
가득하다 ADJS. /HM
가들막가들막하다 ADJS. /HM
가들막하다 ADJS. /HM
가뜩가뜩하다 ADJS. /HM
가뜩하다 ADJS. /HM
가뜬하다 ADJS. /HM
가랑가랑하다 ADJS. /HM
가량맞다 ADJS. /MM
가량스럽다 ADJS. /SM
가련하다 ADJS. /HM
가렵다 ADJS. /RM
가마노르께하다 ADJS. /HM
가마득하다 ADJS. /HM
가마말쑥하다 ADJS. /HM
가마무트름하다 ADJS. /HM
가마반드르하다 ADJS. /HM
가마반지르하다 ADJS. /HM
가맣다 ADJS. /RM
가무끄름하다 ADJS. /HM
가무대대하다 ADJS. /HM
가무댕댕하다 ADJS. /HM
가무레하다 ADJS. /HM
가무숙숙하다 ADJS. /HM
가무스럼하다 ADJS. /HM
가무스레하다 ADJS. /HM
가무스름하다 ADJS. /HM
가무잡잡하다 ADJS. /HM
가무족족하다 ADJS. /HM
가무칙칙하다 ADJS. /HM
가무퇴퇴하다 ADJS. /HM
가물가물하다 ADJS. /HM
가뭇가뭇하다 ADJS. /HM

< figure 3 - Extract of [DECOS-ADJS / V01] >

Affixed forms and compound forms constitute other lexicons [DECOA / DECOC]. Given that some of the affixes (prefixes and suffixes) produce a considerable number of affixed forms, especially affixed **nouns**, we need complete lists of affixes in order to construct a lexicon of affixed items in a systematic way. The number of affixes taken into account in the current version is as followings <figure 4>:

Prefixes	*Suffixes*	*Pseudo-Nouns**	Total
950	900	180	2 030

< figure 4 - Numbers of *Pfx*, *Sfx*, and *PN* >

* Pseudo-Nouns are units that only occur in combinations with other nouns.

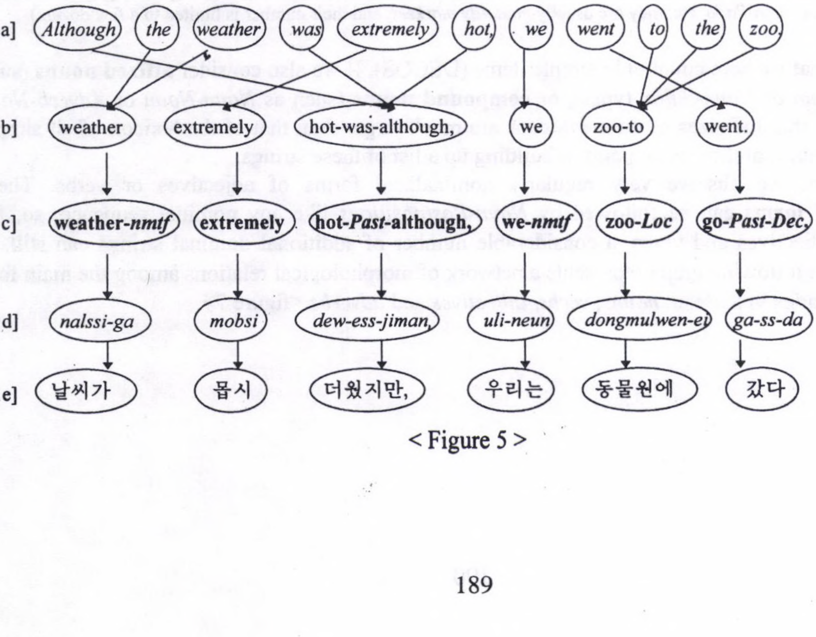
Notice that using a large corpus is indispensable for the construction of lexicons of affixed and compound items. However, remember that we can not obtain *automatically* the lists of these complex forms by combinating the lexicon of simple items with that of affixes, since there are too many homographs and therefore too many errors. In this case, we could try to establish syntactic or morphological rules that control wrong analyses and generations, but it seems to us that constructing valid general rules about all derivations and compositions would be a much more complicated task than describing all combined forms for each basic item. For the moment, the lexicons DECOA and DECOC are being constructed in such a procedure that their lists are estimated as much more complete than those we can find in existing dictionaries or in a large scale corpus: for example, the prefix '여 *ye-* [woman]' can be attached to nouns containing a semantic feature "*human*" such as 선생 *sensaing* [teacher], 간첩 *gancheb* [spy], 사장 *sajang* [boss]. We emphasize that, however, we do not search syntactic or semantic rules to list up these nouns, since all of the nouns with "*human*" feature do not admit the prefix '여 *ye-*': nouns denoting family relations or status such as 어머니 *emeni* [mother], 삼촌 *samchon* [uncle], 과부 *gwabu* [widow] do not accept this prefix: they already contain a gender marker; likewise, nouns describing human qualities such as 바보 *babo* [fool], 깍쟁이 *ggagjaingi* [miser] do not admit the prefix '여 *ye-*' either. Therefore, the list of nouns prefixed by '여 *ye-*' should be built up by examination of all simple nouns with "*human*" feature.

2. Lexicons of *N-Postpositions*, *A-Postpositions* and *V-Postpositions* DECO-POST

2.1. Strings delimited by separators

Let us recall that, in Korean, there are grammatical function markers such as *nominative*, *accusative*, *dative* or *locative* (we call them ***Noun*-Postpositions** [*PostN*] 명사 활용어), which are linked to nouns without any blanks. Thus, as we mentioned above, a sequence composed of two strings like *'in Paris'* corresponds to one string *'Paris-Locative'* in Korean. Likewise, verbs appear as conjugated forms like in English or French, but the inflectional suffix sets (we call them ***Verb*-Postpositions** [*PostV*] 동사 활용어) include several types of suffixes such as tense marker, modality marker, aspect marker, sentence or string type marker or politeness marker. Besides, the order and combinational constraints are extremely complex. Adjectives in Korean also should be followed by inflectional suffix sets (we call them ***Adjective*-Postpositions** [*PostA*] 형용사 활용어): suffixes indicate all grammatical functions of adjectives, whereas it is a copulative verb such as *'be'* or equivalent verbs in English, or such as *'être'* or equivalent verbs in French that takes the markers indicating grammatical functions of adjectival strings.

Thus, whereas the following sentence in English [1a] contains 9 strings separated by blanks, the corresponding sentence in Korean is composed of 6 strings as shown in [1e] <figure 5>:



< Figure 5 >

It is obvious that an automatic analyzer in Korean could not recognize canonical forms of nouns, verbs or adjectives without information about associable postposition types. (look at the phase [1d] in the graph above. Except the adverbial string '몹시 *extremely*', all strings are composed of a basic item and (a) grammatical suffix(es): '날씨 ***weather*** - 가 *Nominative Postposition*', '더우 ***hot*** - ㅓㅆ *Past* - 지만' *Conjunctive Postposition*', '우리 ***we*** - 는 *Nominative Postposition*', '동물원 ***zoo*** - 에 *Locative Postposition*' and '가 ***go*** - ㅆ *Past* - 다 *Declarative Postposition*').

Therefore, a machine-readable dictionary (MRD) should provide information about all these strings. One could intend to represent a complete list of all conjugated forms in a MRD, given that finding out general rules that cover all cases is much more complicated than listing them out.

However, the number of sequences of postpositions for each basic item is considerable: a simple noun can be followed by around 1 500 different sequences of postpositions, since several postpositions can combine with one another (e.g. *Dative-Modality-Modality* such as '에게-만-이라도'); a verb and an adjective can be linked to around 6 000 types of postposition combinations. Hence, for a dictionary containing 35 000 basic items (*cf. Korean electronic dictionary of simple items* **DECOS**), we can observe around 100 million strings as shown in the following table <figure 6>: it will be too huge to be presented in the form of a list in a MRD.

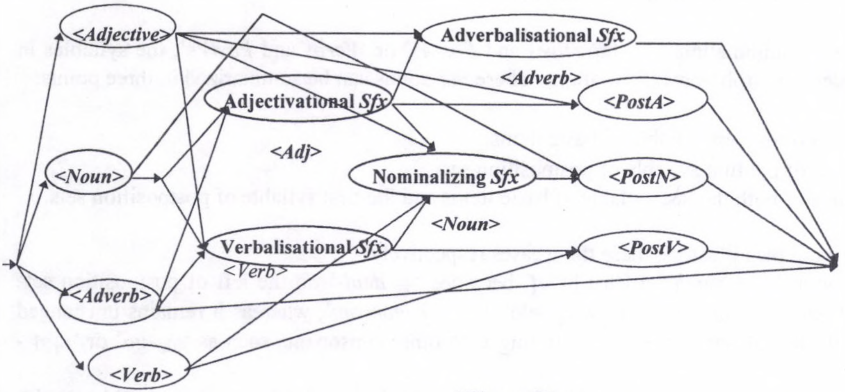
	Entry numbers in DECOS	Estimated String numbers
Simple Nouns	15 000	15 000 x 1 500 = 2.2 x 10^7
Simple Adjectives	5 300	5 300 x 6 000 = 3.2 x 10^7
Simple Verbs	7 500	7 500 x 6 000 = 4.5 x 10^7
Simple Adverbs / *Functional units*	7 000 200	7 000 * 200
Total	35 000	10^8

< Figure 6 >

(* We here do not take into account some *Adverb-Postpositions*, such as 도 *do*, 는 *neun*, 만 *man*, 만은 *man-eun*, 만이라도 *man-ilado*, etc: they are usually *modality markers*, and their number is limited to a few dozen.)

Remember that we here count only simple items (DECOS). If we also consider **affixed nouns** (such as *Prefix-Noun* or *Noun-Suffix* types), or **compound nouns** (such as *Noun-Noun* or *Adverb-Noun* types), given that the sizes of these lexicons are much larger than that of the lexicon of all simple items, it is clear that there is no point in building up a list of these strings.

Moreover, we observe very regularly nominalized forms of adjectives or verbs. These **nominalized forms** can be followed by *Noun-Postpositions* like any nominal sequence; so, for almost all adjectives and verbs, a considerable number of additional nominal strings can still be made up. The following graph represents a network of morphological relations among the main four lexical categories in Korean: *nouns*, *verbs*, *adjectives* and *adverbs* <figure 7>.



< Figure 7 >

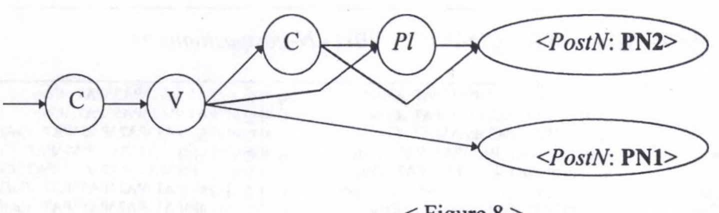
Therefore, for the time being, the method we have adopted in our lexical system DECO is to construct a lexicon of sequences of postpositions apart (DECO-*POST*), and to indicate morphological and inflectional information around each basic item, i.e. *nouns*, *adjectives* and *verbs*, in the lexicon DECOS: the sequences of postpositions are represented in the form of finite state automata (FSA).

2.2. Pinciples of sub-classification of postposition sets in DECO-POST

2.2.1. Classes of N-Postpositions

The combination of a *Noun* with a *PostN* is regular: the elements do not undergo morphological variations in connection. In the morphological view point, we can divide *PostN*s into two series: a series of *PostN*s that attaches to nouns with a **vocalic** ending; the other series of *PostN*s that attaches to nouns with a **consonantal** ending. We do not observe any morphological changes in the syllables in connection: neither in basic items themselves (nouns) nor in postposition sets.

Currently, *PostN* sets are sub-classified by morphological information types. The class **PN1** can be associated with **vowel ending** nouns, while the class **PN2** can follow **consonant ending** nouns. When a singular noun becomes a plural form (i.e. followed by plural marker '들 *deul* [pl]': it is the only grammatical marker of plurality in Korean), the postposition sets will be **PN2** type, since this marker ends in a consonant. We can represent these combinations as in the following graph <figure 8>:



< Figure 8 >

2.2.2. Classes of A-Postpositions and V-Postpositions

In the case of combinations of '*Adjectives* and *PostA*s' or '*Verbs* and *PostV*s', the syllables in connection undergo morphological variations. These variations can be summarized in three points:

A. Variations of the last syllable of basic items;
B. Variations of the first syllable of postposition sets;
C. Variations of both the last syllable of basic items and the first syllable of postposition sets.

The following examples illustrate these three cases respectively:

Ex-A. A verb '듣다 *deud(da)* [(to) hear]' becomes '들 *deul-*' on the left of postposition sets starting with a null consonant such as '어라 *-ela*' or '으면 *-eumyen*', whereas it remains unchanged '듣 *deud-*' on the left of postposition sets starting with other consonants such as '고 *-go*' or '다가 *-daga*';

Ex-B. An adjective '착하다 *chagha(da)* [(to be) kind]' does not change when combined with postposition sets, but it requires a particular variant of the sequences of postpositions, when this sequence begins with a null consonant. Thus, postpositions starting with '여 *-ye*' such as '여서 *-yese*' or '였으므로 *-yesseumeulo*' only occur after verbs ending in '하 *ha*': '하여서 *ha-yese*', '하였으므로 *ha-yesseumeulo*';

Ex-C. A verb '굽다 *gub(da)* [(to) bake]' changes when the following postposition sets start with a silent consonant such as '으면 *-eumyen*' or '어 *-e*': the verbal string containing the first type of postposition will be '구우면 *gu-u-myen*', and the string with the second type will be a different type of fusion '구워서 *gu-we-se*'.

In the current version of the lexicon DECO-POST, we have integrated the morphological variants of postpositions [Ex-C type] and [Ex-B type] (the number of all postposition combination sets reaches to about 42000 in each of the cases of *PostA* and *PostV*). This dictionary provides information about the morphological types of basic items, i.e. adjectives and verbs. Here are samples of [DECO-POST / V01] <figure 9> and <figure 10>:

```
E   \PN1 \PN2 .nmtf.Acc .Postp
가  \PN1 .nmtf.Postp
가라도  \PN1 .nmtf.Postp
가보다 \PN1 .nmtf.Postp
가보다는 \PN1 .nmtf.Postp
가보다도 \PN1 .nmtf.Postp
가보단 \PN1 .nmtf.Postp
같이  \PN1 \PN2 .Postp
같이가  \PN1 \PN2 .Postp
같이까지가  \PN1 \PN2 .Postp
같이까지나 \PN1 \PN2 .Postp
같이까지나마 \PN1 \PN2 .Postp
같이까지는 \PN1 \PN2 .Postp
같이까지라도 \PN1 \PN2 .Postp
같이까지만 \PN1 \PN2 .Postp
같이까지만도 \PN1 \PN2 .Postp
같이까지만이 \PN1 \PN2 .Postp
같이까지만이라도 \PN1 \PN2 .Postp
같이까지야 \PN1 \PN2 .Postp
같이까진 \PN1 \PN2 .Postp
같이나 \PN1 \PN2 .Postp
같이나마 \PN1 \PN2 .Postp
같이는 \PN1 \PN2 .Postp
같이는커녕 \PN1 \PN2 .Postp
같이도 \PN1 \PN2 .Postp
같이라도 \PN1 \PN2 .Postp
같이를 \PN1 \PN2 .Postp
```

< Figure 9 - Extract of [DECO-POST / V01] - ***N-Postpositions*** >

```
E(아) \PA1 .TmDec .Conj
E(아)? \PA1 .TmInt
ㄴ \PA1 \PA2 \PA5 \PA7 .Dtm
ㄴ가 \PA1 \PA2 \PA5 \PA7 .Conj
ㄴ가? \PA1 \PA2 \PA5 \PA7 .TmInt
ㄴ가가 \PA1 \PA2 \PA5 \PA7 .Conj
ㄴ가나 \PA1 \PA2 \PA5 \PA7 .Conj
ㄴ가나마 \PA1 \PA2 \PA5 \PA7 .Conj
ㄴ가는 \PA1 \PA2 \PA5 \PA7 .CoDis
ㄴ가는 \PA1 \PA2 \PA5 \PA7 .Conj
ㄴ가도 \PA1 \PA2 \PA5 \PA7 .Conj
ㄴ가라도 \PA1 \PA2 \PA5 \PA7 .Conj
ㄴ가마저 \PA1 \PA2 \PA5 \PA7 .Conj
ㄴ가마저도 \PA1 \PA2 \PA5 \PA7 .Conj
ㄴ가만 \PA1 \PA2 \PA5 \PA7 .Conj
ㄴ가만도 \PA1 \PA2 \PA5 \PA7 .Conj
ㄴ가만은 \PA1 \PA2 \PA5 \PA7 .Conj
ㄴ가만이 \PA1 \PA2 \PA5 \PA7 .Conj
ㄴ가만이라도 \PA1 \PA2 \PA5 \PA7 .Conj
ㄴ가뿐만아니라 \PA1 \PA2 \PA5 \PA7 .CoDis
ㄴ가뿐만아니라 \PA1 \PA2 \PA5 \PA7 .Conj
ㄴ가뿐아니라 \PA1 \PA2 \PA5 \PA7 .CoDis
ㄴ가뿐아니라 \PA1 \PA2 \PA5 \PA7 .Conj
ㄴ가야 \PA1 \PA2 \PA5 \PA7 .Conj
```

< Figure 10 - Extract of [DECO-POST / V01] - ***A-Postpositions*** >

2.3. Final syllable types of adjectives and verbs

As we mentioned above, we have 5 300 adjectives and 7 500 verbs in the current version of our lexicon DECOS. We have classified them according to the final syllable types: we have obtained 151 types for adjectives and 325 types for verbs. These classes can be regrouped according to the required postposition types. The following tables represent respectively the first entries of these types: <figure 11> and <figure 12>.

Type	A-1	감	Type	A-11	곧	Type	A-21	길	Type	A-31	낫
Type	A-2	갑	Type	A-12	곱	Type	A-22	깊	Type	A-32	낮
Type	A-3	같	Type	A-13	괠	Type	A-23	깜	Type	A-33	넓
Type	A-4	갛	Type	A-14	교	Type	A-24	깝	Type	A-34	넘
Type	A-5	걸	Type	A-15	굳	Type	A-25	껌	Type	A-35	녹
Type	A-6	검	Type	A-16	굵	Type	A-26	껍	Type	A-36	높
Type	A-7	겁	Type	A-17	굽	Type	A-27	꼽	Type	A-37	눅
Type	A-8	겋	Type	A-18	궂	Type	A-28	나	Type	A-38	늘
Type	A-9	결	Type	A-19	글	Type	A-29	낡	Type	A-39	늙
Type	A-10	겹	Type	A-20	기	Type	A-30	납	Type	A-40	늦

< Figure 11- Final syllable types of *Adjectives* >

Type	V-1	가	Type	V-11	겪	Type	V-21	굳	Type	V-31	기
Type	V-2	갈	Type	V-12	견	Type	V-22	굴	Type	V-32	긷
Type	V-3	갉	Type	V-13	곁	Type	V-23	굶	Type	V-33	길
Type	V-4	감	Type	V-14	고	Type	V-24	굽	Type	V-34	깁
Type	V-5	갖	Type	V-15	골	Type	V-25	귀	Type	V-35	까
Type	V-6	갚	Type	V-16	곪	Type	V-26	그	Type	V-36	깎
Type	V-7	개	Type	V-17	곯	Type	V-27	글	Type	V-37	깔
Type	V-8	건	Type	V-18	곱	Type	V-28	긁	Type	V-38	깜
Type	V-9	걸	Type	V-19	괴	Type	V-29	금	Type	V-39	깨
Type	V-10	게	Type	V-20	구	Type	V-30	긋	Type	V-40	꺾

< Figure 12 - Final syllable types of *Verbs* >

III. Perspectives

So far, we have discussed the method we have adopted in constructing Korean electronic lexical system DECO. Given that existing editorial dictionaries are hardly reliable, the use of a large Korean corpus should be required. For the construction of the lexicon of **simple items** [DECOS / V01], we have consulted existing editorial dictionaries, but formal and explicit principles have been used: accurate attribution of parts of speech is done; morphological and syntactic information is indicated in a coherent way.

A large corpus is required especially when we build up lexicons of **affixed items** and **compound ones**. However, even if the scale of the corpus is considerably extended, on one hand, we can never avoid lack of lexical items; on the other hand, we cannot only expect appropriate identification of affixed and compound forms. Therefore, the exhaustive description of combinations for a given item will be indispensable to obtain all correct sets related to each item and only them: powerful general grammars that cover all cases do not exist.

The current version of our lexical system DECO provides, on one hand, all simple items, classified by parts of speech: *nouns* (NS), *adjectives* (ADJS), *verbs* (VS), *adverbs* (ADVS), *functional units* (FUS); and the affixes: *prefixes* (PF), *suffixes* (SF), *pseudo-nouns* (PN). On the other hand, it contains a lexicon named DECO-POST that provides all postposition sets, i.e. *Noun-Postpositions* (PostN), *Adjective-Postpositions* (PostA), *Verb-Postpositions* (PostV): these sets are represented in the form of finite state automata.

The lexicons of affixed items and compound ones should be developed in an exhaustive and coherent way, by using combinatorial procedures based upon the lexicons of simple items and that of affixes. Besides, all information about the conjunction of lexical items (DECOS) and grammatical items (DECO-POST) has to be described in detail.

Reference

Clemenceau, David, 1993, *Structuration du lexique et reconnaissance de mots dérivés*, PhD thesis, Paris: Univ. Paris7.

Courtois, Blandine, 1987, *Dictionnaire éléctronique du Laboratoire d'Automatique Documentaire et Linguisique pour les mots simples du français* (DELAS), Rapport Technique of LADL, n°17, Paris: Univ. Paris7.

Courtois, Blandine, 1989, DELAF: *Dictionnaire éléctronique du LADL pour les mots fléchis du français*, Rapport Technique of LADL, N°20, Paris: Univ. Paris7.

Grand Dictionnaire Encyclopédique Larousse, 1982, Paris: Larousse.

Grand Robert de la Langue Française, 1986, Paris: Le Robert.

Gross, Maurice, 1987, The use of finite automata in the lexical representation of natural language, *Lecture Notes in Computer Science* 377, Springer-Verlag.

Gross, Maurice, 1989, La construction de dictionnaires élétroniques, *Annales des Télécommunications,* tome 44 N°1:2, Issy-les-Moulineaux / Lannion:CNET.

I, Hi-Seung, 1988, *Guge Dai Sajen* (Korean Dictionary), Seoul: Minjungselim.

Kim, Myung-Cheol; Seo Kwang-Jun; Jun Kyung-Heon, 1992, *Development of Natural Language Interface*, Report in ETRI, Korea.

Nam, Jee-Sun, 1991, *Etablissement du corpus des adjectifs coréens*: Rapport technique N° 30, Paris: Institut Blaise Pascal, University Paris 7.

Nam, Jee-Sun, 1992, Corpus des adjectifs coréens: Constitution et classification, *XVème Congrès International des Linguistes*, Québec: University Laval.

Nam, Jee-Sun, 1994, Représentation de la combinatoire des variantes consonantiques et vocaliques et de la combinatoire des suffixes de conjugaison des adjectifs en coréen, *Papers in Computational Lexicography Complex '94*, ed. by Ferenc Kiefer, Gabor Kiss et Julia Pajzs, Budapest: Linguistics Institute, Hungarian Academy of Sciences.

Nam, Jee-Sun, 1994, *Dictionnaire des noms simples du coreen*, Rapport technique N°46, LADL.

Nam, Jee-Sun, 1996, *Dictionary of Korean simple verbs : DECOS-VS / V01,* Rapport technique N° 49, Laboratoire d'Automatique Documentaire et Linguistique, Université Paris 7.

Nam, Jee-Sun, 1996, *Lexicon of Korean predicative terms classified by morphological ending forms, Vol. II. : Predicates except* Hada *ending ones*, IGM Rapport N° 96-9, Institut Gaspard Monge, Université de Marne-la-Vallée.

Nam, Jee-Sun, 1996, *Dictionary of N-Postpositions, A-Postpositions and V-Postpositions in Korean :DECO-POST / V01*, Rapport technique N° 51, LADL, Université Paris 7.

Perrin, Dominique, 1989, Automates et algorithmes sur les mots, *Annales des Télécommunications,* tome 44 N°1:2, Issy-les-Moulineaux /Lannion:CNET.

Roche, Emmanuel, 1993, *Analyse syntaxique transformationnelle du français par transducteurs et lexique-grammaire,* PhD thesis, Univ. Paris 7.

Silberztein, Max, 1993, *Dictionnaires éléctroniques et analyse automatique de textes - le systeme Intex,* Paris: Masson.

Sin, Gi-Chel; Sin, Yong-Chel, 1990, *Sai Ulimal Keun Sajen* (New Korean Dictionary), Seoul: Samseng Chulpansa.

Identification of Terms and Linguistic Patterns on Unrestricted Texts

OUESLATI ROCHDI

Abstract

In this paper we describe a term acquisition system which uses texts and specific tools to identify terms denoting concept labels. The design of term identification tools would be a great aid to a terminologist (particulary in specialised domains, where terms denote unambiguous concepts) or to a knowledge engineer, analysing a new domain. Our approach to term acquisition uses repeated word sequences and word distributions to help locate meaningful entities in text. First, the program collects all repeated word sequences and uses a stop list made up of grammatical words to refine the data obtained. Secondly, it produces a structured list of terms: head plus extensions. On the basis of some example texts and the system output, we will illustrate how these terms can be exploited to identify linguistic patterns which has the following form: term1 verb term2 and which are frequently used to describe linguistic relationships between terms.

1. Introduction

A terminologist is mainly concerned with terms and with relating them appropriately via relationships and definitions. A term can be defined as a linguistic sign made up of one or several words which denote unambiguous concepts in a specialised domain. A concept is a mental representation of an entity.

In NLP systems, a terminology is often built by employing texts and a set of methods which facilitate term identification. Texts used in terminology may be technical documents, texts which introduce the domain, guide-books, etc.

2. Term acquisition tools

Corpora-based techniques have been widely used for term acquisition (Delisle and Szpakowics, 1991) , (Delisle, et al., 1994) , (Möller, 1989) . Classical methods which use grammars and domain dictionaries (Mars, et al., 1994) cannot be used in totally new domains where domain dictionaries and conceptual hierarchies have not yet been established. Furthermore, they need to have a process that determines unambiguously syntactical categories. Such a process is time consuming.

The design of term identification tools would be a great aid to a terminologist (Czap and Nedobity, 1990) , (Meyer, et al., 1992) or to a knowledge engineer (Skuce, 1993).

Many tools perform term acquistion (Enguehard, 1992) , (Reinert, 1995) , (Justeson and Katz, 1995) , (Bourrigault and Lepine, 1994) . For instance, Bourigault uses syntactic markers as boarders to locate terms. Compounds are located and structured into a terminological network. Smadja uses statistical methods to collect relevant pairs of co-occuring words, which are then used to reconstruct n-word collocations (called n-grams).

3. Our Approach to Term Acquisition

Our approach to term acquisition uses repeated word sequences and word distributions to help locate meaningful entities in text. Repeated word sequences (Lebart and Salem, 1994) are n-word strings (n > 1) occurring at least twice in the text (strings containing punctuation are not considered). The semantic hypothesis behind repeated word sequences is based on Harris (Harris, 1968) . According to Harris, language offers discrete and arbitrary linguistic units which may be clustered according to linguistic constraints (i.e., terms are made up of linguistic units choosen to cluster according to linguistic constraints).

The corpus processed by our system is a 35000-word medical corpus in French, consisting of medical reports on patients, and a description of coronarography:

"Patient âgé de 52 ans, adressé par le Dr. C. pour coronarographie diagnostique. Il a bénéficié il y a deux ans d'une angioplastie coronaire droite avec un bon résultat contrôlé et malgrè ce, a constitué il ya un an un IDM inférieur par occlusion de cette artère. Depuis il est asymptomatique sous traitement béta-bloquant .."

3.1. MANTEX: the Term Acquisition Module

The MANTEX module collects all repeated word sequences (RSs) and use a stop list (a filter) to refine the data obtained. The stop list is made up of grammatical words such as prepositions, conjunctions, articles, etc and a domain verb list (LV). Only RSs begining and ending with words from the stop list are discarded (Frath, et al., 1995) . Figure 1 shows examples of RSs before and after the filtering process.

RSs before the filtering process	RSs after the filtering process
Après infarctus du myocarde *les*	infarctus du myocarde
de tension artérielle *et*	tension artérielle
*l'*épreuve d effort *est*	épreuve d'effort
la chirurgie coronarienne *ne*	chirurgie coronarienne
par voie intramusculaire *avant*	voie intramusculaire
un angor d effort *qui*	angor d'effort

Figure 1. Examples of RSs before and after the filtering process

MANTEX builds a domain verb list (LV) automatically (Oueslati, et al., 1996). It uses a set of morphological rules : Let (L) = (é, ait, aient, ant, er) be a list of verb endings.

If a word (w) ends with a member (m) of (L) then the module supposes the string consisting of (w) without (m) is a radical candidate. If this candidate is also found with another member of (L) then it is retained as a verb (for example, "montr(é)" and "montr(ait)" --> the verb is :"montr(er)"). MANTEX produces 75 verbs from the medical corpus: (déceler, traiter, dilater, présenter, ...).

A problem which must now be dealt with is noise (RSs which are not terms). Some RSs produced by the system are included in others. Such RSs are not considered and are discarded by applying an inclusion rule. For example MANTEX finds out that ((myocarde récent) . 4) is included in ((infarctus du myocarde récent) . 4) (numbers between brackets indicate the RSs frequency).

In addition, noise includes RSs containing verbs. Such RSs are discarded by using the list of domain verbs (LV). At this point the system has produced a list of candidates to be domain terms.

MANTEX produces a structured list of terms: generally terms can occur in texts with different extensions, the main term can be seen as the head of a tree and extensions as branches. For instance, in French, the first noun of a noun phrase usually expresses the main concept, while the rest expresses specification. For instance, in "artère coronaire" the main concept is "artère" and "coronaire" is used to specify the main concept. example of a term tree (Figure 2) (Barthelemy, 1995):

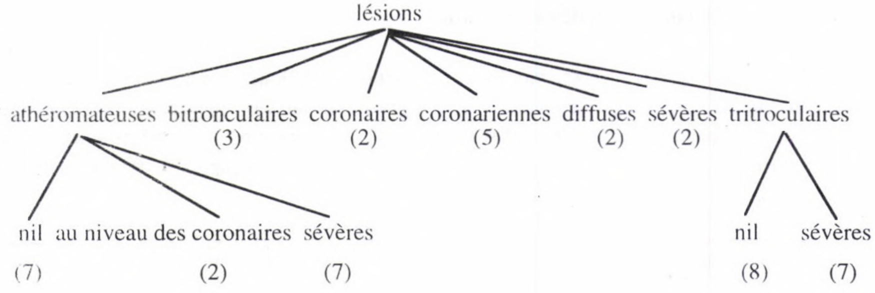


Figure 2. Example of a term tree

In addition, heads in the singular and in the plural can be put together (lemmatization) which means a small number of heads can unfold into a larger number of terms.

3.2. Evaluation

The following quality ratio suggests that our method needs some refining but can be considered as rather efficient:

Total Heads (TD)		343
Irrelevant Heads (I)		52
Relevant Heads (RD)	291	
Quality (RD/TD)	0.84	

Quality is the ratio between relevant collected data (RD) and total collected data (TD) according to the formula Q=RD/TD (Q=1 means all collected data is relevant). In addition, a measurement on several randomly chosen texts from our corpus has shown only about 15% of missing terms.

4. Linguistic Patterns Acquisition

In this section we shall describe how linguistic patterns may help study external relations which hold between terms. Czap and Meyer (Czap and Nedobity, 1990) , (Meyer, et al., 1992) think that a terminology should be added knowledge (encyclopedic or expert knowledge) which can be useful in understanding a new domain. We are interested in linguistic patterns which may describe external relations which hold *between* two terms:*term1* and *term2* . Our main idea is that a verb occurring with two terms: *term1* **verb** *term2* may describe a domain-specific relation.

We propose an incremental process called the VERB process which help study relationships between terms.

