Papers from the Seventh Scandinavian Conference of Computational Linguistics

Reykjavík 1989

Edited by

JÖRGEN PIND
and
EIRÍKUR RÖGNVALDSSON

Reykjavík 1990
Institute of Lexicography
Institute of Linguistics

These proceedings are published with financial aid from IBM

Contents

Part III: Computational Lexicography

Preface

The present volume is a collection of papers that were read at the Seventh Scandinavian Conference of Computational Linguistics (Nordiska Datalingvistikdage) and the Symposium on Computational Lexicography and Terminology in Reykjavík, June 26th–28th, 1989.

The Conference and the Symposium were jointly organized by the Institute of Linguistics and the Institute of Lexicography, University of Iceland. In addition to the editors of this volume, Sigurður Jónsson, cand.mag., of Iðunn Publishing Co., was a member of the organizing committe.

The book is divided into three sections: Morphology, Syntax and Discourse Analysis (8 papers), Machine Translation (15 papers), and Computational Lexicography (7 papers).

A few papers were presented at the conference but not received for publication.

The editors would like to thank Sigurður Jónsson for taking the initiative in holding the Conference and the Symposium in Iceland, and for his assistance in organizing these events. Thanks are due to Björn Þór Svavarsson, and Friðrik Magnússon, of the Institute of Lexicography, for help in bringing out these proceedings.

Jörgen Pind and Eiríkur Rögnvaldsson

Part I

Morphology, Syntax, and Discourse Analysis

STEFÁN BRIEM

Automatisk morfologisk analyse af islandsk tekst

Abstract

Automatic Morphological Analysis of Icelandic Text

One of the projects worked on at the Institute of Lexicography at the University of Iceland is a frequency analysis of Icelandic vocabulary and grammar. The most time-consuming part of the work consists in morphological analysis of text samples containing in all more than half a million running words. For every single word the analysis results in registration of the word class, the flexion form and the lemma to which the text word belongs.

If manually performed, this kind of analysis would be enormously monotonic work requiring high precision. A method has been developed to perform the analysis to a great extend automatically using a computer. However the manual work can not be eliminated, but it has already been reduced significantly, and at the same time the character of the manual work is altered to be mainly a matter of correcting activity.

The method of automatic analysis is based on a corpus of tags and word forms originating in a previous manually performed analysis of more than 54,000 text words and on a set of rules for possible relations between words of the same sentence. On the basis of frequencies compiled by the previous analysis and of points given by the rules when fulfilled, a computer program automatically selects the probably 'best sentence' among a (usually great) number of homograph sentences. Furthermore, in case of words not found in the collection of word forms, the program makes use of a collection of more than 5,000 back parts of word forms in order to make an intelligent guess.

At the current stage the result of the automatic analysis is completely correct for about 70% of the text words and partly correct for about 15%. Our experience shows that the manual effort is reduced by about 2/3. By extension of the word form collection and by improvement of the relation rules and points giving, a significant improvement of the automatic analysis is expected in the near future, e.g. leading to 85%–90% of text words being correctly analysed.

1 Indledning

Automatisk morfologisk analyse af islandsk tekst hører til en af Leksikografisk Instituts opgaver. Det drejer sig om en fase af et større projekt, som er udarbejdelse af en islandsk frekvensordbog. Bogen skal give oplysninger om brugen af islandsk nutidssprog, det skriftlige sprog, i form af forskellige slags oversigter over hyppigheden af ord, ordformer, bøjningsformer, ordklasser o.s.v.

Det kan give en idé om projektets omfang at det vil omfatte ca. en halv million tekstord. Langt den største del af arbejdet ligger i den morfologiske analyse, som bliver særlig omhyggeligt udført. I den sidste ende vil analysen blive manuelt udført eller i hvert fald manuelt kontrolleret. Men det har allerede vist sig at arbejdet reduceres i betydelig grad ved at man i første omgang udfører analysen automatisk ved hjælp af en datamat.

2 Formål

Formålet med den morfologiske analyse uanset om den bliver udført manuelt eller maskinstøttet er følgende.

For det første skal analysen for hvert enkelt tekstords vedkommende føre til registrering af ordklasse og bøjningsform.

Desuden registrerer man hvilket kasus verber og præpositioner styrer. De fleste ord, som man plejer at klassificere som præpositioner, bruges i nogen tilfælde som adverbier, og omvendt, mange ord, som traditionelt klassificeres som adverbier, bruges også som præpositioner. I dette projekt har man derfor valgt at behandle præpositioner og adverbier under ét, hvilket medfører, at man også registrerer adverbiers kasusstyrelse, når den forekommer.

Endvidere registrerer man for hvert tekstord det tilhørende leksikonsord, også kaldt lemma.

Som et eksempel tager vi følgende korte tekst og det tilstræbte resultat af dens analyse:

TEKST:

```
Það er þriðjudagur í dag. Magnús kemur á
morgun. Hann dvaldist ásamt dr. Jósteini
Samúelssyni alllengi í Danmörku. Félagi
hans hefur orðið eftir.
```

ANALYSE:

f p h e n	það	það
s f g 3 e n	er	vera
n k e n	þriðjudagur	þriðjudagur
a o	í	í
n k e o	dag	dagur
n k e n s	Magnús	Magnús
s f g 3 e n	kemur	koma
a o	á	á
n k e o	morgun	morgunn
f p k e n	hann	hann
s f m 3 e þ	dvaldist	dvelja
a þ	ásamt	ásamt
n k e þ	dr.	dr.
n k e þ s	Jósteini	Jósteinn
n k e þ s	Samúelssyni	Samúelsson
a a	alllengi	alllengi
a þ	í	í
n v e þ s	Danmörku	Danmörk
n k e n	félagi	félagi
f p k e e	hans	hann
s f g 3 e n	hefur	hafa
s s g	orðið	verða
a a	eftir	eftir

I den midterste kolonne har vi tekstordene, ét i hver linie, og til højre de tilhørende leksikonsord. Til venstre har vi så tegn for de grammatiske oplysninger som analysen har ført til. Det første tegn står for ordklasse. For hvert tekstord er der højst 7 grammatiske tegn i en bestemt rækkefølge. Den kan man kalde en grammatisk streng. En blank linie betegner begyndelsen af en ny sætning.

3 Den automatiske analyse

Den metode, som man her benytter til automatisk morfologisk analyse, er baseret på en tidligere manuelt udført analyse af godt 54.000 tekstord (Friðrik Magnússon 1988) og fungerer ved hjælp af et sæt af morfologiske regler og ved betragtning og vurdering af sandsynligheder.

Hovedanalysen udføres med én sætning ad gangen. Det må derfor være helt klart, hvor hver sætning begynder. For at opnå det udføres der en foranalyse som har til formål at registrere startfeltet for hver sætning af den tekst, som skal analyseres.

3.1 Foranalyse

Foranalysen udføres halvautomatisk, d.v.s. i samarbejde mellem menneske og maskin.

TEKST:

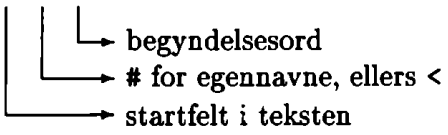
```
Það er þriðjudagur í dag. Magnús kemur á
morgun. Hann dvaldist ásamt dr. Jósteini
Samúelssyni alllengi í Danmörku. Félagi
hans hefur orðið eftir.
```

BEGYNDELSESORD:

```
< Það
< Hann
< Í
< Ég
< Og
< En
< Þá
< Hún
< Þegar
< Við
< Á
< Þar
< Þetta
< Nú
< Um
< Þú
< Að
< Ekki
< Hér
< Af
......
......
```

POINTERREGISTER:

```
   0 < Það
  26 # Magnús
  49 < Hann
 115 < Félagi
```



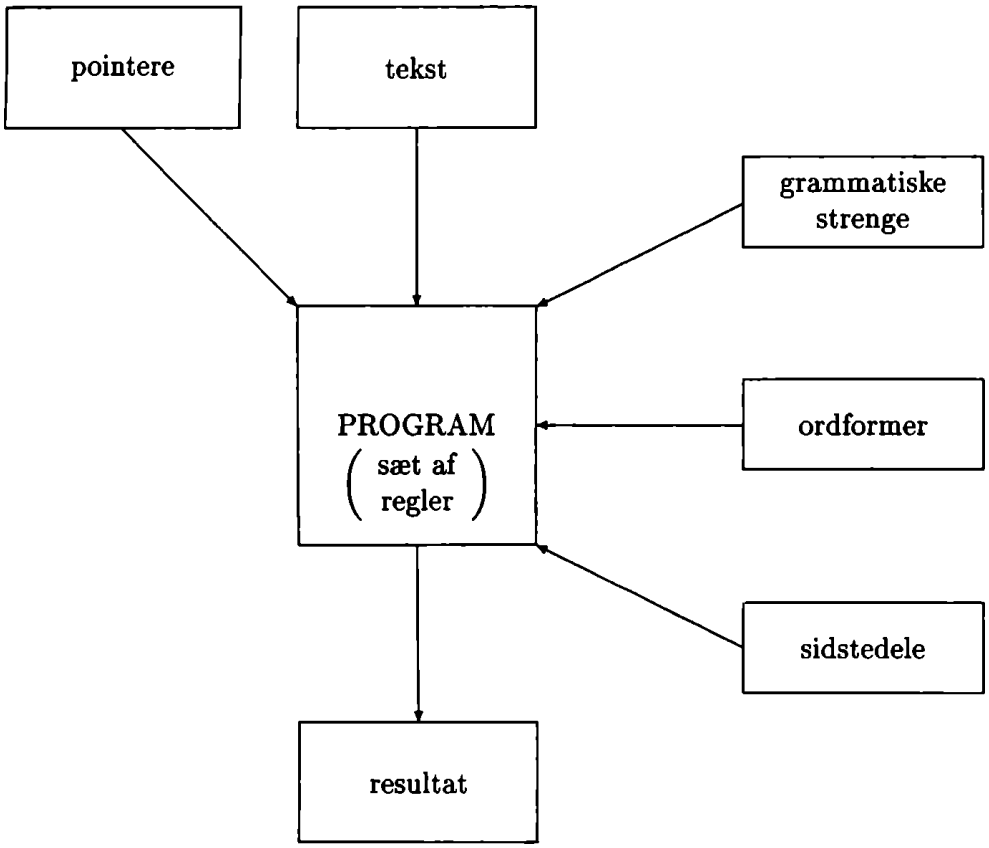
Her har vi den samme tekst igen. Til højre er vist de første 20 ord i en samling af 649 begyndelsesord, ordnet efter hyppighed. Disse 20 ord er altså de hyppigste i begyndelsen af islandske sætninger, i hvert fald i den tekst som samlingen er baseret på. Det program, som udfører foranalysen, bruger denne samling til automatisk at finde ud af de fleste sætningsbegyndelser i en tekst. Når der er tvivl stopper programmet op, inviterer mennesket til en afgørelse og fortsætter når en beslutning er truffet.

I dette eksempel må et menneske træde til for at afgøre, at Magnús er et proprium og at Jósteinn ikke er begyndelsen af en ny sætning.

Resultatet, et pointerregister, bruges så sammen med teksten under hovedanalysen, den egentlige automatiske morfologiske analyse, som vi nu går over til at betragte.

3.2 Den automatiske hovedanalyse

SYSTEMETS BESTANDDELE



Dette billede viser systemets bestanddele. I centrum er et program, hvori der er indbygget et sæt af 75 regler, især morfologiske regler. Programmet gør brug af en samling af ordformer og en samling af sidstedele af ord. Det er programmets opgave for hvert enkelt ord i en given tekst at finde frem til den mest sandsynlige streng blandt de 566 forskellige grammatiske strenge, samt at finde det tilhørende leksikonsord.

Vi vil nu betragte de enkelte bestanddele nærmere.

3.2.1 Støtteregistre

Lad os begynde med de 3 støtteregistre som programmet bruger uændrede fra tekst til tekst.

Det første støtteregister indeholder 566 grammatiske strenge i alfabetisk rækkefølge. Tallene er hyppighedstal.

GRAMMATISKE STRENGE:

```
  434 a a
   27 a a e
   34 a a m
   17 a e
   40 a o
   13 a u
.....
.....
   25 n v f þ g
    1 n v f þ g s
   24 s b g 2 e n
    1 s b g 2 f n
   57 s f g 1 e n
.....
.....
```

Det andet støtteregister indeholder knap 18.000 ordformer i alfabetisk rækkefølge sammen med det tilhørende leksikonsord og længst til venstre nummeret på en grammatisk streng. Hver ordform kan optræde mange gange p.g.a. forskelle i leksikonsord og forskellige bøjningsformer. Tallene i anden kolonne er hyppighedstal fra den tidligere analyse.

ORDFORMER:

```
.....
.....
  421     1  vélstjórum   vélstjóri
  453     5  vélum        vél
  455     3  vélunum      vél
  116     4  vér          ég
  116   171  við          ég
    0    52  við          við
    4   498  við          við
    6    27  við          við
  484     1  viða         viða
  215     1  viðamesta    viðamikill
  208     1  viðamikið    viðamikill
.....
.....
```

Og det tredje støtteregister indeholder godt 5.000 sidstedele af ordformer i baglæns alfabetisk rækkefølge, hver sammen med den tilsvarende sidstedel af et leksikonsord og nummeret på en grammatisk streng. Tallene i anden kolonne er sandsynlighedstal som man kan knytte til hver sidstedel for at styre programmets anvendelse af dette register.

SIDSTEDELE:

```
  377   0  trjáa      tré
  287   0  ædda       ældur
  311   0  ædda       ældur
   41   0  alda       öld
  287   1  falda      faldur
  311   1  falda      faldur
  377   0  halda      hald
  377   0  elda       eldi
.............
.............
  377   0  efla       efli
  429   0  regla      regla
  377   0  bila       bil
  373   0  heimila    heimili
  377   0  heimila    heimili
  287   0  mikla      mikill
  311   0  mikla      mikill
  409   0  jökla      jökull
  417   0  jökla      jökull
  377   0  falla      fall
  377   0  fjalla     fjall
  409   0  valla      völlur
.............
.............
```

3.2.2 Sæt af regler

Lad os nu betragte det regelsæt som er direkte indbygget i programmet. Til hver regel er der knyttet et antal points, som gives hver gang reglen er opfyldt.

Programmet arbejder med én sætning ad gangen. Det slår tekstordene op i samlingen af ordformer. I de fleste tilfælde findes der nogle muligheder for hvert tekstord. Det kan føre til et stort antal af homografe sætninger, der bliver kandidater til stillingen 'den bedste sætning'. Hvis tekstordene betragtes hver for sig uanset deres stilling i sætningen, er det umuligt at afgøre hvilken af de homografe sætninger er den rigtige eller den bedste.

Men her træder reglerne til. For hver enkelt af de homografe sætninger kalkulerer programmet det totale antal points som reglerne giver hver gang de er opfyldt. Den sætning, som får de fleste points, anses for at være den bedste og bliver valgt som analysens resultat.

I praksis kan antallet af homografe sætninger blive så enormt, at det ville tage datamaten måneder eller endog år at betragte dem alle. Men da antallet er kendt tidligt under processen, løses dette problem nemt ved optimalisering af kun en del af sætningen ad gangen. Det medfører en nedsættelse af analysens korrekthed på kun ca. 1%.

De fleste af reglerne er meget enkle og kan derfor nemt omskrives til et programmeringssprog. Her er vist nogen få af reglerne i dansk oversættelse. Tallene er de tilhørende points.

- Hvis der eftir et faldstyrende adverbium følger et ikke faldstyrende adverbium og derpå følger et faldbøjet ord, så vil det faldbøjede ord stå i det fald som det første adverbium styrer. **400**

- Det er sandsynligt at *að* er en konjunktion, hvis et ukendt ord følger efter. **500**

- Hvis der efter et adjektiv følger et substantiv, så har de næsten altid samme køn, tal og fald. **1000**

- Hvis der efter et adjektiv følger et substantiv i bekendt form, så er der større sandsynlighed for at adjektivet har bestemt form end ubestemt. **10**

- Et verbum i perfektparticipium er sandsynligt, hvis det følgende ord er verbet *vera* eller hvis verbet *vera* er et af de to foranstående ord. **200**

Kongruensbøjning er en af de stærkeste støttepiller for den automatiske analyse. Et eksempel på kongruensbøjning har vi her i tredje regel i tilfælde af substantiv og tilhørende adjektiv. Andre analoge regler for kongruensbøjning omfatter også pronominer og talord.

3.2.3 Resultat

Korrektheden af den automatiske analyse fremgår af sammenligning mellem den på næste side anførte og den tidligere viste korrekte analyse af samme tekst. I dette eksempel har programmet bl.a. taget fejl af tekstordet **félagi**. Ifølge den automatiske analyse skulle det være singularis dativ af substantivet *félag* som betyder *forening*, men i virkeligheden drejer det sig om singularis nominativ af substantivet *félagi* som betyder *kammerat*. Lidt senere får vi flere eksempler på homografi.

AUTOMATISK ANALYSE:

f p h e n	það	það
s f g 3 e n	er	vera
n k e n #	þriðjudagur	þriðjudagur
a o	í	í
n k e o	dag	dagur
n k e n s	Magnús	Magnús
s f g 3 e n	kemur	koma
a o	á	á
n k e o	morgun	morgunn
f p k e n	hann	hann
	dvaldist	dvaldist
a þ	ásamt	ásamt
	dr.	dr.
n k e þ s #	Jósteini	Jósteinn
n k e þ s #	Samúelssyni	Samúelssonur
a a	alllengi	alllengi
a þ	í	í
n v e þ s	Danmörku	Danmörk
n h e þ	félagi	félag
f p k e e	hans	hann
s f g 3 e n	hefur	hafa
s s g	orðið	verða
a þ	eftir	eftir

betyder at analysen er baseret på ordets sidstedel.

3.3 Præstation og kvalitet

Resultatet af analysen af en prøvetekst på 5.000 ord blev:

Præstation:

Maskin/menneske:	Foranalyse	2– 5	min.
Maskin:	Hovedanalyse	15–20	min.
Menneske:	Korrektur	20	timer

Kvalitet:

70%	tekstord korrekt analyseret
15%	tekstord ukorrekt analyseret
15%	tekstord slet ikke analyseret

Foranalysen tager næsten ingen tid. Hovedanalysen tager heller ikke lang tid og det er jo datamatens tid. Størstedelen af arbejdstiden er stadigvæk den menneskelige arbejdstid som kræves til korrekturlæsning af den automatiske analyses resultat.

Kvaliteten af den automatiske analyse på det nuværende stadium er vist her i procenter.

3.4 Problemer

De største problemer hidrører fra følgende tre faktorer:

1. Mange ukendte tekstord
2. Uregelmæssig interpunktion
3. Homografer blandt hyppige ordformer

Ukendte ord medfører at det bliver svært at analysere de nærmest liggende ord korrekt og præcist.

På grund af uregelmæssig brug af interpunktion i islandsk har man i den automatiske analyse helt set bort fra interpunktionen, selv om den selvfølgelig i mange tilfælde kunne give værdifulde oplysninger.

Jeg vil nu give et par eksempler på homografer blandt hyppige ordformer, som tit bliver fejlagtigt analyseret under den automatiske analyse.

ordform	*leksikonsord*		*dansk*
við	*við*	pron. pers. 1. p. pl. nom.	*vi*
	við	præp./adv.	*ved*
	(*viður*	sb. masc. sg. akk.	*ved*)

Ordformen **við** har tre helt forskellige meninger. De mest almindelige er personligt pronomen og præposition. Den tredje er her sat i parenteser, fordi den er ikke nær så hyppig som de andre; og den er faktisk slet ikke med i samlingen af ordformer.

ordform	*leksikonsord*		*dansk*
orðið	*verða*	vb. p.p./sup.	*blevet*
	orð	sb. neutr. sg. nom./akk. bek.	*ordet*
orðin	*verða*	vb. p.p.	*blevet*
	orð	sb. neutr. pl. nom./akk. bek.	*ordene*

De to andre ordformer, **orðið** og **orðin** har hver for sig to helt forskellige meninger.