In a first step, the user (a linguist or a terminologist) selects a verb from the (LV) list which may describe a binary relation. The program collects automatically all contexts where term1 verb term2 co-occur. The hypothesis is that all these contexts express the current relation. The program then builds two sets of terms as arguments of the verb. For instance, let us consider the following verbs (Figure 3) from the (LV) list: "montrer" "présenter" (SHOW). The program builds the following list:

Verb	term1	term2
montrer	bilan	examen clinique
	bilan	sténose
	coronarographie	calcifications diffuses
	coronarographie	lésion significative
	coronarographie	lésions bitroculaires
	coronarographie	lésions coronariennes
	coronarographie	lésions sévères
	coronarographie	lésions tritronculaires
	coronarographie	minimes irrégularités
	...	
présenter	patient	accident vasculaire cérébral
	patient	angor d'effort
	patient	angor spastique certain
	patient	angor spontané prolongé
	patient	douleur précordiale
	patient	douleur thoracique
	patient	infarctus du myocarde
	...	

Figure 3. Example of terms as arguments of a verb

The program then builds a morphosyntactic pattern from the list of contexts. For instance, let us consider the above "montrer" verb. The program finds contexts (> 2) where: "coronarographie" montre "lésions" co-occur. The program builds the following morphosyntactic pattern:

X "montre" Y

which expresses the "montrer" relation.

To describe the "montrer" morphosyntactic pattern the program provides a frame with predefined attributes:

```
(montrer
    X: ("coronarographie" ..)
    Y: ("lésions" ..)
    definition: ( X montrer Y))
```

The attributes: X and Y describe the arguments of the verb. The attribute: definition describes the pattern built by the program).

Figure 4. describes the VERB process. It shows the different modules performed in order to search for linguistic relationships between terms. A module may be automatic or interactive.

Modules	Automatic	Interactive
1. Select a verb from the (LV) list		X
2. collect all the contexts term1 verb term2	X	
3. build the pattern X verb Y	X	
4. enter a relation hypothesis		X
5. apply the pattern term1 X term2 to the corpus	X	
6. collect new contexts verifying term1 X term2	X	
7. synthesize the contexts and build new patterns: X Exp Y	X	
8. select the new patterns verifying the relation hypothesis		X
9. search for new arguments X, Y where X Exp Y co-occur	X	
10. Module 5		

Figure 4. The VERB process description

In a second step, the program applies the pattern term1 X term2 (based on the couple (term1,term2)) to the corpus in order to collect new contexts verifying the current relation. For instance, let us consider the couple (coronarographie, sténose) which represents the two arguments of the SHOW relation. The program finds strings containing: "a confirmé l'existence de" and strings containing: "met en évidence des":

term1	morphosyntactic expression	term2
(coronarographie)	*a confirmé l'existance de*	*(lésions)*
(coronarographie)	*met en évidence des*	*(lésions) tritronculaires sévères*

The program then synthesizes these contexts in order to build the following morphosyntactic patterns which express the SHOW relation (X Exp Y):

X "confirmé l'existence" Y and
X "met en évidence" Y.

These patterns are used to search for new arguments (the program starts a new cycle search based on these patterns). The final result is a frame which consists of a set of morphosyntactic patterns which express the current relation, and two sets of terms as lexical arguments (arg-1 and arg-2) of the current relation:

```
(montrer
    arg-1: ("coronarographie" ..)
    arg-2: ("lésions" ..)
    patterns: ("montrer" mettre en évidence" "confirmer l'existence" ..))
```

5. Conclusion

The system outlined in this paper encompasses a number of tools which are effective for identifying terms and studying linguistic relationships between terms. While our system has specifically addressed only French terminology we think it can easily be transported to other languages. We are working on extracting terms from a genetic engineering corpus. In English the problem is that the main noun in a noun phrase is often the last, as in *(DNA strands)* but may sometimes be the first as in *(strands of DNA)* . Finally, our system can easily integrate new other tools. For example a set of tools have been developed to analyse a corpus consisting of Frequently Asked Questions in order to provide a structured overview on the contents and to give the user hypertext facilities.

6. References

(Barthelemy, 1995) T. Barthelemy. "Apprentissage automatique de schémas relationnels à partir de textes. Rapport de stage de DEA. Laboratoire ERIC ULP strasbourg." 1995) Technical Report .

(Bourrigault and Condamines, 1995) D. Bourrigault et A. Condamines. "Réflexions sur le concept de Base de Connaissances Terminologiques." *Actes des Journées du PRC-IA* (Nancy, 1995) .

(Bourrigault and Lepine, 1994) D. Bourrigault et P. Lepine. "Méthodologie d'utilisation de LEXTER pour l'acquisition des connaissances à partir de textes." *Actes de JAVA* (Strasbourg, 1994) .

(Czap and Nedobity, 1990) H. Czap et W. Nedobity. "Terminology and Knowledge Engineering." *Proceedings of the 2nd International Congress on Terminology and Knowledge Engineering* (Indeks Verlag Frankfurt, 1990) .

(Delisle, et al., 1994) S. Delisle, K. Baker, J.F. Delannoy, S. Matwin, et al. "Du texte aux clauses de Horn par combinaison de l'analyse linguistique et de l'apprentissage symbolique." *Actes de JAVA 94 Strasbourg* (Strasbourg, 1994) .

(Delisle and Szpakowics, 1991) S. Delisle et S. Szpakowics. "A broad coverage parser for knowledge acquisition from technical texts." *Proceedings of 5th International Conference on Symbolic and Logical Computing ICEBOL5 MAdison SD* (USA, 1991) pp.169-183.

(Enguehard, 1992) C. Enguehard. "ANA: Apprentissage Naturel Automatique d'un réseau sémantique ." (UTC Compiègne et CEA Cadarache, 1992). Thesis.

(Frath, et al., 1995) P. Frath, R. oueslati et F. rousselot. "Identification de relations sémantiques par repérage et analyse de cooccurences de signes linguistiques." *Actes de JAVA 95 Grenoble* (Grenoble, 1995) .

(Harris, 1968) Z. Harris. "*Mathematical structure of Language* ." (Wiley Interscience, New York, 1968).

(Justeson and Katz, 1995) J.J. Justeson et M. Katz. "Technical terminology: some linguistic properties and an algorithm for identification in text." *Natural Language Engineering* Vol.1(1), (1995) pp.9-27.

(Lebart and Salem, 1994) L. Lebart et A. Salem. "*Statistique textuelle* ." (Dunod, 1994).

(Mars, et al., 1994) N. Mars, H.D. Jong, P.H. Speel, G. Wilco, et al. "Semi-automatic knowledge acquisition in Plinius: an engineering approach." *Proceedings of the 8th Banff Knowledge Acquisition for Knowledge Based Systems Workshop* (1994) .

(Meyer, et al., 1992) I. Meyer, D. Skuce, L. Bowker et K. Eck. "Towards a new generation of terminological resources: an experiment in building a terminological knowledge base." *Proceedings of COLING* (1992) pp.956-960.

(Möller, 1989) J.U. Möller. "Knowledge acquisition from texts." *Proceedings of EKAW88 St Augustin* (1989) pp.25-1, 25-16.

(Oueslati, et al., 1996) R. Oueslati, P. Frath et F. Rousselot. "Tools for acquisition and exploitation of terms." *AISB Workshop on Language Engineering for Document Analysis and Recognition* (Sussex University, England, 1996) .

(Reinert, 1995) Reinert. "*ADT manuel de l'utilisateur (société Image)* ." (1995).

(Skuce, 1993) D. Skuce. "A multi-functional knowledge management system." (Knowledge Acquisition, 1993) pp.305-346.

(Smadja, 1993) F. Smadja. "XTRACT: an Overview. In Computer and the Humanities." *Computer and the Humanities* Vol.26, (1993) .

A Study of Word Ambiguity in French-English MT

MORRIS SALKOFF

The problem of the resolution of the ambiguity of words under translation is presently one of the major difficulties in the path of constituting a program of machine translation. It is shown here that this problem may be amenable to a satisfactory approximation after an empirical study has been carried out on the occurrence of ambiguous words in running text. It turns out that many more occurrences of such words appear in nonambiguous contexts (idioms, collocations, frozen expressions) than appear in a potentially ambiguous context. In the latter context, an approximation is available in which the ambiguity can be represented by means of a parallel translation presenting the reader with the two most likely translations in that context. This turns out to be satisfactory in most cases. It is improved by making reference to the semantic domain in which the sentence occurs, and by the use of semantic sub-classes of the grammatical categories to specify the semantic context in which the ambiguous word appears.

A study of word ambiguity in French-English MT

MORRIS SALKOFF
L.A.D.L.

1. The problem

It is a common observation that very many words of a natural language are ambiguous, i.e., they receive different paraphrases in the various contexts in which they appear. This can be seen quite clearly when translating into another language; the entries of any bilingual dictionary invariably list multiple contexts for a given word, and indicate its translational equivalent(s) in each context. Simple inspection of such contexts and their associated translations shows that the choice of such translational equivalents by the determination of the meaning of the word in context will be extremely difficult or impossible: in many contexts the choice of translation depends on the meaning of the word, which in turn depends on the meaning of the context. Hence the initial problem of choosing the meaning of a word depends on a problem of meaning choice that is just as difficult as the original problem. In such contexts, the word is said to be ambiguous. Taken together, these data pose a difficult problem for any program of machine translation, since it is precisely such a program that requires formal rules for choosing an appropriate translation. We shall see below that an approximation is available for the translation problem in this case that does not attempt to choose a meaning directly, and so circumvents this difficulty. In many other contexts, however, the word appears in a fixed expression, i.e., it appears with other words for which little or no substitution is possible without changing the meaning of the expression. These are frequently called collocations or idioms, and their translation usually poses much less difficulty.

It would then seem that the question of word ambiguity represents an almost insuperable obstacle for the constitution of a program of MT. Note, however, that word ambiguity has been defined above abstractly, by reference to the number of translations to be found in a bilingual dictionary. The greater the number of translations in the lexical entry for a given word, the greater is the possibility of finding it in an ambiguous context. But it is not clear how this definition of ambiguity is related to the ambiguity to be found in texts containing the word. The question then arises of just how many such ambiguous contexts are actually encountered in running text. What is the ratio of the number of occurrences of a given word in collocations, where its translation is unambiguous, with respect to the number of its occurrences in ambiguous contexts? To answer this question, we must examine the concordance of a given word in a large corpus, i.e., all of its occurrences in, for example, a year of the newspaper *Le Monde*. We then divide such a concordance into occurrences in collocations, and a remainder. If the remainder, which is the number of occurrences in possibly ambiguous contexts, is not too great, then the appropriate translation can be approximated by a parallel translation giving two possibilities. For example, in those contexts where the adjective *dur* modifies a concrete noun, it can be translated approximately as *stiff/tough*, and when it modifies an abstract noun, as *hard/harsh*. Attempting to define formally those contexts where *dur* means *stiff*, and those where it means *tough* is so difficult a project that little progress in this direction has so far been made. Hence, instead of trying to do this, I present the reader with both translations (or, what amounts to the same thing, both meanings) and allow him to choose the appropriate one according to context. This approximation by parallel translations will be used throughout.

The collocations consist essentially of support expressions, compound nouns, and various frozen adverbs and prepositions. A support expression consists of a support verb[1] followed by a predicate noun: *Vsup Npred*, e.g., *faire une allusion (à) (allude to)*, in which *Vsup = faire*, and *Npred = allusion.* All these collocations must be available in a lexicon, so that appropriate programs can identify them and remove the sentencing containing them from the concordance. The compound nouns and other frozen expressions can be identified by a lexical program, e.g., INTEX[2]; the support expresssions can be treated correctly only by a syntactic parser. The remaining occurrences are not all ambiguous; for many of them, the appropriate translation can be chosen on the basis of the syntactic environment in which they occur, in particular, in the environment of specific sub-classes of noun and verb. These occurrences too can be identified only by parsing the sentences. When these occurrences are removed, we have a remainder whose translations are possibly ambiguous. At this point, the approximation mentioned above can be used, and the translation of ambiguous occurrences of a word can be represented as a choice between two possibilities.

A group of 6 words with multiple translations into English has been studied in this way. A concordance of sentences in *Le Monde* 1992 containing occurrences of these words was established for each of them, and all sentences containing a collocation, idiom, or frozen expression in which the word appeared were eliminated. A large collection of French compound nouns, frozen adverbs and frozen prepositional locutions is available in the Laboratoire d'Automatique Documentaire et de Linguistique (L.A.D.L.); the program INTEX uses it to detect and highlight these collocations in the sentences of the concordance so that these sentences can then be discarded. These occurrences are listed in §2 under the heading Compound Expressions. In the remaining sentences, I find particular syntactic patterns containing the potentially ambiguous word in which its translation is unambiguous. Such patterns cannot be detected by purely lexical means; they would require the intervention of a parser, like the string parser of French[3]. These are listed in §2 under the heading Syntactic Expresssions. Assuming that the problem in which this study of word ambiguity is embedded is that of MT, then these sentences can also be discarded, for these patterns can be identified by the string parser and used to determine the unambiguous translation of the word. Note also that many of these syntactic patterns are observed only when such a concordance has been established, for it is not a simple matter to list such patterns without reference to a body of occurrences. However, this is not the case for the support expressions, which have been well studied in the L.A.D.L, and such lists are presently available[4] .

When all these sentences have been eliminated, the remainder contains those occurrences of the given word that may be ambiguous under translation into English. If there are no obvious syntactic or semantic means of lifting the ambiguity, then a parallel translation, indicating the two principal translational equivalents, must be presented to the reader. It turns out that in many cases only a relatively small percentage of the occurrences of a potentially ambiguous word actually require such a parallel translation; the lexical and syntactical means mentioned above enable an appropriate program to provide unambiguous translations for a considerable percentage of occurrences. This is the principal result of this study of word ambiguity in running text.

The corpora of occurrences taken from *Le Monde* are of a particular nature, since the bulk

[1] The so-called light verb; cf. Jespersen, 1965, Vol. 6, p. 117

[2] M. Silberztein, 1993

[3] M. Salkoff, 1973, 1979

[4] Cf. Giry-Schneider, J. 1987, Labelle, Jacques, 1983, Vivès, Robert 1983, Gross, Gaston, 1989.

of this newspaper contains articles on political and economic issues, and on world affairs in general. Hence it is to be expected that some technical uses of words may appear very infrequently here, as compared with the number of such uses in more technical or scientific articles. This study would have to be extended to corpora taken from scientific journals in order to verify whether the results obtained here are also valid in scientific domains.

2. The data

A concordance was established for each word, containing all its occurrences in the sentences of *Le Monde* for the year 1992. Each occurrence of the word gave rise to an entry, and a multiple occurrence of the word in a sentence gave rise to multiple entries in the concordance. Each concordance was then compared with a pre-established lexicon of compound nouns, frozen adverbs, and frozen prepositional sequences, whose translations are unambiguous. This lexicon is based on the lexicons of compound nouns and frozen expressions that have been constructed in the L.A.D.L. All sentences containing such occurrences of lexical compounds were eliminated from the concordance. Support expressions also have unambiguous translations, but the sentences containing them can be identified automatically with certainty only by using a parser. Parsing such a large corpus of sentences by machine was impractical, hence I eliminated them 'by hand' using the search program GREP. There remained many sentences containing the word in specific syntactic contexts that allowed for an unambiguous translation; these are presented in the discussion of each word.

When all such unambiguous cases have been deleted, there remains a variable percentage of the initial total of occurrences which may appear in an ambiguous context. These can be represented by a parallel translation containing the two most likely translations of the word, allowing the reader to choose the appropriate one in that context. The need for such parallel translations usually arises for a reasonably small number of occurrences of the word in running text.

It is not always feasable, in the presentation of the data below, to list all the lexical compounds separately, for they can be very numerous (more than 100) for some words. Hence, only the total number of sentences containing lexical compounds is indicated, except for one favorable case where the word appears in only a small number of compounds.

2.1 *suite*. This word can be translated as *suite* (hotel or music), *retinue*, *continuation* or *sequel*. There were 4.965 occurrences of *suite* or *suites* in *Le Monde* during the year 1992. The following fixed expressions were found; the number of occurrences is shown on the left.

Compound Nouns

2	suite infinie	infinite series
3	suite présidentielle	presidential suite
7	droit(s) de suite	right of pursuit
35	la suite des évènements	what followed
19	suites judiciaires	judicial consequences
3	suites juridiques	" "
3	la suite et la fin	conclusion

Frozen Adverbs

|	383	tout de suite	right away
|	9	par suite	subsequently
|	23	sans suite	incoherent
|	48	(et) ainsi de suite	and so forth
|	8	à la suite	one after the other
|	7	de suite (revenir de suite)	right away

Frozen Prepositions

2173	à la suite de	following
62	par suite de	owing to
119	suite à	following
13	dans la suite de (Nabs)	in the continuation of
4	dans la suite logique de	as an obvious result of

Frozen expressions

3	suite et fin	final episode
27	la suite logique	the obvious result (of)
425	par la suite	subsequently/later on
14	pour la suite,	for the rest,

This is a total of 3390 unambiguous occurrences of *suite*, the great majority of which is due to the frozen prepositional sequence *à la suite de*. There are also many support expressions containing *suite*, as follows:

Support Expressions

4	avoir de la suite dans les idées	be single-minded/show singlemindedness
12	pas avoir de suite (pas av. bcp. de suites)	lack coherence (not have much coherence)
2	avoir des suites	have consequences
25	classer sans suite	shelve (a project; a dossier)
280	faire suite à; faire une suite à	follow upon
102	donner (une) suite (à); suites à donner; quelle suite donner	follow up; pursue
4	les suites Adj à donner	the follow-up
2	manquer pas de suite dans les idées	have a certain coherence
27	prendre la suite de; prendre la suite	succeed to

This is a total of 458 occurrences of support expressions with unambiguous translations, so that the total of unambiguous occurrences of *suite* is now 3848, or 77% of the total number of occurrences.

Just as the parser is needed to analyze support expressions, so it can also be used to analyze other expressions containing unambiguous occurrences of *suite*. The context of these expressions can be described in terms of semantic sub-classes of nouns. In what follows, *Nt* is a time noun, *Q* is a quantifier (number), *Nabs* is an abstract noun, and *Tposs* is a possessive article (*his*, *her*, *its*,...):

12	Q fois de suite	Q times in a row (in succession)
34	Nt (ans, mois, jours, etc.) de suite	(days, months,...) in succession
117	une Adj suite de Npl[5] ; deux suites de Npl	a Adj series of Npl two series of Npl
5	suite de Npl (in apposition)	series
7	cette suite de Npl	this series of Npl
281	les suites de Nabs (Ndisease, accident, virus, désastre,..)	the consequences of Nabs
119	la suite de Nabs	the sequel to Nabs
28	à Tposs suite (à sa suite...)	following him, them, ...
13	dans la suite de (Nabs)	in the continuation of

This represents an additional 616 unambiguous occurrences of *suite*, for a total of 4464, or 90% of

[5] The noun phrase *la suite de Npl* is ambiguous (unlike *une suite de Npl*): either *the series* or *the continuation of Npl*.

all the occurrences of *suite*. There remain 501 occurrences, or about 10% of the total, whose translation may be ambiguous between *series, continuation, retinue* and *suite* (hotel suite or musical suite). The sentences in which each of the senses of *suite* occur are of the following kind:

suite = hotel suite	Translation
(1) salons, fumoirs et suites	salons, smoking rooms and suites
dans leur suites de palace	in their palatial suites
dans les suites de ministres	in the ministerial suites
suite = (musical) suite	
(2) suite en sol majeur	suite in G major
Il composa sa suite	He composed his suite
suite = series	
(3) dans ses suites d'encre de chine sur papier	in his series of India inks on paper
constitué de suites de danses andalouses	made up of series of ... dances
suite = continuation / the rest	
(4) ainsi que la suite intitulée...Fanny	as well as the continuation entitled Fanny
il savait la suite par coeur	he knew the rest by heart
suite = retinue	
(5) Charlemagne et sa suite	Charlemagne and his retinue
quand le ministre et sa suite arrivèrent	when the minister and his retinue arrived

Many of these occurrences of *suite* are ambiguous. Thus, in (1), *les suites des ministres* could also mean *the ministerial retinues*. Hence, in many of these sentences, it will be necessary to present several translations of *suite* in parallel. With the help of some reasonable approximations, it may be possible to reduce the number of such cases:

(i) in (4), the permanent ambiguity between *continuation* and *rest* can be represented by the *passe-partout* translation *continuation*.

(ii) the reference to the semantic domain 'music' allows *suite* to be translated as musical suite in (2). Similarly, *il composa sa suite* there might mean *he constituted his retinue*, but if the sentence is occurring in a text on music, this second translation is very unlikely.

(iii) the use of semantic sub-classes can be of help here. Thus, the most likely translation of *Det suite à/de Ntext*, where *Ntext* is a sub-class of nouns referring to texts, reports, books, etc., is probably *Det sequel to Ntext*: *une suite au célèbre roman de.. -> a sequel to the famous novel of..*; *la suite naturelle des Chaussons rouges -> the natural sequel to the Red Shoes*; *écrire une suite de Don Quichotte -> write a sequel to Don Quixote*.

(iv) the conjunction *Nh et sa suite* is more likely to be *Nh and his retinue* than *Nh and his hotel suite*: (*Charlemagne + le ministre*) *et sa suite ->* (*Charlemagne + the minister*) *and his retinue*.

When all these approximations are used, the remanent ambiguity of the word *suite* is considerably less than the 10% quoted above. Nevertheless, some intractable ambiguities remain:

(6)a Ce ne sera pas sans doute une nouvelle suite avec le même personnage -> It will undoubtedly not be a new (continuation + sequel + retinue) with the same character
b ne s'est pas satisfait de la suite. Il le dit dans son pamphlet -> did not satisfy himself with the (continuation + sequel + suite). He says so in his pamphlet

The major difficulty in approximating the translation of *suite* in these cases is that the sentences are taken from a newspaper, *Le Monde*, in which the sentences containing *suite* can appear in any semantic domain. The latter will change according to the topic being written about: the semantic domain of music in the art pages, where *suite* is more likely to be a musical suite, and the domain of politics and economy in the news section where *suite* is more usually a continuation or a retinue. And even in texts on music or politics, it is still possible for *suite* to refer to a hotel suite or a series, as well as to its more likely translation in that domain.

2.2 *feu*. There were 3.588 occurrences of *feu, feux* in Le Monde 1992. If context is not taken into account, it can be translated as *fire*, (*traffic* + *head*) *light*, or (*kitchen*) *burner*. The unambiguous collocations and expressions are distributed as follows[6]:

Compound Expressions: 68. *feu d'artifice* (*fireworks*), *arme à feu* (*firearm*), *feu de joie* (*bonfire*), etc.

Frozen Expressions: 4. *au feu les pompiers* (*call the firemen!*), (*plein*) *feu sur N!* (*spotlight on N*), etc.

Adverbs: 6. *feu* (*le* + *la* + *l'* + *son*) -> *the late..*; *his late* (*father*); *à feu*(*x*) (*doux* + *vif*) -> *on a* (*low* + *high*) *fire*, etc.

Frozen Preposition: 1. *dans le feu de N* (*in the heat of* (*the discussion*))

Support Expressions: 28. (*donner* + *recevoir*) *le feu vert* -> (*give* + *get*) *the go-ahead; ouvrir le feu -> open fire; mettre le feu -> set fire; mourir à petit feu -> die by inches*, etc.

Altogether, the number of sentences containing one of these occurrences amounts to 3005, leaving a remainder of 583, or 16% of the total number, that may be ambiguous. In these, the translation as *burner* occurs entirely in a particular semantic domain, that of culinary articles. Also, most of the occurrences of *feu* as (*traffic*)*light* occur here in the compound nouns *feu orange*, *feu rouge* and *feu vert*, so that the ambiguous (*traffic* + *head*) *light* can be represented by the *passe-partout* translation *light*. Hence, for the majority of this remainder, which contains a few cases of untreatable ambiguity, the parallel translation *fire/light* will suffice:

(7) a le signal orange lui indiquant que le *feu* suivant, qu'il rencontrerait ... -> the orange signal indicating to him that the following *light/fire,* which he..
b lui qui aimait le *feu* pour sa flamme, son éclat,... -> he who liked the *fire/light* for its flame, its brilliance
c Le *feu* n'a pu non plus jouer un rôle... -> The *fire/light* could also not play a rôle..
d Le *feu* dans la ville? -> The *fire/ traffic light* in the city?

[6] In the case of the concordance for *feu*, only the number of different collocations and expressions was noted, together with a final count of the total number of occurrences of all such expressions. For all the other words treated here, I also counted partial totals, i.e., the number of occurrences of each type of collocation or expression.

e Non-respect de la priorité, d'un *feu* ou d'un panneau stop -> Disregard of the priority/right of way, of a *fire/ light* or of a stop sign

f une quarantaine de *feux* étaient encore actifs -> about forty *fires/lights* were still active

g pour chevauchement de la ligne continue et maintien de *feux* gênants pour .. -> for crossing over the solid line and keeping on *lights/fires* which bother..

2.3 *courant*. Of the 2060 occurrences of this word in *Le Monde* 1992, 181 are adjectives, which can translate as *current*, *standard*, or *common*, and 1879 are nouns, which translate as *current*, *wave* or *movement*.

A. The noun *courant*. The following occurrences are unambiguous:

618 Compound Expressions: *pratique courante -> standard practise*; *courant de population -> population shift*, etc.

206 Support expressions: *tenir au courant* -> *keep up-to-date*; *remonter le courant -> get back on (one's) feet*, etc.

Many more expressions are unambiguous after parsing:

562 Syntactic Expressions

294 *courant Adjabs -> Adjabs movement*: *courant* (*antieuropéen, contestataire, ...*) -> (*anti-European, dissident, ..*) *movement*. Some would be better with *current*: *courant* (*musical, socioculturel, ...*) -> (*musical, sociocultural*) *current*.

124 *courant* (*de Nh-propre, de Nh, Npropre*) -> *Nproper movement*, *movement of Nh*: courant (de Fabius, des chrétiens) -> Fabius movement, movement of christians

71 *courant de Nabs -> current of Nabs*: *courant d'* (*opinion, influence, opposition, ...*) -> *current of* (*opinion, influence, opposition*). Some would be better with *wave*: *courant de* (*sympathie, comédie satirique, ..*) -> *wave of* (*sympathy, satirical comedy*), others with *movement*: *courant de* (*centre-gauche, sociologie, ...*) -> *movement of* (*the center-left, sociology*). But *current* will do as a *passe-partout*.

72 *courant Nt* (*during July*); *pour le courant de Nt* (*during the month, year..of April, 1992*); *dans le courant Nt* (*during April, 1992.*); *le 27 courant* (*27th of the month*); *au courant de l'année* (*during the year*)

1 *courant de Q volts -> current of Q volts*

When these unambiguous occurrences are subtracted, there remain 489 occurrences of *courant*, i.e., 26% of the total number. Most of these refer to a political movement, given the nature of the journal *Le Monde*, and a small number refer to electric current. Hence the parallel translation *current/movement* will suffice. Note that in a non-political context, perhaps in some technical journal, the translation as *current* would almost certainly be more frequent.

B. The adjective *courant*.

In the object of *être* (i.e., as an attribute), *courant* translates as *common:*

Il était courant de... -> It was common to...

Il est devenu courant de.. -> It has become common to...

As the right adjunct (modifier) of a noun, it is either *current, standard*, or *common*:

(résultat + dollars +emploi) courant -> current (result + dollars + use)
(entretien + fonctionnement + symptôme) courant -> standard (upkeep + functioning + symptom)
(argument + nom de famille + procédé) courant -> common (argument + family name + procedure)

The parallel translation *current/standard* may serve as a useful *passe-partout* translation.

2.4 *échelle*. There were 857 occurrences of *échelle* in *Le Monde* 1992; its translations are *scale, ladder*, and *run* (in a stocking). The unambiguous occurrences were as follows:

250 Compound Expressions: *économies d'échelle -> (large) scale economies*; *sur une grande échelle -> on a large scale*, etc.

349 Syntactic Expressions: *sur une ADJ échelle -> on an ADJ scale; à l'échelle DE -> on the scale of; faire la courte échelle -> give a boost*, etc.

There now remain 258 occurrences, which constitute 30% of the total. Many of them are of the form *à ADJ échelle, à une ADJ échelle, PREP DET échelle*: *à échelle humaine -> on a human scale, à une très large échelle -> on a very large scale, sur une échelle ADJ -> on an ADJ scale, sur l'échelle de gravité -> on the scale of gravity*, etc., where the translation is always *scale*. There are a few occurrences where the correct translation is *ladder*:

grande machine de guerre en bois avec *passerelle, échelle et poulie*, [*gangway, ladder and pulley*] qui vient chercher la prisonnière sur ...
La fermeture inattendue de la petite usine d'*échelles en aluminium* [*aluminum ladders*], la seule de la région, les quinze
aviation légère américaine ou des constructeurs d'*échelles et d'échafaudages* [*ladders and scaffoldings*], a engendré une multiplication

These ambiguities may be resolved by using two semantic sub-classes to disambiguate the context: *Nm*, nouns referring to a metal, and *Nc*, a concrete noun. Then in the context *échelle en Nm*, or where *échelle* is conjoined to an *Nc*: *pulley, scaffolding*, etc., the translation is *ladder*.

In a few occurrences, reference must be made to the semantic domain of music texts:

inventeur du concept de world music), il garde les *échelles défectives* [*defective scales*], la stabilité harmonique, l'absence de
mis aujourd'hui en musique selon *les modes et les échelles* [*modes and scales*] traditionnelles. Soliste de chorales réputées

Only such a reference can prevent *échelles défectives* from being translated here as *defective ladders*, a translation which is possible in a different semantic context. The reference to the domain of music also allows *modes et échelles* to be translated correctly, and not as *methods and ladders*.

There may also be occurrences in which the ambiguity of translation of *échelle* may be hard, even impossible, to resolve:

existe en trois tailles *adaptées aux différentes échelles du décor* [adapted to the different scales/ladders of the scenery] reçoivent des soins constants et assid

obant la totalité du grand bassin parisien. *Cette échelle est la mieux adaptée* [This ladder/scale is the best adapted] à la concurrence entre les g

cette reconstitution du Tricorne, sauf à *trouver l'échelle qui convienne à* [except by finding the scale/ladder that suits] l'immense plateau de Garnier.

Once use has been made of semantic sub-classes and reference to semantic domain, the parallel translation *scale/ladder* is needed for fewer occurrences than the 258 in this remainder of 30%.

2.5 *gain.*There are 647 occurrences of *gain* in *Le Monde* 1992. It can be translated as *increase*, *saving*, *winning*, *profit*, and *earning/wage*. The unambiguous occurrences are the following:

157 Compound expressions: *âpre au gain* (*greedy*), *gain de temps* (*saving of time*), *gain de place* (*saving of space*), etc.

12 *V sur un gain de* -> *V with a profit of*; *V* = *terminer*, *conclure*, *achever*, *finir*.

77 Support expressions: *avoir*, *obtenir gain de cause* (*win the case*); *donner gain de cause* (*decide in favor of*)

The remaining 401 occurrences of *gain*, 62% of the total, are almost all ambiguous, but the ambiguity of many occurrences can be lifted in two ways: by an examination of the syntactic and semantic context of *gain*, and by reference to the semantic domain in which the sentence occurs.

Syntactic context. (i) If all the expressions of the form *gain de N* whose translation is *saving of N* can be listed *in extenso*, then the ambiguity of *gain* is reduced by one. This must be investigated.

(ii) It seems possible to separate the translations *increase* and *profit* according to the measure nouns *N* in the *PN* that follows *gain*:

gain de Q %; gain de Q N --> increase of Q%, increase of Q N:

...dans le cadre de l'Uruguay Round, soit un *gain de 0,5 %* du produit intérieur brut -> in the framework of the Uruguay Round, i.e., *an increase of 0.5%* of the gross internal product

En fin de séance, le CAC 40 s'adjugeait un *gain de 1,19 %* -> At the end of the session, the CAC40 won *an increase of 1.19%*

avec un *gain de dix points* par rapport au précédent sondage -> with *an increase of 10 points* with respect to the previous poll

Avec un *gain de cinq sièges*, le groupe socialiste ... peut s'enorgueillir d'un *gain de près de 7 points* pour le PS.-> With an *increase of five seats*, the socialist group can boast of an *increase of almost 7 points* for the PS.

gain de Q Nmonnaie --> profit of QNmonnaie:

avec un *gain de 19 millions de dollars* entre 1990 et 1991 -> with a *profit of 19 million dollars* between 1990 and 1991

procure à l'Etat un *gain de 3,8 milliards de francs* en 1993 -> procures for the State a profit of 3.8 billion francs in 1993

(iii) When *gain de P N* contains an abstract noun, the most frequent translation is *increase*:

la totalité de ce *gain de pouvoir d'achat* -> all of this *increase in purchasing power*
Le *gain de puissance* ainsi obtenu -> The *increase in power* thus obtained
un *gain d'espérance de vie* de 3,5 mois par an -> an *increase in life expectancy* of ...