Disse bestemte homografer udgør måske ikke ret store problemer. Men der kræves i hvert fald mere præcise regler end de hidtidige til at skelne i mellem dem.

3.5 Forbedringer

Til slut skal vi betragte de muligheder der gives for at forbedre den automatiske analyse.

- Større korpus, d.v.s. flere ordformer og flere sidstedele
- Flere og mere præcise regler
- Præcisering af pointsgivning
- Udnyttelse af interpunktion

Den forbedring som man venter at opnå uden større besvær skulle føre til 85%–90% korrekt analyse.

Litteratur

Magnússon, Friðrik. 1988. Hvað er títt? Jón Hilmar Jónsson [ed.]. I *Orð og tunga 1*: 1–49. Orðabók Háskólans. Reykjavík.

EVA EJERHED

A Swedish Clause Grammar and Its Implementation

Abstract

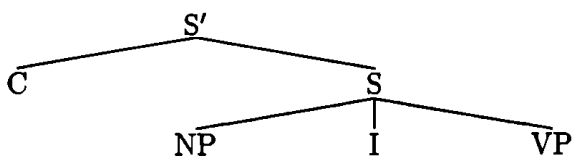
The paper is concerned with the notion of clause as a basic, minimal unit for the segmentation and processing of natural language. The first part of the paper surveys various criteria for clausehood that have been proposed in theoretical linguistics and computational linguistics, and proposes that a clause in English or Swedish or any other natural language can be defined in structural terms at the surface level as a regular expression of syntactic categories, equivalently, as a set of sequences of word classes, a possibility which has been explicitly denied by Harris (1968) and later transformational grammarians. The second part of the paper presents a grammar for Swedish clauses, and a newspaper text segmented into clauses by an experimental clause parser intended for a speech synthesis application. The third part of the paper presents some phonetic data concerning the distribution of perceived pauses (Strangert and Zhi 1989, Strangert 1989) and intonation units (Huber 1988) in relation to clause units.

1 What is a Clause in Linguistic Theory?

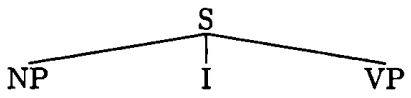
In traditional grammar a clause is defined as a unit consisting of a subject and a predicate. The terms suppositum and appositum were used in scholastic grammar to denote the syntactic functions of these two basic parts of a clause. Traditional grammar makes a distinction between main clauses and dependent clauses.

In current transformational grammar as presented by Radford (1988), three types of clauses are recognized (see (1)).

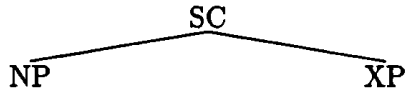
(1) (a) Ordinary Clauses



(b) Exceptional Clauses

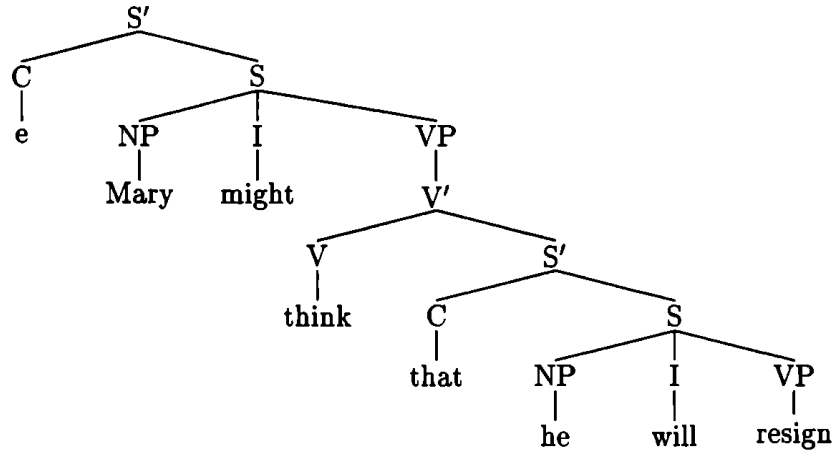


(c) Small Clauses

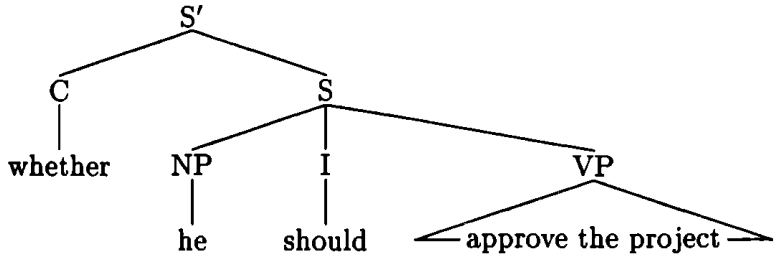


According to Radford (1988) "the three Clause types differ principally in that Ordinary Clauses contain both I and C, Exceptional clauses contain I (=infinitival to) but not C, and Small Clauses contain neither C nor I. Moreover, both Exceptional Clauses and Small Clauses are highly restricted in their distribution: for example, Exceptional Clauses typically occur only as the Complements of certain specific types of verbs; and Small Clauses occur mainly as the Complements of a subset of Verbs and Prepositions ..." It should be noted that I here is tense, modal, or infinitival to, and C is complementizer. Examples of ordinary clauses are given in (2), (3) and (4) below.

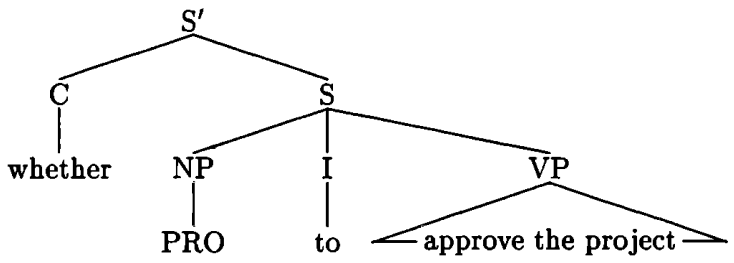
(2)



(3)



(4)



In computational linguistics, there is no single answer to the question of what a clause is, since this depends on the particular grammatical theory chosen in a given computational framework.

In order to illustrate one particular and explicit notion of clause, or more precisely predication, in computational linguistics, I want to quote an interesting study by Henry Kučera (ms, 1985) on the computational analysis of predicational structures in the Brown Corpus.

He considers a predication to be, first of all, any verb or verbal group with a tensed verb that is subject to concord (for person and number) with its grammatical subject. These verbal constructions he calls finite predications. In addition to that, he also includes in his analysis non-finite predications, consisting of infinitival complements, gerunds and participles. What he did in his study was to identify and classify all the predications, which were 145,287 in all the 54,724 sentences of the Brown Corpus.

Table 1 shows for each genre in the corpus, the mean sentence length (words

Genre	Words per Sent.	Pred. per Sent.	Words per Pred.
A. Press, report.	20.81	2.65	7.85
B. Press, edit.	19.73	2.74	7.20
C. Press, reviews	21.11	2.65	7.96
D. Religion	21.23	2.90	7.32
E. Skills	18.63	2.60	7.17
F. Pop. lore	20.29	2.82	7.20
G. Belles lett.	21.37	2.94	7.27
H. Misc.	24.23	2.82	8.59
J. Learned	22.34	2.87	7.78
K. Fiction, gen.	13.92	2.41	5.78
L. Mystery/detect.	12.81	2.29	5.59
M. Science fict.	13.04	2.23	5.85
N. Adv./Western	12.92	2.30	5.62
P. Romance	13.60	2.45	5.55
R. Humor	17.64	2.84	6.21
CORPUS	18.49	2.65	6.97

Table 1:

per sentence), sentence complexity (predications per sentence), and mean predication length (words per predication).

Table 2 below shows that whereas sentence length varies a great deal between a mean of 21 words per sentence in informative prose (INFO) and 13 words per sentence in imaginative prose (IMAG), sentence complexity does not vary that much between genres: 2.80 versus 2.38 predications per sentence.

Measure	INFO	IMAG	CORPUS
Words/Sent.	21.12	13.55	18.49
Pred./Sent.	2.80	2.38	2.65
Words/Pred.	7.54	5.69	6.97

Table 2:

Table 3 below shows how the finite (F) and non-finite (NF) predications were distributed in the genres of informative and imaginative prose.

Group	Type	No.	Pred. per Sent.	Percent
INFO	F	68,157	1.91	68.09%
	NF	31,935	0.89	31.91%
		100,092	2.80	100.00%
IMAG	F	34,329	1.81	75.96%
	NF	10,866	0.57	24.04%
		45,195	2.38	100.00%
CORPUS	F	102,486	1.87	70.54%
	NF	42,801	0.78	29.46%
		145,287	2.65	100.00%

Table 3:

What Kučera considers as the main result of his study is the lack of correlation between sentence length and sentence complexity, and it is indeed surprising.

Kučera's study was concerned with finding, counting and classifying predications units (verbal groups) in the Brown Corpus. It was not concerned with what would have been an even more difficult goal, that of finding entire clause units, in the sense of demarcating their beginnings and endings. There is an obvious relation between predications and clauses, in that a reasonable definition of clause, I think, would be one in which there is one predication, in Kučera's sense of the term, per clause.

In Ejerhed (1988), which is a computational linguistic study of clauses in English, done in collaboration with Ken Church when I visited ATT Bell Laboratories 1986–87, I used a definition of clause that differed somewhat from the one considered in the previous paragraph. In my definition of clause in English,

only finite and to-infinitival predications are criterial for clausehood. Other infinitival predications, gerunds and participles are not taken to imply the presence of a clause unit.

Another feature of my definition of clause that was used in parsing clauses in unrestricted text, is that the opening of a new clause always implies the closure of the previous clause unit, whether or not this unit is complete with subject and predicate, or complete with respect to the argument structure of its predicate. To illustrate this no-nesting of clauses, the sentence in (2) is reproduced in (5) below with clause boundaries inserted where the clause parsers described in Ejerhed (1988) would place them.

(5) `[Mary might think] [that he will resign]`

There are several reasons for the move to adopt the hypothesis that clauses do not nest, at a very superficial level of syntactic structure.

The first reason is that the hypothesis makes possible an exceedingly simple definition of, and recognition algorithm for, clauses: a clause can be defined as a set of permissible sequences of word classes by means of a regular expression, i.e. by using the operations of concatenation, union and Kleene star on elements that are word classes.

That such a simple definition of clauses, or sentence forms as he called them, was possible, was something Harris considered, but rejected in the following passage from Harris (1968:31–32):

> ... in English a *wh*-clause can be away from its noun (usually if no other noun intervenes):
>
> Finally the man arrived whom they had all come to meet.
>
> In describing sentences, one can still say that there is a constituent, even though with non-contiguous parts: the subject above is MAN with adjoined THE on the left and WHOM ... after the verb on the right.[23] But the difficulty lies in formulating a constructive definition of the sentence. For if we wish to construct the sentence by defining a subject constituent and then next to it a verb (or predicate) constituent, we are unable to specify the subject if it is discontiguous, because we cannot specify the location of the second part (the adjunct at a distance). At least we cannot specify the location of the distant adjunct until we have placed the verb constituent in respect to the subject; but we cannot place the verb in respect to the subject as a single entity unless the subject has been fully specified.[24]

[23] And one can specify that it can be at a distance primarily if no noun intervenes.

[24] *To the extent that such problems did not arise, it would be possible to define sentence forms as short sequences of morpheme classes (or word classes), each class being expandable by a certain neighborhood of other classes* (my emphasis EE).

The sentence discussed in the passage above would be parsed as indicated in (6), given the clause grammar of Ejerhed (1988).

(6)
[Finally] [the man arrived] [whom they had all come to meet]

The second reason for the hypothesis that clauses do not nest has to do with performance considerations, i.e. observational data from studies in psycholinguistics and phonetics.

For a review of the clausal hypothesis in psycholinguistics and studies relating to it, the reader is referred to Flores d'Arcais and Schreuder (1983:14–19). They present the clausal hypothesis as a view of sentence comprehension that is characterized by two major features. First, clauses are taken to be the primary units of normal speech perception. Incoming material is organized in immediate memory clause by clause; the listener or reader accumulates evidence until the end of a clause. Second, at the end of a clause, working memory is cleared of surface grammatical information and the content of the clause is represented in a more abstract form. They point out that these two major properties of the hypothesis are logically independent.

Phonetic evidence for the segmentation of speech (in perception as well as production) at the level of clauses, as structurally defined units, will be discussed in the last section of the paper, after a presentation and illustration of a structural definition of Swedish clauses.

2 A Swedish Clause Grammar

This grammar for Swedish clauses has the same structural units as targets as the grammar for English clauses in Ejerhed (1988), modulo the difference between the two languages, i.e. finite (tensed) clauses and infinitival clauses introduced by *att* are clauses. In addition, there are three types of clause fragments: verb phrase fragments, noun phrase fragments and adverb fragments.

In an appendix to this paper, there is a Swedish newspaper text from April 1984 which has been segmented into clauses and clause fragments, labelled to the right according to the type of unit in the grammar that they instantiate. The categories that are criterial to the identification of a clause or clause fragment according to the grammar, have been labelled underneath.

```
GRAMMAR

Main clause (mc)

1. mc-noninv          (COORD)  NP'  VFIN  ...
2. mc-inv             (COORD)  VFIN  (SADV)  NP  ...
3. mc-coord            COORD  VFIN  ...

Subordinate clause (sc)

4. sc-comp            (COORD)  (PREP)  COMP  ...
5. sc-coord            COORD  (SADV)  VFIN/VSUP  ...
```

```
6. sc-nocomp          (COORD)  NP'  (SADV)  VFIN/VSUP  ...

VP-fragment

7. mc vp-fragment     VFIN  ...
8. sc vp-fragment     (SADV)  VFIN/VSUP  ...

NP-fragment

9.                    (COORD)  (COMP)  NP'  ...
10.                   NP'  COORD  NP'

ADV-fragment

11.                   (COORD)  PP/ADVP/SADV*
```

A few words on the notation used in the grammar are required. For readability, concatenation is represented simpy by juxtaposition. Union (i.e. alternatives) is represented by /, and the special case where something alternates with nothing (i.e. optionality) is represented by (). Kleene star is represented by *, which has scope over /. The three dots ... should be read as a variable over any word class.

- COORD is the category of coordinating conjunctions, *och*, *eller*, *men*.
- NP is a non-recursive noun phrase consisting of any prenominal modifiers plus head noun. NP does not include any postnominal modifiers. For the concept of such a noun phrase as applied to English, see Church (1988).
- NP' consists of a non-recursive NP followed by postnominal modifiers that are non-clausal, i.e. prepositional phrases PP, or adverbs ADV. Thus, NP' = NP PP/ADV*
- VFIN is the category of finite verbs, active or passive, and VSUP is the category of supinum forms of verbs occurring after the auxiliary *hava*. Because finite forms of *hava* can be optionally deleted in subordinate clauses in Swedish, it is necessary to allow occurrences of VSUP in such cases to count as finite.
- COMP is the category of subordinating conjunctions, including *att* as infinitive marker.
- SADV is the category of sentence adverbs, *inte*, *ofta*, *aldrig*.
- ADVP is the category of adverbial phrases.
- PREP is the category of prepositions.

Each of the regular expressions 1 through 11 constitutes an alternative definition of clause or clause fragment. The way that these alternative definitions interact in the processing of a text is very important. In cases where two or more alternative analyses compete, *the regular expression that matches the longest substring wins.* This can be illustrated by considering how the first sentence of the text in the appendix is processed. The sentence is repeated below with numbers indicating linear positions in the string of words.

(7) 0 **Allting** 1 **verkar** 2 **så** 3 **okontrollerat** 4
NP **VFIN** ...

The regular expression 9, NP-fragment, matches the string of words from 0 to 1.

The regular expression 7, VP-fragment, matches the string of words from 1 to 4.

The regular expression 1, non-inverted main clause, matches the string of words from 0 to 4. This is the expression that matches the longest substring, and it wins over the alternative analyses of the string from 0 to 4.

The status of the implementation of this particular clause grammar for Swedish is that it is in the process of being implemented. What that means, is that I do not yet have a running program for Swedish that automatically decides the location of boundaries between clauses and clause fragments in unrestricted text. This is an ambitious and long range goal, and the biggest problem in developing such a program is lexical. Each word in a text has to be labelled with a unique syntactic category (including information about the form of the word) before any matching against the regular expressions in the grammar can take place. The category label assigned to a word has to be the one that is correct for the word in its context of occurrence.

A successful approach to the problem of automatically assigning unique and correct syntactic categories to English words in context is probabilistic (Church 1988, DeRose 1988, Eeg-Olofsson 1985). This is one of several approaches that will be applied to Swedish in the context of a joint corpus based research project between the universities of Stockholm and Umeå (Källgren, Ejerhed) that will start in the fall of 1989.

Another approach to the disambiguation of the syntactic category and form of a word in context is rule based, constraint based or heuristic, and the disambiguation between alternative analyses of a word is done as an integrated part of the parsing of a text, rather than as a separate subroutine completed before parsing begins. A version of this approach has been applied to Swedish with successful (95% correct) results (Brodda 1983, Källgren 1984a, 1984b).

Fred Karlsson claimed in his paper at this conference, on the basis of his recent research on disambiguation, that more than 60% of the consecutive words in a Swedish text are at least two-way ambiguous, as compared with 45% in English according to DeRose (1988), and 11% in Finnish. Karlsson's figure for Swedish tallies with what is reported in Allén (1970:XV, XXV): 645,000 out of the 1,000,669 words of the Swedish corpus Press-65 were homographs, and that amounts to 64.5%.

What I have by way of implementation at this time is a modification of the finite state parser for Swedish, described in Ejerhed & Church (1983), Ejerhed & Bromley (1985), and Ejerhed (1986). Subject to the limitations of its lexicon, which is currectly being expanded, the modified parser, in its parsing of orthograhpic sentences as input, is capable of identifying and assigning constituent structure to substrings that can be put in direct correspondence with the 11 different clauses and clause fragments enumerated in the new clause grammar described here.

3 Phonetic Data concerning Clause Boundaries

There are two recent phonetic studies of spoken Swedish, based on recordings of several different speakers reading the same texts aloud. One is by Eva Strangert (Strangert and Zhi 1989, Strangert 1989) and the other by Dieter Huber (1988).

Strangert's research project, which is still going on, studies perceived pauses in 2 texts of a total of 810 words read aloud by 10 different speakers at 3 different speech rates, and the acoustic and grammatical properties of such pauses. The first of the two texts is identical to the text in the appendix of this paper. Acoustically, a perceived pause can be signalled in several different ways: by final lengthening, a special fundamental frequency contour, silence, and/or voice quality irregularities. Strangert and Zhi (1989) reports findings primarily concerning these acoustic properties of the pauses perceived by two different judges. Strangert (1989) is also concerned with the distribution of the perceived pauses in relation to the following kinds of boundaries: paragraph, sentence, clause and phrase.

Using the definition of clause presented in this paper, I have segmented the two texts used in Strangert's study and found that they consist of a total of 115 units that are clauses or clause fragments. The number of perceived pauses at these 115 clause boundaries is presented in Table 4 below, for which I am indebted to Eva Strangert. A perceived pause is here a pause judged by both of the two judges to be present in the speech of at least 5 of the 10 speakers. For the purposes of this table, all clause boundaries have been included, whether they are sentence internal, or happen to coincide with sentence boundaries or paragraph boundaries. In Strangert (1989) these three boundary conditions are treated separately.

Speech rate	Number of clause boundaries with perceived pauses	Percent (N = 115)
Fast	57	50
Normal	78	68
Slow	97	84

Table 4: The frequency of clause boundaries where pauses were perceived.

The study of Huber (1988) is concerned with intonation units in recordings of 3 newspaper texts read aloud by 4 different speakers of Swedish, a total of 2.2 hours of connected speech. He defines the concept of intonation unit in purely acoustical terms, related to fundamental frequency only, and devises a method of automatically segmenting connected speech into such intonation units. The advantage of this segmentation procedure is that it makes no reference to either higher level linguistic information concerning syntax, or to lower level physiological information concerning pausing, breathing, phonation onset or offset etc. He arrives at a total of 1664 intonation units in the accumulated text material (3 texts, 4 speakers). Table 5 shows the grammatical correlates of the 1664 intonation units, averaged across four speakers and three texts. For the exact definitions of the grammatical units, see Huber (1988:78). Of interest here is that he defines as sentences "graphic sentences that begin with a capital letter and end with a full stop (or some other mark of 'final' punctuation)". And he defines as clauses "units of linguistic organisation smaller than the sentence and consisting of at least one subject and one finite verb".

Grammatical Unit	Number of intonation units	Percent
SENTENCE	299	18.2
CLAUSE	662	39.7
SUBJECT	83	4.8
VERBPHRASE	76	4.5
ADVERBIAL, init.	35	2.0
ADVERBIAL, final	141	8.5
PARENTHETICAL	132	8.0
MISCELLANEOUS	238	14.3
Total	1666	100.0

Table 5: Frequency of intonation units corresponding to different grammatical categories.

Unfortunately, these figures cannot be directly related to the notions of clause and clause fragment discussed in this paper, because the definitions of the grammatical categories do not agree. However, it is likely that we can equate monoclausal *sentences* (which accounted for 63.6% of the 1-IU-per-sentence that occurred) with a subset of main clauses (Rules 1–3 in the Swedish clause grammar), *clauses* with either a subset of main clauses (in the case of multiclausal sentences) or a subset of subordinate clauses (Rules 4–6), and *initial adverbials* with adverb-fragment (Rule 11), and these three categories together account for 60% of all intonation units. It is also likely that *subject* corresponds to NP-fragment, and *verbphrase* to VP-fragment on the basis of the illustrative examples of these categories in Huber (1988:83–85). If so, close to 70% of Huber's intonation units would correspond to a clause or clause fragment in the sense of the present paper. In order to establish the exact extent to which the notions of clause and clause fragment proposed here correlate with the intonation units found in Huber's study, a separate study is being undertaken in collaboration with Huber.

Acknowledgement

The work on this paper was done while the author was a member of the Speech group headed by Bertil Lyberg, Department of Research and Development, Swedish Telecom, Stockholm, as well as of the Department of Linguistics, University of Umeå. I am indebted to Swedish Telecom in Stockholm for the use of the resources of its Speech Lab, and to Eva Strangert in Umeå for collaboration on pauses.

References

Allén, S. 1970. *Nusvensk frekvensordbok baserad på tidningstext. 1. Graford, homografkomponenter.* Data linguistica, 1. Almqvist & Wiksell, Stockholm.

Brodda, B. 1983. An experiment with heuristic parsing of Swedish. *Proceedings of the First Conference of the European Chapter of the Association for Computational Linguistics*:66–73. Pisa.

Church, K.W. 1988. A stochastic parts program and Noun Phrase parser for unrestricted text. *Proceedings of the Second Conference on Applied Natural Language Processing*:136–143. Association for Computational Linguistics, Austin, Texas.

DeRose, S.J. 1988. Grammatical category disambiguation by statistical optimization. *Computational Linguistics*, 14(1):31–39.

Eeg-Olofsson, M. 1985. A probability model for computer aided word class determination. *ALLC Journal*, 5(1 & 2):25–30.

Ejerhed, E. 1986. A finite state parser for Swedish with morphological analyzer and semantics. Proceedings of SAIS-86. Institutionen för Datavetenskap, Linköpings universitet.

Ejerhed, E. 1988. Finding clauses in unrestricted text by finitary and stochastic methods. *Proceedings of the Second Conference on Applied Natural Language Processing*:219–227. Association for Computational Linguistics, Austin, Texas.

Ejerhed, E. and H.J. Bromley. 1986. A self-extending lexicon: description of a word learning program. F. Karlsson [Ed.] *Papers from the Fifth Scandinavian Conference of Computational Linguistics*:59–72. Publication No. 15, Department of General Linguistics, University of Helsinki.

Ejerhed, E. and K.W. Church. 1983. Finite state parsing. F. Karlsson [Ed.]. *Papers from the Seventh Scandinavian Conference of Linguistics*:410–432. Publication No. 9, Department of General Linguisics, University of Helsinki.

Flores d'Arcais, G.B. and R. Schreuder. 1983. The process of language understanding: A few issues in contemporary psycholinguistics. G.B. Flores d'Arcais and R. Jarvella [Eds.]. *The Process of Language Understanding*:1–41. Wiley, Chichester.

Harris, Z. 1968. *Mathematical structures of language.* Wiley, New York.

Huber, D. 1988. Aspects of the communicative function of voice in text intonation— Constancy and variability in Swedish fundamental frequency contours, Department of Computational Linguistics, University of Göteborg, Department of Information Theory, Chalmers Institute of Technology, Göteborg, and Department of Linguistics and Phonetics, University of Lund.

Karlsson, F. 1989. The resolution of morphological ambiguities. Paper presented at the Scandinavian Conference of Computational Linguistics, Reykjavík, June 26–28, 1989.

Kučera, H. n.y. Computational analysis of predicational structures in English. Brown University, Providence, R.I. (unpublished).

Kučera, H. 1985. The analysis of the English verbal group. Paper presented to the ICAME Sixth International Conference on English Language Research on Computerised Corpora, Lund (unpublished).

Källgren, G. 1984a. HP-systemet som genväg vid syntaktisk märkning av texter. *Svenskans beskrivning*, 14:39–45, Lunds universitet.

Källgren, G. 1984b. HP—A heuristic finite state parser based on morphology. A. Sågvall Hein [Ed.]. *De nordiska datalingvistikdagarna 1983*:155–162. Centrum för Datorlingvistik, Uppsala Universitet.

Radford, A. 1988. *Transformational Grammar: A First Course.* Cambridge University Press, Cambridge.

Strangert, E. and M. Zhi. 1989. Pause patterns in Swedish: A project presentation and some data. Fonetik-89. *Speech Transmission Laboratory Quarterly Progress and Status Report*, 1:27–31. KTH, Stockholm.

Strangert, E. 1989. Pauses, syntax and prosody. Paper presented to the Nordic Prosody V meeting in Turku, Finland, August 23–25, 1989 (to appear).

Department of Linguistics
University of Umeå
S-90187 Umeå
Sweden
EJERHED@SEUMDC51 (Bitnet)

Text A1

The text is divided into paragraphs by consecutive numbering. The paragraphs are divided into orthographic sentences by sentence final punctuation marks. The sentences are divided into non-recursive clauses or clause fragments marked by [ ], and each such unit is labelled according to the Swedish clause grammar presented in this paper.

```
Paragraph 1
[Allting verkar så okontrollerat.]                mc-noninv
 NP      VFIN
[Det tycks]                                       mc-noninv
 NP  VFIN
[som om ingen längre håller i styret.]            sc-comp
 COMP
[Framför allt]                                    adv-fragment
 P       NP
[verkar läget vara okontrollerat inne i Tripoli] mc-inv
 VFIN   NP
[där ungdomar i femtonårsåldern på något sätt
sc-comp  COMP
har fått tag i skjutvapen.]

Paragraph 2
[Det sade en ung spanjor]                         mc-noninv
 N   VFIN
[som var en av de 113 personer]                   sc-comp
 COMP
[som lyckades komma ut ur Libyen                  sc-comp
 COMP
med den första flygningen]

[sedan USA bombade Tripoli och Bengazi            sc-comp
 COMP
i början av veckan.]

[Den unge spanjoren fanns ombord                  mc-noninv
 NP                 VFIN
på det reguljärplan från Libyan Airlines]

[som kraftigt försenat landade på                 sc-comp
 COMP
den internationella flygplatsen utanför Rom
sent på torsdagen.]

Paragraph 3
[Planet återvände aldrig till Tripoli             mc-noninv
 NP     VFIN
på torsdagskvällen.]
```

```
[En väntande skara journalister fick                mc-noninv
 NP                               VFIN
officiellt beskedet]

[att besättningen helt enkelt var för uttröttad.] sc-comp
 COMP

Paragraph 4
[Libyan Airlines flygning 167 tillbaka till         mc-noninv
 NP                              ADV      P
den libyska huvudstaden uppsköts därför till
NP                       VFIN
någon gång under fredagen.]

Paragraph 5
[Ingen av de 113 passagerarna på den första         mc-noninv
 NP    P  NP                    P  NP
utflygningen från Tripoli var svensk.]
P    NP           NP      VFIN
[Det finns omkring 200 svenskar i Libyen]           mc-noninv
 NP  VFIN
[varav ungefär hälften bor i huvudstaden Tripoli.] sc-comp
 COMP
[Den svenska ambassaden har rekommenderat]          mc-noninv
 NP                       VFIN
[att de svenskar]                                   np-fragment
 COMP
[som arbetar i Libyen]                              sc-comp
 COMP
[skall evakuera sina familjer]                      vp-fragment
 VFIN
[så snart tillfälle ges.]                           sc-comp
 COMP

Paragraph 6
[Den unge spanjoren,]                               np-fragment
 NP
[som ville vara anonym,]                            sc-comp
 COMP
[talade om en skräckstämning i Tripoli]             vp-fragment
 VFIN
[där ingen egentligen vet]                          sc-comp
 COMP
[vem som bestämmer.]                                sc-comp
 COMP

Paragraph 7
[En vild ryktesflora grasserar också                mc-noninv
 NP                    VFIN
om ledaren Muammar Gadaffi.]
```

```
[Det har även under torsdagen förekommit          mc-noninv
 NP  VFIN
skottlossning i den militärförläggningen i Tripoli]

[där Gadaffi och hans familj bodde]               sc-comp
 COMP
[när de amerikanska bombplanen slog till          sc-comp
 COMP
natten till tisdagen.]

Paragraph 8
[Det osäkra läget befästes                        mc-noninv
 NP               VFIN
på torsdagen ytterligare]

[av att minst tre passagerarplan från Spanien,    sc-comp
 P  COMP
Rumänien och Jugoslavien avbröt sina flygningar
till Tripoli.]

[Planen startade]                                 mc-noninv
 NP     VFIN
[men fick återvända till sina hemorter.]          mc-coord
 COORD VFIN

Paragraph 9
[Då det gällde Libyan Airlines första utflygning] sc-comp
 COMP
[florerade också ryktena.]                        mc-inv
 VFIN      SADV  NP
[Då planet skulle ha startat återfärden           sc-comp
 COMP
från Rom kl 17]

[hade det ännu inte lyft från utgångspunkten      mc-inv
 VFIN NP
Tripoli.]

[Flera passagerare dementerade dock uppgifter     mc-noninv
 NP                VFIN
om skottlossning i samband med starten utanför
Tripoli.]

Paragraph 10
[Men de bekräftade]                               mc-noninv
COORD NP VFIN
[att det råder kaotiska förhållanden i            sc-comp
 COMP
den libyska huvudstaden.]
```

```
[De flesta passagerarna var från öststater.]      mc-noninv
 NP                      VFIN

Paragraph 11
[De flesta håller sig inomhus även under dagtid,] mc-noninv
 NP         VFIN
[sade en polsk medborgare.]                       mc-inv
 VFIN NP
[Ute på gatorna]                                  adv-fragment
 ADV P  NP
[är det alldeles för osäkert.]                    mc-inv
 VFIN NP
[Det finns alldeles för många ungdomar med gevär] mc-noninv
 NP  VFIN
[för att man skall kunna känna sig säker.]        sc-comp
 P   COMP

Paragraph 12
[Och ryktena om överste Gadaffi]                  np-fragment
 COORD NP    P  NP
[och vad som har hänt honom]                      sc-comp
 COORD COMP
[är lika många som fantastiska.]                  vp-fragment
 VFIN
```

LARS M GUSTAFSSON

Händelsestyrd textgenerering

Abstract

This paper describes the system RADAR, an event-driven text generator. The system reads input from a radar-system i.e. *time,id,position*, and generates comments in Swedish about the objects on the screen. The problem as such is quite easy to grasp and is not discussed much, but the techniques that have been utilized are described in more detail. The system is written in an object-oriented environment, implemented as a meta-interpreter in Prolog. Another technique that plays a major role is Data Driven Execution, this is also implemented on top of a Prolog-system. The source code for the entire system is available in C-Prolog and fully portable. The system makes an object-instance for every new physical object on the Radar-screen and lets the objects themselves generate comments about their situation. The Data Driven Execution rules generate comments on the simplest level, i.e. X appears, Y disappears. Other rules try to combine these simple observations with the comments generated by the objects themselves to more complex phrases and sentences.

1 Inledning

Detta papper beskriver uppbyggnaden av programmet RADAR, som är implementerat helt i C-Prolog. En ursprungsversionen till programmet skrevs på Inst. för Allmän Språkvetenskap vid Lunds Universitet. Den nuvarande versionen av systemet är utvecklat i huvudsak vid Carnegie-Mellon University i Pittsburgh och CoTech AB i Lund. Det problem som studerats är att utifrån informationen från en övervakningsradar generera kommentarer i realtid. Som framgår av titeln så är det beskrivna systemet baserat på att textgenereringen sker fortlöpande. Det finns alltså ingen möjlighet att planera längre sammanhängande yttranden där hänsyn tas till senare händelser. Problemet som sådant är ganska lättfattligt och inte mycket att orda om. Däremot så kommer dom tekniker som använts att beskrivas mer i detalj och deras förtjänster vid denna typ av textgenerering kommer förhoppningsvis att framgå.

2 Använda tekniker

Det är i första hand två metodiker jag använt mig av.

1. OOP–Object Oriented Programming
2. DDE–Data Driven Execution

Jag kommer först att beskriva (motiveringen för och implementeringen av) objektorienteringen. Därefter kommer jag att beskriva den datadrivna exekveringen, för att till sist komma in på hur dessa båda tekniker kan knytas samman till ett system.

3 Varför objektorientering?

1. Det är i det här fallet naturligt med ett objekttänkande eftersom det finns en naturlig korrespondens mellan objekten i yttervärlden och deras representation i en objektorienterad programmeringsmiljö.
2. Modulariteten ligger på klassnivån (konceptnivån), där man i klassbeskrivningen anger de variabler och metoder som tillsammans med ärvda variabler och metoder utgör objektbeskrivningen.
3. Sen bindning, dvs. dynamisk allokering av nya objektinstanser under exekveringen. Men även möjligheten att fortlöpande lägga till funktioner och datatyper som inte kunnat förutses vid den ursprungliga programkonstruktionen.
4. Ärvning, detta är visserligen inte nödvändigt för OOP, men väldigt naturligt och arbetsbesparande. Ärvning skapar också en genomgående konsistens i programmet. Detta kan naturligtvis (som traditionellt) till en viss nivå uppnås genom en hård disciplin hos programmeraren, men det är vare sig önskvärt eller effektivt.

Problemets realtidskaraktär gör att multipel ärvning är av stor nytta. Man kan då specificera objektrepresentationen mer i detalj efterhand som informationen kommer in. Tex. kan man låta ett okänt objekt först ärva egenskaper från klassen Flygplan. Därefter när man fått nya indikationer (fart, höjd osv.) så kan man precisera objektet genom ärvning från klassen Jetplan. Denna precisering kan fortgå under hela objektinstansens livslängd genom ytterligare observationer, tex. visuella rapporter. Möjlighet finns även att ta bort ärvningar som visat sig bero på felaktiga informationer.

4 Implementationer av OOP

Smalltalk-80
Simula
Ada
C++
Objective-C
Scheme
Flavours
Prolog meta-interpretator

Detta är några av dom möjligheter som finns för den som vill använda sig av objektorienterad programmering. De olika språken har naturligtvis sina fördelar respektive nackdelar, ADA tex. saknar en naturlig ärvningsmekanism. Smalltalk-80 är den mest renodlade implementationen och tillåter inget annat än objekt-programmering, vilket försvårar kontruktion av hybridsystem.

För min implementation så har jag skrivit en objektmekanism som en metainterpretator i Prolog. Detta gör det möjligt att konstruera hybridsystem där vissa delar är objektorienterade medan andra delar av programmet utnyttjar andra problemlösningsparadigmer.

5 OOP i Prolog

Objekten representeras som enhetsklausuler i Prolog med metoderna och variablerna i en lista.

```
object(name,[metod_1, metod_2,..., metod_n]).
```

Den hierarkiska informationen lagras i en separat klausul.

```
isa( Obj, Obj_Super ).
```

Möjligheter finns också att definiera andra typer av relationer, tex.

```
partof( Obj, Obj_0 ).
```

All kommunikation mellan objekten sker med predikatet **send**, tex.

`send(Name, show).`	Metoden **show** gör att objektet ritar ut sig på skärmen.
`send(Name, alert(A)).`	Returnerar värdet på alert i A.
`send(plane, create_instance(Name)).`	Skapar en instans av klassen plane, ett unikt namn (id) returneras i Name ; om Name är instansierat så blir istället detta objektets namn.
`send (Name, set(description, viggen_1).`	
`send(Name, kill(description).`	Sätter/tar bort variabler.

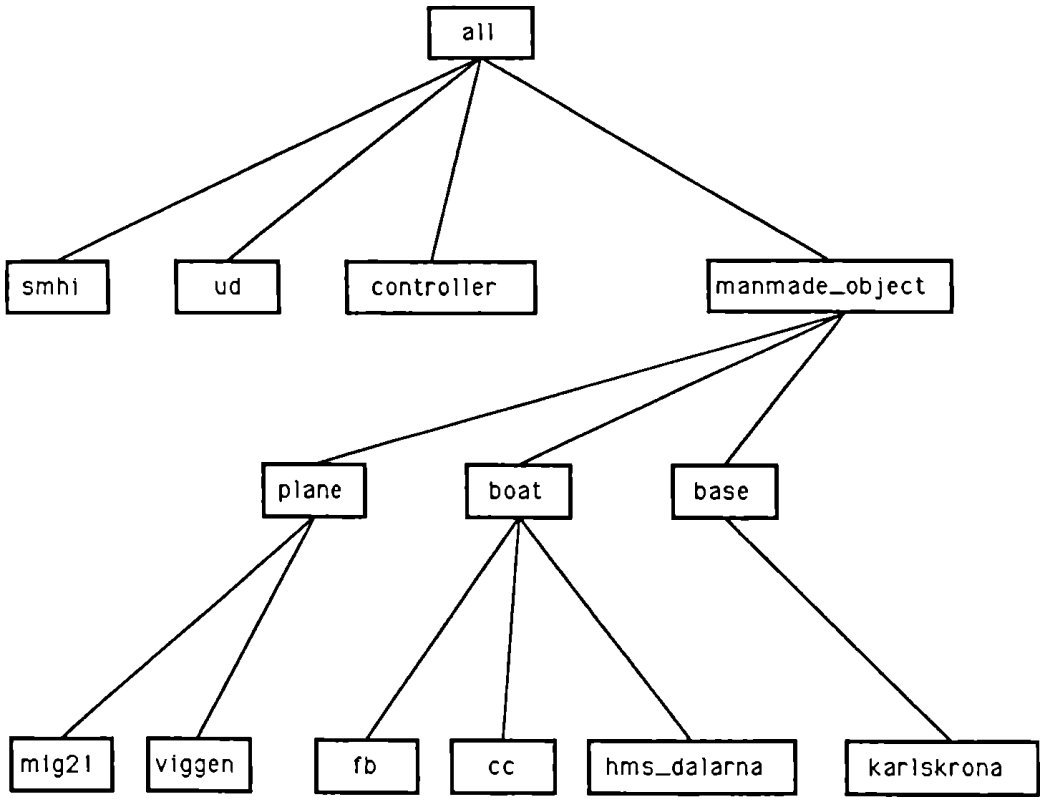


Figure 1: Object hierarchy in Radar

6 Objekthierarkien

Detta är en förminskad version av den objekthierari som jag använt mig av i systemet. *Controller* är huvudobjektet som kontrollerar resten av systemet och innehåller metoder för anpassning till mekanismen för mönsterdriven exekvering som kommer att beskrivas senare. Andra exempel på objekt är *smhi*, som anropas för väderinformation och ud som hanterar information om det politiska läget. Alla relationerna i denna hierarki är av 'isa'-typ, alla löven är alltså objektinstanser.