When *gain de P N* contains a concrete noun, the best translation is *winning*:

le *gain d'un globe* en or -> the *winning of a globe* in gold
l'annonce d'un *gain de 10 000 cacahuettes* ou d'un voyage -> the announcement of the *winning of 10.000 peanuts* or of a trip

Semantic context (iv) In a semantic context of sports or politics, *gain* is most frequently *winning*:
qui auraient dû lui donner le *gain de la deuxième manche* -> which should have given him the *winning of the second game*
Première place au classement général, *gain de l'étape*, record du parcours -> First place overall, *the winning of the stage*, the record of the distance
Avec un *gain de cinq sièges*, le groupe socialiste.. -> With the *winning of five seats*, ...

Note then that *gain* is ambiguous in *gain de cinq sièges*: it translates as *increase* in (ii) above, and as *winning* here in a political context.

(v) In the context of finance or economics, the translation is either *increase* or *profit*:

Les deux boutiques et *un gain annuel* de 2,5 millions de dollars -> The two stores and an *annual increase/profit* of 2.5 million dollars
Résultat : *un gain* de 5 millions de dollars par an -> Result: *an increase/profit* of 5 million
soit *un gain immédiat* de près de 11 000 francs par an. -> an *immediate increase/profit* of almost 11.000 francs per year
Un *gain énorme*, car la cotation est devenue entre-temps .. -> An *enormous increase/profit*, for the quotation has become meanwhile..

These two translations cannot be disambiguated without an examination of the wider context. Such an examination would be extremely difficult to carry out, so that the best approximation in this semantic context is the parallel translation *increase/profit*.

(vi) The semantic domain in which *gain* is translated as *earning/wage* is found in texts concerning unions, workers and salaries. Even in such a context, problems arise:

En effet, *ses gains* sont limités à la prime acquise au départ, et.. -> In effect, his *earnings/wages* are limited to the bonus..
sur les revenus actuels des salariés, mais de *gains supplémentaires* à eux ainsi attribués, -> on the present income of salaried people, but some additional *earnings/wages*...
En contribuant à réaliser de *tels gains*, les salariés scient une branche qui.. -> By contributing to making such *earnings/wages*, salaried people cut off a branch..

In the first two examples, we can safely translate *gains* as *wages* in a marked context. In the last example, even in a context of a salary discussion, *gains* is ambiguous between *profits* and *wages*. In the following example: *par un gain salarial de 600 à 700 francs*, the adjective *salarial* translates either as *salary* (as noun epithet) or *of salaries*, independently of the semantic context. In either case, *gain salarial* can translate only as *salary increase* or *increase in salaries*, as in (iii) above. Such a trańslation will be obtained only by a systematic study of adjectives like *salarial.*

When the decision as to the correct translation of *gain* cannot be made on the basis of its syntactic context (as in i-iii above), but depends on an examination of the semantic context, only approximations can be given:

Sport or Politics: *gain -> increase/winning*
Finance, Economics: *gain -> increase/profit*
Unions, salaries, etc.: *gain -> profit/wages*

The removal of the occurrences that can be identified syntactically from the still outstanding 60% of the total mentioned above leaves a smaller remainder, of the order of 50%, in which *gain* must be translated by one of the parallel translations shown above. If the semantic context is a non-specific one, as is frequently the case in a newspaper such as *Le Monde*, the translation *increase/profit* is likely the best approximation.

2.6 *affaire*. There were 15.845 occurrences of *affaire(s)* in Le Monde 1992. It can be translated as *case*, *matter*, *affair*, *business*. The non-ambiguous occurrences are in the majority:

7.646 Compound expressions: *mauvaise affaire*, *monde des affaires*, *homme d'affaires*, etc. The majority of occurrences here are due to *affaires étrangères* (*foreign affairs*), *chiffre d'affaires* (*turnover*), and *homme d'affaires* (*businessman*). This is to be expected in a newspaper devoted essentially to politics and economics.

2.566 Syntactic Expressions. Half of these are composed of *l'affaire* ou *l'affaire DE* followed by a proper name: *l'affaire Habache -> the case Habache*; *l'affaire de Mitterand -> the case of Mitterand*. Only one intercalated word, *dite*, is possible: *l'affaire dite du sang contaminé -> the so-called case of the contaminated blood*. The other half contains *l'affaire DU N*: *l'affaire des comptes du OM -> the case of the accounts of OM*; *l'affaire des fausses factures -> the case of the false invoices*; *l'affaire du Conseil constitutionnel -> the case of the constitutional Council*, etc.

Note that *les affaires de Npropre* does not translate as *cases*, but rather as ***business*** or *matters* (cf. below): *les affaires de la Chine -> the matters/business of China.*

354 Support and Frozen Expressions: *avoir affaire à -> deal with*; *faire l'affaire* (*de*): *will do* (*for*); *classer l'affaire -> shelve*, etc.; *la belle affaire -> big deal*; *une affaire en or -> a gold mine*; *c'est une affaire entendue -> it's a deal*; etc.

These non-ambiguous occurrences amount to 9.402, or 59% of the total. The remainder consists of 6.443 occurrences, or 41% of the total, of which 4.186 contain the singular *affaire*, and 2.221 contain the plural *affairs*[7]. The singular (*la + cette + une*) *affaire* can be represented by the parallel translation *matter/business*. If the article is *son*, *notre*, *leur*, etc., the translation is almost certainly *business*. An inspection of the occurrences yields some that are best represented by ***matter,***

[7] This adds up to a total of 6.407 occurrences; the discrepancy with the first figure of 6.443 is due to the appearance of a variable number of formatting lines in the RTF (Rich Text Format) used in the files containing the occurrences. The variation is never as much as 1%.

some by *business*, and some that are ambiguous:

Probably *business*: Quelques mois après avoir *cédé son affaire* à l'italien Ferruzzi, -> A few months after having *transferred his business* to the Italian Ferruzzi,

il a *monté son affaire* avec des *fonds* recueillis -> he *set up his business* with funds..

a part des "*commissions* " versées pour *obtenir l'affaire*. -> apart from the 'commissions' paid to *obtain the* matter/*business*

un résultat gonflé par la cession d'une *affaire américaine,Pennysaver,* pour 40 millions -> a result inflated by the transfer of *an American* matter/*business*, Pennysaver, for..

ils ont *investi dans l'affaire* 150 000 F chacun. -> they invested i*n the* matter*/business* 150.000 Francs each

Probably *matter*: le rapporteur du tribunal chargé de *résumer l'affaire* a affirmé qu' -> the court reporter in charge of summarizing *the matter/*business asserted..

Comme d'habitude, *l'affaire a dégénéré* en prise de bec entre... -> As usual, the *matter/*business degenerated into a row between..

faire toute la lumière sur cette affaire", -> get right to the bottom of *that matter/*business

Ambiguous: Cette *affaire* a été trop *onéreuse, financièrement* et symbolique -> This *matter/business* was too costly, financially and symbolically

dernier *emprunt* de la Confédération suisse, *une affaire* de 500 millions de francs -> the last loan of/from the Swiss Confederation, *a matter/business* of 500 million francs

Le groupe Expansion a *perdu dans l'affaire* 1 million de francs. -> The group Expansion lost in *the matter/business* 1 million francs

Tout compris, frais et commission inclus, *l'affaire revient à 9,31 %* l'an à l'emprunteur. -> All included, expenses and commission included, *the matter/business* amounts to 9.31% per year to the borrower

The plural. There are many syntactic expressions containing *affaires* whose translation is not ambiguous:

(i) When preceded by one of the determinanats such as *nombre de, infiniment de*, or by a noun determinant such as *série de*, *kyrielle de*, etc., the translation is *affairs*: (*nombre d' + infiniment d' + une série d' + une kyrielle d'*) *affaires* -> (*a number of + an immense number of + a series of + a stream of*) *affairs*.

(ii) The noun phrase *affaires Adj-pays*, where *Adj-pays* is the adjectivalized name of a country, always translates as *Adj-pays affairs*: *affaires* (*africaines + algériennes + albanaises*) -> (*african + algerian + albanian*) *affairs*.

(iii) In the noun phrase *affaires Adj-abs*, the best approximation is *matters/affairs*: *les affaires* (*constitutionnelles + communautaires + agricoles + consulaires*) -> (*constitutional + community + agricultural + consular*) *matters/affairs*.

When no syntactic clues like these are present, the plural *affaires* can be represented by the parallel translation *matters/business* (*deals*):

on trouve *des affaires à vendre* pour le franc symbolique. -> one finds *matters/business* (*deals*) for sale for the symbolic franc

de décider *des affaires* qui les concernent localement. -> to decide *matters/business* (*deals*) which concern them locally

moyens pour réguler le *flux des affaires* qui lui sont confiées. -> ways to regulate the flow of *matters/business* (*deals*) that are confided to him

son mode de conduite *des affaires* qui a été reproché -> his way of conducting *matters/business* (*deals*) which was reproached

3. Conclusions

In dealing with the ambiguity of words via their translations in a French-English lexicon, the idea of separating the occurrences containing nonambiguous lexically frozen entries from those occurrences in which the translation of the word is potentially ambiguous is not new. What is novel here is the systematic use of syntactic and semantic context in order to augment the number of nonambiguous cases, followed by the use of an approximate parallel double translation when the ambiguity cannot be lifted in a simple manner. In this last case, the reader, who has intimate knowledge of the semantic domain in which the translated text appears, has no difficulty in choosing the correct translation.

This separation of nonambiguous from ambiguous occurrences also yields an unexpected result when all the occurrences of a given word in a relatively large corpus are examined. In this study, I used all the issues of *Le Monde* for the year 1992 as my corpus. The examination of word concordances in this corpus has shown that a large percentage of the occurrences of a word are not ambiguous. The correct translation of many of those that are potentially ambiguous can frequently be decided on the basis of the syntactic and semantic context, where the latter is expressed in terms of semantic sub-classes of nouns, adjectives, etc., leaving a relatively small residue of potentially ambiguous occurrences. The translation of the ambiguous word in this residue can be approximated satisfactorily in most cases by the method of parallel translations. The least satisfactory case examined here is that of *gain*, almost half of whose occurrences are ambiguous and required different parallel translations in various semantic contexts. But for many other words, the number of potentially ambiguous occurrences is surprisingly small: less than 10% for *suite*, about 16% for *feu*, about 25% for *courant* and *échelle*. This study would have to be extended to a much larger number of words (particularly nouns and adjectives) in order to determine just how much of a problem word ambiguity in running text represents for a program of machine translation. One important point would be the discovery of how many words represent difficult cases, like *gain*, and how many represent favorable cases, like *suite*.

Note that if this study were to be extended to include other more technical corpora, it might be the case that difficult words like *gain* in such domains would be almost entirely nonambiguous. For example, in biological or physical texts, the correct translation might almost always be *increase*, and never *profit* or *winning*, whereas in electronic texts it might appear almost entirely as a technical term whose translation is *gain*.

What is required then to provide a satisfactory resolution to the problem of word ambiguity is an empirical study like the present one. The establishment and examination of concordances using the appropriate lexical tools, i.e., dictionaries of compounds, collocations, etc., constitutes an empirical study, much as the observation of stars in astronomy is empirical. And as for any empirical study, the results cannot be known in advance. In this connection, the case of the word *gain* in a very general domain like the newspaper *Le Monde* is instructive. We have seen that the specification of the semantic domain in which the sentence containing *gain* occurs is not sufficient to resolve its ambiguity under translation. For example, in a text on politics the ambiguity of *gain de cinq sièges -> increase/winning of five seats* cannot be resolved on the basis of imparting to the program the knowledge that this expression is in the semantic domain of politics. Even in such a domain, only an examination of a much larger context than the sentence itself would allow one to understand whether it is a question, after some election, of an increase or of a winning of seats. It is clearly impossible at present, and will be so for a long time to come, to carry out an examination of the meaning of the larger context and then codify it so as to enable one to write formal rules that could decide that question. The further problem then arises of how many words present the same difficulty as *gain*, and whether their translation can be approximated satisfactorily, as is the case with *gain*. The answer to this question will be obtained from an extension of the present study.

References

Giry-Schneider, J. 1987 *Les prédicats nominaux en français*, Genève:Droz

Gross, Gaston, 1989 *Les constructions converses du français*, Genève:Droz

Jespersen, Otto 1965 *A Modern English Grammar* London: George Allan & Unwin

Labelle, Jacques, 1983 Verbes supports et opérateurs dans les constructions en *avoir* à un ou deux compléments. *Linguisticae Investigationes* 7:2, pp. 237-260, Amsterdam/Philadelphia: J. Benjamins

Salkoff, Morris 1973 *Une grammaire en chaîne du français*. Paris:Dunod

Salkoff, Morris 1979 *Analyse syntaxique du français*. Amsterdam/Philadelphia: J. Benjamins

Silberztein, Max 1993 *Dictionnaires électroniques et analyse automatique de textes* Paris:Masson

Vivès, Robert 1983 Avoir, prendre, perdre: *constructions à verbe support et extensions aspectuelles*. Thèse de troisième cycle, LADL, Université Paris 7

Morphemic and Morphological Analysis of the Latvian Language

UĢIS SARKANS

Abstract

In this paper we describe a system of morphemic analysis of Latvian texts and outline further work aimed towards implementing morphological analyser. The basis for our system is regularity of the Latvian language in the sense of word formation. Morphemic analysis is based on morphotactic rules and simple lexicons of morphemes instead of extended traditional lexicons. We describe the process of segmentation, principal constituents of the system and the formal language used for rule description. The system is compared with some other formalisms of morphological analysis and their implementations.

1. Introduction

In this article we discuss a formalism for description of morphemic segmentation rules of Latvian and its implementation in the form of linguist's workbench. In the field of natural language processing it is a usual practice that performance and linguistic "plausibility" are at the opposite ends of scales: if the system is constructed with high performance in mind, it is very often *ad hoc* with descriptions far from being natural from linguist's viewpoint; and, if the formalism is natural, it is hard to achieve good performance. The Latvian language and its structure from word formation viewpoint seems to be, according to our investigations, suitable for efficient and at the same time natural description of morphemic segmentation rules.

The best approach for morphological analysis for languages that are not highly inflected, like English, is to store all possible word-forms in lexicon, or to use some pattern matching techniques to deal with common affixes. Of course, this applies only if the morphological analyser is the end result and there are no concerns about "the means", i.e., how the morphological descriptions look like. We have tried this method also for Latvian, using the first above-mentioned approach, i.e., storing all possible word-forms. In Latvian there can be about 100 graphically different word-forms of adjectives and, if we consider participles as word-forms of verbs, more than 200 for verbs in the Latvian language. We used a simple method for compressing the lexicon: for every word and all its word-forms the largest common prefix was found, and the remaining parts were stored in a mini-lexicon (c.f. [Koskenniemi 1983]). Thus we had a lexicon of "pseudo-roots" and many mini-lexicons of "pseudo-endings", allowing us to store the whole lexicon in a manageable space without penalty on retrieval time. We are using such morphological analyser in a monolingual Latvian - Latvian electronic dictionary to provide the feature of cross-referencing words from dictionary entries to the main word list.

Still such systems have several drawbacks, the main one being inability to cope with new words that are not present in the lexicon. If we want to link together a morphological analyser with a parser through providing the output of the morphological analyser as the input of the parser, this is an unacceptable situation, especially if the parser is not very robust.

The approach discussed in this paper is more profoundly rooted in the linguistic knowledge; we use the rules of word formation of Latvian.

Modern written Latvian is a comparably new language, and it is quite regular. First, it is regular in the sense of forming inflected forms. For the most part of Latvian words all inflected forms can be obtained from the base form and little supplementary information. Almost the only exception here is certain verbs that require infinitive, past and present stems in order to correctly generate all inflected forms.

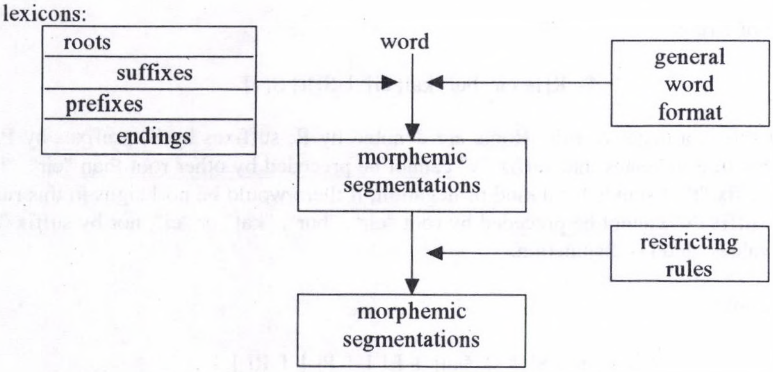
Second, our approach is heavily relying on regularity of the Latvian language in the sense of word derivations. Our basic source of linguistic information is [Metuzāle-Kangere 1985] that is one of only a few dictionaries of such kind that have been compiled in the world (an indirect evidence of Latvian having exceptionally regular derivational system). Our analyser does not use a lexicon in the traditional sense, i.e., a lexicon containing word stems together with some information necessary for obtaining all inflected forms. We have only lexicons of morphemes, and words are analysed down to the level of single morphemes, not just "stem + inflectional features".

What is very important for our approach of analysing running text without aid of a large lexicon, new words that appear in the language (usually of foreign origin) always are regular from the viewpoint of inflection formation. The group of verbs that require 3 stems (see above) is closed, i.e., new verbs never belong to it.

What is different in our approach compared to the widely known two-level morphology [Koskenniemi 1983]? At the heart of two-level morphology is phonotactics, while we because of our primary concern of morphemic segmentation needed mainly morphotactics. There are several implementations of Koskenniemi's formalism, among them PC-Kimmo [Antworth 1990]. We looked also at PC-Beta [Brodda, Karlsson 1980] that is a general purpose text processing tool. Both these tools require from linguist knowledge of finite state automata; we think that it is easier for linguists to think in terms of morphemes, their adjacency and so on, not in terms of states and automata head movements. Therefore our language for describing morphotactic rules was designed as easy to understand and use as possible without sacrificing generality.

2. Morphemic Analysis

The general schema of our approach to morphemic analysis of Latvian words:



The lexicons of roots, suffixes, prefixes and endings have been taken from [Metuzāle-Kangere 1985] and made computer-readable. The general word format of Latvian words (excluding rare compound words consisting of more than two roots) is

<prefix>* <root> <suffix>* [<ending>]

for single-rooted words and

<prefix>* <root> <suffix>* [<ending>] [<prefix>] <root> <suffix>* [<ending>]

for compound words (the possibility of zero or more occurrences of element A is denoted by A*, and optional elements are included in square brackets).

During the first phase of analysis our algorithm produces all possible morphemic segmentations of the word with the structure according to one of the rules and morphemes belonging to the appropriate lexicon.

During the second, more interesting analysis phase new segmentations are not generated; instead, all segmentations obtained during the first phase are validated against a set of morphotactic rules that filter out some of the solutions.

There are rules of general nature, as well as rules dealing with some specific prefixes/ roots/ suffixes/ endings included in the rule set that is used during the second phase. A significant part of rules are included in order to restrict segmentation of compound words (with 2 roots; compound words with more than 2 roots are not analysed by our algorithm). There are about 700 rules at the moment.

The rule language was designed to be both easy to learn and concise. Rules are ordered, and the ordering is important; segmentations are matched against rules always in that fixed order, and processing of one segmentation is interrupted after the first successful match. There are two kinds of rules possible, positive and negative ones. If a hypothetical segmentation matches a positive rule, it is accepted as a correct one. If a segmentation matches a negative rule, it is rejected.

An example of a rule:

$- R[!# cir; bur; kal; ei] | S[!t] S[v]

$- indicates that this is a negative rule. Roots are denoted by R, suffixes by S, prefixes by P and endings by E. This rule indicates that suffix "v" cannot be preceded by other root than "cir", "bur", "kal" or "ei" or suffix "t". ! stands for a kind of negation; if there would be no ! signs in this rule, it would state that suffix "v" cannot be preceded by root "cir", "bur", "kal" or "ei", nor by suffix "t". # indicates list of values, and | is disjunction.

Another example:

$- R[mē] S[!# sl; šan] | E[ ] | P[ ] | R[ ] | .

This rule states the only suffixes that can follow root "mē" are "sl" and "šan"; no endings, prefixes or other roots can follow it, and it can not be at the end of word.

P[3] matches all prefixes with length equal to 3, and S[>2] matches all suffixes with length greater than 2. S[MAX] matches only the longest suffix possible in the respective place.

A problem with such rules is that the rule-set quickly grows and becomes hard to manage, even with extensive commentaries. The task of compiling the rule-set is time-consuming and requires work of highly qualified linguists with deep understanding of diachronic and synchronic aspects of morphotactics.

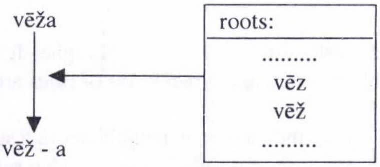
At the initial stage the work is productive, i.e., there are numerous cases of wrong morphemic segmentations discarded with introduction of every new rule. Later the progress in this sense stalls, there are more and more specific rules needed, even for single words. In principle, after adding each successive rule the rule-set should be tested on all previous examples as well, because interaction between rules is so complicated that it is virtually impossible for the rule set developer to imagine all possible consequences a single rule can cause.

At present we have achieved about 90% accuracy on "real" texts (newspaper and magazine articles). By accuracy we understand the ratio of words that were correctly segmented and the correct segmentation was the only segmentation obtained.

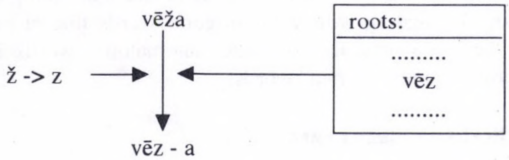
The main practical positive side of our approach is the ability to recognise new words as they appear in modern everyday language. New prefixes, suffixes and endings appear in the laŋguage very rarely, if they do at all. The only thing that appears often are roots, and our system can automatically raise hypothesis about new, unseen roots and with approval of the human linguist add them to the list of known roots. The other serious advantage over other methods is that the set of rules developed in the course of work can give valuable insights into morphotactics of the Latvian language because of the easily readable form of rules.

3. Phonemic Variations

Our system is heavily oriented towards morphotactics, and at this stage it is weak in the respect of phonotactics. Our first try at solving the problem of phonemic variations was storing several variations of the same morpheme in the corresponding lexicon. The analysis of, for example, word "vēža" (singular genitive of "a lobster") proceeded like this:

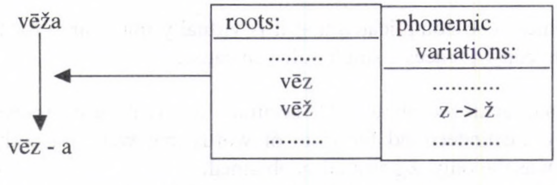


This approach was not satisfactory, both from the viewpoint of linguistic "preciseness" of the underlying formalism and from practical considerations. The next (present) implementation could be shown schematically this way:



Here phonemic variations are not regarded as separate morphemes, testing the possibility of a variation is built into the program.

We think that for Latvian the best approach is to introduce one more lexicon (besides lexicons of prefixes, roots, suffixes and endings) - the lexicon of phonemic variations. Then this very important aspect of language would not be hidden as in the current implementation:



We intend to regard phonemic variations in rules just like other "normal" morphemes. Then it will be necessary to change the rule of the general structure of Latvian words:

.... <root> [<phonemic variation>] <suffix>* ......

4. Morphological Analysis

At present our system performs only morphemic segmentation of Latvian words, it gives no information on possible morphological attributes. We intend to add morphological analysis rules in the form, e.g.,

<noun in genitive> =
= <noun in nominative> - <ending> [+ <phonemic variation>] + <genitive ending>

Together with morphotactic rules we already have such rule system will be adequate for accurate morphological analysis.

Another direction we are investigating is, how to add higher level, i.e., syntax rules and how they can aid morphological analysis. Here again two kinds of rules are applicable:

1. negative (e.g., words of what kind cannot be neighbours in a sentence);
2. positive (i.e., what is the structure of a well-formed Latvian sentence).

Such rules should considerably reduce the number of cases where morphemic segmentation/ morphological analysis is ambiguous, i.e., returns several possible solutions.

In order to decrease the number of rules (both morphotactic and morphological) we intend to add to our system a mechanism for dealing with very irregular words that at present need very specialised rules. It seems to be more natural to include "anomalous" words in some kind of sublexicon and treat them differently from "normal" words.

5. Using Some Ideas of Statistical Language Learning

Statistical language learning is a comparably new and promising field (see [Charniak 1993] for an overview), and there are some approaches we are investigating regarding morphemic and morphological analysis as well.

One of the possible applications of statistical language learning methods can be used for easing the rule development process. Statistics about adjacency of various morphemes can be collected from pre-segmented texts (segmentation performed by hand or by some other method). These statistics can serve for automatic preparation of the first version of the rule set, or for morphemic segmentation entirely based on statistics (i.e., without any rules).

The second idea could be used for reducing segmentation ambiguity caused by the fact that there is no information *a priori* attached to morphemes on possible word classes where morphemes can be used. Let us explain this idea on an example.

"zut-is" in Latvian means "an eel", and "zus-t" means "to get lost". Both roots "zut" and "zus" are contained in the root lexicon. Suppose the systems has to segment word "zuša". "zuš" is a phonemic variation of both "zut" and "zus", therefore an ambiguity arises. One solution would be to add some *ad hoc* rule dealing with these words. The other solution for such situations comes from statistical language learning. While analysing texts the system can remember which roots have been seen as constituents of which word classes. If there is a word "zut-im", (singular dative of "an eel"), it is obvious that "zut" here is a root of a noun. If there is a word "zus-t", "zus" is obviously a root of a verb. Now, if "zuša" appears in a place where only a noun is appropriate, most probably "zuš" here is a phonemic variation of "zut" rather than "zus".

6. Conclusion

The Latvian language with its high level of inflections on one hand and regularity on the other hand seems to be a very appropriate language for rule-based morphemic and morphological analysis, as well as for testing various machine learning ideas. It remains to see whether the approach reported here can be used for other languages as well.

Acknowledgements

The work described in this article has been carried out in the framework of a joint research project together with Baiba Kangere from Stockholm University who has been providing most of the linguistic insights; it was financed by Stockholm University. Several researchers of the Artificial Intelligence laboratory, lead by Andrejs Spektors, took part in both theoretical work and practical implementation, especially Baiba Krūze-Krauze, Sandris Imbovičs, Inguna Greitāne.

References

Evan L.Antworth. PC-KIMMO: A Two-level Processor for Morphological Analysis. Summer Institute of Linguistics, Dallas, Texas, 1990.

Benny Brodda, Fred Karlsson. An Experiment with Automatic Morphological Analysis of Finnish. Institute of Linguistics, University of Stockholm, 1980.

Eugene Charniak. Statistical Language Learning. The MIT Press, 1993.

Kimmo Koskenniemi. Two-level Morphology: A General Computational Model for Word-Form Recognition and Production. Academic dissertation, Helsinki, 1983.

Baiba Metuzāle-Kangere. A Derivational Dictionary of Latvian. Helmut Buske Verlag, Hamburg, 1985.

Towards an Active Dictionary for Upper Sorbian

JANA SCHULZE – IRENE ŠĚRAK – EDUARD WERNER

Abstract

Because of the waxing need for a monolingual dictionary for Upper Sorbian the project presented here has been started. The aim is an active dictionary with about 20,000 entries usable as an all-purpose dictionary providing synonyms, idioms, and stilistic hints. This will be the first monolingual dictionary for Upper Sorbian.

Introduction

Sorbian is the smallest Slavonic language with about 60,000 speakers. The language situation of Sorbian with regard to the dictionaries has been characterized in WERNER 1994. Since then, the need for a monolingual Upper Sorbian dictionary has been emphasized by various people and institutions (schools, writers, linguists). A monolingual dictionary project is being worked on since the start of this year and shall be presented here.

The Corpus

The text corpus we retrieve our examples from is a compilation of classical Sorbian literature and dictionaries that are currently being scanned and read by OCR-software.

To retrieve examples we use common UNIX-tools like grep, agrep, perl, etc. to search with regular expressions. The texts are not tagged since there's no full implementation of Sorbian morphology yet that could be used to fulfill this task. This will change in the long run because a tagger would allow us to detect words in texts that have no corresponding entries in the existing dictionaries.

A Synonymic Dictionary

A crucial point of a dictionary that wants to provide synonyms is how to provide them. To justify our approach we will shortly compare two synonymic dictionaries:

The approach of the DUDEN 1986 aims at the native speaker who just has to be reminded of synonyms he already knows at least passively. Therefore, no explanations whatsoever are given. This approach has the merit of being very concize, but the correct usage of the synonyms depends totally on the competence of the user.

The ACTIVATOR 1994 on the other hand aims at the advanced student who is not a native speaker. He has to be told in extenso about the usage and restrictions of a given expression (compare also KILGARRIFF 1994).

A synonymic dictionary for Upper Sorbian is in the first place a synonymic dictionary for native speakers. On the other hand, the language situation implies—due to the influence and omnipresence of German—uncertainty about native synonyms as can be shown by the following example: *dać* and *wostajić* are both equivalents for 'to let'; while *dać* is 'to let/allow someone do s.th.' *wostajić* is more 'to let s.th. be'. We have the following possibilities:

Daj	jemu	rěčeć
to let$_{imp}$	he$_{Dsg}$	to talk$_{inf}$

Wostaj	to	ležo
to let$_{imp}$	that$_{Asg}$	to lie$_{ger}$

There is, however, a contact zone of *Wostaj*$_{imp}$ *jeho*$_{Asg}$ *ležo*$_{ger}$ 'let him lie' and *Daj*$_{imp}$ *jemu*$_{Dsg}$ *ležeć*$_{inf}$ 'let him lie' which leads to contaminations like *Wostaj*$_{imp}$ *jeho*$_{Asg}$ *rěčeć*$_{inf}$ 'let him talk'.

Since words and idioms can often be used synonymously a dictionary of synonyms should contain idioms and proverbs as well (as does the ACTIVATOR 1994). For Sorbian the providing of idioms and proverbs is especially important since code-switching and word-for-word translations are very common when it comes to proverbs and idioms. To give an example:

In	der Nacht	sind	alle	Katzen	grau
in	night$_{Dsg}$	to be$_{3pl}$	all$_{Npl}$	cats$_{Npl}$	grey$_{Npl}$
W	nocy	su	wšě	kóčki	šěre
in	night$_{Lsg}$	to be$_{3pl}$	all$_{Npl}$	cats$_{Npl}$	grey$_{Npl}$

Apart from the government (the preposition *w* governs the locative in Sorbian which doesn't exist in German), the structures are identical. Nonetheless, the Sorbian idiom is quite different but commonly unknown among the young(er) generation(s):

Howno	wěš,	što	póćmje	jěš,	hač	mušku	abo	rózynku/pawka
shit$_{Asg}$	to know$_{2sg}$	what$_{Asg}$	after dark$_{adv}$	to eat$_{2sg}$	whether	fly$_{Asg}$	or	raisin$_{Asg}$/spider$_{Asg}$

Another for us crucial point of traditional monolingual dictionaries along the lines of the Oxford Advanced Learner's Dictionary is that they can't activate words. A word unknown to a student must first occur in a text before it can be found and learned. The concept of the ACTIVATOR 1994 has an edge here, too.

So we regard the approach of the ACTIVATOR 1994 to be principally appropriate for our goals and adopted its concept of *key words*, *sections*, and so on. To serve as an all-purpose dictionary, however, the concept has to be extended. There will be no entries, however, that do not belong to a concept which is the main difference to the approach of LONGMAN 1995.