7 Metoder

Här är några av dom metoder som objekten i programmet är utrustade med.

Metoderna i objektet 'all' ärvs av alla andra objekt i systemet. Dessa metoder skulle visserligen kunna vara inbyggda systemfunktioner men på detta sätt blir systemet 'renare' och man kan även modifiera dessa metoder om så skulle önskas. Metoden internals hos klassen manmade_objects finns hos alla objekt i systemet. Det är denna metod som ger objekten möjlighet att själv generera kommentar-

Objects	Methods
all	create_instance
	set(NewMethod)
	kill(Method)
smhi	visibility(Visi)
ud	alert(Alert)
controller	start
manmade_object	show
	see(List, Sdist)
	firing_range(Range)
	country(Country)
	internals

Figure 2: Methods in the classes of Radar I

Objects	Methods
plane	moving_dimensions(3)
boat	moving_dimensions(2)
base	moving_dimensions(0)
	internals*
	show*

* Overides the inherited methods.

Figure 3: Methods in the classes of Radar II

er utifrån varje objekts egna speciella förutsättningar. Metoden internals finns alltså hos alla objekt, men funktionen varierar beroende på objektets typ.

8 Data Driven Exekvering

Grundprincipen med Data Driven Exekvering är att man har en gemensam dataarea —"blackboard", där alla fakta lagras. Runt denna area ligger ett antal regler som kan påverka innehållet i dataarean om vissa triggvilkor är uppfyllda. Dessa regler kan triggas av "IF-ADDED" eller "IF-ERASED" vilkor. Dom väntar alltså på att ett visst mönster av information skall dyka upp eller försvinna och på så sätt trigga actiondelen av regeln. Dessa regler skulle direkt kunna överföras till en ren parallell maskinarkitektur eftersom det inte finns någon speciell ordningsföljd mellan reglerna.

Detta gör att man kan trigga outputregeln (Rule_5) vid vissa tidpunkter så att systemet genererar "den bästa" output det hunnit skapa fram till denna tidpunkt. Vid komplexa scenarios kan detta vara nödvändigt för att systemet skall kunna jobba i realtid.

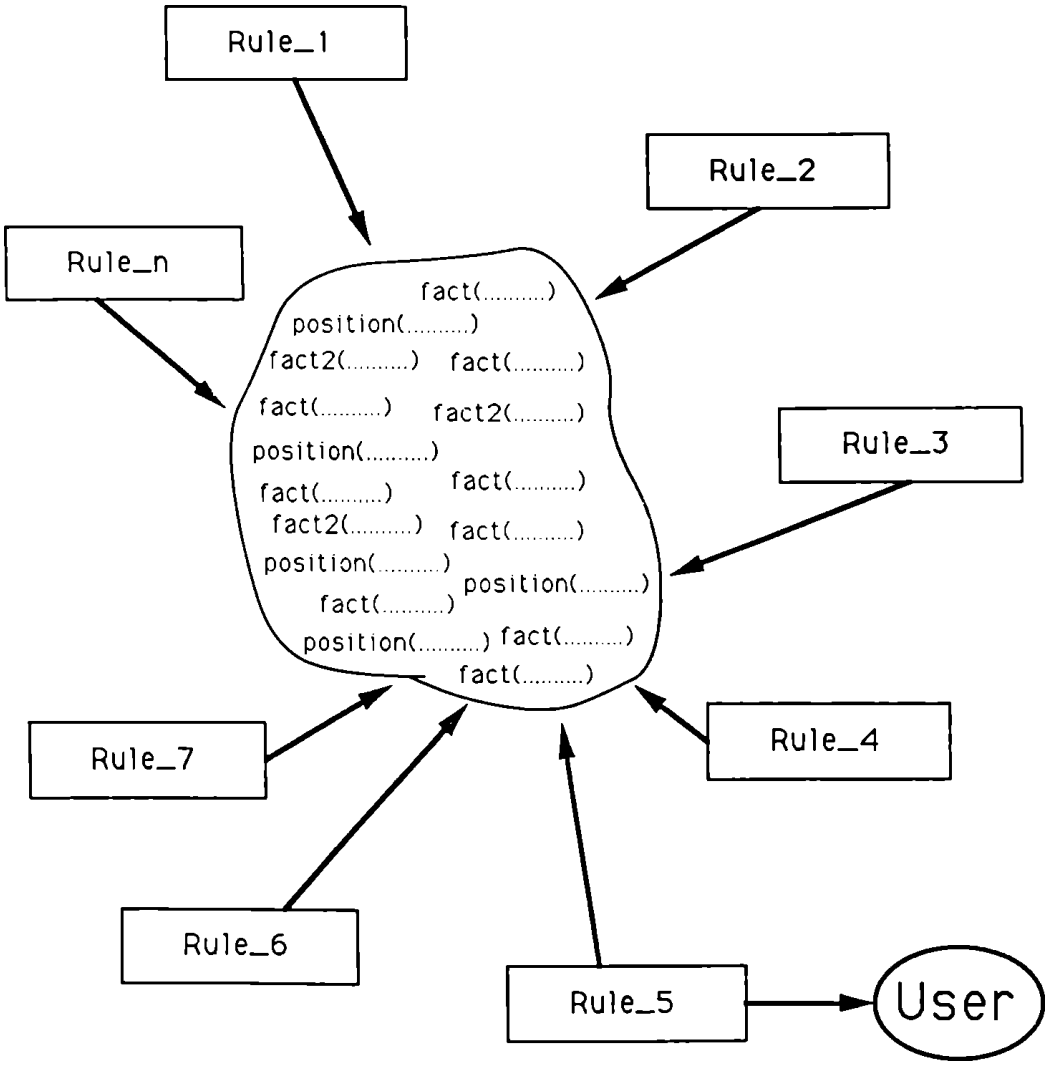


Figure 4: The structure of the pattern-matcher

```
% Rule_1
[condition_1, condition_2,...,condition_n]
     --->
     [action_1,action_2,...,action_n]
-
-
% Rule_n
[condition_1, condition_2,...,condition_n]
     --->
     [action_1,action_2,...,action_n]
%End Rule
[]   ---> [stop].

All rules are tried starting from the top,
whenever a rule triggers, the matcher restarts
from the first rule.

All rules are tried until no more matches
are possible (and the matcher reaches the
End Rule).
```

Figure 5: Rule format in pattern-matcher

9 Regelformat

I figur 5 vises regelformatet där "condition" kan vara förekomsten eller frånvaron av ett visst mönster i dataarean. Action är de operationer som utförs då regeln triggas. Dessa operationer påverkar innehållet i dataarean så att ett nytt tillstånd uppstår som eventuellt kan trigga nya regler. Principen är alltså att man lägger in fakta om objekten på den lägsta nivån (fysisk position) och startar därefter matchningsmekanismen och låter den stegvis bygga upp mer komplexa uttryck ända upp till det slutliga yttrandet.

10 Generering av enkla observationer

Genereringen av enkla observationer kan ske på tre olika sätt.

1. Generering utifrån matchning från position-nivån.
2. Generering av en instans av manmade_object via metoden internals.
3. Generering av ett objekt av klassen "base" som har en speciell variant av internals som kontrollerar alla objekt som närmar sig och passerar osv.

Om man efter programkonstruktionen vill lägga till ett objekt som ställer speciella krav på på att nya kommentarer genereras, så är det bara att se till

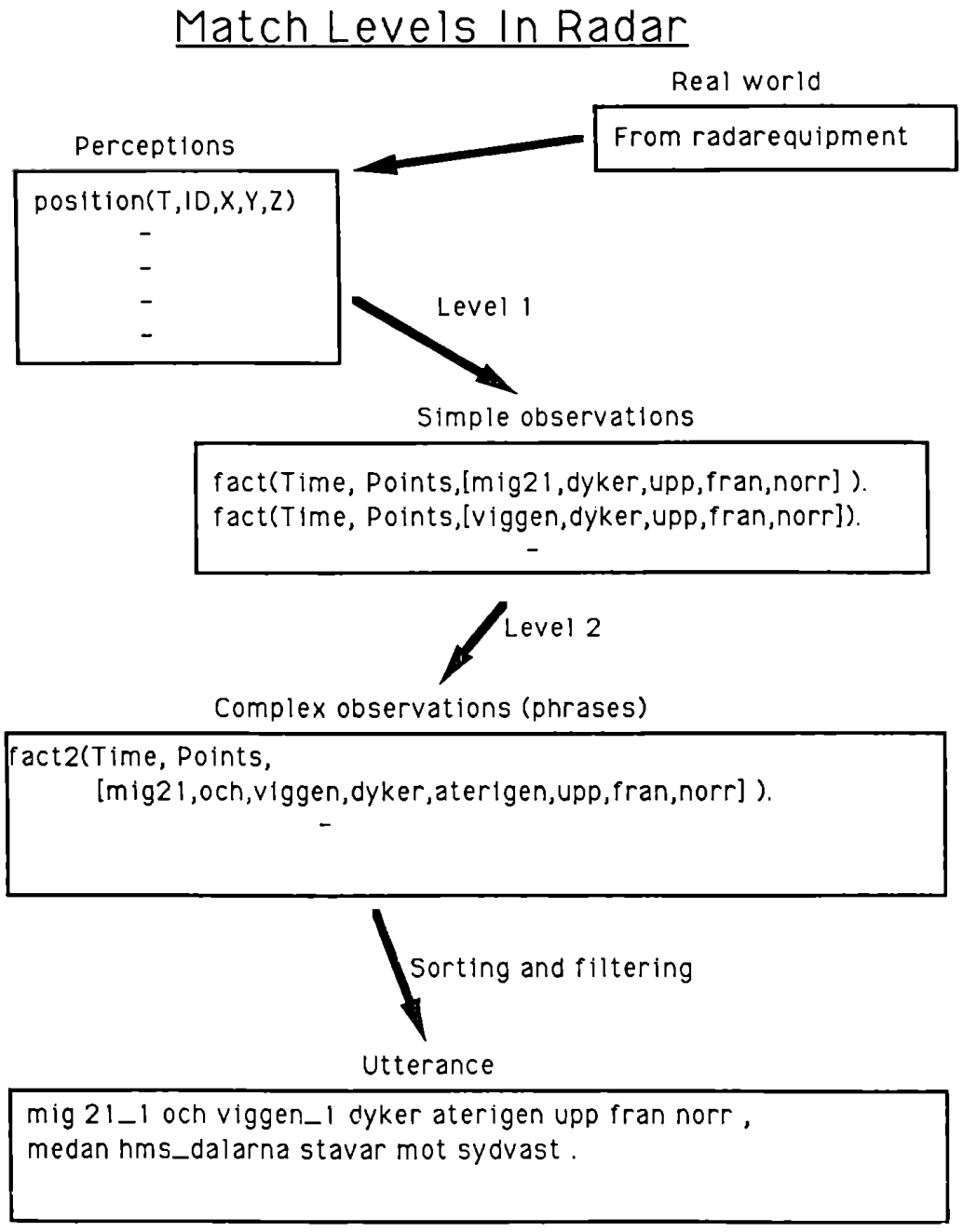


Figure 6: Match levels in Radar

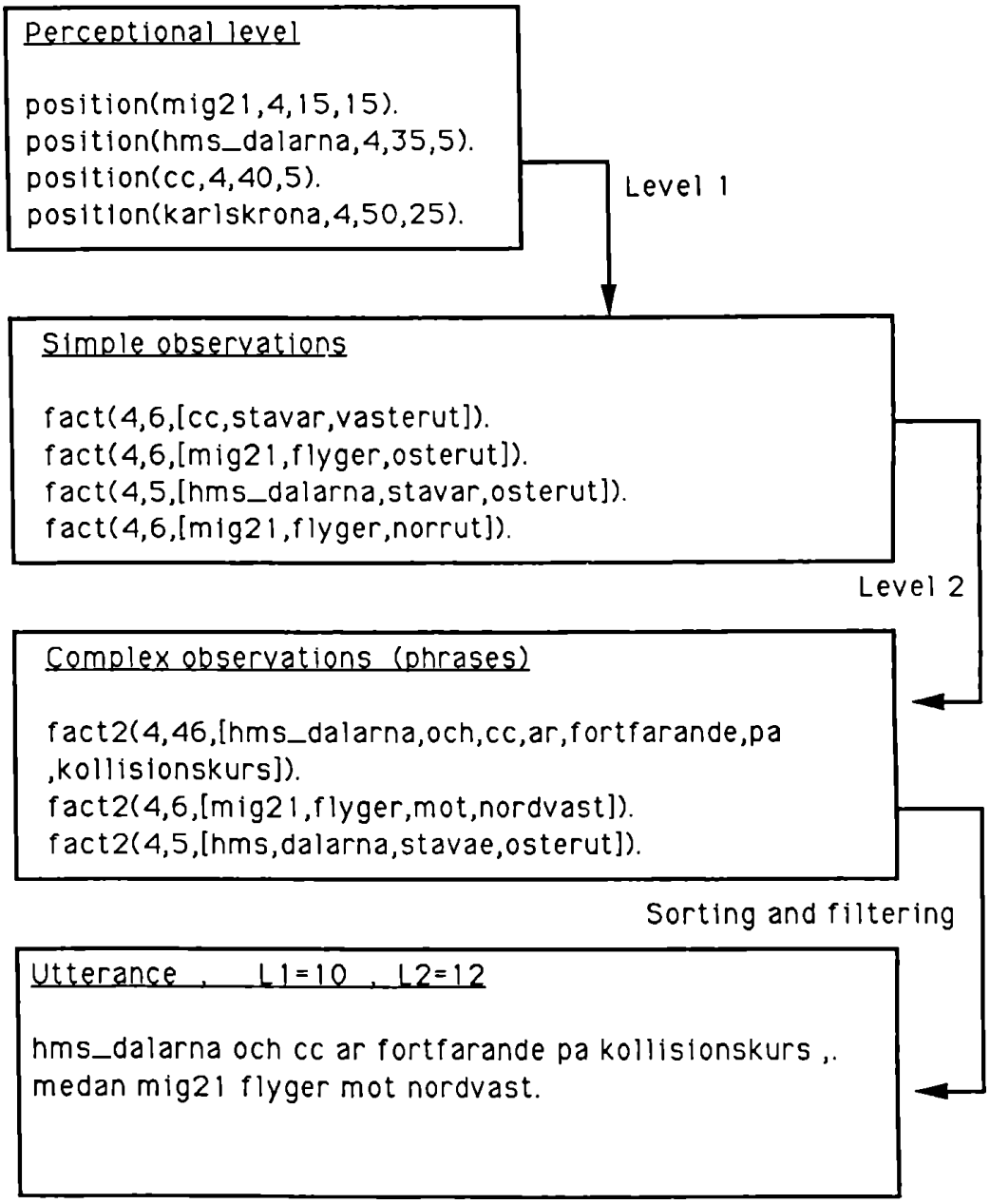


Figure 7: Sample stages in the matching process

att dessa genereras av objektet själv i metoden internals. Det övriga systemet behöver alltså inte modifieras.

11 Faktanivåer i systemet

Systemet utgår från informationen från en övervakningsradar, bestående av *tid,id,position.* Från denna information finns det ett antal regler som detekterar uppdykande och försvinnande objekt. Vidare finns det på denna nivå andra regler som detekterar kurs och hastighet osv. Dessa regler skapar dom enkla observationer som syns i figur 6 på nivån "simple-observations".

Där variabeln "Points" indikerar hur intressant denna information bedöms vara. Detta räknas ut ifrån objektets egenpoäng, händelsens poäng, nationalitet, beredskapsgrad osv. Dessa enkla observationer kombineras sedan ihop av en annan uppsättning regler till kompletta fraser. Poängen från den nedersta nivån förs upp till överliggande nivå genom att man tar hänsyn till bl.a. dom ingående objektens poäng, händelsens poäng samt eventuella samordningspoäng. Utifrån dessa fraser bildas sedan ett yttrande mha. ett antal sorterings och filtreringsregler. De egenskaper hos objekten som behövs vid matchningen fås naturligtvis

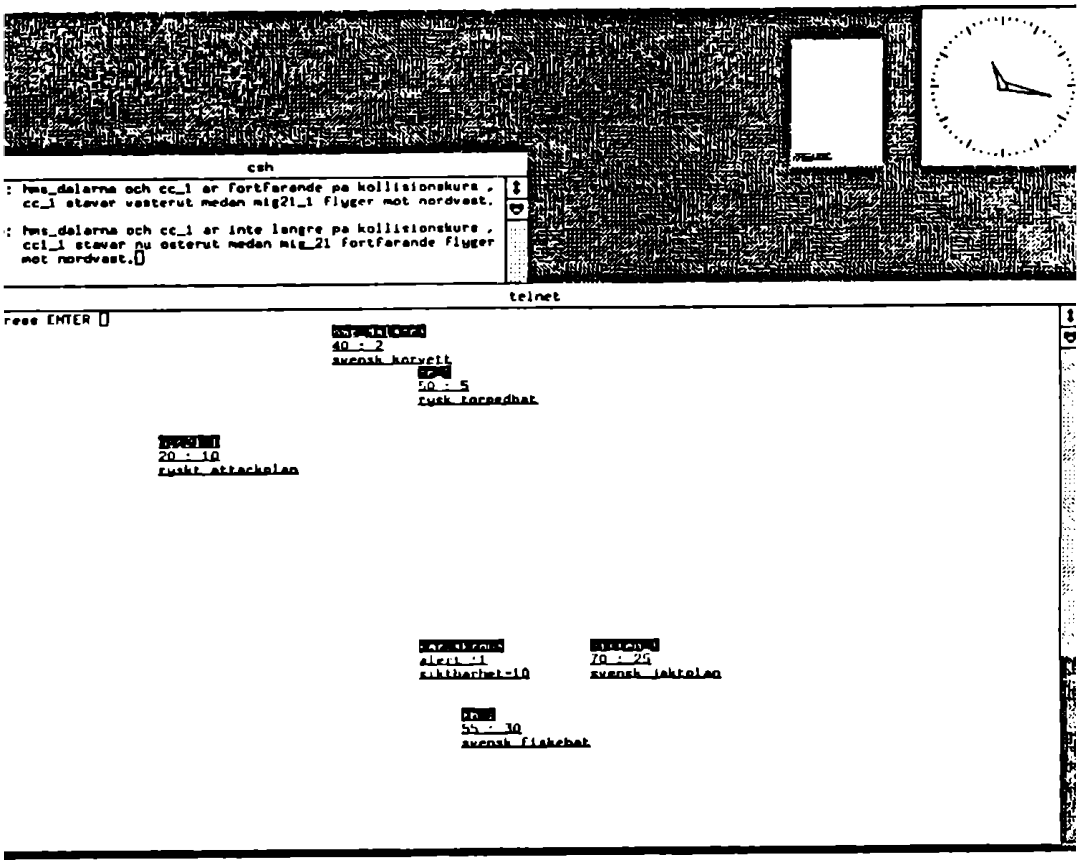


Figure 8: Interacting with Radar

genom att man via "send" till respektive objekt frågar om den önskade egenskapen.

I figur 7 ser man hur hela genereringsprocessen går till vid en viss tidpunkt.

12 Systemets användargränssnitt

I figur 8 ser man ett exempel på hur bildskärmen kan se ut vid en given tidpunkt i prototypsystemet.

Litteratur

Cox, Brad J. 1986. *Object Oriented Programming, An Evolutionary Approach.* Addison-Wesley, Reading, Massachusetts.

Covington, Michael A., Donald Nute, Andre Vellino 1988. *Prolog Programming in Depth.* Scott, Foresman and Co., London.

Bratko, Ivan. 1986. *Prolog Programming for Artificial Intelligence.* Addison-Wesley, Wokingham.

Clocksin, William F., Christopher S. Mellish. 1981. *Programming in Prolog.* Springer, Heidelberg.

Goldberg, and Robson. 1983. *Smalltalk-80: The Language and its Implementation.* Addison-Wesley, Reading, Massachusetts.

Fornell, Jan. Sigurd, Bengt 1983. Commentator. Praktisk Lingvistik, 8. Lund.

Cognitive Technology AB
Box 1691
221 01 LUND
lmg@hum.gu.se

STEFFEN LEO HANSEN

På vej mod en fagsproglig tekstfortolker?

Abstract

The FAGFLADE Project

"FAGFLADE" is short for Danish "fagsproglig grænseflade" ("special purpose language interface"). The aim of the FAGFLADE project is to develop and test theories and methods for automatic interpretation of texts written in special purpose language. We use the expression "text interpreter" to designate a program which transfers the information contained in a natural language text into knowledge representations in a knowledge base. Thus a text interpreter is meant to perform part of the task of a knowledge acquisition system in an expert system.

The FAGFLADE project takes as its basis the text of the Danish Companies Act ("Lov om aktieselskaber"). It is not our ambition to build an expert system, nor do we aim at the construction of a complete text interpretation system, fit for use, and capable of translating statutes into legal knowledge bases. Rather, we take the development of a specific interpreter to be an ideal goal which defines an overall project capable of giving rise to a handful of interesting subprojects for the investigation of general theories and principles concerning interpreters, e.g. in the domain of syntactic and semantic analysis, parsing strategies, knowledge representation, dictionary databases, and terminological analysis.

The choice of the Danish Companies Act as a text basis is motivated partly by our desire to work with an LSP central to industry and commerce, partly by the evident practical perspectives which will open up if text interpreters for legal directives become a realistic possibility.

This paper deals with the initial phase of the project. In this phase we have constructed a framework for the syntactic and semantic analysis of simple sentence structures, and we have developed a program, written in Quintus Prolog, which carries out syntactic and semantic parsing of these structures.

1 Introduktion

På Institut for Datalingvistik på Handelshøjskolen i København har vi siden instituttet blev oprettet i 1985 udviklet en tradition for et forskningssamarbejde,

der i starten formede sig som en række datalingvistiske seminarer over udvalgte emner, men som siden 1987 har ført til arbejdet på et egentligt forskningsprojekt med deltagelse af alle forskningsmedarbejdere, nemlig projektet FAGFLADE, hvilket står for '*fag*sproglig grænse*flade*'.

Målet for dette projekt er at afprøve og udvikle teorier og metoder bag konstruktionen af en tekstfortolker, dvs. et system som analyserer en tekst og overfører viden herfra til vidensrepræsentation i en vidensbase. En sådan tekstfortolker, forestiller vi os, kan indgå som komponent i et ekspertsystem som en del af et vidensindlæringsmodul.

Den tekst vi har valgt som grundlag for projektet er den danske „Lov om aktieselskaber", og det sprog som er repræsenteret i loven er dermed det sublanguage som fortolkeren skal kunne analyser og forstå.

Projektet sigter ikke mod at udvikle et færdigt ekspertsystem, men er udgangspunkt for en række mindre projekter der fokuserer på problemer og områder knyttet til udviklingen af en tekstfortolker, det vil først og fremmest sige grundlaget for den syntaktiske og semantiske analyse, vidensrepræsentation, udvikling og afprøvning af parsing strategier, den maskinlæsbare ordbogs struktur og omfang samt terminologiske problemer så som identifikation og repræsentation af flerledede termer, synonymer, definitioner og begrebsrelationer.

I dag er projektet nået dertil at der er udviklet en parser i Quintus Prolog som kan klare simple sætningsstrukturer, dvs. sætninger der kun indeholder valensbundne led som optræder på den forventede position i sætningen. Vi kan altså ikke på nuværende tidspunkt klare spørgebisætninger eller topikaliseringer, heller ikke adverbialer, ledsætninger og adskillige andre ting.

Det output vi får fra parseren er dels sætningens syntaktiske struktur i form af et konstituentstrukturtræ, dels en semantisk struktur. Den syntaktiske struktur anvendes ikke i øvrigt af tekstfortolkeren, hvorimod den semantiske struktur skal danne udgangspunkt for den vidensrepræsentation som skal overføres til vidensbasen.

Et input som

Denne aftale påfører selskabet en forpligtelse

vil derfor give flg. output:

(1) s(np(det(denne), n(aftale)), vp(v(påfører),
np(n(selskabet)), np(det(en), n(forpligtelse))))

(2) påføre(agent(aftale), theme(forpligtelse),
locus(selskab))

Grundlaget for den syntaktiske analyse og den semantiske repræsentation er dels en konstituentstruktur grammatik, dels et leksikon med en leksikalsk beskrivelse af verberne ud fra en valensbaseret kombination af grammatiske funktioner og roller. Jeg skal i det følgende komme nærmere ind på de overvejelser der ligger til grund for denne leksikalske beskrivelse og på dens implementering i selve parserprogrammet.

2 Konstituenterne

Den parser vi har bygget er en 'left-corner, bottom-up' parser. Den starter således med det første ord i en input-sekvens og forsøger at opbygge en konstituentstruktur for et S som har dette første ord som sit venstre hjørne. Grammatikken der rummer flg. genskrivningsregler, hvoraf 2–9 repræsenterer de simple sætningsstrukturer som parserne kan klare netop nu:

```
(1)     S  -->   NP   VP
(2)     VP -->   V
(3)     VP -->   V    NP
(4)     VP -->   V    PP
(5)     VP -->   V    AP
(6)     VP -->   V    NP   NP
(7)     VP -->   V    NP   PP
(8)     VP -->   V    NP   AP
(9)     VP -->   V    PP   PP
(10)    PP -->   P    NP
(11)    PP -->   P    AP
(12)    NP -->   N
(13)    NP -->   DET  N
(14)    AP -->   A
(15)    AP -->   A    PP
```

I leksikon er hvert enkelt ordform forsynet med en oplysning om konstitutenttype. Subkategoriseringen af VP'erne genfindes i leksikon som en subkategoriseringsramme, der repræsenterer de konstitutenttyper som verbet kan forbindes med som valensled. Denne konstituentramme skal unificeres med en identisk ramme i en Prolog regel og tjener derfor det formål at effektivisere selve parsningen. Hvis unificationen lykkes, vil parseren vælge den pågældende regel.

Vi kan således begynde at se på opbygningen af en post i leksikon der har form af en Prolog klausul med prædikatet d/6 hvis argumenter alle er sammensatte termer:

```
d(word(_), lexeme(_), lexcat(_), gram_form(_),
     synt_spec(_,_), sem_spec(_,_)).
```

For verbet 'påføre' som optræder i eksempelsætningen ovenfor ser udfyldningen af de første argumenter således ud:

```
d
     word:              påfører
     lexeme:            påføre
     gram_form:         [pres]
     lexcat:            v
     synt_spec
          const_frame:       v_np_np
          adj:               [nil]
```

Argumentet *lexcat* indeholder oplysning om konstituenttype, *synt_spec* har to sammensatte termer som argumenter: dels *const_frame*, der angiver konstitutentrammen, dels *adj* (adjekt), der som argument har en liste over tilladte præpositioner hvis der kan optræde et valensbundet PP. Notationen `[nil]` angiver at det ikke kan forekomme ved dette verbum.

3 Funktioner

De funktioner der opereres med i FAGFLADE er hentet fra den valensteori som Herslund og Sørensen (1985, 1988a, 1988b) har udviklet i deres arbejde med franske verber. Der kan her kun blive plads til at skitsere hovedtrækkene i denne teori, der bygger på en distinktion mellem sætningsled der hører til hele sætningen, frie tilføjelser, og sætningsled som er dikteret af det konkrete verbum som optræder i sætningen. Et valensled defineres som det syntaktiske udtryk for de relationer der består mellem et verbum som semantisk prædikat og dets argumenter og mellem argumenter indbyrdes, og det er derfor kun verbalafhængige sætningsled med argumentstatus der kan opfattes som valensled. Endvidere antages det at valensled alene kan beskrives i form af funktioner, og at der til enhver leksikalsk beskrivelse af et verbum hører et valensskema med angivelse af hvilke funktioner, resp. valensled, det pågældende verbum forbindes med. Der opereres med et inventar på tre grammatiske funktioner: subjekt, objekt og adjekt. Hertil har vi i FAGFLADE for egen regning føjet en fjerde funktion, agential, for også at kunne håndtere passiv.

Sammenlignet med antallet af valensled i traditionelle valensgrammatikker er der sket en drastisk reduktion i antallet af funktioner. Funktionen adjekt repræsenterer på den ene side en valensgrammatisk nytænkning, og er på den anden side den funktion som rummer nøglen til rationaliseringen i antallet af funktioner.

Traditionelt er man vant til at arbejde med valensled som DO (dativobjekt), IO (indirekte objekt), A eller LO (adverbial/lokativobjekt), SP og OP (hhv. subjekts- og objektsprædikat). Ved anvendelsen af et givet verbum kan man således komme ud for at det er muligt at vælge mellem en konstruktion hvor verbet optræder med DO eller IO: man kan fx. 'give *nogen* noget' eller 'give noget *til nogen*', eller det kan være et valg mellem DO og LO: 'jeg har efterladt ham mine bøger' overfor 'jeg har efterladt mine bøger hos ham'. Hvad man derimod ikke kan komme ud for, er en kombination af begge funktioner samtidig, dvs. at de ovennævnte, traditionelle valensled gensidigt udelukker hinanden. Derfor antages det, at de alle repræsenterer en og samme ledfunktion som imidlertid kan realiseres på forskellig måde. Vi kan illustrere det med flg. eksempler:

(1) Han efterlod $\underbrace{\text{børnene}}_{\text{DO}}$ $\underbrace{\text{en betragtelig formue}}_{\text{O}}$

(2) Han efterlod $\underbrace{\text{en betragtelig formue}}_{\text{O}}$ $\underbrace{\text{til børnene}}_{\text{IO}}$

(3) Han efterlod $\underbrace{\text{sin bog}}_{\text{O}}$ $\underbrace{\text{hos sin broder}}_{\text{LO}}$

(4) Han efterlod $\underbrace{\text{sin bil}}_{\text{O}}$ $\underbrace{\text{på banegården}}_{\text{LO}}$

(5) Han efterlod $\underbrace{\text{hende}}_{\text{O}}$ $\underbrace{\text{dypt frustreret}}_{\text{OP}}$

Argumentationen for at sammenfatte i dette tilfælde DO, IO, LO og OP i een og samme funktion, *adjekt*, er, at uanset hvilken betegnelse man vælger at anvende så gælder det i alle konstruktioner at funktionen som sådan etablerer en ny relation i sætningen, i ovennævnte tilfælde en relation mellem *objekt* + *adjekt*. Denne relation opfattes som en sekundær prædikation der kan parafraseres med enten VÆRE eller HAVE og som supplerer verbets betydning ved at placere referenten for O i forhold til referenten for adjektet.

Placeringen kan være enten meget konkret som i eksemplerne (3): bogen er hos hans broder, og (4): bilen er på banegården, eller mere abstrakt som i (1): børnene har en betragtelig formue. Der kan imidlertid også være tale om en placering i forhold til en egenskab eller en klasse, nemlig i de tilfælde hvor adjektet svarer til en prædikativ konstruktion med enten SP eller, som i (5), med OP.

Begrundelsen for at indføre adjekt er derfor på den ene side at det er afhængigt af sætningens verbal som argument i logisk forstand, og på den anden side at det som argument gør det muligt at relatere et af de to fundamentale valensled, S og O, til et andet argument i sætningen og at denne relation mellem valensled indbyrdes er udtryk for en sekundær prædikation. Udfyldningen af adjektet kan være forskellig, men den sekundære prædikation er konstant.

Som tidligere nævnt omfatter den leksikalske beskrivelse af et verbum et valensskema som angiver hvilke funktioner der kan forekomme sammen med et givet verbum. Et sådant valensskema findes også i det leksikon som vores parser benytter, men inden vi skal se nærmere på dette vil det være nødvendigt at behandle de tematiske roller, idet de grammatiske funktioner i valensskemaet associeres med en semantisk rolle.

4 Roller

Med udgangspunkt i opfattelsen af verbet som et prædikat med tilhørende argumenter kan de *semantiske relationer* mellem argumenterne beskrives vhja. semantiske roller. Den semantiske analyse støtter sig ligeledes på Herslund og Sørensen (1985) samt endvidere på Korzen et alii (1983) og deres arbejde med den såkaldte PK-grammatik, idet den ovenfor omtalte valensteori og de syntaktiske funktioner er udgangspunkt for at associere funktioner med roller.

Antallet af roller synes at være en temmelig ubestemmelig størrelse, men vi har foretrukket at reducere antallet i henhold til de nævnte arbejder og i vores projekt indskrænket det til tre: agent, tema og locus.

agent svarer til den traditionelle opfattelse af agent og kan kun associeres med funktionerne *subjekt* og *agential.*

locus optræder kun ved prædikater med mindst to argumenter. Denne rolle relaterer referenten for et andet argument til den location som locus-argumentet udtrykker. *Locus* kan associeres med funktionerne *subjekt*, *adjekt* og *agential.*

tema er semantisk set den meste generelle og den mindst præcise af de tre roller. Referenten for tema er den 'entity' som enten påvirkes af handlingen eller placeres mht. en location. *Tema* kan associeres med funktionerne *subjet*, *object* og *adjekt.*

Distribution og kombination af roller associeret med funktioner kan således se ud som vist i flg. skemaet:

VERB	SUBJ	OBJ	ADJ	AGL
anse	agent	tema	locus	none
anses	tema	none	locus	agent
bestemme	agent	tema	none	none
bortfalde	tema	none	none	none
eje	locus	tema	none	none
ejes	tema	none	none	locus
finde	agent	tema	locus	none
påføre	agent	tema	locus	none
være	tema	none	locus	none

Af skemaet fremgår at fx. verbet *anse* optræder med en funktionsramme 'subjekt + objekt + adjekt' og at der dertil svarer rollerammen 'agent + tema + locus'. Udtrykket 'none' betyder at den pågældende funktion ikke kan forekomme ved det anførte verbum.

Jeg vil gerne knytte en enkelt kommentar til rollen locus med udgangspunkt i disse sætninger:

(1) spørgsmålet *bortfalder*	`sub(theme)`	
(2) ministeren *bestemmer reglen*	`sub(agent)`	`obj(theme)`
(3) stifterne *ejer* dette selskab	`sub(locus)`	`obj(theme)`
(4) disse aktier *er* fondsaktier	`sub(theme)`	`adj(locus)`
(5) bestyrelsen *finder* disse	`sub(agent)`	`obj(theme)`
undersøgelser nødvendige		`adj(locus)`
(6) denne aftale *påfører*	`sub(agent)`	`obj(theme)`
selskabet en forpligtelse		`adj(locus)`
(7) disse undersøgelser *anses*	`sub(theme)`	`adj(locus)`
for nødvendige af bestyrelsen		`agt(agent)`

I sætning (4) og (5) er locus associeret med funktionen adjekt, som her svarer til hhv. det traditionelle subjektsprædikativ (4) og objektsprædikativ (5), og i (6) er locus associeret med et adjekt der svarer til det traditionelle dativobjekt.

Jeg anførte tidligere at rollen locus kun optræder sammen med et andet valensled og at den denoterer en location for referenten denoteret af det andet

valensled. Ser vi på de tre sætninger ovenfor så er det subjektet, *disse aktier*, der i (4) localiseres ekstensionalt i forhold til en klasse, klassen af *fondsaktier*, adjektet, og i (5) er det objektet, *disse undersøgelser*, der localiseres intensionalt med hensyn til egenskaben at være nødvendig, udtrykt i adjektet. Ligesom for funktionernes vedkommende kan locationen også være mere abstrakt sådan som det er tilfældet i (6) hvor objektet, *en forpligtelse*, localiseres i forhold til *selskabet*, adjektet, forstået på den måde at det er selskabet som har en forpligtelse.

Vi kan hermed afslutte udfyldningen af argumenter i leksikonposten og indsætte valensskemaet i form af de semantiske specifikationer:

```
d
     word:                påfører
     lexeme:              påføre
     gram_form:           [pres]
     lexcat:              v
     synt_spec
          const_frame:    v_np_np
          adj:            [nil]
     sem_spec
          func_to_role
               sub:            agent
               obj:            theme
               adj:            locus
               agl:            none
            aspect:       nil
```

Jeg vil ikke komme yderligere ind på rollerne her. På nuværende tidspunkt i projektet fungerer denne opfattelse af roller kombineret med valensteorien udmærket når det gælder om at producere en semantisk repræsentation af de simple sætninger vi arbejder med. Når vi skal i gang med at etablere selve vidensbasen kan det imidlertid meget vel vise sig at denne semantiske repræsentation er utilstrækkelig, og at fx. antallet af roller skal suppleres.

5 Opbygningen af den semantiske repræsentation

De oplysninger der ligger i leksikon som vist ovenfor indtastes manuelt ved hjælp af en særlig editor, DICTED, udviklet af Henrik Kersting (under udgivelse) i forbindelse med FAGFLADE projektet. De danner udgangspunkt for en procedure som genererer et nyt leksikon i form af en prolog database, hvor lexemnavnet associeres med en semantisk repræsentation i form af et lambdaudtryk beregnet på baggrund af argumenterne i den sammensatte term `func(tion)_to_role`. Det skal her indskydes, at kun verber og adjektiver, herunder også participier, på nuværende tidspunkt forekommer med en specifik semantisk repræsentation. Alle andre ordklasser, dvs. determinativer, præpositioner og substantiver, har det pågældende leksemnavn som semantisk repræsentation. Sammensatte NP'er har

kerneleddets leksemnavn som semantisk repræsentation og PP'er har styrelsens kerne som semantisk repræsentation. Udtrykkene *aktie*, *aktierne*, *disse aktier* og *af disse aktier* har således alle en og samme repræsentation, nemlig *aktie*.

Argumenterne i termen `func_to_role` er de fire syntaktiske funktioner i ordnet rækkefølge: subj, obj, adj, agl, afbildet på de semantiske roller agent, tema og locus. Alle fire funktioner optræder, også i de tilfælde hvor de rent faktisk ikke kan forekommer og derfor som værdi har 'none' som vi så det ovenfor. De anvendes ved konstruktionen af lambdaudtrykkket, hvis argumenter ligeledes forekommer i kanonisk rækkefølge: `agent(_)`, `theme(_)`, `locus(_)`.

Verbet 'påføre' har således flg. `func_to_role` repræsentation:

```
func_to_role( sub(agent), obj(theme), adj(locus), agl(none))
```

som konverteres til flg. lambdaudtryk:

```
X^Y^Z^påføre(agent(Z), theme(Y), locus(X) )
```

Dette udtryk beregnes af flg. regel:

```
% Verb + adject + object (V NP NP).
% Verb + object + adject (V NP PP & V NP AP).

build_LambdaExpr(Lexeme,
  func_to_role(sub(RoleS),obj(RoleO),adj(RoleAd),agl(none)),
  X^Y^Z^Expr) :-
                RoleS \== none,
                RoleO \== none,
                RoleAd \== none,
                RoleSTerm =.. [RoleS,Z],
                RoleOTerm =.. [RoleO,Y],
                RoleAdTerm =.. [RoleAd,X],
                sort_RoleTerms([RoleSTerm,RoleOTerm,
                                RoleAdTerm], SortedRT),
                Expr =.. [Lexeme|SortedRT].
```

Når parseren skal bygge en sætnings semantiske struktur har den da for verbernes vedkommende den type informationer til sin rådighed som vist på flg. side.