School/Learner's Requirements

A monolingual dictionary to be used at school has to be a full equivalent of the existing bilingual Sorbian-German school dictionaries, i. e.:

- it has to cover the same scope (all-purpose lexicon)
- it must have about the same size (about 20,000 words)
- it must contain grammatical information; we use the number code and the abbreviations of the existing school dictionaries as references to paradigms and to provide information about gender and aspect

Furthermore, phonetic transcription is desirable since in some case the pronounciation is not obvious. Sorbian speakers tend to pronounce some words in a funny way in official situations because there is no standardized pronounciation for Sorbian and so they pronounce it "the way it is spelled". So we decided to include phonetic transcription. Although this is common for school dictionaries for English or French, it is not an obvious decision for Sorbian as can be seen from the fact that our project is the first Sorbian dictionary to do that. Because of the lack of a standardized pronounciation our phonetic transcription is only a hint of how to pronounce a word in an unmarked, "normal" way. The phonetic transcription is analogous to the transcription used in school for English and French.

A sample entry for *z drapawku sčesać* 'to comb with a thistle (= to straighten s.o. out)' looks like this (the explanatory text is translated below):

z drapawku (z hrabjemi) sčesać (někoho) ['zdrʌpaʊku ('zrʌːbˌɛmˌi) 'stʆɛsatʆ] 39 *p*: Někomu doraznje njekorektne abo njezamołwite zadźerženje porokować. Wuraz woznamjenja wuslědk čina a njemóže so z wurazami kaž *dwě hodźinje, dołho* a pod. wužiwać, kiž poznamjenjeja dołhosć čina. *Wón wšak bě do toho přišoł, zo so ta knjenina rozprawa na njeho měri, zo by z drapawku sčesany był.* (Radyserb)

To criticize s.o. because of incorrect or irresponsible behaviour. The expression emphasizes the result of an action and can't be used with expressions like *two hours, for a long time* and so on which say something about the duration of the action.

Stilistic advice and examples from literature is also something missing in school dictionaries that should be included. In the examples especially such forms shall be shown that are known to be difficult for native speakers, such as homonyms, little used, and irregular forms. Words and expressions that are error-prone are marked with a dangerous bend sign; so e. g. the verb *boleć* 'to hurt' is often used with dative government instead of the accusative:

☡ **boleć** ['bɔlɛtʆ] 37 *ip*: Kedźbu na rekciju; *boleć* sej žada akuzatiw, nic datiw: *Nětko woni ćahnu, moji křižerjo! Ow, kak* ***mje to boli****, hdyž tole haleluja słyšu.* (Ćišinski) | *Mój žiwjenja chód krasny je, wšak nimam strowja tradać, stawčk* ***njeboli*** *nic žadyn* ***mje****; što chcu sej wjace žadać?* (Zejler)

Pay attention to the government; *boleć* requires the accusative case, not the dative [the NPs_{acc} are printed in bold face].

To enable the user to find something better than a common poor expression (that might be the only one he knows) he should most definitely not use such expressions are marked with a stop sign and other solutions are suggested; our example is a quite unacceptable calque of the weird German derivation *andenken* 'to think about s.th. a little bit, but not thoroughly':

STOP **namyslić** ['namɨslitʃ] 38 *p*: Słowo ma jenož woznam *sej něšto wumyslić*, přir. *nabać, napowědać, naręčeć*. Pod wliwom němskeje twórby *etwas andenken* wužiwa so wot 90ych lět též w serbskich žurnalistiskich tekstach. Město *smy to namyslili* praj radši *smy wo tym trochu přemyslowali/rozważowali/ smy to wobmyslowali*. Imperfektiwny aspekt tutych słowow hižo signalizuje, zo njeje so hišće doskónčny wuslědk docpěł.

The word means only *to make s.th. up*, compare *nabać, napowědać, narěčeć*. Under the influence of the German derivation *etwas andenken* it is used since the 90s in Sorbian journalistic texts. Instead of *smy to namyslili* say better *smy wo tym trochu přemyslowali/rozważowali/smy to wobmyslowali*. The imperfective aspect of these expressions already signalizes that the action has not been brought to a final result.

Names

First names, names of animals, cities, villages, rivers, mountains, and lakes shall be integrated into this concept. They have to be in the dictionary for the following reasons:

- the school dictionaries contain them
- the derivation of adjectives and names of inhabitants of villages is sometimes difficult
- there is no source of information e.g. how to typically name a cat it Sorbian

So mountain names will be integrated in a concept *mountain* with the sections *mountain*, *parts of a mountain*, *names of mountains*, *mountain range*, *names of mountain ranges*, *hill*, and *names of hills*. Into the sections *names of* ... we will enter especially the names of hills and mountains in Upper Lusatia with a short description where the hill or mountain can be found.

Affixes

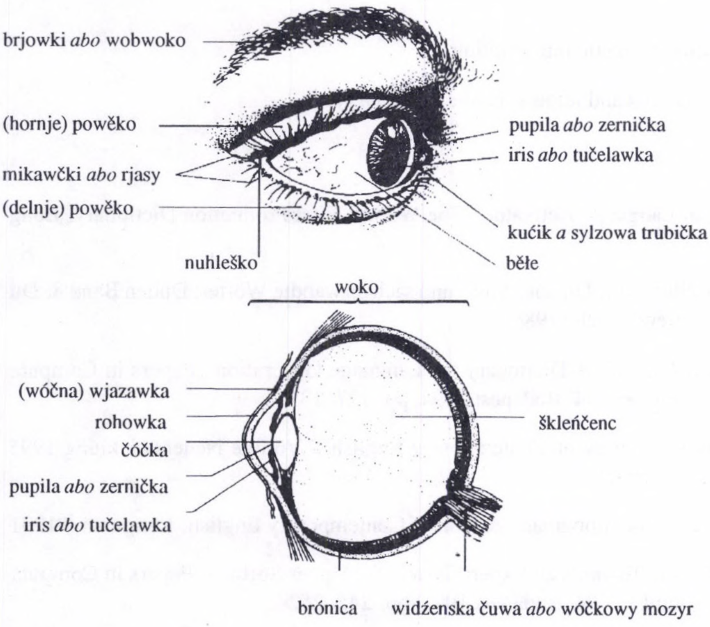
Due to the limited size of our dictionary we have decided to integrate morphemes and morpheme complexes. We will concentrate on indigene Sorbian morphemes, not on international morphemes like *kontra-* 'counter-' and similar. Affixes will be integrated into the concepts according to their semantics, so suffixes for deriving place names will be found in a section of the concept *place*.

Pictures[1]

For many items—especially from the world of science—the German terma are much more commonly known than the Sorbian equivalents (that might not even exist). To explain such items pictures will be used. These pictures always belong to a section of a key word, never to a single entry only; the items on the pictures are explained and the entries are ordered alphabetically. Non-alphabetical ordering—as e. g. in MCARTHUR 1991–strikes us as user-unfriendly since frequency seems an inappropriate criterion when you have already found the word and just look up the definition to be sure. Our example is the second section from the concept *eye*:

[1] We don't have any proper pictures yet. For the purpose of this article the pictures have been taken from MCARTHUR 1994.

2. Dźěle wóčka



běłe ['bɨəʊɛ] 21 *n*: Běły dźěl wóčka wokoło tučelawki.
brjowki ['brˌoʊkˌi] 8 *plt*: Kosmički, kiž w formje wobłuka nad wóčkomaj rostu. *Wona měješe njewšědne kuzłotne čorne brjowki (na njewšědnych čornych brjowkach sym so pozdźišo hišće časćišo popadował), wobličo brune kaž zrału brěšku a powabne běłe zubički.* (NOWAK-NJECHORŃSKI)
brónica ['brʊnˌitsa] 9 *f*: Kóžka w zadnim dźělu woka ze swětłočućiwymi čuwowymi bańkami, kiž optiske signale přez wóčkowy mozyr k mozam dale wjedu.

. . .

Summary

The approach of ACTIVATOR 1994 is quite appropriate for Upper Sorbian as far as our goals are concerned. But the concept of the ACTIVATOR 1994 has to be extended to suit our needs since we definitely need an all-purpose dictionary. These extensions are:

- providing of "normal" words as well as idioms, proverbs, and synonyms
- including of non-autosemantic entries (prefixes)
- extensive grammatical information; especially critical, i. e. error-prone forms are to be included in the examples
- pictures for explaining technical and similar items

- icons for discouraging usage or drawing attention to difficulties

As far as Sorbian is concerned this project is an innovation with regard to the following aspects:

- first monolingual dictionary for Sorbian (apart from small studies for academical purposes like WERNER 1994)
- first dictionary providing phonetic transcription
- first dictionary for synonyms and idioms (active dictionary)

References

ACTIVATOR 1994 Longman Language Activator : The World's First Production Dictionary. Longman ²1994

DUDEN 1986 Wolfgang Müller (ed.): Duden : Sinn- und sachverwandte Wörter; Duden Band 8. Dudenverlag Mannheim/Wien/Zürich ²1986

KILGARRIFF 1994 Adam Kilgarriff: A Dictionary for Language Generation. Papers in Computational Lexicography CompLex '94, Budapest 1994, pp. 127–135

LONGMAN 1995 Longman Dictionary of Contemporary English : Völlige Neuentwicklung 1995. Langenscheidt-Longman ³1995

MCARTHUR 1991 Tom McArthur: Longman Lexicon of Contemporary English. Longman ¹⁶1991

WERNER 1994 Eduard Werner: Towards an Expert System for Upper Sorbian. Papers in Computational Lexicography CompLex '94, Budapest 1994, pp. 245–252

Creating a Morphological Dictionary of Bulgarian Language

KIRIL IV. SIMOV – DIMITAR G. POPOV

Abstract:

A methodology for acquisition of morphological rules and morphological dictionary from machine-readable dictionaries with the help of minimal initial morphological grammar is`presented. The methodology includes the following stages: extracting of the relevant information from the available machine-readable resources; explication of the hidden grammar information; grammatical generalisations over the data; checking the results. A set of software modules supporting the methodology have been developed: a parser of the dictionary entries; an explicator of the morphological information with respect to the initial grammar; a generalisator of the grammar data; an editor of the lexical information.

1. Introduction

In this paper we are concerned with methodology for extraction of morphological data about the inflectional morphology of the Bulgarian language from machine-readable dictionaries. In what follows when we are using morphology we understand inflectional morphology. According to [Ritchie et al., 1992] the inflectional morphology is characterised by the following features:

- Preservation of a category: the words are grouped over a bunch of common grammatical categories and common semantics (usually carried by the common stem). These categories are the same for all members (called word forms) of the group (called lexeme). For instance, the noun "father" will be still noun in plural form --- "fathers";
- Systematic: the way how the structure of the paradigm's member is obtained is the same for a large class of lexemes (the same rule applies to formation of the plural form of a given class of nouns);
- Productivity: the new words in the language inherit the whole paradigm from the corresponding class.

These characteristics of the inflectional morphology are widely used in the linguistic theory and practise and particularly in the construction of different kinds of dictionaries for human use. One of the main assumptions is that the users of such kind of dictionaries are able to recognise and produce each word form of each lexeme in the language.

The morphology as a computer module for identifying and/or synthesising of any particular word form in the speech process is an obligatory part of any system for processing of a natural language with claim to cover a wide variety of language phenomena.

Here presented process of preparing a morphological data of Bulgarian language (a formal morphological grammar and the corresponding dictionary) is based on the following sources of information: **(1)** A morphological grammar - a description of the structure of the word forms in Bulgarian language and the Bulgarian inflection; **(2)** A dictionary of Bulgarian language in machine-readable form; **(3)** An orthographic (spelling) dictionary of contemporary Bulgarian language in machine-readable form. It should be mentioned here that all three sources of information are originally designed and worked out for human use only, which means that a great part of the morphological information, needed for the automatic analysis and synthesis is only implicit in their different parts.

An important aspect of the creation of the morphological dictionary is its orientation to human user, because of which, together with the explicate manifestation of the entire morphological information for every word, is realised also the possibility of a strictly formal classification of all lexical units in the dictionary with a view to its easy use by man.

There are two approaches to the creation of a morphological dictionary: (1) a hand encoding of the class for each lexeme; (2) an automatic extraction of necessary information for automatic classification of the lexeme. In the first case a relatively full description of the inflectional morphology is needed and the linguist determines for each lexeme (possibly with appropriate software tools) to which class the lexeme belongs. This approach was utilised in the creation of the MORPHO-ASSISTANT system — [Simov et al., 1990], [Simov et al., 1992], [Paskaleva et al., 1993]. The main problem here is that the initial morphological grammar has to be full, otherwise some lexemes will be impossible to be classified properly. In the second case it is not necessary to define a full morphological grammar, it is enough to give only a minimal one and provide appropriate means for valuable linguistic generalisations over the available morphological data. The main problem in this approach is to find sources containing enough information and the second problem is that some of these sources of information may not exist in machine-readable form. In our view depending on the situation in some case it is better to create a machine-readable form of missing resource(s) than to encode the whole information manually.

2. Sources of Information

2.1. Morphological grammar

The underlying morphological grammar (in our case the morphology part of the introduction to the yet unpublished Learners' dictionary of Bulgarian language, compiled by D. Popov [Popov, forthcoming]) represents a full description of the system of Bulgarian inflection. For all the inflected parts of speech are formulated here general principles of the inflection and rules of generating the different grammatical forms of all the words of a particular class are fully described, after which patterns of the inflection classes of words are presented.

These classes are formed on the principle of unifying the different lexical units on the basis of one or more rules of generating their different word forms and all the elements of the paradigm. On these lines, for example, we have particular rules for generating singular and plural forms of the noun and on the basis of this there could be different classes of nouns, which form their plural forms in one and the same way but they have different forms in singular.

Other criteria for classification of lexemes, besides the group of affixes, could be the number of syllables and the position of the stress on different parts of the paradigm.

2.2. Dictionary of Bulgarian language (one-volume Bulgarian to Bulgarian explanatory dictionary)

The dictionary of Bulgarian language we have chosen, in which about 60_000 words are explained, is a reference book, full enough for our purposes [Andreichin et al., 1994]. The entry provides grammatical, stylistic and semantic information for the explained words according to the basic principles of the traditional lexicography. The relevant to the morphology information is situated in strictly defined places in the structure of the dictionary entries. The possible combinations of the grammatical information given on the different locations in one•entry are determined by the standard rules. This information is in italic and underline style in the following examples:

мàйка *ж*. **1.** Жена по отношение на своите деца. *Майка на две деца*. ...

нов *прил*. **1.** Който е направен, купен, станал, явил се е или е възникнал ...

воèнен 1. *прил*. Който се отнася до войска или война. ....

отѝвам *нсв*., **отѝда** *св*. *нпрх*. **1.** Вървя, движа се ... **отѝва ми** *нпрх*. Прилича ... —_**си** *нпрх*. **1.** Напускам ...

The last entry contains grammatical information for the lexemes:

отѝвам *нсв*. *нпрх*.

отѝда *св*. *нпрх*.

отѝвам си *нсв*. *нпрх*.

отѝда си *св*. *нпрх*.

отѝва ми *безл*. *нсв*. *нпрх*

Notice that a part of the grammatical information is not represented explicitly in the dictionary.

2.3. Orthographic (spelling) dictionary of Bulgarian language

Authors of this reference book ([Georgieva et al., 1983]) had based their work on the assumption that all the users of the dictionary have Bulgarian as a mother tongue or know the language well enough to speak it fluently and have in their minds a structure of the basic rules and material elements of the inflection and are able to recognise the grammatical categories and the meaning of words. Bearing in mind this virtual scheme, at every entry of the dictionary they have presented, beside the basic form of the particular word, some minimal additional information, which is estimated as necessary and enough for the user of the dictionary to supplement the missing part of the paradigm. E. g.:

мàйка

нов

воèнен, -нна

отѝвам (си), -аш

отѝда (си), -еш; отѝдох, отѝде, отѝшъл, -шла, отѝшли; отѝдех, отѝдел

It should be mentioned here that the set of classes of lexemes, described in the morphological grammar (see 2.1. above), could be rather different from the one the authors of the spelling dictionary have had in mind. That is why the two schemes could in some cases differ from one another. Another problem is that the classes of lexemes are not represented explicitly in the dictionary. It is not straightforward task to answer questions like: "Are the lexeme A and lexeme B belonging to the same class?" or "Which lexemes belong to a particular class C?".

3. Stages of Processing

The goal is to create a correspondence between the implicit classes of lexemes in the spelling dictionary of Bulgarian language and the explicit classes of lexemes described in the morphological grammar and to connect each lexeme in the dictionary. There are two possible decisions of this problem.

The first decision comprises the following steps: (1) building a full formal representation of the morphological grammar; (2) creation of a set of rules which depending of available information for each lexeme in the spelling dictionary determines the right class to which this lexeme belongs. This decision suffers from the same shortcomings as the MORPHO-ASSISTANT approach: the morphological grammar has to be full. Another problem is the complexity of the system of rules. The creation of such a system of rules is a time consuming and error pruning task.

The second decision includes a simple initial morphological grammar and clear rules for linguistically motivated generalisation over the available data. The implicit information is made explicit in several steps so that each step is simple and the result of each step can be checked easily for its correctness. Some of the intermediate results may have their own value. The final result is a set of automatic generated classes of lexemes connected explicitly with the corresponding lexemes. Of course, the so created classes will be different from the initially defined classes in the morphological grammar, but now the building of correspondence between the two sets of classes is much easier task because the both sets are represented explicitly.

In our case the whole process includes the following stages:

- Extraction of the relevant lexeme information from the Bulgarian to Bulgarian explanatory dictionary;
- Combining of the information in the spelling dictionary with the extracted information from the previous stage;
- Creation of new lexical items that contain explicitly all morphological inflectional knowledge. On this stage we can automatically generate the whole paradigm of each lexeme. This information is also enough for building of a morphological analyser;
- Linguistically motivated generalisation over the new lexical items. Creation of classes of lexemes and classification of the lexemes with respect to them. Checking the result;
- Building of correspondence between the new classes of lexemes and these described in the morphological grammar, to produce a Morphological dictionary of Bulgarian language for human use;
- Building of a morphological computer module that is able to automatically analyse and/or generate arbitrary Bulgarian word form.

3.1 Parsing of lexical entries in the Bulgarian to Bulgarian explanatory dictionary

The goal on this stage is an automatic extraction of the relevant grammatical information from the lexical entries in the dictionary. We built a small parser for the entries of the dictionary using the DCG facilities of the PROLOG. The information that the parser can recognise is the typeset information (the different fields in the dictionary entries are marked as **bold** or *italic*, see the examples above), abbreviations of the grammatical categories, and some special means to mark out lexical information inside the lexical entry.

3.2 Combination of the information from the two dictionaries

This task is very easy, it simply matches the main words of the entries of the two dictionaries and produces new entries of the following kind:

мàйка *ж.*

нов *прил.*

воѐнен *прил.* , воѐнна

воѐнен *м.*

отѝвам (си) *нсв. нпрх.* , отѝваш

отѝда (си) *св. нпрх.* , отѝдеш; отѝдох, отѝде, отѝшъл, отѝшла, отѝшли; отѝдех, отѝдел

3.3 Explication of the Morphological Knowledge

Our goal at this stage is to create new lexical items containing explicit morphological information which is enough to generate and analyse each word form. Here we use a formal representation of the rules in the morphological grammar in order to extend the entry information from the previous stage with all relevant morphological data. The importance here is the explicitness of the data but not the parsimony of the representation so the result contains a lot of redundant information and very little morphological generalisations. To keep things simple, all problems concerning the alternations in stems are left implicit in the lexical items.

3.3.1 Formal Account of Morphological Knowledge

We use a formal word grammar the rules in which have the following form:

$$\mathrm{WForm[LexInf,WFGrF] \text{ -> } Stem[LexInf,WFGrFG]\ Affix_1[GrF_1,AN_1]...Affix_n[GrF_n],AN_n]}\ (1)$$

where LexInf represents the lexeme information and it is shared by the word form and the stem. This ensures that the stem and the word form belong to the same lexeme. WFGrF represents the grammatical information, concerning a particular word form. WFGrFG is a generalisation over the grammatical information for several word forms in the paradigm of the lexeme and it guarantees that the right stem is chosen in the case of alternation. WFGrFG and WFGrF may be the same in which case the stem is appropriate only for this word form. GrF_1,...,GrF_n represent the grammatical information connected with the corresponding affix. AN_i is a mnemonic name of the corresponding affix and it is a part of lexeme information LexInf. In this way the right affixes for a given lexeme are determine. For some affixes the name is not specified because they are appropriate for the whole class of lexemes. Appropriateness of some affixes is depending not on the concrete lexeme, but on the other affixes in the rule. In order to capture these dependencies, a part of the information connected to the each affix ensure the agreement between the affixes. The grammatical information WFGrF is determined uniquely by WFGrFG and GrF_1,...,GrF_n.

Here is an example:

WF[[n,masc,BF],[pl,def]] -> S[[n,masc,BF],[pl]] A_1[pl,BF_1,Ag]A_2[pl,def,Ag]

where n stands for noun, masc for masculine, BF represents the common base form and the list of appropriate affixes, BF_1 is the mnemonical name of the affix A_1 and it is a member of the list in BF, pl stands for plural, def for definiteness, and Ag represents the agreement information between the affix for the plural forms and the plural definite article. The information connected to the stem ensures that the right stem and the right affix for the plural forms are used. As it is clear from the example the same stem is using for define and indefinite plural forms.

We also have rules of the following forms:

WordForm[LexInfo, WFGrF+GrF] -> WordForm[LexInfo, WFGrF] Affix[GrF]

which rule out some word forms that are not relevant for the determination of the lexeme class. For example, some of the participles take the whole paradigm of the standard class of adjectives, but only four of them are relevant to the classification.

Additionally, the grammar contains a list of affixes. The entry for each affix includes grammatical information, a mnemonic name and agreement information. Some examples follow:

"те", [["plural","definite"],"те","еи"]
"ове", [["plural"],"ове","еи"]
"л", [["participle","past","active"],"л","я"]

As it was mentioned on this phase of processing, we don't have any rules for the alternation of stems. If some word form is constituted by means of an alternated stem then this stem is stored in the lexical entry with an appropriate information (see WFGrFG above). Sometimes a group of word forms in the paradigm of a lexeme share the same alternated stem. In this case the common stem is recorded only once. In this way, the paradigm of each part of speech is divided in parts. For example, the paradigm of Bulgarian noun is, naturally, divided in *singular*, *plural*, *vocative* and *count* part.

The lexical entries in the dictionary here have one of the following formats:

Base_Form, LexInfo[LGR,BFS,AffixList], StressMoving, StemAlternation (2)
Base_Form, LexInfo[LGR,BFS], [ParadigmPart_1, ...,ParadigmPart_l]

where Base_Form is the base form for a given lexeme, LexInfo is the lexeme information which includes LGR - the lexeme grammatical information, including the stress position; BFS the stem of the base form; AffixList is a list of mnemonic names for the affixes that are appropriate for a given lexeme. The order of the mnemonic names is in correspondence with the structure of the paradigm of a given class of lexemes.

StressMoving represents the new positions of the stress for the members of the corresponding part of paradigm. The possible values are: an empty list when there are not changes in the position of the stress with respect to the base form; or, a list of pairs GrCh:StressPos where GrCh represents the grammatical characteristics, determining for which word forms the moving of the stress has place and StressPos is the new position.

StemAlternation is an empty list if there are no alternations in the stems for the specified part of the paradigm, or it is a list of pairs GrCh:AlternatedStem where again GrCh represents the grammatical characteristics, determining for which word forms the alternated stem is appropriate, AlternatedStem is the alternated stem. Each part of the paradigm has the same members for each lexeme of a given class of lexemes and it is recognised by its position in the above structure. In the second format above each ParadigmPart$_i$ has the following format:

GrCh:Stem$_i$:StressPos:StressMoving:StemAlternation (3)

where GrCh characterises the corresponding part of the paradigm, Stem$_i$ is the stem for this part of the paradigm, StressPos is the related stress position, StressMoving and StemAlternation has the same format and meaning as it is describe above, but with respect to the given part of the paradigm.

Here are some examples:

заплес, [[n,masc,1], заплес, ['a', 'и', 'a':0, 'e':0]], [], [].
бялка, [[n,fem,1], бялк, ['и', 'o' : 0]], [], [[pl] : белк].
букна, [[v,impr,intr,1], букна],
[[pres] : букне : 1 : [] : [], [aorist] : букна : 1 : [] : [], [past_imp] : букне : 1 : [] : []].
блея, [[v,impr,intr,1], блея],
[[pres] : блее : [] : [], [aorist] : бля : [] : [[pl] : бле, [partis] : бле], [past_imp] : блее : [] : []]

[[pres] : блее : [] : [], [aorist] : блея : [] : [[partis] : блее], [past_imp] : блее : [] : []].

The last example shows the way in which we treat the doublets. The mark :0 after some of affixes in the first two examples means that the corresponding word forms are only theoretically possible like the plural form of uncountable nouns in English.

3.3.2 Process of Explication

We start with the lexical entries obtained after the combination of the information from the two dictionaries at the stage 2:

BaseForm, GramChar, ListOfWordForms

The process of explication consists of several steps:

1. On the base of grammatical information GramChar, the program determines the sort of the new lexical entry (see (2) above). It gives us the parts in which the paradigm of the lexeme is divided and the set of the possible rules. For instance, if the lexeme is noun, masculine then we expect first to find the singular definite word forms, afterwards the plural indefinite form, following by the count and vocative forms. The rules from the grammar are ordered according to this expectation.

2. In the next step, the program tries to analyse the word forms listed in ListOfWordForms, using the rules. The program starts with the word form given as a string and a guess about the grammatical characteristics of the word form made on the base of grammatical characteristics of the corresponding part of the paradigm. The result is a segmentation of the word form into one or more affixes and the right word form stem.

3. On the base of analyses of all word forms, the program constructs a lexical entry in one of the two format given in (2) above. This means that the corresponding list of affixes is create together with lists of stress moving and alternated stems or the value for each part of the paradigm is determined.

In the case when for some part of paradigm no word forms are listed a default rule is applied. Any default rule comprises two parts - Condition and ParadigmPartCharacteristics. Condition determines when the rule can be applied. The condition includes such characteristics as: grammatical information, the length of the base word in syllables, the stress position, and the grammatical characteristics of the corresponding part. ParadigmPartCharacteristics is the default value for this part of the paradigm. When for some part of the paradigm several rules can be applied, the one with the most specific conditions is chosen.

The resulting lexical entry contains enough information to generate the full paradigm of the lexeme without the help of default rules and therefore it is possible one to construct a morphological parser of Bulgarian.

3.4 Linguistic motivated generalisation

After the process of explication we have a morphological dictionary that demonstrates the right morphological information for each lexeme, but it still contains a lot of implicit information. It is still impossible to answer questions like the mentioned above ones: "Which lexemes are from the same word class as the lexeme A?" Our next step to ensure this possibility and to give a more explicit classification of the morphological knowledge of Bulgarian language.

As a model for treating of the alternations in the stems we choose a model of rules very close to the Two-level model [Koskenniemi, 1983], [Trost, 1991]. Our rules actually are not in exactly the same format of the two level rules, but they are good for (1) a morphological classification, the task we have in hands, (2) a good starting point for creation of real two level rules for Bulgarian Language. The rules have the following format:

NameOfRule, ListOfChanges

where Name is the name of the rule and ListOfChanges is a list of the alternations for the given rule. The format of each alternation is as follow:

lStr -> SStr, Condition

here lStr is a string on the lexical level, SStr is the corresponding surface string and Condition is a description of the grammatical information when a given alternation takes a place. The lexical entries have now a changed format:

Base_Form, LexInfo[LGR,BFS,AffixList], StressMoving, NameOfRule (4)

the meaning of the fields are the same as in (2) above, the only difference is that the list of alternated stems is replace by the name of an appropriate rule.

It is important that the rules are generated automatically in one optimal way so that there are no conflicts among the alternations that belong to the same rule.

The all morphological information in the new lexical items allows a meaningful linguistic classification. A set of classes of lexemes according to this information is built. We developed a database over the lexical entries of the whole dictionary. The database is connected with the formal grammar so the developers can easy check the result from the first three stages and also to classify the lexemes with respect to different criteria and in this way to produce dictionaries with different formats of the lexical entries, according to their intended use. In case of errors in some entry it is possible the whole information connected with this entry to be edited and afterwards the lexeme is classified with respect to the new information.

3.5 Towards a Morphological Dictionary for Human Use

Connecting the formal description of the word classes that are result of the whole process with the human oriented descriptions in the morphological grammar (section 2.1 above), we are creating a morphological dictionary of Bulgarian Language.

The morphological dictionary in its human user oriented form will be an enlarged, enriched and highly improved version of the existing spelling and orthoepic dictionaries of Bulgarian language put together. It will provide full and comparatively easy access to information about the orthographic and orthoepic characteristics of all the words and their forms. It could be published as a book and could also be used in a computer-readable form.

3.6 Automatic Morphological Processing

The results from different stages can be used for construction of software modules for automatic processing of Bulgarian morphology.

The first possibility is a compilation of a full form dictionary in which the lexical items have the following form:

WordForm, LexInfo, WFGrF, CorpusTag

where LexInfo is a pointer to the lexeme information, WFGrF is word form grammatical features, CorpusTag is a tag for corpus tagging. This module will be used for corpus tagging, but if the lexical information is enriched then it can be used as a morphological component in other systems.

A modified variant of this is a dictionary of the alternated stems with a set of appropriate rules for generation and analysis.

At the end a module along the lines of the Two-level model is under development.

4. Conclusions and Future Work

We presented a methodology for extracting of morphological information from machine-readable sources. The final and some of the intermediate results have there value both for automatic processing of the natural language and for using by human beings.

We consider the result of our work not as a final product but as a resource for further development. We plan to use it for investigations of the Bulgarian derivational morphology and in a system supporting dictionary making process as a morphological software module and as a base list of lexemes.

References

[Andreichin et al., 1994] Andreichin L. et al., *Dictionary of the Bulgarian Language*. 4th revised edition, prepared by D. Popov. Sofia, 1994.

[Georgieva et al., 1983] Georgieva El. et al., *Orthographic Dictionary of the Bulgarian Language*. Sofia, 1983.

[Koskenniemi, 1983] Koskenniemi K., *Two-level Model for Morphological Analysis*, IJCAI-83, 683-685, Karlsruhe, Germany, 1983.

[Paskaleva et al., 1993] Paskaleva, Elena, Kiril Simov, Mariana Damova, Milena Slavcheva, *The* *long journey from the core to the real size of large LDB*. In Proc. of a Workshop: Acquisition of Lexical Knowledge from Text. (Branimir Boguraev and James Pustejovski eds.). Ohio State University, Columbus, Ohio, USA. 1993.

[Popov, forthcoming] Popov, Dimitar, *Learners' dictionary of Bulgarian language*, forthcoming.

[Ritchie et al., 1992] Ritchie, Graeme D., Graham J. Russell, Alan W. Black, and Stephen G. Pulman. *Computational Morphology: Practical Mechanisms for the English Lexicon*, A Bradford Book, The MIT Press, Cambridge, Massachusetts, 1992.

[Simov et al., 1990] Simov, Kiril, Galia Angelova and Elena Paskaleva. *MORPHO-ASSISTANT: The proper treatment of morphological knowledge*. In Proc. COLING'90, vol.3, 453 - 457. 1990.

[Simov et al., 1992] Simov, Kiril, Elena Paskaleva, Mariana Damova, Milena Slavcheva, *MORPHO-ASSISTANT - a knowledge based system for Bulgarian morphology*. Demo description, Third conference on Natural language application, Trento. 1992.

[Trost, 1991] Trost, Harald, *X2MORF: A Morphological Component Based on Two-Level Morphology*. DFKI Research Report RR-91-04, Saarbrücken, Germany, 1991.

The formalization of collocations for Natural Language Processing: the Syntagmatic Lexical Functions model

AGNES TUTIN

Abstract

In this paper, the Lexical Functions Model of the *Explanatory and Combinatorial Dictionary* is examined (Mel'chuk 96; Wanner 96) for the formalization of collocations in natural language processing. We propose a classification of the different kinds of functions used to account for collocations. A formal grammar which accounts for the grammaticality and the syntactic characteristics of complex LFs is then sketched. Lastly, we highlight syntactic properties which have to be encoded in the lexicon (either within the base entry or within the collocate entry) if the collocations are to be used in a working system and show that inheritance mechanism have to be taken into account to avoid redundancy.