Kolonnen med funktioner og roller kan læses på to måder: dels som udtryk for en afbildning af syntaktiske funktioner på roller, dels kan man for hvert verbum aflæse en funktionsramme og en rolleramme ved alene at læse funktioner og roller. Rammerne udtrykker tilladte kontekster for et verbum i form af konstituenter, funktioner eller roller. Strukturerne, den syntaktiske og semantiske, bygges af parseren ved at relatere ordene i den konkrete sætning til konstitutenttyper, roller eller funktioner.

	Konstituent ramme	Func-to-role mapping	Prolog lambda udtryk
VERBER			
anser	v_np_pp	sub(agent) obj(theme) adj(locus)	X^Y^Z^anse(agent(Z),theme(Y),locus(X))
anses	v_pp_pp	sub(theme) adj(locus) agl(agent)	X^Y^Z^anse(agent(Y),theme(Z),locus(X))
bestemmer	v_np	sub(agent) obj(theme)	Y^Z^bestemme(agent(Z),theme(Y))
bortfalder	v	sub(theme)	Z^bortfalde(theme(Z))
ejer	v_np	sub(locus) obj(theme)	Y^Z^eje(theme(Y),locus(Z))
ejes	v_pp	sub(theme) agl(locus)	Y^Z^eje(theme(Z),locus(Y))
er	v_ap	sub(theme) adj(locus)	Y^Z^være(theme(Z),locus(Y))
er	v_np	sub(theme) adj(locus)	Y^Z^være(theme(Z),locus(Y))
finder	v_np_ap	sub(agent) obj(theme) adj(locus)	X^Y^Z^finde(agent(Z),theme(Y),locus(X))
følger	v_pp	sub(theme) adj(locus)	Y^Z^følge(theme(Z),locus(Y))
påfører	v_np_np	sub(agent) obj(theme) adj(locus)	X^Y^Z^påføre(agent(Z),theme(Y),locus(X))
undtager	v_np_pp	sub(agent) obj(theme) adj(locus)	X^Y^Z^undtage(agent(Z),theme(Y),locus(X))

Under parsningen af sætningen „Denne aftale påfører selskabet en forpligtelse" vil parseren—når den skal bygge den semantiske struktur for sætningen—starte med at identificere *denne aftale* som et NP der opbevares i en variabel. I leksikon under *påfører* vil den få oplysninger om den konstituentramme der gælder for verbet samt det lambdaudtryk der svarer til verbet. Ud fra en matching af konstituentrammen finder den frem til den regel som skal anvendes og identificerer herefter en konstitutentstruktur med to NP'er efter verbet. Det første NP, *selskabet*, associeres med rollen locus som er tildelt adjektet, det andet NP, *en forpligtelse*, med rollen tema som er tildelt objektet. Til sidst vil den udfylde rollen agent med subjektet, det første NP den læste, og aflevere det færdige resultat:

```
påføre(agent(aftale), theme(forpligtelse), locus(selskab))
```

Den semantiske repræsentation vi kan få frem nu er baseret på rollestrukturen fordi vi anser den for et centralt element når den information sætningen indeholder skal fastholdes som vidensrepræsentation i vidensbasen. Vi er godt klar over at der er flere forskellige semantiske fænomener som ikke er repræsenteret i den nuværende fase, tidsrelationer, modalitet eller fx. adverbialsemantik. Vi har også en liste over punkter som skal løses og andre punkter som forestår. Det

er også derfor jeg har kaldt indlægget „På vej mod en fagsproglig tekstfortolker", og forhåbentlig kan vi når vi mødes igen om to år præsentere en yderligere udbygning og forbedring af vores tekstfortolker.

Hvis der er nogen der er interesseret i en mere udførlig rapport om projektet samt en detaljeret gennemgang af programmet, så kan man orientere sig nærmere i LAMBDA NR. 11.

Litteratur

Hansen, Steffen Leo (forthcoming), FAGFLADE: Text, Types and Tokens, *Lambda*.

Hansen, Steffen Leo and Carl Vikner. 1989. FAGFLADE: The Initial Phase of a Project in Natural Language Interpretation. *Lambda*, 11. Institut for Datalingvistik, HHk.

Herslund, Michael. 1988a. On Valence and Grammatical Relations. *Copenhagen Studies on Language*, *11*:3–35, Copenhagen.

Herslund, Michael. 1988b. *Le datif en français*. Louvain-Paris, Editions Peeters.

Herslund, Michael and Finn Sørensen. 1985. *De franske verber. En valensgrammatisk fremstilling. I. Verbernes syntaks*. Romansk Institut, Københavns Universitet.

Kersting, Henrik. 1989. DICTED: An Editor for Dictionaries Stored as Prolog Databases. *Lambda*, 12.

Korzen, Hanne, Henning Nølke, Henrik Prebensen and Finn Sørensen. 1983. PC-Grammar: An Alternative?, *Acta Linguistica Hafniensia*, *18*:5–53, Copenhagen.

Sørensen, Finn. 1988. Om rollen Locus, Notat. Institut for Datalingvistik, Handelshøjskolen i København.

Institut for Datalingvistik
Dalgas Have 15
DK-2000 København F.
Danmark

JANNE BONDI JOHANNESSEN

Is Two-level Morphology a Morphological Model?

Abstract

This paper contains a close look at Koskenniemi's Two-level morphology from a linguistic point of view. The model will be compared to three other traditional, linguistic morphological models, IA, IP and WP. It will be shown that there are linguistic phenomena that can hardly be handled by some of the just mentioned models, and not at all in a linguistically satisfactory manner by the Two-level morphology.

1 Introduction

Koskenniemi's Two-level morphology (TM) has become well known since it was developed in 1983. One reason for this is probably that it is one of the few models within computational linguistics that has taken morphology seriously. To store full wordforms, inflected and derivated, in the lexicon may be possible for a language like English, with relatively poor morphology. But Koskenniemi saw that for Finnish, where a single verb can have between 12.000 and 18.000 different graphemic forms (included clitics), such a solution would not work. If the American computational linguists had been Red Indians speaking the Cherokee-language Oneida, instead of white and English-speaking, then they too would probably have developed a morphological model that could handle their verbs with up to 100.000 forms each.

I assume the Two-level morphology to be well-known, and I will thus only give a very short description of it, before I procede to the main task; to compare the Two-level morphology with other morphological models, and to see if this model can be said to be a morphological model.

2 A Short Description of Two-level Morphology

The Two-level morphology is designed to perform both analysis and synthesis on the basis of more or less the same data. It has at its disposal a rule module and

a lexicon module. The rule module takes care of one-segment correspondences, mostly phonological ones. The lexicon module may consist of several lexicons, one or more for stems and others for affixes. From each lexicon there is a pointer to the next possible lexicons. The entries in the lexicon may look different from their surface representation, which the rule module takes care of. (1) and (2) are examples of lexicon entries in two sublexicons for Norwegian:

(1)
```
LEXICON Nouns
vintEr  /MNoun  Lexeme=WINTER
gutt    /MNoun  Lexeme=WINTER
```

(2)
```
LEXICON /MNounSg
0       /Genitive   Num=sg/Defin=ind/Gender=m/.
en      /Genitive   Num=sg/Defin=def/Gender=m/.
```

The information that we get about a word-form that is analyzed, is the information that is accumulated through all the lexicons that have been consulted. Thus if we analyze *vinteren*, we get the information from both the stem- and the suffix lexicon:

(3) `vinteren: Lexeme=WINTER  Num=sg/Defin=def/Gender=m/`

(This accumulation is the reason for the seeming zero-inflectional morph that is apparent in (2). It is not meant as a suffix, it is just there to ensure that the information about singular and indefinite is collected. This information could not have been represented in the stem lexicon, even if the stem is identical to the word form of indefinite singular, because the stem lexicon also points to lexicons for plural and definite forms. Since information is accumulated on its way through the lexicons, we would, if we had given the singular indefinite information in the stem lexicon, have gotten absurd results like *vintrene = singular, plural, indefinite, definite.* In other words: The stem lexicon can only include information that is common for all the wordforms belonging to one lexeme.)

The lexical form of the entry we have looked at is `vintEr` (1), but the surface representation should be as in (4), of course:

(4) vinter

The default alphabet then includes a lexical `E` that corresponds to a surface `e`, (`E:e`), (in addition to the usual `e:e`). The reason for this cumbersome representation is that *vinter* and many other Norwegian lexemes go through a morphophonemic change that deletes the `e` before certain morphological endings:

(5)
```
Singular:  vinter - vinteren
Plural:    vint_rer - vint_rene
```

If we want to have the same lexical entry for all wordforms of one and the same lexeme, which is obviously the most satisfactory solution from a linguistic point of view, we have to make the 'e' which can go away, a little different from other 'e'-s that can not be deleted (e.g. in *vinteren*), so that we can later formulate a rule that refers only to the appropriate 'e'. Only then can we keep one lexical entry for this lexeme, `vintEr`, instead of two, e.g. as in (6):

(6)
```
vinter      /MNounSg      Lexeme=WINTER
vintr       /MNounPl      Lexeme=WINTER
```

We then formulate a rule that overrules the lexical default values:

(7)
```
"E-deletion in stem before plural"
E:0 <= _ Liquid PlSuffix ;
```

(The rule context consists of names that refer to certain sets and definitions that we have predefined.)

3 How Can Two-level Morphology be Characterized When Compared to Other Linguistic Models for Morphological Description?

In our century traditionally there have been three models for morphological analysis; IA (Item and arrangement), IP (Item and Process) and WP (Word and paradigm). For a discussion of these models, see Hockett 1954, Matthews 1972, 1974, Robins 1970. A fourth model can also be mentioned, which I shall not go into here; NM (Natural morphology), see Wurzel 1982 or Bybee 1985. Below I shall compare the Two-level morphology with each of the three models. (The discussion will to a large degree be built on Johannessen 1988.) As they have not existed quite simultaneously, I will start with the oldest one and then end with the newer one.

3.1 Item and Arrangement

The main characteristic of this model is that there are minimal units, morphemes, that can be arranged in a number of ways to form bigger units. The morphemes are abstract units that are represented through their allomorphs. Since at the time of IA (approximately 1930–1950) the view held that syntax and morphology should be described in the same way; that there is ideally a one-to-one relationship between morpheme and allomorph, more precisely a relationship where one morpheme has one surface realization and vice versa:

(8) IA:

Morphemes:	Allomorphs:	'Word'	
{gutt} + {indef pl}	gutt-er	gutter	(= boys)
{hus} + {indef pl}	hus-0	hus	(= houses)

IA and TM have in common that the different elements are arranged lexically, as we see. But the elements of IA (morphemes) are abstract, so that in (8) we have the same second element in both words, it is just realized differently (different allomorphs). In TM on the other hand, the two plural formatives have nothing in common because of their different realization in the lexicon. In TM they are actually two different endings, since they are different graphemically:

(9) TM:

Lexical (stem) entries:	Lexical (affix) entries:	'Word'
gutt	er	gutter (= boys)
hus	0(nothing)	hus (= houses)

We do not see any morphosyntactic information here, since it is irrelevant for the model. The grammatical features that are present in the lexicon entries, can not be made use of by the rules. TM does not get past the concrete level of allomorphs, it can thus not be equivalent with the IA-model.

3.2 Item and Process

The IP model was popular until the 1960s. Like IA this too is a model based on the morpheme-allomorph distinction. The difference from the IA model is that the IP model allows processes, that is, it allows elements to undergo a metamorphosis to gain a shape different from the original one. This is possible both at phoneme and morpheme level. The model allows rules of both sorts.

When it comes to the process part, we can say that IP and TM have something in common. We have seen the rule part of TM, and even if there we deal with pairs of segments that correspond with each other in certain circumstances, the idea could be that the correspondence looks like a process. (In fact: The rule formalism is designed to take care of morphophonemic changes that are abundant in Finnish (vowel harmony and consonant gradation)). I can also cite Karlsson and Koskenniemi (1985:127): What is described by rules is "fairly natural one-segment modifications; mostly automatic, transparent, productive, exceptionless alternations between phonologically closely related single phonemes in predominantly phonological contexts." Phonological rules are then taken care of in TM. Morphological rules, on the other hand, i.e. processes that form e.g. plural word-forms from stems, are not possible in TM, which handles all formatives in the lexicon part.

Morphophonemic changes can thus be described in TM in a manner similar to IP (when we ignore the lack of morphemic level in TM) :

(10) IP:

Morphemes:	Allomorphs:	Morphophonemic rule:	'Word'
{bok} + {indef pl}	bok-er	o -> ø/ _ C er	bøker

(11) TM:

Lexical (stem) entry:	Lexical (affix) entry:	Two-level rule:	'Word:'
bok	er	o : ø/_ C er	bøker

The two models are similar so far, but only as long as the rule-context is purely phonological (graphemical). Morphological context is impossible in TM, but possible in IP :

(12) IP:

Morphemes:	Allomorphs:	Morphophonemic rule:	'Word'
{bok} + {indef pl}	bok-er	o -> ø / _ C {+pl ind}	bøker

The only way TM can use morphological information, is to make the morphological information 'phonological' by adding extra characters in the rule context. The extra character will then symbolize the morphological class or feature. E.g. can we put a dollar sign in front of the affix (which must of course also be present in the lexicon where the affix has its entry) or the morphological ending may itself get a different lexical shape, to satisfy the need for a context that can be morphologically unique:

(13) TM:
```
+ morphophonemic rule: o -> ø / _ C  $er
  or
+ morphophonemic rule: o -> ø / _ C  Er
```

Now one might want to reply that it is not important. But in natural language it is often necessary to distinguish between phonological and morphological conditioning. A number of Norwegian dialects have productive palatalization of /k/ and /g/ in front of noun suffixes, but not otherwise:

(14)
/stok/	(= stick)
/stoc-en/	(= the stick (nom.))
/drek-e/	(= (to) drink)
/stoc-a/	(= the stick (dative))
/tak-a/	(= thanked)

It would be a mistake to phonologize this type of morphophonological process. It is the morphological category of the suffix that conditions the alternation of the stem, and not the phonological shape.

3.3 Preliminary Summary

We have looked at two linguistic models for morphological analysis which both have the distinction morpheme-allomorph, i.e. which take the segmental side of natural languages very seriously. One of them, IP, is a little more flexible in that it accepts segmental changes triggered by some phonological or morphological feature.

When we compared TM to these models, we saw that it seems to be inspired by them. It too emphasizes the segmental side, through the linked lexicons. Also it seems to be inspired by the rule module of IP, although TM only allows "phonological" conditioning for the triggering of rules.

The serious defect of TM, however, is that it lacks a conceptual, morphological level. It operates only at the concrete, phonological (graphemic) one, which is of course the reason for the just mentioned phonological triggering.

We have seen that IA and IP are not fashionable today. The reason is that they are too limited to account for all facts about natural language. But the knowledge that morphological processes can be more than just elements arranged in a certain order is not new. E.g., the American linguist Edward Sapir in his book "Language" in 1921 distinguished between six different processes,

(i.e. ways of expressing morphological characteristics) where he included things like "internal modification of the radical or grammatical element", reduplication, accentual and quantitative processes.

It is this knowledge that led to a revision of morphology by Matthews in 1972.

3.4 Word and Paradigm

The WP is a model that attempts to take morphology seriously in that a grammatical feature can be realized in many different ways, like Sapir suggested. In this model the underlying representation is even more abstract than in the two preceding morpheme models. Any wordform is represented through its lexeme (an invariant representation of the word) with the grammatical (morphosyntactic) information represented as an unorded set:

(15) GUTT_{N} masculine, indefinite, plural

To reach the correct wordform, the stem, which is the starting point, can go through various processes:

(16) GUTT_{N} masculine, indefinite, plural:

Stem:	gutt
+ operation:	suffix -er
= Word	gutter

The number of processes is potentially infinite, the reason for this is that it is the word which is the basic unit in this model: If a wordform differs in more than one way from any other wordform in the same paradigm, then it goes through more than one process to reach its final shape. And all the processes will be exponents of the same grammatical (morphosyntactic) feature. We shall look more closely at three linguistic phenomena that are problematic for the two other linguistic models and for TM, but not for WP.

The first and most important difference between the WP model and the morpheme models is that while the morpheme models need a one-to-one relationship between morphological contents and its realization, WP accepts a many to one/one to many-relationship:

(17) BOK_{N} feminine, indefinite, plural

Stem:	bok
+ operation:	suffix -er
+ operation:	change stem vowel
= Word:	bøker

As we have seen previously, the morpheme models and TM necessarily must give priority to one of the realizations, and let the other(s) be conditioned by it. This poses strong constraints upon the linguist, who will have to give arbitrary priority to one realization, e.g. to let an affix trigger a vowel change.

A second problem is that while there still might be some universal claims about the priority of affixes to 'internal modifications', so that the first problem may look smaller, a worse case occurs when there are more than one *affix* that represents a grammatical feature. This is the case in German past participles:

(18) ge-sag-t

In the German case we might still argue to give priority to the suffix, though, since the other verbal features in the language are marked by suffixes, but consider the Kubachi dialect of Dargwa from the Northeast Caucasus, where each adjective agrees with the noun's gender and number both initially and finally, in addition to agreeing with number penultimately:

(19) Kubachi (dialect of Dargwa, Northeast Caucasus):
b-īk'a-zi-b qalč'e 'little bird'
d-īk'a-žu-d qulč'-ne 'little birds'
(The example is from Anderson 1988:32)

The morpheme models that we considered previously would have to give one affix priority over the other, or make use of the concept circumfix. The TM on the other hand, does not present a satisfactory solution.

It could represent the prefixed affixes in a separate lexicon, and have pointers from there to the further lexicons. This however would mean that the stem would occur as many times in the stem lexicon as there are prefixes in the language, since the suffixes have to agree with the prefixes. Other equally inelegant solutions are also possible.

But even if TM has problems in representing phenomena like the above, it can do it in an inelegant way. The third problem is more serious, however:

The third area is phenomena that are not 'segmental' in their nature. We recall Sapir who allowed internal modification as a means for representing morphological features. A typical example of this is the Germanic umlaut and ablaut, which in many cases is the sole distinguishing factor between two wordforms of the same lexeme:

(20) mann (indef sg)–menn (indef pl) (man–men)
se (infinitive)–så (preterite) (see–saw)
mouse (sg)–mice (pl)

Both the morpheme models that we looked at and TM have problems in describing such phenomena as vowel alternation, when it is not the biproduct of some other segmental, morphological process, but rather the main exponent of that morphological feature.

The main problem for TM is that all grammatical (morphosyntactic) information is only represented in the lexicon part of the system, and not in the rule part. If a two-level rule should take care of this information, it would 1) need a segment to trigger the rule, and 2) let the morphological information be accompanied by the trigger, and not by what is really the difference between the two forms—the vowel alternation:

(21) TM:

Lexical (stem) entry:	Lexical (affix) entry:	Two-level rule:	'Word':
`mann`	`-0`	`a:e <=> _ C* -0`	menn

4 Conclusion

The question which is the title of this paper, 'Is Two-level morphology a morphological model?', can from the previous discussion be answered quickly and clearly negatively. The reason for this is that a minimum to be demanded from a morphological model, is for it to accept morphological features and categories as primitives. In that way it could allow morphological conditioning on stem variation. But TM has to make use of artificial null-segments and other triggers, i.e. it has to make the originally morphological context 'phonological', segmental. Anything that is morphological—like the information in the lexicons—can be used only by the linguist, not by the model. In this way it does not have any possibility of making generalizations independently of phonological shape. It is even worse than the morpheme models, as it can not say that both a suffix and a vowel alternation could represent 'plural', e.g..

It therefore seems fair to say that TM is not a morphological model. There are, however, languages that can be well described by it, viz. the languages that are usually called agglutinative in traditional typology. These are languages like Finnish, whose morphology consists of easily separable affixes that each corresponds to one morphological unit. The phonological alternations in Finnish are not realizations of morphological features, only phonologically determined automatic alternations, which the two-level rules handles well. The important rule-part of the model makes the TM more phonological than morphological.

We may ask a last question: Is it important that a computational model has linguistic qualities? From the above discussion I think the answer should be positive.

References

Anderson, S.R. 1988. Inflection. In Hammond, M. & M. Noonan (eds.), *Theoretical Morphology*, 23–44. Academic Press Inc., San Diego.

Bybee, J. L. 1985. *Morphology. A Study of the Relation between Meaning and Form.* John Benjamins Publishing Company, Amsterdam.

Hockett, C.F. 1954. Two Models of Grammatical Description. *Word*, 10, 210–231. New York.

Johannessen, J. 1988. *Automatisk morfologisk analyse og syntese.* Lingvistisk institutt, Universitetet i Oslo, Oslo. Hovedoppgave.

Koskenniemi, Kimmo. 1983. Two-level Morphology: A General Computational Model for Word-form Recognition and Production. Publ. no. 11, Department of General Linguistics, University of Helsinki, Helsinki.

Källgren, G. 1983. Computerized Analysis and Synthesis of Finnish Nominals. In *Papers from the Seventh Scandinavian Conference of Linguistics*, 433–444. Publ. no. 10, Department of General Linguistics, University of Helsinki, Helsinki.

Karlsson, F. and K. Koskenniemi. 1985. A Process Model of Morphology and Lexicon. *Folia Linguistica*, 207–231, Haag.

Matthews, P.H. 1972. *Inflectional Morphology.* Cambridge University Press, Cambridge.

——1974. *Morphology. An Introduction to the Theory of Word-structure.* Cambridge University Press, Cambridge.

Rankin, I. 1986. SMORF—An Implementation of Hellberg's Morphology System. In *Papers from the Fifth Scandinavian Conference of Computational Linguistics*, 161–172. Publ. no. 15, Department of General Linguistics, University of Helsinki, Helsinki.

Robins, R.H. (ed.) 1970. *Diversions of Bloomsbury. Selected Writings of Linguistics.* North-Holland Publishing Company, Amsterdam.

Sapir, E. 1921. *Language.* Harcourt, Brace & World, New York.

Wurzel, W. U. 1982. *Phonologie - Morphonologie - Morphologie.* In *Linguistische Studien*, 93, Reihe A, Arbeitsberichte. Akademie der Wissenschaften der DDR, Zentralinstitut für Sprachwissenschaft, Berlin.

Institutt for humanistisk informatikk,
Universitetet i Oslo,
P.b. 1102 Blindern, N-0317
Oslo 3.

GUNNEL KÄLLGREN

Automatic Indexing and Generating of Content Graphs from Unrestricted Text*

1 Introduction

For quite some time, I have been exploring the surface signals of language, and trying to put them to as much use as possible, primarily in morphology-based part-of-speech assignment (Källgren 1984a,b,c, 1985) and pattern-based syntactic analysis (Källgren 1987). This kind of large-scale, probabilistic parsing on the basis of morphological and syntactic patterns has lately come to use in several projects. Some models that have been documented are the UCREL parser in England for the Brown and LOB corpora (Garside & Leech 1987), the VOLSUNGA parser in the USA for the Brown corpus (DeRose 1988), Ken Church's stochastic models (Church 1988), as well as other work in the US (Black 1988), but I am sure work along these lines is going on in several places.

The impetus behind the works just mentioned is mainly a need for analyzing large amounts of unrestricted text in a way that is not too resource-demanding, either on time or on computing power. As a secondary goal, I have seen the needs of large-scale information retrieval. Keeping my original surface-orientation, I have gone further from the analysis into parts-of-speech and constituents and started to look at the extraction and representation of some kind of 'content' from the surface of texts, without any kind of knowledge base support. This might seem quite impossible. Many of the other papers in this volume deal with how unavailable inferences are to the comprehension of text, and of course they are right. If the aim is to build a computerized system that will in any way simulate language understanding, it is necessary to have a large knowledge base and mechanisms for making inferences from it, but there are also applications where the human knowledge and inferencing capacities can be used instead. My approach in the experiment to be reported here has been to let each one do

*My sincere thanks to Boris Prochaska and Sten-Erik Bergner, formerly at Ericsson Telecom, who wrote the original version of the graph drawing program that I have used, and to Howard Gayle who set me in contact with them. Sune Magnberg has done a great job in transferring the program to a PC environment and has also written the part of the whole system that checks for collocational pairs. I also wish to thank Benny Brodda, Kari Fraurud, and Sune Magnberg for valuable comments on earlier drafts of this article.

what he/she/it is good at; let the computer store, sort, and find facts, and let the human being do the inferencing.

This is actually the way it normally is in the field of information retrieval, where it is the human user that (interactively, at best) decides whether she/he has got the desired material. The problem is then to produce an optimal basis for that decision. I have so far been testing automatic indexing of texts, i. e. to find central concepts in texts automatically (Källgren 1984c). This is not all that difficult, but not all that effective either. In order to be covering, the index lists will easily grow too long; if shortened, important descriptors may disappear. This is an instance of what is known as the conflict between recall and precision. Still it is clear that simple word lists can be quite informative, though a bit boring.

I have also gone to the other extreme and tried to actually generate coherent abstracts automatically (Källgren 1988). This would certainly be desirable, and though it is far away, I do not think it is impossible.

What I will report on here is something in between. It is a way of showing central concepts and their interrelations in what I have called 'content graphs'. It is then up to the user to interpret the relations and make the inferences that are needed in order to get a picture of the content of the concerned text.

In principle, it may be wrong to talk about content graphs. The graphs picture 'what a text is about', rather than its content, but to call them 'aboutness graphs' is just too clumsy. What these graphs actually do is to give a hint about the content of a given text, not a full and true representation of it.

Given these limitations, the graphs still have their justification. The derivation of them from texts is an interesting task, for a set of reasons: The process can be fully automatized. It can be run on unrestricted text without manual preprocessing. The output can often be strikingly accurate. It also seems to have some interesting psycholinguistic implications.

2 Surface-Oriented Indexing and Information Retrieval

Of course, different kinds of surface-oriented methods have been used extensively through the years in research on automatic information retrieval. Salton & McGill (1983) give a broad overview of the field and also report interesting data on the notoriously difficult evaluation of information retrieval systems. Many of the systems described make use of adjacency and term frequency features in different combinations, and some systems take into account not only terms that are immediately adjacent but also terms that appear within a limited distance of each other (ibid. p. 33). Sophisticated methods for computing frequency and relative weight of terms occuring in documents are also described. Frequency of immediately adjacent terms is used as a means of finding complex terms (such as 'information retrieval'), but the authors do not report any work dealing with frequency of more loosely connected terms. The results of their experimentation is encouraging in that they show that well-constructed automatic

indexing systems may perform quite as well as manual indexing, and also that simple surface-based procedures can be as good as or better than more refined methods (ibid. p. 102).

The original inspiration for this work comes from Phillips (1985) which relates to some very early work within computational linguistics (e.g. Sinclair et al. 1970). Many of the ideas suggested at that time would deserve a renewed interest today, when computational power as well as linguistic knowledge has increased (the former considerably more than the latter, however). A good old idea that turns up every now and then is the concept of collocation. Collocations are words that appear together considerably more often than would be expected on purely statistical grounds. They can either be immediately adjacent or appear within a limited distance from each other. This distance, in terms of number of words, can be called a span. 'Collocation' and 'span' are basic concepts in my method for generating graphs.

To search for collocations can be a way of finding the idioms of a language, both those that are entirely fixed, like 'red tape', and those that contain slots, like 'pull someone's leg'. It can also be a way of finding relations between words. In the kind of content analysis that is carried out in the social sciences, cooccurrences between predesigned pairs or sets of words have sometimes been investigated. My treatment of the collocations is related to both uses; to the former in regarding all words in a text as liable to enter collocative relations, to the latter in assigning some kind of semantic load to the relations. This amounts to saying that the fact that two words cooccur suspiciously often carries some meaning in itself.

There is, however, one limitation on what words can form collocations in my system. To avoid uninteresting collocations, such as article plus noun etc., I only take content words into regard, not form words. The distinction between content words and form words is of course not totally clear (which linguistic distinctions are?), but clear enough to be operationalizable. There are some rare instances of homography, as when 'out' can be a noun in connection with baseball, and some adverbs can be felt to be 'content heavy'. Disregarding this, form words can be given as lists of words from the closed categories: pronouns, prepositions, adverbs, auxiliary verbs, articles, particles, and conjunctions. Removing such words from running text, or placing them on so-called 'stop lists', is a much used practice in automatic indexing, and it is estimated that about 250 common words cover 40-50 percent of an average English text (Salton & McGill p. 71). Lemmatization and rank ordering, as described in steps 2 and 3 in the algorithm below, are also well-established techniques.

3 The Algorithm

The method can now be presented in the form of an algorithm, which I will proceed to describe and examplify.

(1) THE ALGORITHM:

1. Eliminate all form words.
2. Lemmatize the remaining words, i. e. disregard differences at the end of words that either belong to the sets of derivational and inflectional endings or are admissible combinations of those.
3. Rank the lemmas in order of frequency.
4. Decide on a lowest frequency of lemmas and exclude the lemmas below that level. The level is dependent on the length of the text and the degree of recall wanted. The lemmas above the frequency threshold form the set of INDEX words.
5. Decide on a SPAN length. The length of the span is not dependent on text length or wanted recall, but might be language dependent.
6. Find all instances where two words from the index set appear within the same span. These are the COLLOCATIONAL PAIRS.
7. Find all pairs that are identical, disregarding order, as the pairs in themselves are unordered.
8. Rank the collocational pairs in order of frequency.
9. Decide on a lowest frequency of collocational pairs, based on the same principles as for the lemma frequency. Pick out all pairs above that frequency.
10. Construct ADJACENCY LISTS, i. e., for each lemma, list all other lemmas with which it forms a pair.
11. Use the adjacency lists as input to the GRAPH-drawing program.

Alternative version with graphs drawn by hand:

10′ Try to find optimal orderings of the pairs, look for central concepts that occur in many pairs.

11′ Draw the GRAPH.

4 Implementation

The system has been tested for Swedish, and the programs for removing form words and lemmatizing content words so far only exist in Swedish versions. They are a set of Lisp procedures running on PCs. For the purpose of demonstration, I will however use an English text where the lemmatizing has been done manually. The full English text is given in Appendix A and all examples below are taken from that text. Appendix B contains three shorter Swedish texts and Appendix C their respective graphs. These have been produced in a wholly automatical way as specified in the algorithm.

The removal of form words is, as stated above, simply done by removing all words on a pre-specified list from the text. Lemmatization is normally a far from

trivial process, but can in this connection be done in a simplified manner. The text, devoid of its form words, is treated as a word list and sorted alphabetically. Words that start in a similar way are compared as to their endings. If two words are identical all the way, they clearly belong together. If the parts where they differ belong to the pre-defined set of endings, they are also regarded as belonging together. This matching can be done in more or less sophisticated ways. Either the pairs of matched endings must signal the same part-of-speech and be morphologically connected, as when *berry* and *berries* form a lemma *berr* with matching endings *y* — *ies*. Otherwise, anything 'endinglike' will do, as when *favorite*, *favored* and *favorable* are matched. Actually, I think the latter alternative, lemmatizing across part-of-speech boundaries, should be preferred, as we are primarily looking for semantic relations, regardless of how they are expressed. In this way, the truncated stems (see below) that represent each lemma come to refer to a concept more than just a word. This kind of lemmatization has been called 'root lemmatization' and a linguistically sophisticated way of doing it is described in Fjeldvig-Golden (1984).

Semantically erroneous lemmatization can of course not be avoided, as when *late*, which in the text is used in the sense of *deceased*, is lemmatized with a temporal *lately*. This is however not such a big problem as one might suspect, as such infelicitious pairings rarely reach a frequency where they will influence the outcome of the entire process. To solve the problem, on the other hand, would demand a very large apparatus based on not only semantic but also pragmatic knowledge.

What is left when possible endings have been removed is a truncated stem, where the truncation process has sometimes been quite brutal. The truncated stems can now be sorted according to frequency and those below a certain level are removed. For the short sample text of two typed pages, a frequency level of two was settled. This is of course the minimal frequency. A frequency of one has no discriminatory effect whatsoever, as those lemmas can never occur in more than one pair. The lemmas with a frequency of two or more in the sample text are given in (2). They are called *index words* and are saved on a separate file to be matched against the full text in the next step of the process. For a longer text, a higher frequency level might have been preferred in order to limit the set of index words. This is a typical instance of balancing recall and precision to reach a result that is felt to be adequate.

(2) INDEX WORDS. (LEMMAS WITH FREQUENCY >= 2.)

acre	hair	orchard
berr	I	past
brown	includ	plant
captivat	North_Island	produc
chance	Jim_Macloughlin	seed
Chinese	kiwi	ship
commercial	kiwifruit	sold
countr	late	success
develop	lemon	tast
ear	like	Te_Puke
egg	market	thousand
favor	me	vine
five	millionaire	white
flesh	most	wild
fruit	new	world
green	New_Zealand	year

Next, a span length has to be settled. This does not, however, seem to be connected to recall and precision in the same way as the frequency limits. Rather, there seems to be an optimal span length. Increasing or decreasing the span length in relation to the optimal length will increase/decrease recall in the way that would be expected, while both increase and decrease of span length, interestingly enough, seem to reduce precision. An increase in span length will give more of accidental and thereby uninteresting collocations and also a higher relative frequency of such uninteresting collocations among all collocations above the critical threshold that is to be set in step 9 of the algorithm. At the same time, increased span length seems to give surprisingly few new 'hits', while the old hits run a risk of being outnumbered by the new accidental collocations. A decrease in span length will remove many wanted collocations, while the relative proportion of hits among the remaining collocations will not increase. Any variation of the span length thus seems to give a reduced proportion of semantically significant collocations. This is, however, only subjective impressions from small-scale tests with varying span length. Similar results have been reached by others (Sinclair et al. 1970, referred in Phillips 1985), and have led to establishing a span length of four orthographic words as optimal.

This is a point that would deserve a more thorough investigation. It probably has something to do with the normal size of common constructions: modifier and noun will almost always appear within less than four words distance, as will mostly subject—verb and verb—object, while e. g. more peripheral adverbials will not occur that close to the nexus part of the sentence.

In the sample application, the span length is settled to the optimal 4. The original text, including form words, is searched for occurences of the (truncated) stems of the index words. Whenever a word containing such a stem is found, a span of four words is scanned for more occurences of (stems of) index words. If

any are found, the resulting pairs are stored and the search goes on. In (3) a clause from the text is given with all index words capitalized.

(3) THOUSANDs OF ACREs ARE NEWLY PLANTED EACH YEAR IN A DOZEN OR MORE COUNTRIES, ...

Here, *thousand* collocates with *acre* and *new*, but not with *plant*. *Acre* collocates with *new* and *plant*, *new* with *plant* and *year*, and *plant* with *year*. *Countr* has no collocations in this instance. The internal order of the collocational pairs is of no importance, so the stems within each pair are stored in alphabetical order. The pairs are then sorted alphabetically and the frequency of each collocational pair is calculated.

The next step is again to decide on a lowest frequency, this time of collocational pairs. This decision governs which pairs, and consequently which lemmas, are to be regarded as representative of the content of the text. As this has such great impact on the output, it may well be that it should be possible to vary the frequency threshold for collocational pairs interactively, in order to facilitate closer inspection of interesting findings. A way of making expansions of the sets of lemmas and relations will be described below.

In (4), all collocational pairs with a frequency equal to or above 2 in the sample text are given in alphabetical order.

(4) COLLOCATIONAL PAIRS WITH FREQUENCIES.

berr — kiwifruit	2
commercial — kiwifruit	2
develop — kiwifruit	2
flesh — green	2
I — kiwifruit	3
I — New_Zealand	2

The idea is that this can give a more or less accurate picture of concepts and relations that are central to the text, at least in the sense that they show a high frequency. Mostly, this is sufficient to provide a hint about what the text is about. In some cases there may however be a need for enlarging the basis of the representation. This can be done by setting a lower minimal frequency level for lemmas or collocational pairs or both, but this means redoing parts of the processing. A better way can be to use a set of expansion operations as defined below.

Without changing the given frequency levels for lemmas and collocational pairs, we can derive the following sets of concepts and relations between concepts:

(5) EXPANSIONS

Primary concepts: the lemmas occuring in the pairs originally picked out by the algorithm.

First expansion: all collocations between primary concepts.

Second expansion: all other lemmas collocating with primary concepts. This gives the set of secondary concepts.

Third expansion: all collocations between secondary concepts.

Fourth expansion: all lemmas collocating with secondary concepts.

Etc.

The second and fourth (generally: all even) expansions are 'opening' expansions, as they bring in new concepts. The first and third (and all odd) expansions are 'closing', as they establish relations between existing concepts and make the corresponding graph more closely knit.

In (6) below, we see for each of the primary concepts the collocations it enters: a) with other primary concepts and with a frequency above the minimal level; b) with other primary concepts but with a frequency below the minimal level (first expansion); c) all collocations between primary concepts and other lemmas from the set of index words (second expansion). To construct all interrelations between all those items would in its turn give the third expansion.

From (6) we can also see that another characteristic of the collocations is their ability to delimit the interpretation of polysemous words. The pairs that a word can enter will often signal the specific meaning in which the word is used in a particular text. This is not so striking in this text as in some others, but looking at e g *green* we will see that we have to do with the green of fruit, not that of green paint, and *commercial* does not directly refer to e. g. banking, but to commercial aspects of growing fruit.

(6) COLLOCATIONAL PAIRS WITH FREQUENCIES: A) PRIMARY CONCEPTS, B) FIRST EXPANSION, C) SECOND EXPANSION

Primary concepts			Possible expansions		
kiwifruit:					
a)	3	kiwifruit I	b)	1	kiwifruit New_Zealand
	2	kiwifruit berr	c)	1	kiwifruit captivat
	2	kiwifruit commercial		1	kiwifruit chance
	2	kiwifruit develop		1	kiwifruit includ
				1	kiwifruit Jim_Macloughlin
				1	kiwifruit millionaire
				1	kiwifruit adjacency
				1	kiwifruit plant
				1	kiwifruit sold
				1	kiwifruit tast
				1	kiwifruit vine
				1	kiwifruit year
berr(y):					
a)	2	berr kiwifruit	c)	1	berr brown
				1	berr like
				1	berr tast
				1	berr wild
commercial:					
a)	2	commercial kiwifruit	b)	1	commercial New_Zealand
			c)	1	commercial orchard
develop:					
a)	2	develop kiwifruit	b)	1	develop New_Zealand
			c)	1	develop taste
				1	develop vine
flesh:					
a)	2	flesh green	b)	1	flesh I
				1	flesh kiwifruit
			c)	1	flesh tast
green:					
a)	2	green flesh	b)	1	green I
				1	green kiwifruit
			c)	1	green fruit
				1	green seed
I:					
a)	3	I kiwifruit	c)	1	I I
	2	I New_Zealand		1	I vine
New_Zealand:					
a)	2	New_Zealand I	c)	1	New_Zealand captivat
				1	New_Zealand lemon
				1	New_Zealand North_Island
				1	New_Zealand produc
				1	New_Zealand tast

Both (4) and (6) can, as said before, give hints about the content of texts if the lists are interpreted by a normally inventive human being. A graphic representation of the same facts seems, however, to be more striking and to facilitate

inference making. To proceed to this, a set of adjacency lists is constructed on the basis of (4). The adjacency lists form the input to the graph-drawing program, where each lemma will correspond to a node in the graph. In an adjacency list, each lemma, i. e. each node, is given a list of all its immediately adjacent nodes. This way, each collocational pair will be represented twice, corresponding to the two possible directions of the arc between the nodes. The graphs resulting from this system are however undirected. It would be possible to have weighted arcs in the graph, corresponding to the frequencies of collocational pairs, but this has not been implemented in the present system. The adjacency lists derived from (4) are shown in (7).