Introduction

The Lexical Functions model, which belongs to the *Explanatory Combinatorial Dictionary* (ECD), constitutes an attractive formalization for collocations in natural language processing :

- Lexical Functions (hereafter LFs) are part of a complete linguistic theory, the Meaning-Text Theory, which the ECD is itself only a component of (Mel'chuk & Polguère 1987; Wanner 96). The Meaning-Text Theory is well-suited for natural language processing and the LFs have already been examined for applications in Machine Translation (Heylen *et al.* 1994) and in Text Generation (Iordanskaja, Kim & Polguère 1996).
- The LFs model provides a semantic and syntactic formalization : the collocations are described by means of a syntactico-semantic label, e.g. the LF called **Magn** means 'very, intensely' and yields modifiers (adjectives or adverbs according to the part of speech of the input word).
- The model is generative : it allows for LFs combinations (for example, **AntiMagn** 'not very intensely'), even if the syntax and semantics of the combinations sometimes remain unclear.
- The model has already been used to encode large amounts of data : the three volumes of the French ECD.

In this study, we attempt to go deeper into the syntagmatic LFs formalism, with the intention of adapting this model for applications in natural language processing in French, completing the important preparatory work performed by Marga Alonso Ramos (1993). We take as a basis for this task the three volumes of the *Dictionnaire Explicatif et Combinatoire du Français Contemporain* (DEC1 1984, DEC2 1988, DEC3 1993), the French ECD, which already constitutes a rich research basis. Our metalexicographic study is directly orientated towards natural language processing and it aims to determine to what extent the actual formalism could be reshaped and deepened.

We will first introduce briefly the different kinds LFs of which account for collocations. We will then offer a descriptive model for the different kinds of functions and for the combinations through a formal grammar. We will lastly review the syntactic criteria that should be taken into account for natural language processing.

1. Syntagmatic Lexical Functions of the ECD

1.1 A brief definition of collocation[1]

According to the ECD perspective, we define a collocation as a **semi-compositional phrase** (e.g. *heavy smoker, to have a bath, ...)* , as opposed to phrasemes (e.g. *to kick the bucket*), which are absolutely non compositional, and to standard phrases (e.g. *a large house*, *to buy a book*). The way a word is verbalized (the *collocate*) depends on the immediate context of this word (the *base*). In other words, the word to express intensity in co-occurrence with *smoker* is *heavy* (the literal translation in French would be *big*), while the action corresponding to *bath* is lexicalized through *have* (in French, *to take*).

1.2 Different Kinds of Lexical Functions

Collocations have been formally described from both a syntactic and semantic perspective by means of the Lexical Functions (hereafter LFs), and more exactly by Syntagmatic Lexical Functions (hereafter SLFs). A Lexical Function is a function (in the mathematical sense) which is applied to a key-word to produce a value or a set of values. In the case of SLFs, the key-word can be considered the base, and the value(s) the collocate(s). The most productive and universal LFs have been labeled and around 50 standard LFs are thus available. For instance, the LF **Magn** (intensity meaning) is applied to *smoker* to produce *heavy*, while the LF $\mathbf{Oper_1}$ is applied to *bath* to yield the light verb *have* whose *bath* is the direct objet. This is encoded in the following way :

Magn(*smoker*) = *heavy*
$\mathbf{Oper}_1$(*bath*) = *have*

All LFs are not used to describe a collocational relationship. Depending on co-occurrence and semantic criteria, three types of LFs can be distinguished :

- **Syntagmatic LFs** like **Magn** or $\mathbf{Oper_i}$ are used to formalize collocations.
- **Paradigmatic LFs** are used to associate a keyword with lexical value(s) that share(s) a non trivial semantic component with the keyword. The value(s) and the keyword do not usually form a phrase. For example, $\mathbf{S_{loc}}$ (typical noun for the location) and $\mathbf{S_3}$ (typical noun for the third actant) are paradigmatic relations : S_{loc}(*skate*) = *skating rink* and S_3(*conference*) = *attendance, audience*.
- **Mixed LFs** share characteristics of the two previous types. They are used to associate a keyword with a set of lexemes which share a semantic component, but they can constitute a phrase, e.g. Mult(*ship*) = *fleet* ('standard word for a collection') or Cap(*school*) = *principal* ('the head of').

1.3 Combined Lexical Functions

Furthermore, LFs can be combined in three ways :

- **Composed LFs** are combined through a regular composition, similarly to mathematical functions. The final value is produced through intermediary values of the intermediary functions. For example, to find the values for S_0(Gener(*étuver* [to steam])), we should first get the values for Gener(*étuver*) = *cuire* [to cook], and then S_0(*cuire*) = *cuisson* [cooking]. Composition is of

[1]) The reader will find a thorough investigation on the topic in Heid (1994).

course not commutative. For the study of collocations, composed LFs are of no interest to us[2]. Firstly, they are mainly used to describe paradigmatic relationships. Secondly, compositions do not have to be stored in the dictionary, because they can be regularly deduced from single LFs.

- **Configurations of LFs** have a key word in common. LFs are here linked by a "+" sign, which is commutative. For example, **Magn** + **Func**$_0$ (*storm*) = *rage* means that *rage* is the value when *storm* is the grammatical subject of a light verb which has no complement, and that the storm is intense. While configurations should be considered for collocational relations, they remain quite rare, and they will not be considered at present.
- **Complex LFs** are combinations which cannot be split up. The value is not obtained through a regular combination, unlike the composed LFs. For example, AntiMagn(*price*) = *moderate*, cannot be deduced from Magn(*price) = high*. *Moderate* cannot be considered as "Anti" for *high*, because *moderate* is a value for AntiMagn only when it is related to *price*. When a combined LF is syntagmatic, it is generally a complex LF, while the same rule does not apply to paradigmatic LFs.

  A complex LF is not a composition : in actual facts, the complex LF is a LF resulting from a combination. In the above example **AntiMagn**, **Magn** can be considered as a function, while Anti can be considered as a "functional", a functional being a second order function applying on functions.

  We can thus define a **Lexical Functional** (LFal hereafter) as a Function which applies to a single or a complex FL to produce a complex LF (in a recursive way). The complex LF itself applies to a lexical item to produce lexical items.

  Many functions are at one and the same time single LFs and LFals. For example, Anti is a single paradigmatic LF (Anti(*slim*) = *fat*) and is also a Functional used to produce complex LFs where it negates the semantic content of a function (**Magn** means 'intensity, a large amount of ' while AntiMagn means 'smallness, a small amount of'. Nevertheless **AntiMagn** should be considered as a whole).

  Not only single paradigmatic LFs but also single syntagmatic LFs are used as functionals. Some functions are almost only used as functionals (e.g. Incep 'to begin to' : IncepPred(*malade*) = *tomber*).

For the implementation of collocations in NLP, the semantics and syntax of complex functions have to be thoroughly defined. At the moment, there is confusion and uncertainty among the "ECD practitioners" that prevent a fully-fledged formalization and implementation of LFs[3].

A first step in this direction would be a thorough description of the Single LFs and of the Functionals. This would enable a second step to be taken, a formal grammar of complex LFs.

[2] We do not intend to mean that they are irrelevant for natural language processing. On the contrary, we think they can very profitably be used in text generation for the choice of an appropriate referring expression (see Alonso Ramos *et al.* (1995); Tutin & Kittredge (1992)).

[3] As is quoted in Wanner 96 (ed.) "Although the notion of lexical function is in itself appealing [...], using the very LFs proposed by Mel'chuk is impractical without the collaboration of an MTT guru" (Robin, 1990, *Lexical Choice in Natural Language Generation*, *CUCS-040-90*. Technical Report. New York : Columbia University)

2. A Detailed Description of SLFs : Single LFs and Functional used to build Complex SLFs

In this section, we will just focus on syntagmatic LFs and Functionals used to produce complex syntagmatic LFs.

2.1 Single SLFs

2.1.1 An inventory of SLFs

First of all, an exhaustive inventory of SLFs must be compiled, based on the DEC volumes. This operation may seem trivial but, surprisingly, we encountered some difficulties while doing it. The syntagmatic/paradigmatic nature can usually be deduced from the definitions given in the DEC (e.g. DEC 2 : 91-94) and from the examples given. For example, Oper$_i$ and Magn are clearly defined as syntagmatic :

- "**Oper$_0$, Oper$_1$, Oper$_2$** ... is the semantically empty verb which takes the impersonal pronoun (*il*) or the name of the first, second, ... actant of the situation C$_0$ as its grammatical subject [...], and the key-word C$_0$ as its main object complement [...].
  Oper$_0$(*vent I.1* [wind]) = *faire* [lit. to make]
  Oper$_1$(*attention* [attention]) = *faire* [to pay]
  Oper$_2$(*attention* [attention]) = *attirer* [to catch]
  ... " (DEC 2 : 93)
- "**Magn**: "very", "intense/intensely", "to a high level".
  Magn(*mémoire I.1* [memory]) = *prodigieuse*, *excellente*, *étonnante*, *d'éléphant* [good]
  Magn(*remercier* [to thank]) = *vivement*, *chaleureusement*, *de tout cœur* [very much]
  ..." (DEC 2 : 92)

Nevertheless, for a few functions, the syntagmatic/paradigmatic status remains unclear. For example, it is not clear whether **Mult** ("regular set of ...") is a syntagmatic or paradigmatic LF. In fact, we found three cases in the French ECD with **Mult** :

1. The value cannot be used without the key-word (with the meaning of Mult) : Mult is syntagmatic.
   Mult(*brebis* [ewe]) = *troupeau* [flock].
   Without the keyword, the meaning of the word is very generic.
2. The value cannot co-occur with the key-word : Mult is paradigmatic.
   Mult(*bleu^2*[rookie]) = *bleusaille* [the rookies].
   Here, the association with the key-word is agrammatical : * *bleusaille de bleus*
3. The value can either appear with the key-word or without it : Mult is mixed.
   Mult(*abeille* [bee]) = *essaim* [swarm]
   essaim is a wider synonym for *essaim d'abeilles*[4].

At present, the codification for Mult is not consistent and we propose to consider this function basically as a syntagmatic function. It would be noted as follows, using the merge sign[5] when it is used paradigmatically :

Mult(*brebis*) = *troupeau*
Mult(*bleu^2*) = // *bleusaille*
Mult(*abeille*) = *essaim*; // *essaim*

[4] *Essaim* can be considered as a prototypical "Mult" for *essaim d'abeilles*. This is not the case for *troupeau* and *troupeau de brebis*.

[5] A merged function is a paradigmatic function whose value is equivalent to both the key-word and the collocate. This is encoded with the '//' sign.

The same ambiguity occurs in the DEC with a few functions like **A**$_0$, **A**$_i$, **Adv**$_i$, **Gener**, **Sing**.

2.1.2 Descriptive parameters

To formally describe the syntagmatic LFs, we propose a set of parameters. The values obtained for the parameters are language specific and the examples provided in this section are taken from French[6].

A given LF can apply to many parts of speech[7]. Since the syntax of the function can slightly vary according to that parameter (e.g. the values do not belong to the same part of speech), we think the LFs have to be defined according to the part of speech of the base.

Our description will include the following parameters :

- The **part of speech of the base** : noun, verb, adjective or adverb.
- The **Deep Syntactic relationship (DSyntR) between the base and the collocate** (is it an attributive or an actancial relationship?) **and the direction of the relation** (is the base the head or a daughter of the phrase?). For example, **Magn** is introduced by an attributive relationship (a modifier) and the base is the head of the collocational phrase, whereas **Mult** corresponds to an actancial relationship where the collocate is the head of the NP and the base is the daughter of the collocate[8].
- The **broad part of speech for the value** : adjectival phrase, nominal phrase, adverbial phrase ...
- The **type of surface verbalization for the value** : part(s) of speech and type of phrases generated by the LF. The values can be lexemes (a), phrasemes (b) or non-frozen phrases (c).

  (a) **Magn**(*pluie* [rain]) = *torrentielle*, *diluvienne* [lashing]
  (b) **Magn**(*chanter* [to sing]) = *à tue-tête* [lit. at the top of one's voice]
  (c) **Magn**(*manger* [to eat]) = *comme un ogre* [like a horse]

  It is crucial to distinguish lexemes and phrasemes from phrases, because the latter do not constitute lexical entries in the ECD. Thus, similes like *comme un ogre* and *comme un loir* (for *dormir*) do not need to be registered as autonomous entries. Nevertheless, the specific connotation should be indicated within these entries.
- The **semantic type of the LFs**. The semantic type is determined according to the combinatorial capacity of the LFs. For example, the "Realization" type includes **Real**$_i$, **Fact**$_i$ and **Labreal**$_{ij}$, the "Support" type includes **Oper**$_i$, **Func**$_i$ and **Labor**$_{ij}$, the "Empty-modifier" type just includes **Epit**[9] ('conventional empty modifier').
- The **kind of subscripts and superscripts** for each function, and whether they are compulsory or facultative. Among subscripts, we can find : (semantic or syntactic) actancial subscripts (which indicate the involved actant, e.g. **Oper**$_1$, **Magn**$_1$), set theory subscript (**Syn**$_\supset$, **Ver**$_\cap$, ...), semantic subscripts (**Magn**$_{\text{'consequence'}}$).

[6]) The description provided here carries on and completes the classification proposed by Alonso Ramos & Tutin (1996), Alonso Ramos (1993).

[7]) For example, **Magn** applies to verbs, nouns, adjectives and adverbs.

[8]) We thus would have :

X --ATTR--> Magn(X) Ex: FUMEUR--ATTR--> GROS
Mult(X) --II--> X Ex: ESSAIM --II-->ABEILLE

ATTR is the modificative relation, while II is an actancial relation.

[9]) The classification proposed by Alonso Ramos & Tutin 96 is not fine-grained enough for building LFs combinations. For example, in that classification, **Epit** is included, like **Magn**, **Bon**, **Pos**$_1$, **Plus**, **Minus**, **Pejor** in the class of "Adjectival and Adverbial Modifiers". Nevertheless, while **Magn** or **Bon** can be combined with **Pred** (**PredMagn**, **PredBon**), **Epit** cannot be easily combined with **Pred** (***PredEpit**).

Among superscripts, we find : semantic superscripts (**Magn**^{temp}, **Magn**^{quant} , **Oper**^{actual}, **Oper**^{usual} ...), and intensity superscripts (**Real**$^{II}{}_3$, **Real**$^{I}{}_3$).

We provide below some LFs applying on nouns :

Parameter	**Magn**	**Epit**	**Real**
Definition	'very', 'intense/intensely', 'to a high degree'	semantically empty modifier	verb which means "to realize" which takes the key-word as the first object and the noun of the ith actant as the subject
Part of speech of the base	noun	noun	noun
Deep Syntactic relationship	- base : head, - collocate: daughter, - relation : ATTR	- base : head, - collocate : daughter, - relation : ATTR	- base : daughter, - collocate : head, - relation : II
Broad part of speech for the value	adjectival phrase	adjectival phrase	verb
Type of surface verbalization for the value	- adjective or adjectival locution - prepositional phrase	- adjective or adjectival locution - prepositional phrase	- verb or phrasal verb
Semantic type of the LF	Evaluation-Modifier	Empty-Modifier	Realization
Kind of subscripts and superscripts	- facultative semantic subscript - facultative actancial subscript - facultative semantic superscript.	none	- compulsory actancial subscript - facultative intensity superscript
Examples	*attention* [attention]: *soutenue* [sustained] *feu* [fire]: *d'enfer* [hellish] *bruit* [noise]: *à crever les tympans* [ear-splitting]	*quenotte* [toothy-peg]: *petite* [small] *numéro* [number]: *d'appel* [phone]	Real$^{I}{}_3$(*conseil I.1* [advice]) : *accepter* [to accept] Real$^{II}{}_3$(*conseil I.1* [advice]) : *suivre* [to follow]

Table 1 : Description of some LFs applying on nouns

2.2 LFals

As already mentioned, functionals are functions which apply to functions either single or complex (the mechanism is recursive) and produce functions. Some functionals can have the same designation as some single paradigmatic and syntagmatic LFs from which they borrow their meaning, but not the surface characteristics, while some functionals, like **Caus**, have no single equivalent LF. We are just interested here in combinations producing complex syntagmatic LFs.

Rather than use the quite imprecise notion of "**head function**"[10] (Alonso Ramos 93; Alonso Ramos & Tutin 96), we prefer to adopt more formal parameters.
For the functionals, the following parameters should be examined :

- **The type of LFs the functional can be combined with**. For example, **S**$_0$ can be combined on any verbal LF or complex verbal LF. On the contrary, **Anti** cannot be combined to pure light verbs like **Oper**$_i$ or **Func**$_i$.
- **The part of speech of the value that the complex LF produces** (Functional + (complex or single) LF). For example, a combination whose first functional is **S**$_0$ will produce a noun (Ex :**S**$_0$**IncepOper**$_1$(*poids*) = *prise*), while **Incep** (which is combined with a verbal LF) is the first

[10] "Every combination of LF has a *head* LF. This LF is the semantico-syntactic pivot of the combination : it expresses the central element of the value and determines the syntactic role of the value" (Alonso Ramos & Tutin 96 : 162). This notion had been taken from Elnitsky (in Mel'chuk *et al.* 1988).

functional of a verbal LF (Ex : **IncepOper$_1$**(*poids*) = *prendre*).

- **The eventual syntactic change induced by the functional**. **S$_0$** changes the syntactic category of the value, but does not generally alter the syntactic relation between the base and the collocate. For example, for **IncepOper$_1$**(*poids*) = *prendre*, the head of the relation is the collocate and the collocate is related with the base by an "actancial II " relation. The same kind of relations will take place with **S$_0$IncepOper$_1$**(*poids*) = *prise*[11]. On the contrary, the LFal **Caus** modifies the actancial structure of the affected LF[12].
- **The semantic class of the functional**. The semantic class is a crucial parameter for LFs combinations. The functionals which possess similar combinatorial properties will be incorporated into the same class. For example, **Incep**, **Fin** and **Cont** will be incorporated into the Phasal Class.
- The **kinds of superscripts or subscripts** the functionals can have. Their use can be quite different from that found on single LFs. **Magn**, for example, as a functional, does not take actancial subscripts. The class of Causatives **(Caus**, **Liqu**, **Perm)** can be succeeded by actancial subscripts (which indicate the semantic actant of the noun involved as the grammatical subject of the combination).

In the following table, we give examples of some functionals described with the above parameters :

Parameter	**Incep**	**S$_0$**
Definition	'To begin to'	Nominalization
LFs affected by the functional	- Support verb LFs : Func$_i$, Oper$_i$, Labor$_{ij}$. - Realization verbal LFs: Fact$_i$, Real$_i$, Labreal$_{ij}$. - Expression verbal LFs : Involv, Manif, Degrad, Son, Excess. - Predicative LFs : Pred. - Complex verbal LFs.	Any single or complex verbal LF
Part of speech of the value produced by the complex or simple LF	verb	noun
Eventual syntactic change induced by the Fal	No syntactic change	No syntactic change
Semantic type of the LFs	Phasal (like Fin and Cont)	Nominalization
Kind of subscripts and superscripts	None	None
Examples	IncepOper$_1$(*alphabétisation* [elimination of illiteracy]) = *entreprendre* [to start] IncepReal$_3$(*école I.1a* [school]) = *entrer* [to go to] IncepInvolv(*fureur 3b* [fury]) = *se déchaîner* [to burst out] IncepPredPlus(*chagrin I.1* [sorrow]) = *s'accroître, augmenter, grandir* [to increase] IncepProxOper$_1$(*objection 1* [objection]) = *concevoir* [to conceive]	S$_0$Oper$_2$(*appel téléphonique* [phone call]) = *réception* [reception] S$_0$Real$_2$(*numéro de téléphone* [phone number]) = *composition* [dialling] S$_0$Son(*tempête I* [storm]) = *hurlement* [howling] S$_0$PredPlus$_1$(*paie I* [wages]) = *augmentation, hausse, majoration* [increase] S$_0$Caus$_1$Func$_0$(*scénario 1* [screenplay]) = *écriture, rédaction* [writing]

Table 2 : Descriptive parameters for Functionals

[11] *Quelqu'un* [I] *prend du poids* [II]. Similarly :
La prise de poids [II] *de quelqu'un* [I].

[12] For example, compare **Func$_1$**(*fatigue I.1a*) = *peser* [*sur* N] and **Caus$_3$Func$_1$**(*fatigue I.1a*) = *entraîner* [ART *chez* N] in the following examples :

La fatigue [I] *pèse sur Marguerite* [II].

Le travail [I] *entraîne une certaine fatigue* [II] *chez Marguerite* [III].

Caus adds an actant and moves forward the actants of the involved LF.

3. A Grammar of LFs : an example with verbal combinations including support verbs and realization verbs

Once the description of single LFs and Functionals is completed, a grammar of complex LFs can be initiated. The purpose of this grammar is twofold :

- To enable combinations to be checked in an electronic editor, in order to make easier the lexicographer's job while encoding the LFs, allowing him to concentrate more on lexico-semantic issues than on formal codifications.
- To enable the implementation of the complex LFs in a natural language processing system, guaranteeing the consistency of the lexical information.

For these purposes to be reached, a simple BNF grammar proves inadequate. The grammar should not only account for the grammaticality of the combinations, but should also enable one to determine the semantic type and the syntactic characteristics of complex LFs. Obviously, exhaustiveness cannot be reached since non standard LFs are always allowed.
In the examples given below, we will just focus on combinations including support verbs (Oper$_i$, Func$_i$, Labor$_{ij}$) and realization verbs (Real$_i$, Fact$_i$, Labreal$_{ij}$).

The grammar can easily be encoded in a unification grammar. A simple LF like **Func**$_i$ will be encoded as follows :

```
actsub → 1
actsub → 2
actsub → 3

lf → func, actsub
lf catbase = noun
lf catcoll = verb
lf synt head = colloc
lf synt rel = 1
lf synt daughter = base
lf typesem = {support}
```

This rule states that **Func** is a LF when succeeded by an actancial subscript. The part of speech of the base for **Func**$_i$ is a noun, the collocate is a verb, the collocate (the head) is related to the base (the daughter) by an actancial relation (the base is the subject) and the semantic type is "support" (as for **Oper**$_i$ and **Labor**$_{ij}$).

The codification for functionals is straightforward : the semantic type alone has to be registered.

```
fal → incep
fal typesem = {phasal}
```

In this rule, **Incep** has the semantic type "phasal" (a phase of the action), like **Fin** and **Cont**.

The complex LFs beginning with a Fal will exploit the semantic type to account for LF combinations. For example, the complex LFs beginning with **Incep** (and **Fin** and **Cont**) will be encoded as follows :

```
lf#1 → fal, lf#2
lf#1 typesem = conc((fal typesem) (lf#2 typesem))
lf#1 catbase = noun
lf#1 catcoll = verb
lf#1 synt = lf#2 synt
fal typesem = {phasal}
lf#2 typesem =   {realization}
            or   {support}
            or   {intensity, realization}
            or   {anti, ...}
            or   {future, ...}
```

The typesem of a complex LF is a list which consists of the concatenation of the Fal typesem and the LF typesem[13] (the LF can itself be either complex or simple). If the Fals belong to the "phasal" semantic class, the syntax of the complex LF will be similar to the syntax of the LF (for example, **IncepOper**$_i$ has the same syntactic characteristics as **Oper**$_i$[14]), while the part of speech for the base of the combination is a noun and the value is a verb. The typesem lists with which the typesem of the functional can be combined is a disjunction of lists.

For lack of space, we will not detail here all the rules involved, but the combination of semantic types for LFs including support verbs or realization verbs can easily be sketched by a finite state automaton, as is shown above.

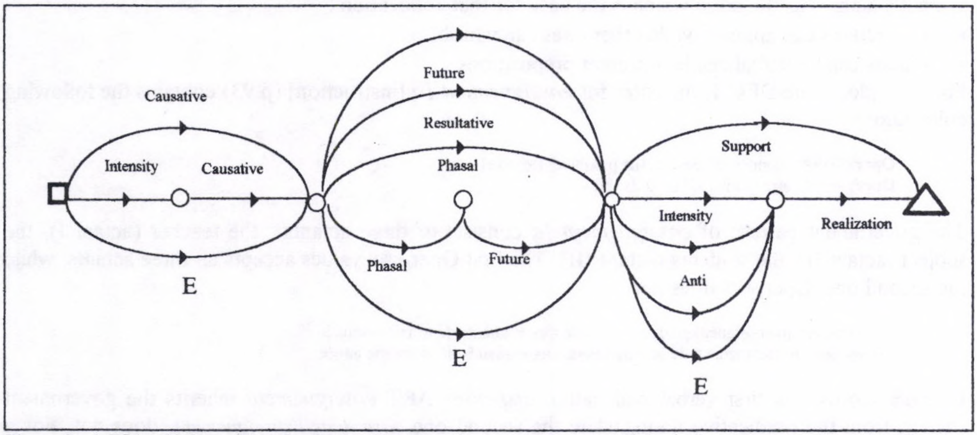


Figure 1 : A FSA which accounts for the combination of semantic types in LFs including support verbs and realization verbs

In Figure 1, **E** stands for empty arc.
The following semantic classes are used :

- **Intensity** : Magn, AntiMagn
- **Causative** : Caus($_i$), Liqu($_i$), Perm($_i$).
- **Future** : Prox, Prepar.
- **Phasal** : Incep, Fin, Cont.

[13]) This basic mechanism should be bettered to account for the semantic composition of complex LFs.

[14]) For example, Oper$_1$(*peur*) = *avoir* has the same syntactic properties than IncepOper$_1$(*peur*) = *prendre* : the base is the second actant of the collocate in both cases.

- **Support** : Oper_i, Func_i, Labor_{ij}
- **Realization** : Real_i, Fact_i, Labreal_{ij}

Acoording to the grammar, **IncepProxOper$_2$** will be allowed, while **MagnOper$_1$** would be considered agrammatical.

4. Syntactic properties of collocations

For natural language processing, syntactic and semantic information concerning collocations should be highly detailed. Though the DEC seems highly consistent and formalized to the human reader (even sometimes a little bit indigestible for the dilettante reader), the syntactic information is not detailed enough for the collocations.

4.1 The government pattern of collocations[15]

In the DEC, the collocates for a given base are stored within the base entry. The government pattern of the base is highly detailed in the DEC, but this is not the case of the collocation itself. The collocate entry often contains a reduced government pattern, which is generally not very detailed. Nevertheless, since the government pattern of the collocation cannot be systematically deduced from the base, this incomplete treatment may raise issues in a computational perspective. For example, the verbal collocations including nouns, e.g. support verbs like **Oper$_i$**, do not necessarily take the same syntactic actants as the noun which is the base for that collocation :

- New actants can appear, while other ones can vanish.
- Actants can be verbalized by different prepositions.

For example, in the DEC 1, the entry for *enseignement 1a* [instruction] (p 93) contains the following collocations :

> **Oper$_1^{\text{actual}}$** : donner, dispenser, **litt** prodiguer [to give]
> **Oper$_1^{\text{usual}}$** : être [dans l'~] [lit. to be in]

The government pattern of *enseignement 1a* consists of three actants : the teacher (actant I), the subject (actant II), the students (actant III). The first $\text{Oper}_1^{\text{actual}}$ values accepts all three actants, while the second one, $\text{Oper}_1^{\text{usual}}$, does not :

> Léa dispense un enseignement de sémantique aux étudiants de troisième année.[16]
> * Léa est dans l'enseignement de la sémantique aux étudiants de troisième année.

In other words, the first verbal collocation *dispenser* ART *enseignement* inherits the government pattern from the predicative noun, while the second one, *être dans l'enseignement*, does not. For a natural language generator to use the collocation *être dans l'enseignement*, the lexicon should contain the information that the actants II and III cannot be verbalized. Besides, in a verbal collocation including a noun, actants can depend either on the predicative noun or on the verb and the type of dependency affects the syntactic behaviour of the collocation in transformations. Let us take the example of the collocation *donner un cours* [to give a lesson]. Three actants can be related to the predicative noun *cours* : the teacher (actant I), the subject (actant II), the audience (actant III). The collocation *donner un cours* (Oper_1(*cours*) + *cours*) also contains three actants (*quelqu'un* (I) *donne un cours sur quelque chose* (II) *à quelqu'un* (III)), but obviously, all the actants are not related to the predicative noun : while the first and the third actants seem to depend on the verb, the second

[15]) The government pattern is the actancial structure of the predicate.

[16]) Loose translation : *Léa is giving instruction in semantics to third year students.*
Léa is in semantics teaching for third year students.

actant seems related to the noun, since in the passive transformation, this complement cannot appear in a postverbal position :

Lulu a donné un cours de mathématiques à Léa.
Un cours de mathématiques a été donné à Lulu par Léa.
* Un cours a été donné de mathématiques par Léa[17].

Nevertheless, in the DEC, the collocate *donner* would be just mentioned as an Oper_1 collocate, within the *cours* entry, without any further precision about the syntactic behaviour about the whole collocation. Such pieces of linguistic information prove nevertheless crucial and show that a reduced government pattern appears inadequate to account for the syntactic behaviour of collocations. In order to lighten the lexicographer's job, some general heuristics could be profitably used as default values. For example, Alonso Ramos (1993) proposed some heuristics for the government pattern of $\mathbf{Oper_i}$ collocations. She found that in three actant collocations like in *donner un cours*, the first and the third actants generally depend on the verb, while the second one generally depends on the predicative noun.

4.2 Distributional and transformational properties of collocations

If collocations happen to be used in real texts, they should be used in all kinds of contexts. Collocations are often not syntactically frozen and the base and the collocate may appear in a non adjacent context. For example, in many noun-adjective collocations, the adjective can be graded or may appear in a predicative position.

Léa a les cheveux très blonds.
Les cheveux de Léa sont très blonds.

This is of course not always the case for noun-adjective collocations, like in the following one where *red* in a predicative position cannot be interpreted in the same sense as in *red wine* (# indicates that the reading is not collocational) :

# This is a very red wine.
# This wine is red.

Besides, some adjectival collocates tend to have a regular syntactic behaviour, whatever base they are associated with. For example, the intensity collocate *gros* like in *gros fumeur* tends to appear in a prenominal position in many collocations (*grosse averse* [heavy rain], *grosse fièvre* [high fever]).

We thus hypothesize that many syntactic properties of the collocate within the collocation are not idiosyncratic for the particular collocation, but are properties of the collocate. We tried to confirm this hypothesis by a small corpus study performed on three French adjectival collocates : *gros*, *invétéré* and *endurci*. We extracted concordances from a one year corpus of the French newspaper *Le Monde* (year 1992) with the INTEX software (Silberztein 1993). This small corpus exhibited interesting properties concerning our hypothesis. For example, we noted that the word *gros* is very common for agent nouns derived from verbs and expresses in that case the intensity of the activity :

gros acheteur, gros annonceurs, gros buveur, gros consommateur, gros constructeur, gros débiteurs, gros dealers ....

In that context, the adjective is often graded, is always in a prenominal position and is never predicative.

[17]) Literal translation : *A lesson has been given of mathematics by Léa.*

Invétéré also displays interesting properties. In our corpus, that collocate is amost always combined with a human noun (25 occurrences out of 26) which generally denotes a bad habit, is always postnominal, attributive and is not graded (and could not be). Here are a few examples :

optimiste, invétéré, ivrogne invétéré, fumeur invétéré, menteur invétéré ...

In some cases, the collocational status of *invétéré* is questionable, since the association does not appear purely conventional (*optimiste invétéré* is far less conventional than *menteur invétéré*). The association is thus productive to a certain extent (? *emmerdeur invétéré,* ? *frimeur invétéré*). Nervetheless, in any case, the adjective always exhibits the same properties. Clearly, these properties are not idiosyncratic to the adjective-noun association, but to the adjective itself.
Endurci turns out to be less (syntactically) frozen than *invétéré*. *Endurci* can involve inanimate nouns (*cœur endurci, organismes endurcis*), though less common than human nouns and, contrary to *invétéré*, *endurci* can be graded.

Examination performed on these three collocates leads us to think that, to a certain extent, some generalizations could be performed on the syntactic properties of the collocates and that the syntactic properties of the collocates are not always purely idiosyncratic. Collocates should have their own entries where syntactic properties are registered. Our pragmatic proposition differs from Mel'chuk's position about the properties of collocate subentries, which are, according to him, always idiosyncratic.

> The obervable peculiarities of L' [the collocate] are phraseological; they are not, strictly speaking, peculiarities of L', but are due to its being part of the collocation L' + L1. These peculiariries must be indicated in the lexical entry of L1 [the base]... - where L' itself is specified. This means that an element L' of the value **f**(L) of the LF **f** given in the entry of **f**'s keyword L can form an *embedded subentry* which carries whatever is individual and idiosyncratic informations for the particular collocation. [Mel'chuk 96 : 75].

This perspective explains that in the current ECD, the syntactic properties of the collocates, even those which seem to be properties of the collocate itself are stored in the base entry, within the collocate subentry. For example, the predicative/attributive nature of the adjectival collocate, the pre/postnominal position of the adjective are encoded within the base entry.

For the sake of economy while encoding the distributional properties, especially in a large scale lexicon, we think properties pertaining to the collocate should be encoded within the collocate entry. The collocation would inherit the default values of the collocate, while the exceptions would be indicated in the base entry and would override the properties encoded in the collocate entry.

Transformational properties of collocations must also be encoded in the lexicon. A large number of them seems to depend on the collocate, as for distributional properties. For example, the support verb *avoir* never accepts the passive transformation, while this is not the case for many support verbs like *donner* :

Lulu a faim.[18]
* Une faim est eue par Lulu.

Un cours a été donné par Léa[19].
La vaisselle a été faite par Léa.

[18]) Literal translation : Lulu has hunger.

[19]) Literal translation : A class has been given by Léa.
The dishes have been done by Léa.

For many support verb constructions, and for the other verbal collocations, a large number of properties should be encoded in the lexicon : passive transformation (a), reduced passive (b), relative transformation (c), *se*-passive transformation (d), impersonal passive transformation (e), ...

(a) Ce cours a été donné par Lulu.
(b) Le cours donné par Lulu.
(c) Le cours que Lulu a donné.
(d) Le cours se donne à l'Université de Montréal.
(e) Il a été donné le même cours à l'Université McGill.

Many syntactic properties, either distributional or transformational, thus happen to be inherited from the collocates. We propose to encode the syntactic properties primarily within the collocate entry, even if this collocate is semantically empty, as is the case for light verbs. Exceptions should be encoded within base entries and would override default values inherited from the collocates.

For example, a collocation like *gros fumeur* will be encoded as shown in the figure below. The collocational adjective *gros* in relation with *fumeur* will be registered within *fumeur* entry. But the syntactic properties of *gros* (prenominal, gradable, attributive, not predicative) will be inherited from the collocate entry.

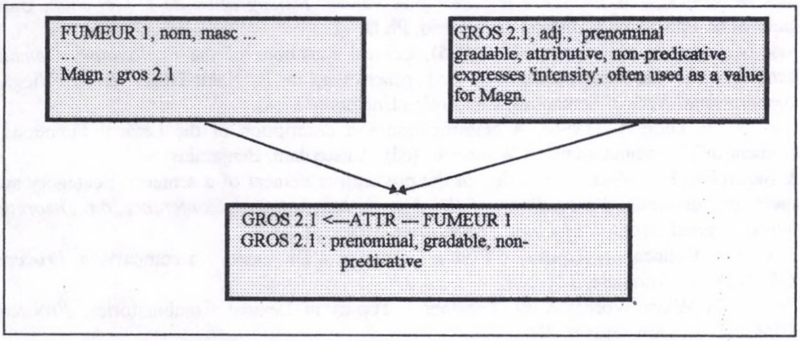

FUMEUR 1, nom, masc ...
...
Magn : gros 2.1
GROS 2.1, adj., prenominal
gradable, attributive, non-predicative
expresses 'intensity', often used as a value for Magn.
GROS 2.1 <—ATTR — FUMEUR 1
GROS 2.1 : prenominal, gradable, non-predicative

Figure 2 : Syntactic properties inherited from the collocate entry

Nevertheless, in the case of *célibataire endurci*, the collocation will not inherit all the collocate properties. For example, while *endurci* can be graded in many collocations (*cœur très endurci*), this is not the case with *célibataire endurci* (*? *célibataire très endurci*). This particular property (non-gradable) will thus be stored within the base entry, within the *endurci* subentry.

According to us, inheritance mechanisms should be applied not only to semantic fields (Mel'chuk and Wanner 1996 showed how a class of German emotion lexemes are likely to collocate with the same verbal lexemes and proposed a technique to account for the lexical inheritance), but also to syntactic properties of the collocations.

Conclusion

In this paper, we examined a few problems raised by the formalization of collocations and we tried to lay the foundations of a few questions that should be tackled for adapting the ECD model to Natural Language Processing. It seems to us that some work remains to be done before the LFs can be implemented in a natural language processing system, in particular in the field of LFs combinations and syntactic properties. A detailed description of LFs (syntax, semantics, combinations) would be a first step in the direction of formalization. Besides, to account for the syntactic behaviour of collocations, many syntactic properties have to be encoded in the lexicon. The actual ECD mentions the collocations through the base entry, but does not account for their distributional and transformational properties. These properties inherit from both the base and the collocate entries and default inheritance mechanisms should be integrated in the formal lexicon.

The fine-grained descriptions and the attempt to formalize in a systematic way the collocational relations make the ECD a complex model. The use of an automatic editor seems indispensable for the consistency of lexical entries to be checked and inheritance mechanisms to be handled. Such an editor, which would be very profitable for making easier the ECD lexicographers' job, necessitates a fully-fledged formalization of the model.

References

Alonso Ramos M. (1993), *Las funciones lexicales y el modelo lexicografico de I. Mel'chuk*, Universidad Nacional de Educacion a Distancias, Madrid, Ph.D.

Alonso Ramos M.,Tutin A. & Lapalme G. (1995), Lexical Functions of the *Explanatory Combinatorial Dictionary* for the lexicalization in text generation, in P. Saint-Dizier & E. Viegas (eds), *Computational Lexical* Semantics, Cambridge University Press.

Alonso Ramos M. & Tutin A. (1996), A classification and description of the Lexical Functions for the treatment of LF combinations, in Wanner L. (ed), Amsterdam, Benjamins.

Danlos L. & Samvelian P. (1992), Translation of the predicative element of a sentence : category switching, aspect and diathesis, *Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation*, Montréal, 21-34.

Fontenelle T. (1992), Collocation acquisition from a Corpus or a Dictionary : a comparison, *Proceedings of EURALEX '92*, Tampere, 221-228.

Heid U. (1994), On Ways Words Work Together - Topics in Lexical Combinatorics, *Proceedings of EURALEX '94*, Amsterdam, 226-257.

Heylen D., Maxwell K. (1994), Lexical Functions and the Translation of collocations, *Proceedings of EURALEX '94*, Amsterdam, 298-305.

Iordanskaja L., Kim M. & Polguère A (1996). Some procedural problems in the implementation of Lexical Functions for text generation, in Wanner L. (ed.)

Mel'chuk I. (1996), Lexical Functions : A Tool for the Description of Lexical Relations in a Lexicon, in L. Wanner (ed).

Mel'chuk I., Polguère A. (1987), A formal lexicon in the Meaning-Text Theory (or how to do lexica with words), *Computational Linguistics*, 13, 261-275.

Mel'chuk I. *et al.* (1984, 1988, 1992), *Dictionnaire Explicatif et Combinatoire du Français Contemporain : Recherches lexico-sémantiques*, Montréal, Presses de l'Université de Montréal.

Mel'chuk I. & Wanner L. (1996), Lexical Functions and Lexical Inheritance for Emotion Lexemes in German, in L. Wanner ed. (1996).

Mel'chuk I. (1994), Collocations and Lexical Functions, in *Proceedings of the Leeds Colloquium on Collocations*.

Smadja F., McKeown K. (1991), Using collocations for language generation, *Computational Intelligence*, 7(4), 229-239.

Wanner L. ed. (1996), *Lexical Functions in Lexicography and Natural Language Processing*, Amsterdam, Benjamins.

Classification syntaxique des verbes de mouvement du hongrois dans l'optique d'un traitement automatique Etude comparative francais-hongrois

LIDIA VARGA

Résumé

Jusqu'à présent, pour la description du hongrois, langue agglutinante, les études lexicographiques sont plus abondantes que les études syntaxiques dans le domaine du traitement automatique.Le marché informatique est très demandeur d'études syntaxiques pour le hongrois aussi.

Notre travail propose une classification syntaxique qui se veut exhaustive des verbes de mouvement du hongrois. Cette classe de verbes montre beaucoup de ressemblances avec la classe de verbes correspondantes du français. Nous donnerons la définition syntaxique des verbes de mouvement du hongrois et étudierons certaines propriétés syntaxiques de cette classe de verbe, notamment la relation préfixe verbal, verbe, et leurs combinaisons possibles.

1. Introduction

Jusqu'à présent, pour la description du hongrois, langue agglutinante, les études lexicographiques sont plus abondantes que les études syntaxiques dans le domaine du traitement automatique. Il existe des dictionnaires électroniques de mots relativement complets, ce qui se traduit par exemple par l'apparition des premiers correcteurs d'orthographe du hongrois (*Helyes-e*, sous la direction de G. Proszéky, *Lektor* sous la direction de L. Seregi), mais les correcteurs tenant compte de la grammaire se font attendre. Le marché informatique est très demandeur d'études syntaxiques pour le hongrois aussi.

L'hypothèse forte était que comme en français et en plusieurs langues non-indoeuropéennes aussi (par exemple le coréen), il est possible de définir la notion de verbe de mouvement pour le hongrois par des propriétés syntaxiques.

Notre travail propose une classification syntaxique qui se veut exhaustive des verbes de mouvement du hongrois. Nous donnerons la définition syntaxique des verbes de mouvement du hongrois et étudierons certaines propriétés syntaxiques de cette classe de verbes, notamment la relation verbe-préfixe verbal.

2. Le cadre de l'analyse

Dans notre étude, nous avons utilisé le cadre théorique du lexique-grammaire de Maurice Gross fondé sur la théorie transformationnelle de Z. S. Harris. Nous considérons comme l'unité de sens la phrase élémentaire et non le mot. Cela suppose la description systématique et la formalisation des phrases élémentaires de la langue, analyse indispensable pour fournir des données linguistiques susceptibles d'être intégrées dans des systèmes informatiques en vue d'un traitement automatique. Dans notre interprétation, la phrase simple se compose d'un élément prédicatif et de ses arguments (M.Gross,1975; Boons, Guillet, Leclère,1976). Pour l'analyse des verbes, plus exactement des emplois de verbes la structure considérée est celle de l'extension maximale en termes de compléments "significatifs".

L'ordre des mots en hongrois varie fortement en fonction de l'élément focalisé de la phrase et selon la topicalisation. Dans notre analyse, nous nous en tenons à l'ordre de la phrase dite neutre où aucun .élément de la phrase n'est focalisé. Cela correspond en général à l'ordre SVO, mais l'ordre SOV est aussi fréquent.

Cette étude constitue le début d'une description syntaxique systématique de tous les verbes du hongrois.

3. La construction des verbes de mouvement du hongrois

3.1. Définition syntaxique des verbes de mouvement (Vmt) du hongrois

En français, les verbes de mouvement peuvent se définir par la possibilité d'accepter un complément phrastique spécifique. Ce complément est défini par des propriétés syntaxiques particulières par rapport aux autres compléments phrastiques. Le verbe de mouvement admet un complément à l'infinitif non précédé d'une préposition. Ce complément a un caractère adverbial et répond à la question *où?* (M. Gross,1975) :

N_0 Vmt $V^o{}_{inf}$ W
Il court voir Marie.
Où court-il ? *Voir Marie.*
V_0 = Vmt (M.Gross,1975)

En hongrois, nous pouvons distinguer également une construction $N_0VmtV^o{}_{inf}$ *W* avec des propriétés syntaxiques similaires :

Dans la construction : ***N_0V_0 (E+N_{locdyn}) $V^o{}_{inf}$ W***

- V_0 = Vmt
- le sujet du verbe principal (V_0) et celui du verbe complément ($V^o{}_{inf}$) est le même.
- le *Vmt* admet un complément à l'infinitif (*$V^o{}_{inf}$*) sans préposition ni suffixe casuel et répond à la question *où? (hova?*dynamique*) :*
  - *W* est le complément non obligatoire de *$V^o{}_{inf}$*
  - *N_{locdyn}* est le complément de lieu non obligatoire (voir 3.1.1.)

Hova szalad Péter? Où court Peter ?

Péter *szalad meglátogatni Marit.*
N_0 Vmt $V^o{}_{inf}$ W
Peter court voir Marie.

- la transformation *N_0 Vmt $V^o{}_{inf}$ W* $\rightarrow$ *N_0 Vmt que P* est possible en hongrois , mais elle n'est pas acceptée en français :

a) *Szaladok megnézni a filmet.* Je cours voir le film.
futni courir
jönni venir
sietni stb. se dépêcher etc.
Vmt $V^o{}_{inf}$ W

b) *Szaladok, hogy megnézzem a filmet.* Je cours voir le film.
Vmt que V^o W

Ainsi, par exemple, la phrase à complément infinitif (1 a) ne répond pas à la question *où ?* (hova?) mais à la question *que ?* (mit?) et la transformation *N_0 Vmt $V^o{}_{inf}$ W* $\rightarrow$ *N_0 Vmt que P* n'est pas possible (1 b); donc le verbe *szeretni* (aimer) n'est pas un verbe de mouvement selon notre classement syntaxique non plus :

(1.) a) *Peter szeret inni.*(N_0 Vmt V^oinfW) Peter aime boire.
b)**Peter szereti, hogy iszik.* (N_0 Vmt que V^o W) *Peter aime qu'il boive.

En hongrois, les verbes exprimant sémantiquement un déplacement partiel du corps comme ***leül*** (s'asseoir), ***letérdel*** (s'agenouiller) etc., qui sont des verbes préfixés acceptent également un complément à l'infinitif (sans préposition ni suffixe casuel) et expriment également une intentionalité. Mais la construction répond à la question *Que fait-il ?* (Mit csinàl ?) Ces verbes représenteront une autre classe de verbes.

Le verbe *siet* (se dépêcher) et ses variantes à préfixe ne sont pas des verbes de mouvement du point de vue sémantique, mais leurs propriétés syntaxiques coïncident avec les propriétés de notre classe de verbes de mouvement :

a) *Péter siet megjavitani a motort.* Peter se dépêche de réparer la moto.
N_0 Vmt V^o_{inf} W

3. 1.1. La directionalité dans *N_0Vmt (Nloc+E) V^o_{inf} W*

Le hongrois distingue nettement dans la syntaxe l'aspect statique ou dynamique d'une action. Les aspects statiques et dynamiques assignent des cas différents dans la phrase. Ainsi le hongrois possède 2 variantes pour la question *où ?* du français :

hol (statique)	pour la localisation
hovà (dynamique)	pour l'orientation, la direction

En hongrois dans la construction correspondante (*N_0 Vmt (Nloc+E) V^o_{inf} W*) seule la question en *hova ?* (où dynamique) est permise. Cette construction exprime toujours une directionalité et une intentionalité. La directionalité est exprimée par le Vmt et son complément (2 a). Le complément locatif (*Nloc*) de la phrase doit être obligatoirement marqué par un suffixe casuel (*rag*) dynamique ou bien prendre une postposition (*névutó*) à la forme dynamique (2 c d) :

(2.) a) *Megyek enni.* (Vmt V^o_{inf}) Je vais manger.
b) ***Ki***megyek enni .(prefVmt V^o_{inf}) Je sors manger.
c) ***Ki****megyek a konyhá****ba*** *enni.* Je sors manger dans la cuisine.
prefVmt N_{locdyn} V^o_{inf}

d) ***Ki****megyek a fa* ***alá*** *enni.* Je sors manger sous l'arbre.
(prefVmt N_{loc} $postp_{dyn}$ V^o_{inf})

En français, une grande partie des structures exprimant un mouvement est représentée par des structures prépositionnelles. La phrase française :

(3.) *Paul court dans la chambre.*

est ambiguë. Elle peut signifier que Paul est dans la chambre et court à l'intérieur de celle-ci ou peut signifier encore que Paul est dehors et court dans (vers) la chambre. En hongrois, grâce à la différentiation syntaxique des aspects statiques et dynamiques, à la phrase (3) correspondent deux phrases différentes selon l'interprétation voulue; ainsi toute ambiguïté est exclue (4 a b) :

(4.) a) *Paul szalad a szobában.* Paul court à l'intérieur de la chambre.
N_0 Vmt Nlocstat
b) *Paul szalad a szobába.* Paul court vers (+dans) la chambre.
N_0 Vmt Nlocdyn

Seule la phrase (4 b) peut accepter un complément infinitif, donc pour le hongrois la question *hová ?* fournit un critère formel aussi pour la définition syntaxique des verbes de mouvement .

a) **Paul szalad a szobában enni.*
N_0 Vmt Nlocstat V^o_{inf}
b) *Paul szalad a szobába enni.* Paul court manger dans la chambre.
N_0 Vmt Nlocdyn V^o_{inf} (Il n'était pas dans la chambre)

3.1.2. Restrictions distributionnelles concernant le sujet N_0 :

Les restrictions distributrionnelles sont les mêmes que pour le français. Le caractère de complément circonstantiel de but de la construction infinitive en hongrois, autrement dit l'intentionalité dans l'action, explique que, le sujet (N_0) doit être animé :

N_0 = animé
Péter kimegy halászni. **A hajó kimegy halászni.*
N_{0hum} Vmt V_{1inf} N_0 Vmt V_{1inf}
Péter sort pêcher. ? Le bateau sort pêcher.

Avec certains verbes, N_0 est obligatoirement au pluriel :

A katonák szétszaladnak búvóhelyet keresni.
N_{0plur} Vmt W V_1
Les soldats se dispersent pour chercher un abri.

3.1.3. Le temps et la construction N_0 *Vmt* (N_{locdyn}) V^o_{inf} *W*

Dans cette construction le temps du verbe principal (Vmt) et le temps du verbe à l'infinitif (V^o_{inf}) doivent coïncider (T_0=T_1). et les deux verbes ne peuvent avoir des compléments de temps différents.

3.1.4. Transitivité, intransitivité

. Pour la construction *N_0 V_{mt} (Nloc+E)V^o_{inf} W* en hongrois les Vmt ne peuvent avoir qu'un emploi intransitif contrairement au français. En hongrois, l'équivalent de la phrase (5 a) avec V_0 transitif n'accepte pas la réduction du complétive (5 b) :

(5.) *a) Max átússza a folyót, hogy találkozzon Luc-kel.*
N_0 Vmt N_{1accus} que $V°_{subj}$ W
Max traverse la rivière à la nage rencontrer Luc.

b) **Max átússza a folyót találkozni Luc-kel.*
N_0 Vmt N_1 V^o_{inf} W
Max traverse la rivière à la nage rencontrer Luc.

4. Le préfixe verbal et les verbes de mouvement

Un problème apparemment organisationnel s'est posé au cours de l'élaboration de la table des verbes: faut-il faire figurer en entrée de table les verbes dans leurs formes préfixées ou seulement les formes non préfixées et leur faire correspondre les préfixes dans les colonnes où aparaissent les propriétés syntaxique non définitionnelle de la construction? Etant donné qu'en hongrois les préfixes verbaux ont des propriétés syntaxiques générales (règles de détachement et de déplacement), cette dernière solution est économique à première vue.

Les préfixes verbaux modifient l'aspect de la phrase dans laquelle se trouve le verbe préfixé ou apportent un sens nouveau aux verbes auxquels il s'ajoutent. Cet autre sens est souvent exprimé en français par un verbe à part .

***Ki**megy* (ki.= dehors+megy=aller)	sortir
***Be**megy* (be=dans+megy=aller)	entrer

En hongrois, la majorité des verbes exprimant une activité à mouvement ou à mode de locomotion exprimé, (*lovagol,* faire du cheval), et se combinant avec un préfixe verbal spatial deviennent des verbes de mouvement (***ki**lovagol*, se rendre à un endroit à cheval) satisfaisant aux critères syntaxiques de notre construction *N_0 Vmt(Nloc+E) V^o_{inf} W* (6 a, b). La phrase (6 c) avec le verbe sans préfixe est correcte, mais elle n'est pas considérée comme neutre :

(6). a) **Lovagol (a hegyre) megnézni a várat.*
Vmt (Nloc+E) W V^o_{inf}
*Il fait du cheval (à la montagne) voir le château.

b) ***Ki**lovagol (a hegyre) megnézni a várat .*
pref Vmt (Nloc+E) V^o_{inf}
Il va (à la montagne) à cheval pour voir le château.

c) *A hegyre lovagol megnézni a várat.*
Nlocdyn Vmt V^o_{inf}
Il va à la montagne à cheval pour voir le château.

Selon notre classement, les verbes sans préfixes comme *üget* (trotter), *vágtázik* (galoper), *csoszog* (marcher d'un pas traînant), *csattog* (claquer), *biciklizik* (faire du vélo), *sétál* (se promener), *ùszik* (nager) etc., ne sont pas des verbes de mouvement du point de vue syntaxique et fonctionnent comme le verbe *lovagol.* Leurs variantes préfixées : ***(kiüget, kivágtázik, kicsoszog, kicsattog, kibiciklizik, kisétál*** etc.,) entrent dans la construction des verbes de mouvement. La construction donne lieu à beaucoup de combinaisons de verbe-préfixe de ce genre. Ainsi des verbes non-préfixés ne figurant pas dans la table peuvent avoir de 8 à 14 variantes préfixées, selon le verbe, acceptées dans la construction.
Nous avons vu ci-dessus que certains verbes avec préfixe fonctionnent comme des verbes de mouvement mais ne sont pas des verbes de mouvement sans leur préfixe. D'autres verbes comme *megy* (aller), *rohan* (se précipiter, se dépêcher) sont des verbes de mouvement avec et sans préfixe. Dans certains cas les préfixes spatiaux se combinant avec les Vmt ne produisent que des changements aspectuels dans la phrase (mise à part les règles de déplacement des préfixes dans la phrase) :

(7.) a) *Peter megy telefonálni.* Peter va téléphoner (non-accompli, duratif)
N_0 Vmt V^o_{inf}

b) *Peter **elmegy** telefonálni.* Peter va téléphoner. (résultatif)
N_0 prefVmt V^o_{inf}

Dans d'autres cas le préfixe verbal apporte une modification sémantique de la phrase :

(8.) a) *Peter megy telefonálni.* Peter va téléphoner
N_0 Vmt V^o_{inf}
megy = aller

b) *Peter **kimegy** telefonálni.* Peter sort téléphoner
N_0 pref Vmt V^o_{inf}
ki =(vers) dehors), megy = aller

Dans la phrase (8 b), le sujet se déplace pour téléphoner en sortant de quelque part.

Les différentes combinaisons de préfixe-verbe peuvent aussi avoir des restrictions distributionnelles sur le sujet et sur les compléments locatifs. Par exemple, le verbe ***szalad*** (courir) avec le préfixe *szét* (dans tout les sens) donne le verbe ***szétszalad*** (se disperser) qui implique un sujet pluriel et avec le préfixe ***ki*** = (vers) dehors nous avons le verbe, ***kiszalad*** (sortir en courant) que l'on peut considérer comme un autre verbe pour lequel le sujet pluriel n'est pas obligatoire.

La combinaison préfixe-verbe n'est pas automatique. Certaines combinaisons ne sont pas acceptées (9. a). Il faut faire la liste exhaustive des combinaisons possibles pour la construction étudiée. Une combinaison de préfixe-verbe acceptée dans une construction ne l'est pas forcément dans une autre. (9 a, b).

(9.) a)**Peter* ***szétment*** *telefonálni.* Peter se disperse téléphoner)
N_0 pref Vmt V^o_{inf}
szét = dans tous les sens

b) ***Szétment*** *a cipöm.* Mes chaussures se sont rétrécies.
pref V_0 N_0

En position de V^o_{inf}, la construction étudiée n'accepte pas les variantes préfixée des verbes du type *tàncol* (danser), *ùszik* (nager) , mais sans préfixe, ils sont acceptés :

a) **Megyek* ***kiúszni*** *a partra.* Je vais rejoindre la plage en nageant.
Vmt prefVmt$_{inf}$ + Nloc

b) *Megyek ùszni.* Je vais nager .
Vmt V^o_{inf}

Ces arguments nous ont amené à faire figurer les formes préfixées d'un verbe dans des entrées différentes de la table de verbe. Nous avons indiqué parmi les propriétés syntaxiques (dans les colonnes) quand le verbe est une forme préfixée ou non. Cette information permet l'application des règles syntaxiques générales des préfixes par des procédures automatiques.

5. Conclusion

Nous avons pu définir une classe de verbe de mouvement du point de vue syntaxique. Ces verbes ont des propriétés syntaxiques caractéristiques et expriment un mouvement et souvent un mode de déplacement. Notre but n'était pas de regrouper tous les verbes du hongrois sémantiquement considérés comme verbes de mouvement, mais de trouver des propriétés syntaxiques formalisables pour un traitement automatique des verbes où la notion de mouvement peut apparaître. C'est la raison pour laquelle nous n'avons pas donné de définition sémantique précise des verbes de mouvement. (Il en existe plusieurs, entre autres celles de : B. Lamiroy, 1983; J-P. Boons, 1987).
Certains verbes qui font partie des verbes de mouvement selon notre définition, sont des combinaisons de préfixe-verbe jusqu'à présent non répertoriées dans les dictionnaires traditionnels. Par exemple, le verbe *sétál* (se promener) apparaît avec le préfixe *ki-* et *be-* aussi, mais le verbe *úszik* (nager) apparaît avec le préfixe *ki-*, mais pas avec le préfixe *be-*, ni dans le Dictionnaire de flexion du hongrois de L. Elekfi (*Magyar Ragozási Szótár)*, ni dans le Dictionnaire de la Langue Hongroise en 7 tomes (*A Magyar Nyelv Ertelmezõ szotàra 7 kötetben*). Les verbes ***kiúszik*** et ***beúszik*** ont pourtant à peu près la même fréquence d'utilisation.
Jusqu'à présent, nous avons répertorié 1922 verbes. La classe correspondante du français en compte 130 verbes (table 2.,Gross, M.,1975). Notre liste est relativement exhaustive, mais d'autre verbes et d'autres critères syntaxiques s'y ajouteront, d'une part parce que la définition syntaxique des préfixes verbaux et adverbe n'est pas évidente en ce qui concerne le hongrois (A. Komlósy, 1992), d'autre part la flexibilité du hongrois permet des

combinaisons verbe-préfixe qui peuvent paraître bizarres aujourd'hui mais qui seront acceptées demain.

Références

Guillet, A. : *Représentation des distributions dans un lexique/grammaire*. Langue Française 1986.

Gross, M. : *Méthodes en syntaxe*. Hermann, 1975.

Gross, M. . *Grammaire transformationelle du français : Syntaxe du verbe*. Libraire Larousse, 1968.

Guillet, A./Leclère, Ch. : *La structure des phrases simples en français*. Libraire Droz S.A., 1992.

Kelemen, J. : *De la langue au style. Eléments de linguistique contrastive français-hongrois*. Akadémiai Kiadó, Budapest 1988.

E. Kiss, K. : *Az egyszerû mondat szerkezete*. Strukturális Magyar Nyelvtan. Akadémiai Kiadó Budapest,1992.

Komlósy A. : *Régensek és vonzatok*. Strukturális Magyar. Nyelvtan. Akadémiai Kiadó Budapest.1992.

Szabolcsi, A-Laczkó T. *A főnévi csoport szerkezete*. Strukturális Magyar. Nyelvtan. Akadémiai Kiadó Budapest.1992.

Dictionnaires :

Elekfi, L. : *Magyar Ragozási Szótár*. (Dictionnaire de flexion du hongrois). MTA, Nyelvtudományi Intézet, Budapest, 1994.

A Magyar Nyelv Értelmezõ Szótára. I/VII. (Akadémia Kiadó, Budapest, 1959-1966). Dictionnaire de la langue Hongroise en 7 tomes).

Liste des verbes du hongrois sur support informatique. Société Morphologic, 1995.

Tuning the Text with an Electronic Dictionary

DUŠKO VITAS – CVETANA KRSTEV

Abstract

In the article the feasibility of a morphological electronic dictionary construction for the language with reach morphology and unstable orthographic system is presented. The variations occurring in text can not be handled satisfactorily neither on the text encoding level nor by the use of the e-dictionary in its standard form as described in [Gross89]. The construction of a meta-dictionary is proposed based on the redefined graphemic system from which the various e-dictionaries in their standard form can be derived by varying a set of appropriately chosen parameters described in this article. The impact of such a dictionary to the revision of traditional Serbo-Croatian lexicographic practice in defining the dictionary entry is discussed, as well as its application in the process of corpus construction.

1. Introduction

Electronic dictionary, in the sense this term is introduced in [Gross89], can be used not only for the corpora processing [Silberztein94], but also in the phase of the construction of corpus for the language with reach morphology and unstable orthography, such as Serbo-Croatian, or, even, Old Church Slavonic. In this article we present one approach to the adaptation of the e-dictionary model in the development and exploitation of corpora for the language with the notion of dictionary entry not stable enough due to the various graphemic, morphographemic, morphological, dialectic and other sources of variations. The main lexicographic task is thus to establish the dictionary entry form that effectively encompasses the possible variations of word forms in text. In the traditional lexicography all these variations are recorded as separate dictionary entries, not always consistently. The starting point of our lexical processing is a model suggested in [Courtois90]. The chosen model with certain modifications not only enables the processing of the source text but leads to more precise specification of dictionary entries in the scope of traditional Serbo-Croatian lexicography.

In this article an application of e-dictionary will be presented which deals with the processing of the collection of proverbs assembled by Vuk [Vuk87], the language reformer whose radical language reform in mid XIXc introduced the phonetically based orthography. This text, together with the other collections by the same author (stories, poems, riddles, etc.) forms the base of both contemporary Serbo-Croatian and literally norms developed from it (particularly, Serbian and Croatian) and is thus an unavoidable part of any corpus of the contemporary languages.

The importance of these texts is not only historical: they illustrate the inherent variability of morphographemic composition of text in comparison to morphographemic system itself [Popović96]. In one sample of modern Serbian prose from 1980–1990 the same phenomenon of morphographemic variations in text is noticed, so it has to be taken into consideration when planning the processing of a contemporary language corpus.

The article presents first the description of some of the variations in the text of proverbs using the SGML encoding (section 2.) and then the morphological e-dictionary underlying the text of proverbs produced according to the traditional lexicographic criteria (section 3.). It can be seen that by using both of these formalisms it is not possible to effectively deal with the whole set of variations of the dictionary entries. In order to overcome these difficulties the modification of graphemic set of Serbo-Croatian is performed as well as the redefinition of the concatenation operator which enables the effective handling of the text in use. In section 4 the model is introduced that establishes the links between the text and the modified e-dictionary. Some of the paths going through a text enabled by this model are emphasized.

2. Electronic text vs. traditional dictionary

The source of the new edition of Vuk's proverbs is its last edition published as a joint venture of Belgrade publishing houses Nolit and Prosveta in 1987. It should be noted, however, that the inventory of proverbs in it and the forms they are presented in have not changed essentially since its first edition in 1849. The last edition thus maintains the old orthography of the original (see example 1) as well as the elements of the Old Church Slavonic alphabet, for instance hard sign, and stressed letters (example 2).

(1) **<pv>**Bog te sačuvao vedra **<corr sic=**'božića' **resp=**'Nolit'**>**Božića**</corr>**
i oblačna Đurđeva dnevi**!</pv>**
<!-- *God saves you from fair Christmas and cloudy Đurđevdan!* **-->**
<pv>Bolje je reći: **<corr sic=**'ne ću' **resp=**'Nolit'**>**neću, no: sad ću.**</pv>**
<!-- *It is better to say: I won't, then I will now.***-->**

```
(2)     <pv>Bez p%adsil;r%adsil; ni u crkvu.</pv><!-- Without money not even to the church. -->
```

In the source, all the proverbs are arranged in the alphabetic order. Some of them are followed by the explanation and some by the link to other proverbs. The original ordering and linking, reproduced in all the later editions, lack both accuracy and thoroughness: there are references to the missing proverbs, links are missing between the proverbs that have basically the same or opposite meaning or does not differ but in minor details as word ordering, changing singular to plural (ex. 3).

```
(3)     <pv>Kakav gost    onaka  mu i   čast.</pv>   <!-- Like guest like feast. -->
        <pv>Kakvi gosti   taka        i   čast.</pv>   <!-- Like guests like feast.-->
```

The new edition does not aim to alter the text significantly except for the adjustments to the contemporary Serbian orthography and alphabet. Its main new feature is the comprehensive index containing all the lexical words reduced to their 'usual' dictionary entries.

The base of the electronic edition is the SGML encoded text. The Document Type Definition (abb. DTD) has been produced based on the encoding proposed by Text Encoding Initiative [Sperberg94]. The developed DTD represents the enhancement of TEI DTD for prose with addition of the data sets for dates, names, and pointers. The DTD has been developed and encoding strategy accepted that enables the fulfillment of the following aims:

- the structural components of a text should be precisely described. For instance, proverbs themselves should be unambiguously separated from the explanation that may follow. This required the addition of new elements to the basic TEI DTD sets (ex. 4);
- all the portions of text, such as terms, names or dates, that may be of use in its future exploitation, possibly not yet foreseeable, should be highlighted. In order to describe the text thoroughly new SGML-elements were introduced to distinguish ethnic names and obscene phrases (ex. 5). Some of these new elements were introduced to describe the variability in the structure of the proverbs: optional parts (ex. 6a), changeable parts (ex. 6b), and open parts (ex. 6c) left to the user to fill them with an appropriate phrase;
- the network should be established that links the proverbs according to their lexical, syntactic or semantic resemblance (ex. 4);
- the encoding should enable the production of both original and revised versions of a main text (see ex. 1).

```
(4)     <divp id=P4185 n=4185>
              <pv>Paze se kao mačka i miš.</pv>
              <!-- Take care about each other as a cat and a mouse. -->
              <pexp>Ili: <s.a target=P4186 resp='Vuk'></s.a></pexp>      <!-- Or: -->
        </divp>
        <divp id=P4186 n=4186>
              <pv>Paze se kao mačka i pseto.</pv>
              <!-- Take care about each other as a cat and a dog. -->
              <pexp>Ili: <s.a target=P4187 resp='Vuk'></s.a></pexp>      <!-- Or: -->
        </divp>
        <divp id=P4187 n=4187>
              <pv>Paze se kao so i oko.</pv>
              <!-- Take care about each other as a salt and an eye. -->
              <pexp>Mrze jedan drugoga.</pexp>                          <!-- Hate each other. -->
        </divp>
```

(5) (a) **<pv>**Bolje **<opt.ph rend=**'() bold'**>**ti**</opt.ph>** je da te ćera
<ethnicName type='narod'**>**Turčin**</ethnicName>** sa sabljom nego
<ethnicName type='narod' **reg=**'Nemac'**>**Švabo**</ethnicName>** s perom.**</pv>**
<!--*It is better for you to be chased by a Turk with a sabre than by a German with a pen.*-->
<pexp>U **<placeName>**Hrvatskoj**</placeName>.</pexp><!--** *In Croatia.* **-->**

(b) **<pv>**Blažene su mnoge ručice, al' su proklete mnoge
<obs.ph value='guzice' **resp=**'CV'**>**g....e**</obs.ph>.</pv>**
<!--*Many hands are blessed, but many b....ms are cursed.*-->
<pexp>Mnogi mnogo urade, ali mnogo i pojedu.**</pexp>**
<!--*Many people can do much, but they eat much as well.*-->

(6) (a) **<pv>**I đavo zna što je pravo**<opt.ph rend=**'() bold'**>**ali ne će da čini**</opt.ph>.</pv>**
<!--*Devil also knows what is right (but doesn't do it)*-->

(b) **<pv>**Igraju se magarci, biće **<c.b.r id=**P1551.t1**>**kiše**</c.b.r>.</pv>**
<!--*Donkeys play, it will rain.*-->
<pexp>Kad se matori ljudi igraju kao đeca. Mjesto **<hi rend=**'italic'**>**kiše**</hi>**jedni reku: **<c.r.s id=**P1551.t2 **rend=**'italic'**>**lijepo vrijeme**</c.r.s>.</pexp>**
<!-- *When the old play like children. Instead of rain some say: nice weather.*-->
<link targets='P1551.t1 P1551.t2' **type=**'exchangable'**>**

(c) **<pv>**Drži se **<opt.ph rend=**'() italic'**>**
<f.gap resp='CV'**>**koga**</f.gap> <n.i.prov>**ili**</n.i.prov>**
<f.gap resp='CV'**>**čega**</f.gap></opt.ph>** kao pijan plota.**</pv>**
<!-- *Stick (to someone or something) as a drinker to the fence.* -->

The index was produced after a arduous lemmatization of concordances of the text of proverbs performed independently by two experienced traditional lexicographers involving the several changes of strategies for handling the numerous graphical and other variations. The example 7a shows the lemmas, that is index entries, proposed by these two lexicographers for different variations of the noun *hleb* 'bread'. The first of them was of the opinion that not all of the variations can be index entries while the other considered them all as such. Both of them, however, thought that the variation *hleb*, although not found in text, should be pointed at by all other variations since it prevails in literary use today. It should be stressed that the list of variations of the noun *hleb* is not exhausted by those occurring in text: [RSANU] also registers the forms *leba*, *lebac* and *ljebac*. The example 7b shows the index that results from this traditional approach.

Yet another glaring example is an attempt to find the corresponding lemma for the word forms of different variations of the verb *sedeti* 'to sit'. The example 8a shows that the word form *siđela* was attached to the lemma *sedeti* although this decision does not have the confirmation in the description of this lemma in the [RMS/MH]. The lexicographers could not decide upon the infinitive for the forms *sjede*, *sjedi*, *sjediš*...—one thought it should be *sjedeti*, the other *sjedjeti* (ex. 8b). Both, however, agreed that *sjeđeti*, although the infinitive form, can not be a lemma (ex. 8c).

7 (a)

i.n.	**w.f.**	**first lexicographer**	**second lexicographer**
d2 45	leb.	ljeb (=leb) (v. hleb)	leb (v. i beškot, ljeb, hlebac, hljeb, hljebac)
d3 60	ljeb.	ljeb (=leb) (v. hleb)	ljeb (v. i beškot, leb, hlebac, hljeb, hljebac)
5076	hlebac.	hljebac (=hlebac) (v. hljeb)	hlebac (v. i beškot, leb, ljeb, hljeb, hljebac)
*			hleb (v. beškot, leb, ljeb, hlebac, hljeb, hljebac)
6027	hljeb.	hljeb (=hleb) (v. hljebac)	hljeb (v. i beškot, leb, ljeb, hlebac, hljebac)
6083	hljeba.	hljeb (=hleb) (v. hljebac)	hljeb (v. i beškot, leb, ljeb, hlebac, hljebac)
2726	hljebac.	hljebac (=hlebac) (v. hljeb)	hljebac (v. i beškot, leb, ljeb, hlebac, hljeb)

(b) leb (v. i beškot, ljeb, hlebac, hljeb, hljebac) &n& 45^2, 29^3
ljeb (v. i beškot, leb, hlebac, hljeb, hljebac) &n& 104^1, 60^3
hleb (v. beškot, leb, ljeb, hlebac, hljeb, hljebac) &n&
hlebac (v. i beškot, leb, ljeb, hljeb, hljebac) &n& 5075
hljeb (v. i beškot, leb, ljeb, hlebac, hljebac) &n& 674, 777, ..., 6082
hljebac (v. i beškot, leb, ljeb, hlebac, hljeb) &n& 2725

8 (a)	i.n.	w.f.	first lexicographer	second lexicographer
	5426	siđela.	sedeti	sedeti (v. i sjedjeti)
(b)	d3 36	sjede.	sjedeti (=sedeti)	sjedjeti (v. i sedeti)
	6319	sjediš.	sjedeti (=sedeti)	sjedjeti (v. i sedeti)
(c)	3766	sjeđeti.	sjedeti (=sedeti)	sjedjeti (v. i sedeti)

As a final result, the index is obtained burdened with many references of a type 'see also' (*v. i* in the examples) in an attempt to link all variations of the same unit. The links of a type 'see' (*v.* in the examples) were introduced as well in order to link a form not occurring in text but prevailing in use today with its numerous variations. The influence of this approach to the index can be seen in ex. 9.

(9) dever (v. i djever) &n&12^1
deverak (v. djeverak) &n&
deverivati (v. djeverivati) &v&
deveričić (v. djeveričić) &n&
devojački (v. djevojački) &a&
devojačko mleko &u izr&1677
devojka (v. djevojka) &n&
devojčina (v. djevojčina) &n&

ded (v. djed) &n&
deda &n&4506
dedak &n&902
delibaša &n&1183
delija (v. i vojak) &n&694,…,4826
deliti (v. dijeliti) &v&
delo (v. dilo, djelo) &v&
deljati (v. djeljati) &v&

In some cases the variations are of a kind that prevents the unambiguous lemmatization if performed on this traditional principles. The inflective paradigm of two variants may overlap making it impossible to decide, for instance, on the form of nominative singular for the nouns, that is on the dictionary entry. The case of this is the noun *breg* 'hill' shown in example 10. One lexicographer chose the nominative singular *breg* for the nominative plural *bregovi*, while the other chose *brijeg*.

(10)	i.n.	w.f.	first lexicographer	second lexicographer
	2956	bregovi.	breg	brijeg
	5589	brijeg.	brijeg (=breg)	brijeg
	295	brijegu.	brijeg (=breg)	brijeg

In the traditional dictionaries this collection of proverbs as well as other Vuk's works are the main source of examples. The check-up in the [RSANU] has shown that out of 39 index entries starting with letter 'a', where proper names and links of a type 'see' were not taken into consideration, one does not appear in a dictionary at all and six have no example form Vuk's work. Out of the remaining 32, 19 have example from this collection and three of them only form it. However, proverbs when cited are not always in their original form (see ex. 11 and picture 2). Both dictionaries have chosen the ekavian variation of noun *devojka* 'girl' and the adjective *pre* 'before' but the other deviations are found too. Moreover, [RSANU] attribute this proverb to [Vuk87].

(11) Ko prije đevojci onoga je đevojka. [Vuk87]
Ko(ji) pre devojci, toga je devojka. [RMS/MH]
Ko pre devojci [onoga je (onome i) devojka]. [RSANU]
Who comes first to the girl will have her.

3. Morphological dictionary of Serbo-Croatian

The development of electronic dictionary according to the model described in [Gross89], [Courtois90] is partly based on the processing of lexical material from traditionally compiled dictionaries. In the case of Serbo-Croatian traditional dictionaries, morphological definition attached to an entry lacks the precision that would enable direct generation of inflective forms [Vitas92]. The problem of morphological processing of an entry can be divided into two independent parts observing the fact that the entries of inflective words, except in a small number of exceptions, can be separated into two strings that we shall call the **invariable and the variable** part. The inflection of an entry is limited to its variable part while the morphographemic variations influence its invariable part.

3.1 Inflection

For the description of the variable part the notion of ***elementary morphographemic class*** (abb. EMC) is introduced. EMC is a regular expression that characterizes the type of morphographemic behaviour for the fixed combination of morphological and graphemic parameters that govern the relations in an inflective paradigm. Entry equipped with the morphological definition, that can also be represented by the inflectional class code, enables the algorithmic generation of all the inflectional forms.

For instance, the nouns ***hleb***, ***hljeb***, ***ljeb***, ***leb*** ... 'bread' (see the ex. 7 and 8) can have the EMC defined either by the code N08.01 or by the code N15.04. Nouns belonging to these classes are characterized by the following parameters: they have the masculine gender, their stem finishes with a consonant, the plural forms are extended by an infix *-ov-* and they are marked as inanimate. In the class N08.01 the ending of a vocative singular is *-e* while in a class N15.04 it is *-u*. The regular expression describing these two classes is given in example 12a. The part of the regular expression enclosed by the second parenthesis defines the corresponding EMC. The noun ***hlebac*** 'bread' belongs to the class N20.18 and its EMC is given in ex. 12b. Similarly, the nouns ***devojka***, ***djevojka***, ***đevojka***, ... 'girl' (see the ex. 9 and 11) are in the class N70.05 to which belong the feminine gender nouns, performing palatalization in dative singular with a fleeting *a* in genitive plural (example 12c).

(12) (a) N08.01:(hleb+hljeb+leb+ljeb+...)
(ε/ns,as+a/gs+u/ds,ls+om/is+e/vs+ov(i/np,vp+a/gp+e/ap+ima/dp,lp,ip))
N15.01:(hleb+hljeb+leb+ljeb+...)
(ε/ns,as+a/gs+u/ds,ls,vs+om/is+ov(i/np,vp+a/gp+e/ap+ima/dp,lp,ip))

(12) (b) N20.18:hle
(bac/ns,as+pca/gs+pcu/ds,ls+pcem/is+če/vs+pc(i/np,vp+e/ap+ima/dp,lp,ip)+baca/gp))

(12) (c) N70.05: (devoj+...)
(ka/ns+ke/gs,np,ap,vp+ci/ds,,,,,ls+ku/as+ko/vs+kom/is+aka/gp+kama/dp,lp,ip)

For the entries described in this way, the parts of e-dictionary **DELAF** were produced [Vitas93] and coupled with the text of proverbs. This process enables that for the given entry all its occurrences in text can be found, e.g. for the lemma *leb* all its inflective forms, and only they, can be retrieved from the text. The variant word forms can be found **only** by using the corresponding lemma (***ljeb***, ***hljeba***, ...). The reverse procedure is also possible. With this approach one remains inside the frames imposed by the traditional lexicography in which the links between graphical variations are established using semantic or historical, rather than formal criteria.

The notion of EMC is important as each morphographemic alternation that control the relation between the entry and its inflective paradigm may, but need not, produce the semantic difference. The assemblage of several elementary morphological classes into one, as generally done in the traditional lexicography, has thus to be performed by hand.

The extension of EMC can be obtained by the factorization of regular expressions for the chosen set of textual words and then attaching to the newly obtained regular expression the **compound** morphological class [Vitas96]. This compound class need not be deducible from the morphographemic definition, as was the case for the EMC. Its use has the advantage of gathering into one compound class, besides the inflective paradigm, the elements of derivational paradigm and doublet forms as well. For instance, the compound class CCN08, showed in example 13a, represents the regular expression obtained by gathering the EMC for *hleb* (ex. 12a) and the EMC for *hlebac* (ex. 12b). Similarly, the doublet form *litar*; N04.02:Nmns/*litra*; N70.02:Nfns 'litre' are described with the regular expressions given in ex. 13b. Applying the same procedure we can obtain the new compound class by concatenating the expressions enclosed in parenthesis to the common string *lit*-.

(13) (a) hleb (N08.01+N15.04) + hlebac (N20.18) ----> hleb, CCN08
CCN08: (b/ns,as+ba/gs+bu/ds,ls,vs+bom/is+be/vs+bov(i/np,vp+a/gp+e/ap+ima/dp,lp,ip)+
(bac/ns,as+pca/gs+pcu/ds,ls+pcem/is+če/vs+pc(i/np,vp+e/ap+ima/dp,lp,ip)+baca/gp)
(13) (b) N04.02 lit(ar/ns,as+ra/gs+ru/ds,ls+re/vs,ap+rom/is+ri/np,vp+ara/gp+ima/dp,lp,ip)
N70.02 litr(a/ns+e/gs,np,ap,vp+i/ds,ls,gp+u/as+o/vs+om/is+ama/dp,lp,ip)
litar (N04.02) + litra (N70.02) ----> litar, CCN04
CCN04: lit(ar/ns,as+ra/ns,gs+ru/ds,ls,as+re/vs,gs,np,ap,vp+rom/is+ri/ds,ls,gp,np,vp+
ara/gp+ima/dp,lp,ip+o/vs+ama/dp,lp,ip)

Continuing in the same way, the regularly produced possessive and relative adjectives can be associated with the noun they are derived from. As an example, the possessive adjective *devojčin* and the relative adjective *devojački* can be attached to the noun *devojka* 'girl' (ex. 14a). Compound class for the corresponding regular expressions is given in ex. 14b, where N denotes nouns, APpα denotes the possessive adjectives in positive indefinite form and ARpβ denotes the relative adjectives in positive definite form.

(14) (a) devojka N70.05 +devojčin A06.04+ devojački A03.01
(14) (b) devoj(ka/Nfns+ke/Nfgs+ci/Nf(d,l)s+... +čin/APpαmns+... +čki/AR pβmns+...)

With this approach, the processing of the entries that the traditional lexicography handles uncertainly, the verbal inflection in particular (as in ex. 8), becomes possible.

3.2 Graphemic variations

The second modification of a morphological e-dictionary tends to neutralize the graphemic variations in the invariable part of the entries. This modification consists of the use of two different alphabets: one for the encoding of a dictionary and another for the encoding of text. One approach to this kind of redefinition of alphabet for encoding a dictionary was described in [Sampson89]. It is of particular interest to Serbo-Croatian that equally uses two alphabets. As the written text can be recorded using either the Cyrillic or the Latin alphabet, that do not correspond to each other unambiguously, the alphabet of e-dictionary must not depend on the alphabet used to record the source.

Although the graphical variations described in section 2 can be historically conditioned, it is possible to effectively process them on the synchronous level using one generalization of the notion of grapheme [Krstev95].

We shall now return to the examples 12a in order to concentrate on the invariable part of the entry *hleb*. If we proceed with the factorization, the part that denotes the morphological class can be extracted as a common factor. The resulting expression is shown in ex. 15 where the first two factors represent the graphical variations in the invariable part of the entry (ε denoting the empty string).

(15) (*h*+ε)(*l*+*lj*)*eb*(ε/ns,as+a/gs+u/ds,ls,vs+om/is+e/vs+ov(i/np,vp+a/gp+e/ap+ima/dp,lp,ip))

.Two alternatives can be suggested here as the possible choice for an entry form. The first one is to choose for the entry the regular expression such as *(h+ε)(l+lj)eb*. Besides the introduction of a large number of regular expressions, and consequently a large number of finite automata, this choice has the drawback of not providing the attributes that govern the realization of one or the other variation.

As a second alternative we suggest the introduction of special graphemes, that we shall call *lexicographemes*, as minimal abstract units of a dictionary encoding system. The lexicographeme must have the following property: in a text it can be realized as one or more graphs or graph strings where the particular choice is governed by the set of attributes assigned to the grapheme itself and by the optional operators that generalize the notion of concatenation [Aho72].

If we continue with the example 15, we can consider the abstract object α that can be realized as *h* or ε (empty string) and β can be realized as *e* or *je* with the property to palatalize the preceding *l*. Attributes assigned to α enable the omission of *h* as an unliterary or archaic possibility. Attributes are assigned to *β* according to the dialect. To certain dialects the operators, such as palatalization, are assigned as well. γ, different from *β*, from the example 16b has four different realizations: *e*, *i*, *je* with the property to palatalize the preceding *d*, and *je* with the property not to palatalize the preceding *d*.

(16) (a) *α l β b* (b) *dγvojka*

For the simplicity, we shall denote α from the previous example as (*h'*) and β as (*e'*). With lemma encoded using these new graphemes, the e-dictionary entry can be derived from the following form:

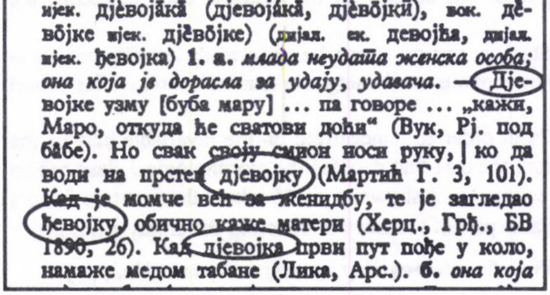
(17) (*h'*) *l* (*e'*) *b*(ε/ns,as+a/gs+u/ds,ls,vs+om/is+e/vs+ov(i/np,vp+a/gp+e/ap+ima/dp,lp,ip))

The above example matches all the inflective forms of the entries *hleb*, *hljeb*, *ljeb*, *leb* ...

This example expanded to the rest of the dictionary of the text of proverbs yields a form of e-dictionary in which the graphical variations are treated in a strictly formal way. As a result of this process we estimate the total number of lexicographemes at few hundred in comparison with 30 standard graphemes of Serbo-Croatian alphabet. The obtained dictionary, although based on the dictionary obtained through the traditional process of excerption, synthesizes the graphical variations and enhances the possibilities of text retrieval. For instance, for an arbitrary entry all its graphical variations can be found (e.g. the form *hleb* returns all the inflective forms of all its graphical variations, *leb, ljeb*,...). Also, the text itself need not be altered, which may often be the case in the phase of his preprocessing.

4. Tuning the text

As has already been stated, neither the precise description of text using the SGML encoding nor the morphological e-dictionary in its standard form are entirely suitable to overcome the limits of text processing imposed by the Serbo-Croatian traditional lexicography. The separation of a dictionary alphabet from a text alphabet and the construction of the e-dictionary based on such a redefined alphabet, described in section 3.2, helps overcoming the problems encountered in corpus construction and exploitation due to permissible orthographic variability of texts. The



дѐвōјка ијек. дјѐвōјка ж (вок. дȅвōјко ијек. дјȅвōјко; мн. ген. дѐвōјāкā (девојáкā, дѐвōјкӣ) ијек. дјѐвојāкā (дјевојáкā, дјѐвōјкӣ), вок. дȅвōјке ијек. дјȅвōјке) (дијал. ек. девојћа, дијал. ијек. ђевојка) **1. а.** *млада неудата женска особа; она која је дорасла за удају, удавача.* — Дјевојке узму [буба мару] ... па говоре ... „кажи, Маро, откуда ће сватови доћи“ (Вук, Рј. под бȁбе). Но свак своју смион носи руку, | ко да води на прстен дјевојку (Мартић Г. 3, 101). Кад је момче већ за женидбу, те је загледао ђевојку, обично каже матери (Херц., Грђ., БВ 1890, 26). Кад дјевојка први пут пође у коло, намаже медом табане (Лика, Арс.). **б.** *она која* suggested format of e-dictionary enables different interpretations of SGML encoded texts depending on the sort and needs of each particular processing. This procedure can be called *tuning of a text*.p

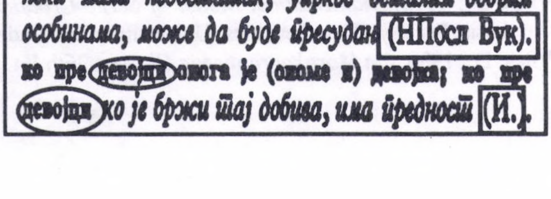
As an illustration, we can examine the processing of the entry *devojka* 'girl' in [RSANU]. The picture shows the fragment of the entry text. Its morphological definition part, enclosed in first parenthesis in the picture, contains an unarranged description of the variations of the entry both in its variable and invariable part. Even such an ehaustive description fails to mention the dative singular form *devojci*, confirmed in [RMS/MH], that is compulsory to establish unambiguously its EMC. The procedure described in section 3 recognizes the entry's invariable part *de'voj-* and its variable part *-ka*. To the variable part the code N70.05 is attached that covers only the inflective forms of *devojka* (as in ex. 12c). Alternatively, the code could be attached that would include the elements of derivational paradigm as well (as in ex. 14). If we consider the first case only, the form of this entry in electronic morphological dictionary would be as presented in ex. 20.

(18) de'vojaka,de'vojka.N70.05:Nfpg+;
de'vojka,.N70.05:Nfns+;
de'vojkama,de'vojka.N70.05:Nfpd+;Nfpl+;Nfpi+;
de'vojke,de'vojka.N70.05:Nfsg+;Nfpl+;Nfpi+;
...

None of the examples from the entry text that illustrate the use of the lexeme *devojka* in its first sense (see the picture 1) use the entry headword but rather the variants that differ from it in its invariable part and that can be obtained by different interpretations of a lexicographeme ***e'***: *devojka, djevojka, đevojka, divojka*, etc (circled in a picture). The exception is the dialect form *djevojća* that does not belong to the same EMC as *de'voj-ka* and that should rather be a separate entry, as, for instance, the entry *devojla* (=*devojka*) already is. On condition that the e-dictionary is constructed on the proposed principles (ex. 20) and that the SGML encoding describes the variations used in the text of a source corpus, the entry could be reconstructed as described in example 19, using the appropriate TEI-like SGML tags.

(19) **<entry>**
<form><metahead>de'voj-ka**</metahead></form>**
<gramGrp><itype>N70.05**</itype></gramGrp>**
</form>
<sense n='1'**><sense n=**'a'**><def>**mlada neudata...**</def>**
<eg variant='jekavian west'**><q>**De'vojke uzmu ...**</q></eg>**
<eg variant='jekavian east'**><q>**...zagledao de'vojku,...**</q></eg>**
<eg variant='jekavian west**><q>** Kad de'vojka...**</q></eg>**
</sense></sense>
</entry>

This encoding enables the authentic visualization of the examples as presented in the picture, but it is also possible to unify all the examples so that they should all use the same form as the entry headword. Moreover, this form of citing is already applied to the proverbs from Vuk's collection in the part of the same entry text that describes idioms, as can be seen in picture 2 and example 11.



НПосл али (ма) девојци срећу квари (губи) *неки мали недостатак, упркос осталим добрим особинама, може да буде пресудан* (НПосл Вук).
ко пре девојци онога је (ономе и) девојка; ко пре девојци *ко је бржи тај добива, има предност* (И.).

It is important to emphasize that the e-dictionary in the format described in [Silberztein93] can be built from the suggested form (ex. 18) by choosing the appropriate attributes either in the extensive or in the reduced form. The future efforts to construct the e-dictionary should thus concentrate on this meta-version of a dictionary in which the variations of the entries would be described in a compact way. In terms of the implementation, the direct use of this meta-dictionary should be possible both for the text retrieval and text analysis on the condition that the source text has been appropriately SGML marked.

5. Conclusion

Facing the situation where, on the one hand, texts include many morphological and orthographic variations and, on the other, the traditional lexicography has not developed the stable criteria for determining the entries, the construction of morphological e-dictionary becomes difficult, if not impossible. The methodological frame suggested in the described model of e-dictionary enables the introduction of the precise formal criteria for the description of both orthographic variations and different morphological phenomena. These criteria are an equally good base for the flexible formal approach to the establishment of entries as well as for the effective encoding of corpora. Thus not only does the construction of e-dictionary become possible, but the framework is set that enables the simultaneous construction of e-dictionary and reconstruction of the traditional dictionary.

6. References

Aho, A.V.; Ullman, J.D.; 1972: *The Theory of Parsing, Translation and Compiling*, vol. 1, Prentice-Hall, New Yersey

Courtois, B.; Silberztein, M. (eds); 1990: *Les dictionnaires électroniques*; Langue française 87, Larousse, Paris

Gross, M.; 1989: *La construction de dictionnaires électroniques*; Annles de télécommunicationes, vol. 44 (1–2), pp. 4–19

Vuk S. Karadžić; 1987: *Srpske narodne poslovice*; Prosveta —Nolit, Beograd

Krstev, C.; Vitas, D; Pavlović-Lažetić, G.; 1995: *Neutralization of Variations in the Structure of a Dictionary Entry in Serbo-Croatian*, I Int.Conf.on Formal Aspects of Slavic Languages, Leipzig

Popović, Lj.; 1996: *Morphosyntactic strings*, in: Homages à Živojin Stanojčić, in press

RMS/MH; 1967: *Rečnik srpskohrvatskoga književnog jezika*, vol. 1–6, Matica Srpska, Matica Hrvatska, Beograd—Zagreb

RSANU; 1950–1990: *Rečnik srpskohrvatskog književnog i narodnog jezika*, vol. 1–14 (A–N), Srpska akademija nauka i umetnosti, Institut za srpskohrvatski jezik, Beograd

Sampson, G.; 1989: *How Fully Does a Machine-Usable Dictionary Cover English Text?*, Literary and Linguistic Computing, vol. 4, No. 1, pp. 29-35

Silberztein, M.; 1993: *Dictionnaires électroniques et analyse automatique de textes: le systeme INTEX*; Masson, Paris

Silberztein, M.; 1994: *INTEX: a Corpus Processing System*, Proc. of COLING'94, Tokyo

Sperberg-McQueen C.M.; Burnard, L. (eds.); 1994: *Guidelines for Electronic Text Encoding and Interchange of Machine-Readable Texts*, TEI P3, ACH–ACL–ALLC, Chicago, Oxford

Vitas, D.; Krstev, C.; 1992: *Interaction between Dictionary and Text in Serbo-Croatian*; Papers in Computational Lexicography COMPLEX'92, Linguistics Institute, Budapest, pp. 333–342

Vitas, D.; 1993: *Mathematical Model of Serbo-Croatian Morphology (Nominal Inflection)*, PhD thesis, Faculty of Mathematics, University of Belgrade

Vitas, D.; 1996: *On Morphographemic Classes*, 26th Int. Slavistic Conf., Beograd (to appear)

Spell-checking and Tagging in a Multilingual Environment with Hairy Input

EDUARD WERNER

Abstract

Spelling checkers are perhaps the most widespread and most straightforward applications of computational lexicography since they obviously need a computer-readable dictionary. For the computational lexicographer on the other hand who has to deal with the problem of tagging languages with strong inflection, taggers for building up computer-readable text corpora are important. Combining both tasks has the advantage of easier getting big dictionaries for a tagger since you can take the dictionaries built up by "normal users" who are using a spelling checker. Furthermore, the multilingual approach makes texts taggable that would otherwise not that easily be usable for automatic analyzing.

Spell-checking and Tagging

The tasks and the structure of a spelling checker and a tagger are mostly identical. Both need a lexicon and rules to derive words from the lexicon entries (in the most simple case, there's only one rule saying that every string matching a lexicon entry is valid word). The assumption about the text, however, are different. While a spelling checker thinks that there are lots of errors in your text a tagger relies on its correctness. Note that this difference, however, mainly affects the largeness of the dictionary. While a tagger should tag anything that *can* be a valid word form a spelling checker is expected to complain about anything "suspicious", that means it should rather think a word is misspelled than accept a really rare form.

Another difference is the user interface. A tagger should normally run non-interactively. Whether a spelling checker should work interactively, can be argued about. It can be tedious to sit around and wait for the program to detect the next word it can't analyze; when the text is large it might be more user-friendly to have the spelling checker run in batch mode and afterwards check the output. So since we want to combine a spelling checker and a tagger, let us assume that both can be run in a sensible way non-interactively.

The Multilingual Environment

A problem often occurring in our texts (in those we want to tag as well as in those we are producing ourselfs) is the multilingual environment we are working in. We may have e. g. German, English, or Sorbian texts with single words or large chunks of other languages, e. g. quotations. It's rather painful to run them through any of the available spelling checkers since these programs usually insist on checking the whole text for one language and on producing garbage for the others. It is possible to build up combined dictionaries, but then there's always the possibility of some of the required languages missing or, even worse, a spelling error might not be detected because it is a valid form in another language. This is the main problem of multi-language checkers like Excalibur (ENGLER 1996) which suffers additionally from the lack of morphological information.

So multi-language support without mark-up is not feasible, at least not in a reliable way. It may be possible to derive the input language by sending the text through several spelling checkers for different languages to see how many errors are produced for which language as suggested in the documentation of ispell (WILLISSON 1993). This is, however, unreliable for short texts and furthermore unapplicable for texts containing words of more than *one* language. Apart from that, the same words may exist in several languages that require different hyphenation. So we do want some mark-up; in our example the parser is tailored to cope with LaTeX input.

In the linguistic department at our institute we are using an adapted LaTeX with mark-up for language switching which is needed by TeX for switching hyphenation patterns, fonts, and captions. We decided to use this mark-up for a multilingual spelling checker and tagger not only because we are using it ourselves, but also due to the assumption that if it can be done for a Turing-complete system like TeX it can be done for everything else, too.

The Hairy Input

The input format is basically LaTeX (i. e. LaTeX2e) with the following two additions: 1) the babel[1] package command \selectlanguage{<language>} switches hyphenation patterns and other language-specific things. We have added the macros \german, \english, \sorbian, \russian, etc. to make language- and font-switching more user-friendly and the characters ^, /, _ and " are active to produce the háček accent, acute accent, the polish characters *ą, ę, ż, ł,* (the last one being also used for Sorbian) and the umlaut accent, respectively. The commands used for language switching have local scope which means the parser must analyze the grouping level correctly. Apart from that, if you want to really cover all the possibilities for TeX input, affairs become notoriously pathological since hišće, hiš/ce, hi\v{s}\'ce, h{{i}{\v s}\'{c}{}}e will produce the same output[2], not to mention the possibility of entering one or more comment lines. Nonetheless, we would want our spelling checker to recognize this stuff. Furthermore, you could use \bgroup instead of { and by saying e. g. \let\foobar\bgroup define new commands with the same behaviour. In all these cases, the spelling checker must have the same (or only unsignificantly differing) notion of the input tokens as TeX.

So a sample input file for our spelling checker may look as follows:

```
\documentclass[a4paper]{article}
\usepackage{everything}
\usepackage{latin2}
\sorbian % the default language
\begin{document}
```

[1] The babel package provides multilingual support for plain TeX and LaTeX.

[2] Well, not entirely. If TeX hyphenates the word they will definitely look the same, otherwise you lose kerning in the last case. But kerning does not affect the spelling, so a spelling checker should still work reasonably.

```
\begin{verse}
Božo, bud\'z mi\l y,\\
zdźer^z strowy a \v cily\\
do skónčenja časow\\
m/o% this will be analyzes as 'mój'
j serbski lud.
\end{verse}
{\russian "Eto tol\6ko primer {\german drei deutsche Wörter}
russkogo teksta.} To je serbsce.
% sorbian again
\english
Here we are at the end of our sample document.
\end{document}
```

This mixed eight-bit/seven-bit input is not merely a theoretical possibility; quite often, texts written in latin-2 encoding or MSDOS code-page 852 style get their final corrections on a terminal not capable of eight-bit input.

The Concept

A spelling checker for the above environment must be modular for several reasons. It must give the user the possibility to add support for other languages. Furthermore, for many (small) languages there is hardly any choice which spelling checker to use, so it should be possible to integrate existing spelling checkers. Therefore, the checker and tagger is divided into a parser configurable for LaTeX, a middle part configurable for language-dependent back-ends, the back-ends themselves, and a post-processor.

The several programs are written in C, Perl, and Expect. For running them your OS should support pipes. Unix/X11 is fine for Tcl/Tk and for Perl, maybe it will also run on MS-Windows. Due to the modularity of the concept, you can have different back-ends for different languages. For adding a new file format (e. g. SGML) you simply have to supply another parser.

The Parser

The parser has the task to analyze the input text and output one line for each word according to the format

```
<fname>:<l-no>:<ftell(w-start)>-<ftell(w-end)>:<lang>:<word>.
```

The format is similar to the standard error format of compilers in order to make it easy to parse the output by emacs. For the word \v ci\l y on line 8 of our input example the following would be produced:

```
test.tex:8:181-188:sorbian:čiły
```

From what has been said above it has become clear that the parser must be highly configurable. It has similar notions about tokens as TeX itself, that is you can assign catcodes, define which commands push or pop grouping levels, which commands are always word delimiters (e. g. \quad is, while \l is not), which environments are to be skipped, and so on. The parser is written in C and based on a spelling checker written some years ago (HANNAPPEL 1991).

A sample (though not very useful) configuration file may look as follows:

```
defaultlang    sorbian
# sets the default language
language       english
language       german
# defines languages
# for the commands \<language> and \selectlanguage{<language>}
letter  Š
# can be part of a word
# the following declares _ an active character with
# four possible parameters and extensions
active  _4
a      ą
e      ę
l      ł
z      ż
# the following declares one command, here \ss to expand to ß
escape_def     1
ss     ß
# a capital letter (note the delimiter @
capital \AE@
# grouping level with no parameters
pushlevel \begingroup
pushlevel \bgroup
poplevel  \endgroup
poplevel  \egroup
# grouping level with one parameter (\begin{<param>} \end{<param>})
pushlevel1 \begin
poplevel1  \end
# the following should not be checked
skipped_command \-
# the following commands always end a word
delimiter_command       \item
delimiter_command       \footnote
# the following environments are not to be checked
skipped_environment     equation
skipped_environment     picture
# the next is mandatory since feof() does not do the right thing
# for unsigned chars
end_of_file
```

This flexibility can also be needed when checking texts written in only one language; we could think of a text containing additional diacritics for marking stress and length of a syllable (e. g. for educational reasons). It would be easy to filter out these diacritics and submit the words to a "normal" spelling checker.

The Middle Part

The middle part consists of a perl script taking the output from the parser (normally through a pipe) and doing two things:

1. writing the parser output to a file
2. deciding which back-end the word from a given line must be sent to (and send it).

It has a simple configuration file saying how to invoke the several back-ends.

The Back-ends

You can take almost anything you like. Whether you have an ispell-dictionary, a list of words grepped with an awk-script or (a)grep, a SQL-database via internet connection or a morphological analyzer along the lines of pc-kimmo, you should only know how to invoke it and how to interpret the output. You can't, however, take a program that wants to interact with the user directly, e. g. via a graphical interface, so that it won't communicate over pipes or a pty.

For each back-end you need a program munching the output of the back-end into one line of information for every word entered. In order to be usable for a spelling checker and a tagger this line should not only say whether the word was found or not but also how it was analyzed. This is a sensible thing to do also for a spelling checker to lessen the probability of spelling errors that produce valid word forms. This output is written to a file.

Tying the Loose Ends

After the session so far we have the following files: a) the parser output written by the "middle part" and b) the munched back-end output (one file per language). We now have a script mangling these files together into one saying which words of which languages couldn't be identified (for the spelling checker and the tagger) and which ones could (for the tagger only). The file is digestible by emacs's compilation-mode, so it looks more or less like the output from the parser with lines containing words successfully checked removed.

Conclusions

Tagging and Checking are combinable. The tagger profits from the fact that you do not have to build up a new dictionary from scratch which is an important point especially for small languages not well supported by commercial spelling checkers. Furthermore, the multi-lingual support enables the tagging of multi-lingual texts that otherwise wouldn't be taggable and, therefor, wouldn't be that easily accessible from a lexicographer's point of view.

The spelling checker has the big advantages of multi-language support and being able to cope with almost pathological input (LaTeX). Due to the modular concept it's easy to add new languages. It also profits from the possibility of working in batch jobs.

References

ENGLER **1996** Tobias Engler: Großes Kaliber. Mac-Rechtschreibhilfe für Text- und LaTeX-Dateien, c't 5/96, p. 82

HANNAPPEL **1991** Jürgen Hannappel, Eduard Werner: Correcting TeXts with TeXorTho, Bonn 1991 (draft)

WILLISSON **1993** Pace Willisson, Geoff Kuenning: ISPELL V3.1, Free Software Foundation 1993

Structure of a Korean Function Word Dictionary for Natural Language Processing

JAEWON YU – JEE EUN KIM – SUNGWON KOO*

Abstract

Korean function words play a fundamental role in NLP. It is because they carry not only morphological information, but also syntactic, semantic and pragmatic information although those function words should be recognized at the morphological processing. Accordingly, it is crucial to store all the required and correct information in the dictionary and to retrieve the information when appropriate. This paper describes how to encode linguistic information in a Korean function word dictionary which is implemented in a natural language processing (NLP) system.

1 Introduction

This paper describes how to encode linguistic information in a Korean function word dictionary which is implemented in a natural language processing (NLP) system. Typologically Korean is an agglutinative language which yields in inflectional morphology. A verb can occur in any of a few thousands forms, and listing all these forms in a dictionary has not been feasible. Therefore, one of the most fundamental tasks for Korean NLP is how to perform the morphological processing effectively in which the ending is separated from the stem. Our approach to accomplish the task was to build the function word[1] dictionary which contains all the well-formed forms and required information. When we constructed the dictionary, the following factors were considered:

- the types of information,
- the structure of the dictionary
- how to retrieve the information.

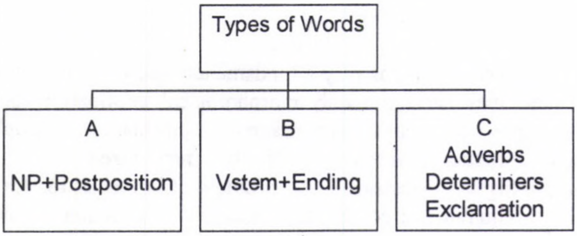
2 Structure of Korean Word

[1] Korean function words are all bound morphemes.

The basic unit of NLP for Korean is a word. Korean utilizes a space for word breaking as in English. The structure of a Korean word, however, is different from that of English one. A word can be classified into three types;

- a word attached by one or more postpositions,[2]
- a word attached by one or more endings,[3]
- a word whose stem alone can be used as a word.

The first type includes noun, pronoun and number, i.e. all types of NPs. . The second includes verb, predicate adjective,[4] auxiliary verb and auxiliary adjective. The third type includes words such as adverbs,[5] determiners such as demonstrative adjectives, exclamations, etc. (cf. Figure 1)



(Figure 1: Structure of Korean Word)

Attaching a function word to a content word is resulted from agglutination which is one of the characteristics of SOV languages according to typology (Greenberg, 1963). Korean, an SOV language also yields free word order[6] by agglutinating a postposition to a stem. Agglutination results in inflectional morphology which can be found in verb and predicate adjective for Korean.

3 Methodology

Korean function words play a fundamental role in NLP. It is because they carry not only morphological information, but also syntactic, semantic and pragmatic information although those function words should be recognized at the morphological processing. Accordingly, it is crucial to store all the required and correct information in the dictionary and to retrieve the information when appropriate. When the information is entered in the dictionary, the structure has to be considered

[2] The postpositions which allows free word order are case markers and/or delimiters.

[3] These will be referred as verbal ending throughout the paper.

[4] Korean adjective can form a predicate in a sentence without utilizing copula.

[5] In Korean, some postpositions are able to be attached to some adverbs . It is very hard to find rules for these cases. The best thing we can do is to list them all in the dictionary. We listed about 1000 such adverbs in our dictionary.

[6] Korean allows relatively free word order by which the predicate has to occur only at the end of a sentence.

carefully for efficient dictionary lookup.

For morphological processing of Korean, implementing automata has been the most popular approach. However, it is not efficient since only 6 morphemes can be concatenated to a stem at the most. In the word,

"salam - eykey-puto-mankum-man-un-ja"(man - to - from - at least - only - contrast- emphasis)
from the man at least

six postpositions are attached to the noun stem. This word includes many redundant expressions and the grammaticality of this word is pretty low at the best. In most cases, the number of the concatenated postpositions is limited to two or three. Furthermore, the recent research shows that the most frequent thirty-five postpositions cover about 97% of the whole occurrences, of which only five are concatenated forms by two postposions. Regarding the verbal endings, the sixty high frequent endings covers 95%, of which no concatenated form is found(S. Kang , 1995).

The second problem of the automata approach is the number of morphemes which has to be identified in the initial state. In Korean dictionary, about 140 postpositions and over 400 verbal endings are listed as entries. But if we consider the allomorphs of endings for the different inflectional classes of verbs, the number of the initial state of the automata will amount to over one thousand. Specifying over one thousand states is complicated and ineffective.

Moreover, the result from the analysis using automata becomes over-powerful, which allows ungrammatical combinations of the morphemes. Ungrammatical combinations arises most easily and frequently between a pre-final ending and a final ending.[7] There are morphotactic constraints between them, but it is very hard to implement these constraints in an automaton.

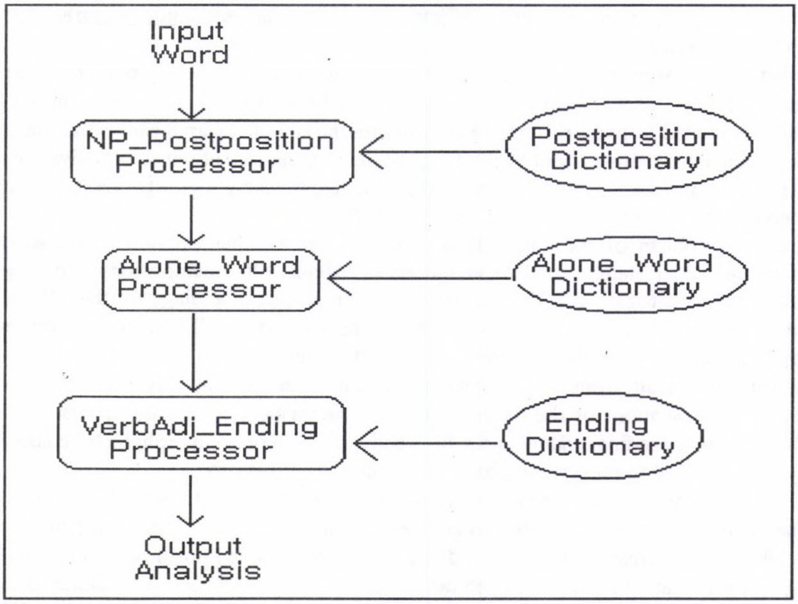
To complement the shortcomings of utilizing automata, we attempted to list up all the possible well-formed combinations of postpositions and verbal endings. First, the combinations were generated automatically using automata. In order to do this task, we classified postpositions into three categories according to their relative positions to the noun stem. The first class can be attached directly to the stem, the second can come between the other postpositions, and the third at the word-final position. All three classes can be attached directly to the stem, but when they concatenate each other, the relative position between them is fixed.[8] For the verbal endings, we made all possible combinations between the pre-final endings and the final endings. The result was manually reviewed and ungrammatical ones are discarded. Since 1992 when the dictionary was constructed, it is updated with 500 entries by running corpora. Most of them are spoken forms, which are not registered in the dictionaries. The current list contains 1084 for postposition and 5510 for verbal ending.

The function word dictionary is further divided into 2 classes; one contains postpositions and the other is for verbal endings. By separating these two, POS information of the stem can be extracted when the function word is detected. If a postposition is identified, then the stem is considered as a noun. The same logic applies to the verbal ending. With respect to the dictionary lookup, the postposition is searched first because 60% of commonly used words are noun. If the input word is not matched with a noun, a word without an ending is searched. The total number of this type is 8,800. Then, the verb is processed.(cf. Figure 2). In the course of identifying a function word, right-to-left

[7] With the term "pre-final ending", we refer to those particles which can be attached directly to the verbal stem, before other verbal endings. In Korean, there are three kinds of pre-final endings; -(u)si-(honorific for subject marker), -ess -(past tense marker), and -kess-(presumption marker).

[8] We had to find this fact heuristically. There ware no researches on this issue.

search method is adopted. For an effective right-to-left search, those function words are listed in the dictionary by reversed order. In addition, the longest match strategy is applied when detecting a function word.



(Figure 2 : Flow Chart)

4 Types of information and the structure of the Action Code

To store the information carried by the function words, one byte is allocated to each dictionary. Each bit is set to true or false; if a certain constraint is required for grammaticality, the bit is set to true. We will explain the types of information and the structure of the "action code" with some examples.

Postposition:	wa	01 110 000
	kwa	10 111 000
	poute	11 111 000
Verbal Ending:	senun	01 110 100
	nikka	01 111 000
	nunta	01 100 000
	nunya	11 100 000
	lkka	01 111 001

For both postposition and verbal endings, first two bits are filled with morphophonemic information, whether the stem ends in a consonant or a vowel. The first code specifies if the postpositions and the verbal endings can be attached to a consonant stem. If the value is set as 1, it can be attached to a consonant stem. If it is set as 0, it cannot. The second code specifies the same information for the vowel stem. The next three bits are used to specify POS of the stem. For the postposition, the third, the forth, and the fifth code specifies if the postposition can be combined with Noun, Pronoun, and Number respectively. For the verbal ending, on the other hand, these three bits specify if the ending can be attached to the Verb, Adjective and Copula respectively. The next three bits are set only verbal ending, specifying verb and predicate adjective conjugation class and aspiration. The sixth bit shows that the ending follows a contract Verb or Adjective. If the seventh bit is set as 1, it tells the ending is used only for those Verbs of Adjectives which have aspirated shortened forms. The last bit specifies if the ending can be attached to Noun stem directly as a copula or not. The last three bits for the postpositions are reserved for the future use. The information is utilized by setting a flag during runtime.

This approach to morphological processing using the function word dictionary has been implemented in various Korean versions of Microsoft Office products such as spelling checker, Answer Wizard which is a help function, etc. Currently it is under implementation for various applications including a stemmer in the Office products and NLPWIN which is a natural language system developed at Microsoft Research.

5 References

H.C. Kwon, Y.S. Chae, and G.O. Jeong (1991) “A Dictionary-based Morphological Analysis”, Proceedings of of Natural Language Processing Pacific Rim Synposium, pp. 87-91.

H.C. Kwon and L. Karttunen (1994) “Incremental Construction of Lexical Tranducer for Korean”, Proceedings of the 15th International Conference on Computational Linguistics(COLING 94), Vol.2, pp. 1262-1266.

J.I. Kwon (1992) “Korean Syntax”, Mineumsa, Seoul.

J.S. Seo (1996) “A Korean Grammar”, Purikipheunnamu, Seoul.

J.W. Yu (1985) “A Reverse Dictionary of Modern Korean”, Jeongeumsa, Seoul.

K. Nam (1985) “The Standard Korean Grammar”, Tap Publishing Company, Seoul.

K.S. Choi, D.B. Kim, S.J. Lee, and G.C. Kim (1994) “A Two-level Morphological Anaysis of Korean”, Proceedings of the 15th International Conference on Computational Linguistics(COLING 94), Vol.1, pp. 535-539.

K.S. Choi & et. Al. (1992) “The Research on Korean Spelling Correction and Parting of Words”. The Second Year Technical Report, KAIST, Daejeon.

“Sae Kuk-Eo-Sajeon” (1994) Dong-A Publishing Company, Seoul.

Sproat, R. (1992) “Morphology and Computation”, The MIT Press, Cambridge, Massachusetts.

S.S. Kang (1993) “Analysis of morphemes of Korean Based on the Syllable Information and Information and Multi-word Information”, Ph.D. Dissertation, Seoul National University, Seoul.

S.S. Kang (1995) “A Construction Josa/Eomi Dictionary using Relative Frequency”, Proceedings of the 7th Conference on Hangul and Korean Computational Linguistics, Korean Society of Cognitive Science, pp. 188-194.

S.S. Kang, and Y.T. Kim (1994) “Syllable-based model for Korean Morphology”, Proceedings of the 15th International Conference on Computational Linguistics(COLING 94), Vol.1, pp. 221-226.

W. Chang, S.H. Yuh, H.M. Jung, T.W. Kim, D.S. Hwang, and D.I. Park (1995) "A Korean Generator using Left-Right Connectivity Information", Proceedings of the 7th Conference on Hangul and Korean Computational Linguistics, Korean Society of Cognitive Science, pp. 121-130.

A. Heo (1996) "Morphology of 20th Century Korean", Saemmunhwasa, Seoul.

List of Participants

I. ADURIZ
UZEI
Aldapeta, 20.
Donostia, THE BASQUE COUNTRY, ES-20009
uzei0005@sarenet.es

Izaskun ALDEZABAL
Department of Computer Languages and Systems University of The Basque Country
Informatika Fakultatea
649 P.K.
Donostia, THE BASQUE COUNTRY, ES-20080

X. ARTOLA
Department of Computer Languages and Systems University of The Basque Country
Informatika Fakultatea
649 P.K.
Donostia, THE BASQUE COUNTRY, ES-20080

Emese BÁLINT
Károli Gáspár Egyetem BTK
Napos u. 5/b.
Budapest, HUNGARY, H-1125

R. BELRHALI
Institut de la Communication Parlée INPG-Université Stendhal
BB 25X, Cedex 9
Grenoble, FRANCE, F-38040
belrhali@grenet.fr

L-J. BOË
Institut de la Communication Parlée INPG-Université Stendhal
BB 25X, Cedex 9
Grenoble, FRANCE, F-38040

Elisabeth BREIDT
Seminar für Sprachwissenschaft Universität Tübingen
Wilhelmstr. 113
Tübingen, GERMANY, D-72074
breidt@sfs.nphil.uni-tuebingen.de

J. COURTIN
Communication Langagière et Interaction Personne-Système
BB 25X, Cedex 9
Grenoble, FRANCE, F-38040

Emőke CSALLOS
Károli Gáspár Egyetem BTK
Napos u. 5/b.
Budapest, HUNGARY, H-1125

Hai DOAN NGUYEN
GETA-CLIPS-IMAG
BP 53, Cedex 9
Grenoble, FRANCE, F-38041
hai.doan-nguyen@imag.fr
Phone: xx33/76/51 43 80
Fax: xx33/76/51 44 05

Daniele DUJARDIN
Communication Langagière et Interaction Personne-Système
BB 25X, Cedex 9
Grenoble, FRANCE, F-38040
daniele.dujardin@imag.fr

Nerea ECEIZA
Department of Computer Languages and Systems University of the Basque Country Informatika Fakultatea
649 P.K.
Donostia, THE BASQUE COUNTRY, ES-20080
jibecran@si.ehu.es

Judith ECKLE
Institut für maschinelle Sprachverarbeitung Universität Stuttgart
Azenbergstr. 12
Stuttgart, GERMANY, D-70174
eckle@ims.uni-stuttgart.de

Helmut FELDWEG
Seminar für Sprachwissenshaft Universität Tübingen
Wilhelmstr. 113
Tübingen, GERMANY, D-72074
helmut.feldweg@uni-tuebingen.de
feldweg@sfs.nphil.uni-tuebingen.de

Gunter GEBHARDI
Humboldt-Universität zu Berlin Philosophische Fakultät II.
Jägerstr. 10/11
Berlin, GERMANY, D-10099
gebhardi@compling.hu-berlin.de

Alexander GEYKEN
Institut für deutsche Sprache Abt. Sprachentwicklung der Gegenwart
Postfach 101621
Mannheim, GERMANY, D-68016
geyken@ids-mannheim.de

Ana GONÇALVES
Museu Nacional / UFRJ
Rua Marques de Abrantes
185/805 Flamengo,
Rio de Janeiro, BRAZIL, BR-22230-060
recanto@unisys.com.br

Gregory GREFENSTETTE
Rank Xerox Research Centre, Grenoble Laboratory
6 chemin de Maupertuis
Meylan, FRANCE, F-38240
grefen@grenoble.rxrc.xerox.com

Maurice GROSS
LADL, Université Paris
2 place Jussieu, Cedex 05
Paris, FRANCE, F-75221
mgross@ladl.jussieu.fr

Sun-Hae HAN
Institut Gaspard Monge, Université de Marne-la-Vallée
2, rue de la Butte verte,
Noisy-le-Grand Cedex, FRANCE, F-93166
han@monge.univ-mlv.fr

Ulrich HEID
Institut für maschinelle Sprachverarbeitung Universität Stuttgart
Azenbergstr. 12
Stuttgart, GERMANY, D-70174
uli@ims.uni-stuttgart.de

Johannes HEINECKE
Humboldt Universität zu Berlin Computerlinguistik
Jägerstr. 10/11
Berlin, GERMANY, D-10099
heinecke@compling.hu-berlin.de

Tove JACOBSEN
French Department
HF-BYGGET, SYDNESPLASS 7
Bergen, NORWAY, NO-5007
tove.jacobsen@roman.uib.no

Primoz JAKOPIN
Institute of the Slovene Language Centre for Scientific Research of the Slovenian Academy of Sciences and Arts
Gosposka ul. 13,
Ljubljana, SLOVENIA, SI-1000
primoz.jakopin@uni-lj.si

Ferenc KIEFER
Research Institute for Linguistics Hungarian Academy of Sciences
P. O. Box 19
Budapest, HUNGARY, H-1250
kiefer@nytud.hu

Jee Eun KIM
Microsoft Corporation
120-6 Yonhee-dong
Sodaemun-ku, 120-111
Seul, KOREA (SOUTH
jeeeunk@microsoft.com

Gábor KISS
Research Institute for Linguistics Hungarian Academy of Sciences
P. O. Box 19
Budapest, HUNGARY, H-1250
kiss@nytud.hu

Lajos KISS
Research Institute for Linguistics Hungarian Academy of Sciences
P. O. Box 19
Budapest, HUNGARY, H-1250

Sungwon KOO
Centre for Computational Linguistics University of Manchester Institute of Science and Technology
Language Engineering Dept., UMIST
P. O. Box 88
Manchester, UK, M60 1QD
sungwon@ccl.umist.ac.uk

Cvetana KRSTEV
Faculty of Mathematics
Studentski Trg 16
Beograd, YUGOSLAVIA, 11000

Ramesh KRISHNAMURTHY
COBUILD, University of Birmingham Cobuild Institute of Research and Development Birmingham Research Park
Vincent Drive
Birmingham, UK, B15 2SQ
ramesh@cobuild.collins.co.uk

Andrea KROTT
MAX-PLANC-Institut für Psicholinguistik
Wundtlaan 1,
Nijmegen, THE NETHERLANDS, NL-6525 XD
akrott@mpi.nl

Udo KRUSCHWITZ
Humboldt-Universität zu Berlin Philosophische Fakultät II., Computerlinguistik
Jägerstr. 10/11
Berlin, GERMANY, D-10099
kruschwi@compling.hu-berlin.de

Heok-Seung KWON
School of English University of Birmingham
Edgbaston, UK, B15 2TT
h.s.kwon@bham.ac.uk

Stefan LANGER
MicroCentre, University of Dundee CIS, University of Munich
Oettingenstr. 67
Munich, GERMANY, D-80538
stef@cis.uni-muenchen.de

Ann LAWSON
Corpus Linguistics School of English Edgbaston
Birmingham, UK, BIS 2TT
a.e.lawson@bham.ac.uk

Péter LÁZÁR
ELTE SEAS Angol nyelvészeti tanszék
Ajtósi Dürer 19.
Budapest, HUNGARY, H-1146
lapid@osiris.elte.hu

Chang Yeol LEE
Institut Gaspard Monge Université de Marne-la-Vallée
2, rue de la Butte verte,
Noisy-le-Grand Cedex, FRANCE, F-93166

Tamás MAGAY
Károli Gáspár Egyetem BTK Angol Intézet
Napos u. 5/b.
Budapest, HUNGARY, H-1125

Petra MAIER
CIS, University of Munich
Oettingenstr. 67
Munich, GERMANY, D-80538
pmaier@cis.uni-muenchen.de

Hansen MALCOLM
c/o ConText Server Group Oracle Corporation
500 Oracle Parkway
Box 659510
Redwood Shores California, USA, 94065
mhansen@us.oracle.com

Louise MANGA
Department of Linguistics University of Ottawa
Ottawa, Ontario K1N 6N5
CANADA
s052277@aix1.uottawa.ca

Baiba METUZĀLE-KANGERE
Department of Baltic Studies Stockholm University
Stockholm, SWEDEN, S-10691
bka@balt.su.se

Simonetta MONTEMAGNI
ILC, CNR
Via della Faggiola 32
Pisa, ITALY, 56126
simo@ilc.pi.cnr.it

Karin MÜLLER
Student at Universitat Stuttgart Institut für Machinelle Sprachverarbeitung
Sonnenbergstr. 14
Stuttgart, GERMANY, D-70184
muellekn@studenten.ims.uni-stuttgart.de

Jee-Sun NAM
Institut Gaspard Monge University of Marne-la-Vallée
2 Rue de la Butte verte
Noisy-le-Grand Cedex, FRANCE, F-93166
nam@univ-mlv.fr

Jürgen OESTERLE
CIS, University of Munich
Oettingenstr. 67
Munich, GERMANY, D-80538
joe@cis.uni-muenchen.de

Csaba ORAVECZ
Research Institute for Linguistics Hungarian Academy of Sciences
P. O. Box 19
Budapest, HUNGARY, H-1250
oravecz@nytud.hu

Judit PAIS
Research Institute for Linguistics Hungarian Academy of Sciences
P. O. Box 19
Budapest, HUNGARY, H-1250
pais@nytud.hu

Júlia PAJZS
Research Institute for Linguistics Hungarian Academy of Sciences
P. O. Box 19
Budapest, HUNGARY, H-1250
pajzs@nytud.hu

Richard PIEPENBROCK
CELEX, Max Planck Institute for Psycholinguistics
Wundtlaan 1,
Nijmegen, THE NETHERLANDS, NL-6525 XD
celex@mpi.nl

Dimitar G. POPOV
Institute for Bulgarian Language Bulgarian Academy of Sciences
bull. Schipchenski prohod 52
Sofia, BULGARIA, BG-1113
dpopov@bgearn.acad.bg
Phone: xx35/22/713 2958

Oueslati ROCHDI
ERIC-ENSAIS
24, bd de la Victoire
Strasbourg-Cedex, FRANCE, F-67084
rochdi@eric.u-strasbg.fr

Ferenc ROVNY
Kossuth Lajos University, Debrecen Foreign Language Centre
P. O. Box 41
Debrecen, HUNGARY, H-4010
rovnyf@tigris.klte.hu

Morris SALKOFF
LADL, Université Paris-7
2, Place Jussieu, Cedex 05
Paris, FRANCE, F-75221
salkoff@ladl.jussieu.fr

Uģis SARKANS
Artifical Intelligence Laboratory Institute of Mathematics and Computer Science
University of Latvia
Raina bulvaris 29,
Riga, LATVIA, LV-1459
usarkans@ailab.mii.lu.lv

Jana SCHULZE
Sorbisches Institut e. V.
Bahnhofstr. 6
Bautzen, GERMANY, D-02625

Dieter SEELBACH
Institut für Allgemeine und Vergleichende Sprachwissenschaft Universitat Mainz
FB 14-20
Mainz, GERMANY, D-55099
Private Fax and Phone: xx49/06192 22925
Fax: xx49/ 6131 395100

Frédérique SEGOND
Rank Xerox Research Center
6, chemin de Maupertuis
Meylan, FRANCE, F-38240
segond@xerox.fr
segond@ grenoble.rxrc.xerox.com

Irene ŠĚRAK
Sorbisches Institut e. V.
Bahnhofstr. 6
Bautzen, GERMANY, D-02625

Kiril Iv. SIMOV
Linguistic Modelling Laboratory Bulgarian Academy of Sciences
Acad. G. Bonchev St. 25 A
Sofia, BULGARIA, BG-1113
kivs@bgcict.acad.bg

Wolfgang TEUBERT
Institut für deutsche Sprache
Postfach 101621
Mannheim, GERMANY, D-68016
telri@ids-mannheim.de

László TIHANYI
Research Institute for Linguistics Hungarian Academy of Sciences
P. O. Box 19
Budapest, HUNGARY, H-1250
tihanyi@nytud.hu

Agnes TUTIN
URA SILEX Université de Lille III
BP 149
Villeneuve d' Ascq Cèdex, FRANCE, F-59653
tutin@univ-lille3.fr

Harald ULLAND
Department of Romance Studies University of Bergen
Sydnesplass 7
Bergen, NORWAY, N-5007
harald.ulland@roman.uib.no

Ruben URIZAR
Department of Computer Languages and Systems University of the Basque Country Informatika Fakultatea
649 P.K.
Donostia, THE BASQUE COUNTRY, ES-20080
jiburenr@si.ehu.es

Giuseppe VALETTO
Rank Xerox Research Center
6, chemin de Maupertuis
Meylan, FRANCE, F-38240
valetto@mailer.cefriel.it

Tamás VÁRADI
Research Institute for Linguistics Hungarian Academy of Sciences
P. O. Box 19
Budapest, HUNGARY, H-1250
varadi@nytud.hu

Lidia VARGA
LADL, Université Paris 7 - BME, Nyelvi Intézet
Erdősor u. 40. IV/13.
Budapest, HUNGARY, H-1214
lvarga@ladl.jussieu
Phone: xx31/1/277 1048

Ildikó VILLÓ
Research Institute for Linguistics Hungarian Academy of Sciences
P. O. Box 19
Budapest, HUNGARY, H-1250

Duško VITAS
Faculty of Mathematics
Studentski Trg 16
Beograd, YUGOSLAVIA, 11000
xpmfl02@yubgss21.bg.ac.yu

Eduard WERNER
Sorbisches Institut e. V.
Bahnhofstr. 6
Bautzen, GERMANY, D-02625
edi@kaihh.hanse.de

Karsten L. WORM
Universität des Saarlandes Computerlinguistik
Postfach 15 11 50
Saarbrücken, GERMANY, D-66041
worm@coli.uni-sb.de

Jaewon YU
Microsoft Corporation & Hankuk University of Foreign Studies
Woosung Apt. 5-605, Bono-dong,
Ansan-si, Kyunggi-do, KOREA (SOUTH
t-jwyu@microsoft.com