(7) ADJACENCY LISTS

kiwifruit(I, berr, commercial, develop)
berr(kiwifruit)
commercial(kiwifruit)
develop(kiwifruit)
flesh(green)
green(flesh)
I(kiwifruit, New_Zealand)
New_Zealand(I)

5 Graph-Drawing

The last step in the algorithm is the drawing of a graph. Automatic drawing of graphs by means of a computer is a demanding task, especially if the work, as in the present case, is to be done on a PC. We have, however, been able to find a satisfactory solution.

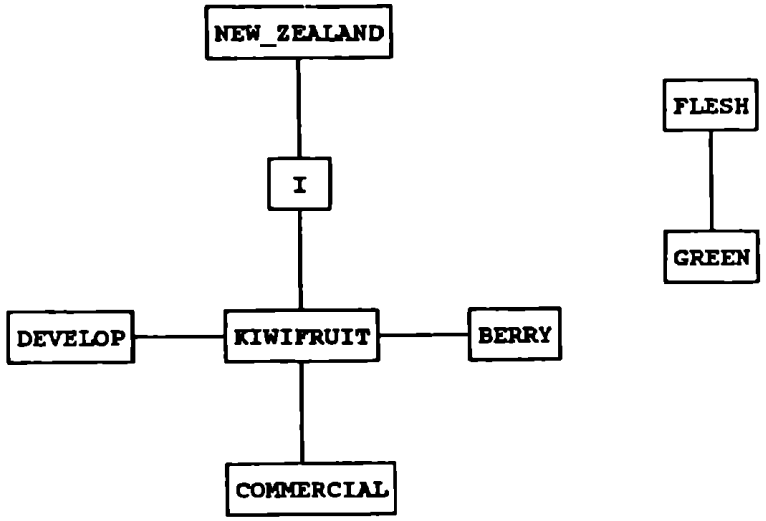
The program consists of two main parts. The first one finds the areas and subareas that together build up the graph. It tries to avoid crossing arcs, but if that is not possible, the program finds the best places to add 'pseudo-nodes', i. e. crossings. Its output is a 'road description' of the graph. The second part of the program performs the computationally heavy task of actually drawing the graph, laying it out nicely on the screen or in a file that can be stored or printed out on paper.

The first part of the program was originally written by Boris Prochaska as a part of his examination at the Royal Institute of Technology in Stockholm (Prochaska 1988), and the second part was written by Sten-Erik Bergner, who was Boris' supervisor during his examination job at Ericsson Telecom. Their version of the program is written in PSL-Lisp and had to be rewritten in GC-Lisp, a subset of Common Lisp, for use on PCs. This non-trivial job has been undertaken by Sune Magnberg, whose programming skills, earlier knowledge of graph theory, and general combination of inventiveness and patience, made the job of transporting the 'portable' Lisp possible.

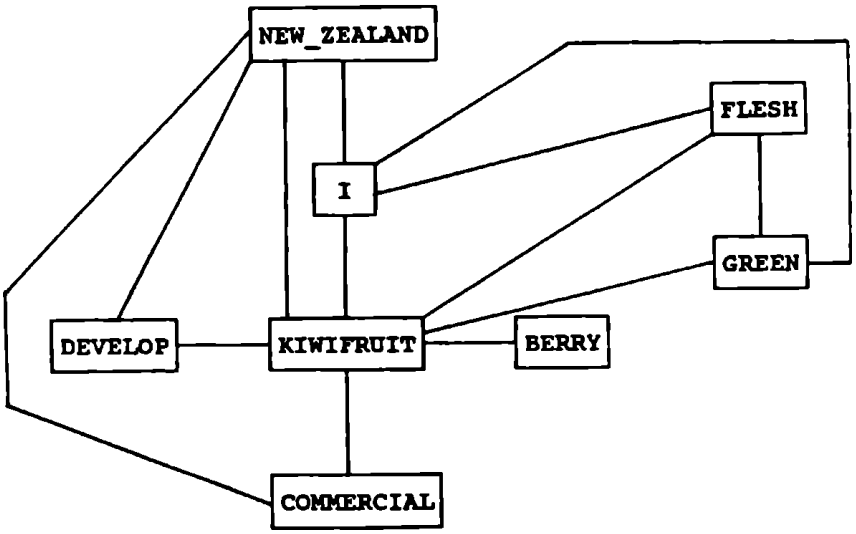
The result of all this is a graphic representation of the lemmas and relations that have a high frequency in a given text and, for that reason, can be assumed

to have strong connections to what the text is about. The graph that, by means of the described system, can be automatically derived from the text in Appendix A is shown in (8), while (9) is the first expansion of that graph (cf. (5)).

8



9



6 Concluding Remarks

These graphs support in principle the same inferences as did the lists of pairs, but in a neater way. The kind of relations that are signalled by the arcs varies considerably and is left to the human user to guess—at the risk of making mistakes. A very natural question to ask is whether all this apparatus gives anything more than would a simple list of high-frequency words. My impression is that it does. Below is a list of the 9 most frequent lemmatized content words in the text, all lemmas with a frequency of 4 or higher. The list should be compared to the words of the adjacency lists, (7), and to the full text in Appendix A.

(10) CONTENT WORDS FROM KIWI-TEXT, LEMMATIZED AND SORTED ACCORDING TO FREQUENCY

13	kiwifruit
10	New_Zealand
5	berr
5	ship
4	Chinese
4	vine
4	I
4	lemon
4	market

Kiwifruit, *New_Zealand*, *berry/ies*, and *I* are also represented among the eight lemmas picked out by the graph-constructing algorithm and the graph clearly shows their centrality. The graph also shows *commercial* and *development* as highly central, while the descriptions *green* and *flesh* are shown to be somewhat less central. The pure frequency statistics, however, has it that *ship/ping*, *Chinese*, and *lemon* are quite as important as *market* and *vine*. But the article (in Appendix A) is certainly not about the shipping of Chinese lemons, it is a subjectively colored boasting about the commercial success of kiwifruit and all that this has meant to New Zealand, interspersed with lyric bursts about the look and taste of the berry. There is no doubt that this is more clearly signalled by the graph than by the frequency list, although both representations need a good deal of human inference making to be added.

The results have not yet been independently evaluated, but the method has been applied to several Swedish texts. Three short Swedish texts are shown in Appendix B and their corresponding graphs in Appendix C. One very interesting finding is that the method seems to be utterly impossible on literary texts, but okey on others. Why this is so is something that has to be investigated more closely. It must also be investigated for which text types the method is best suited and under what circumstances it runs a risk of being seriously misleading.

Another step would be to try the method under realistic circumstances in connection with information retrieval. The idea is something like this: The user sits at a terminal and types in a search question, either in natural language, in which case it has to be parsed, or as a set of key words with or without Boolean operators. The key words are then matched against graphs that have

been previously derived from the texts in the data base to be searched. If the search question was in natural language, the presence of interrelations between key words can also be checked. A measure for when a graph is 'satisfactorily similar' to the information derived from the search question must be defined. Next, one selected graph at a time will be shown on the screen and the user can choose if she/he wants to have the full text. In doubtful cases it may be possible to get one or more of the expansions in order to get a broader basis for decisions. The search can also be carried out in such a way that graphs that are judged as relevant can be used for deriving new, conjoined graphs.

If these ideas can be developed to work well, the practical usefulness of the content graphs is clear, but among the most thrilling questions are why the method works when it works, and why it doesn't work when it doesn't. This is as yet far from clear.

References

Black, E., 1988. Grammar Development for Speech Recognition. Proc. from ELS Conference on Computational Linguistics, IBM Norway.

Church, K. W., 1988. A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. ACL Second Conference on Applied Natural Language Processing, Austin, Texas.

DeRose, S. J., 1988. Grammatical Category Disambiguation by Statistical Optimization. *Computational Linguistics*, 14(1):31–39.

Fjeldvig, T. & Golden, A., 1984. Automatisk rotlemmatisering — et lingvistisk hjelpemiddel for tekstsöking. CompLex no. 9/84. Universitetsforlaget, Oslo.

Garside, R. & Leech, F., 1987. The UCREL probabilistic parsing system. In Garside, R. et al. *The Computational Analysis of English.* Longman, London.

Källgren, G., 1984a. HP-systemet som genväg vid syntaktisk märkning av texter. In *Svenskans beskrivning 14*, p. 39–45. Lunds universitet.

Källgren, G., 1984b. HP — A Heuristic Finite State Parser Based on Morphology. In Sågvall-Hein, Anna (ed.) *De nordiska datalingvistikdagarna 1983*, p. 155–162. Uppsala universitet.

Källgren, G. 1984c. Automatisk excerpering av substantiv ur löpande text. Ett möjligt hjälpmedel vid automatisk indexering? IRI-rapport 1984:1. Institutet för Rättsinformatik, Stockholms universitet.

Källgren, G. 1985. A Pattern Matching Parser. In Togeby, Ole (ed.) *Papers from the Eighth Scandinavian Conference of Linguistics.* Copenhagen University.

Källgren, G., 1987. What Good is Syntactic Information in the Lexicon of a Syntactic Parser? In *Nordiske Datalingvistikdage 1987*, Lambda, 7. Copenhagen, Handelshøjskolen.

Källgren, G. 1988. Automatic Abstracting of Content in Text, *Nordic Journal of Linguistics*, Vol. 11(1-2): 89–110.

Phillips, M., 1985. *Aspects of Text Structure.* North-Holland, Amsterdam.

Prochaska, B., 1988. Automatisk uppritning av grafer. Examination paper, Royal Institute of Technology, Stockholm.

Salton, G. & McGill, M. J., 1983. *Introduction to Modern Information Retrieval.* McGraw-Hill, New York.

Sinclair, J. McH. et al., 1970. English Lexical Studies: Report to OSTI on Project C/LP/08, Dept of English, University of Birmingham.

A The Captivating Kiwifruit

Thirty years ago, growing up in New Zealand, I often sliced into a brown berry that looked like a duck's egg in a bristly hair skirt. Repulsive? Not really, for I knew a secret: The berry's odd appearance disguised an equally exotic interior, a sunburst of neat white streaks radiating from a creamcolored core, past tiny black seeds and into shimmering green flesh (above). Sweet-tart in taste, it seemed a succulent blend of strawberry, banana, melon, and pineapple flavors. Delicious! I loved the kiwifruit.

I still do, and today this peculiar product of a woody vine is captivating palates outside New Zealand at an extraordinary pace. In 1986 more than a billion kiwifruit, once called Chinese gooseberries, were tucked into trays and shipped to at least 30 nations. Thousands of acres are newly planted each year in a dozen or more countries, including the United States, France, Japan, and Italy, the leading producers after New Zealand.

This universal success has uniquely New Zealand roots. The kiwifruit's conversion to a commercial crop occurred in New Zealand, and its name—coined in the 1950s as a marketing tactic—conjures up both that likable country and its whimsical, flightless native bird, renowned for oversize eggs and hairlike brown feathers. Moreover, exports of the fuzzy, four-ounce berry are increasingly important to New Zealand's economy and the creator of more millionaires than anything else in my homeland's history.

The only fruit with such bright green flesh, the kiwifruit is one of just a handful of food plants domesticated within the past thousand years. Originating in the Yangtze Valley, it has long been a favorite of the Chinese, glorified in poetry as early as the eight century. Chinese peasants still gather the wild fruit for sale in rural markets.

The transformation of a small, hard, and wild Chinese berry into fleshier, tastier kiwifruit began about 1904, when a traveler returned from a China visit with seeds for Alexander Allison, a nurseryman on New Zealand's North Island. In the following three decades he and other gardeners developed superior kiwifruit vines through careful selection, pruning, and grafting. Most of these early fanciers were as much interested in the vine's showy white blossoms and attractive fan-shaped leaves as in its berries.

Kiwifruit farming got its commercial start in the 1930s, most successfully at Te Puke on the North Island's east coast. The late James MacLoughlin became the father of the modern kiwifruit—and ultimately a millionaire—by chance.

After he lost his job as a shipping clerk during the Great Depression, Jim's wife's aunt invited them to stay on her lemon orchard at Te Puke. "Later the bottom fell out of the lemon market," he told me, "but a neighbor sold the kiwifruit from a single plant for five pounds (then worth about $20 U.S.). To me that was a lot of money, so I risked putting in half an acre of them."

Luckily for MacLoughlin, the warm, wet climate and volcanic soil at Te Puke favored his vines. Neighbors soon launched their own commercial orchards, which further expanded during World War II when GIs stationed in New Zealand developed a taste for kiwifruit.

Then chance intervened again. In 1952 an English fruit importer ordered a shipment of New Zealand lemons. "To fill spare space in the ship, we included ten cases of

kiwifruit," Jim MacLoughlin explained. "A dock strike delayed the ship five weeks and the lemons arrived rotten, but the kiwifruit were in perfect shape." They sold well, and New Zealanders suddenly realized that they'd opened a world market.

B Swedish Texts

Text 1

Rudi och Renate hyr en liten stuga ovanför sjön, fast de har visst aldrig råd att betala den. Där finns ett rum och kök.

När Malin och jag kommer dit, sätter vi oss på golvet och jag tar av mig skorna också, jag vill vara som hon. Rudi spelar Mozart på en grammofon, som han lånat hem "på prov". Det är alldeles för dyrt att köpa egna grammofoner.

Solen skiner rakt in i köket. Rudi visar sina bilder, Malin röker pipa och ler så gott när hon ser tavlorna och Rudi pratar så mycket att jag slipper.

(Göran Tunström: Prästungen)

Text 2

Metropolen Oslo får en ny profil

Inte på 100 år har så många och omfattande byggprojekt påbörjats i Oslo. När de är klara kommer den norska huvudstaden att få en ny profil och nya möjligheter. Under tiden lider Osloborna.

Nordens högsta hotell, en kongresshall med plats för 10 000 åskådare och Europas längsta gågata under tak är några av de projekt som redan är i full gång.

Det har skett en snabb utveckling de senaste åren. Oslo blir alltmer en metropol. Vad gäller nattliv och restauranger kan staden konkurrera med både Stockholm och Köpenhamn. Den sista sammanräkningen visade 90 nattklubbar och kafeer som höll öppet mellan två och fyra på natten.

Aker Brygge med sin kombination av butiker, restauranger, teater och kontor i läckra omgivningar vid hamnen har blivit något som Osloborna stolt visar upp för tillresande. En bärande tanke har varit att öppna staden mot fjorden igen. Biltrafiken ska läggas så mycket som möjligt i tunnlar.

(DN 1987-12-05)

Text 31

Om batterier och batteribyte

Låt inte ett kvartsur som stannat bli liggande. Batteriet kan börja läcka och skada din klocka.

Vågar man då byta batteri själv?

Några få klockor har ett särskilt batterifack med lock, se bruksanvisningen. Då är det möjligt att själv byta batteri, men eftersom det kan vara svårt att få locket tättslutande igen efter bytet, är det klokt att ändå anlita fackmannen.

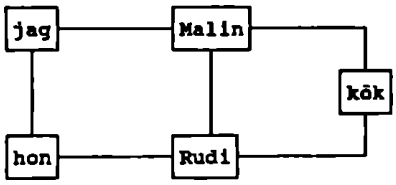
Det är också viktigt att du får rätt sorts batteri och inte ett som är avsett för fotoartiklar eller hörapparater. Då gäller inte garantin som de flesta tillverkare av urbatterier ger.

På de flesta klockor måste boetten öppnas vid batteribyte. Då fordras specialverktyg och stor försiktighet för att inte elektroniken ska ta skada. Och det är viktigt att boetten sluter ordentligt tätt efter batteribytet.

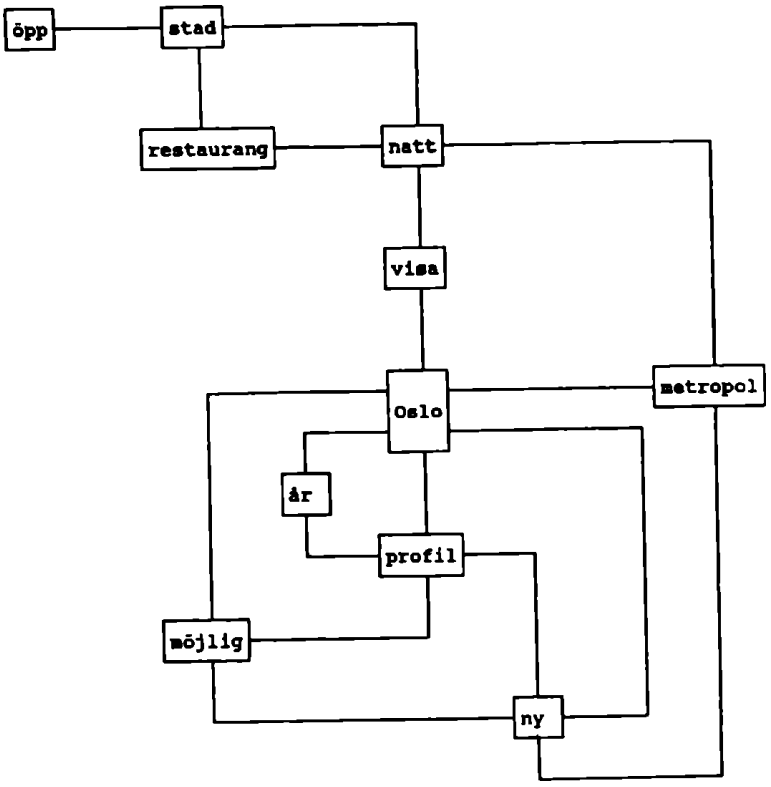
Det är ett arbete du ska överlåta till en fackman.

C Conceptual Graphs of the Swedish Texts

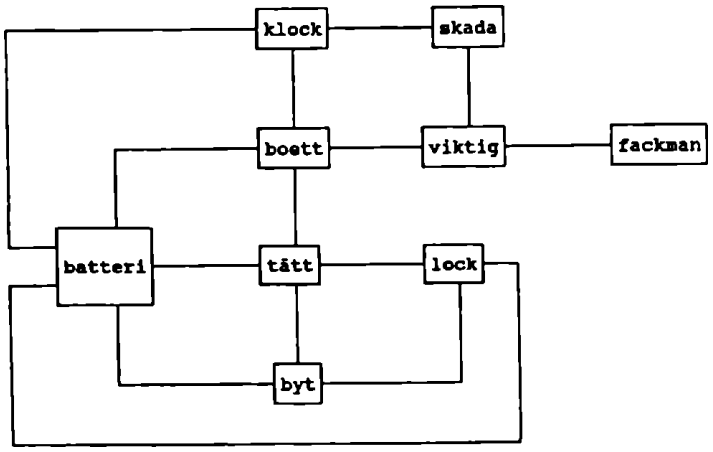
Text 1



Text2



Text3



Department of Linguistics
Stockholm University
S-106 91 Stockholm
Sweden

GREGERS KOCH

Computational Man-Machine Interaction in Simple Natural Language

Abstract

For a wide variety of semantic theories we shall present a common method of calculating the semantic representation when starting from the input text and a grammar covering the syntactic description of the text. It appears that the so-called data-flow trees play a huge and central role in this kind of analysis and translation into a semantic representation. The method here seems particularly well fit for the analysis of natural language queries to database systems. The considerations here are rather tentative and reflect research in progress.

Introduction

This paper investigates methods and tools for developing a specific kind of model of human language learning capability, by presenting a performative simulation model (here termed a computational logico-semantic induction system [16, 18]).

The same methods and tools may be applied for the purpose of implementing a wide variety of computational systems including certain kinds of rule-based expert systems and certain kinds of modern grammars (in particular the so-called unification grammars) [17].

The advantage of logico-semantic induction is its applicability in the context of constructing natural language interfaces as well as a variety of other user-friendly types of interfaces to expert systems and other computer systems.

We are studying the problem of constructing language acquisition models from specific data. That is, we could be claimed to be modelling an extremely advanced type of information processing systems, viz. human beings in the role of acquiring language capabilities. However, we are modelling the performative aspects only. No claim whatsoever is made as to the possible descriptive power of the resulting models from a psychological point of view (so we might call it purely antropomorphic information technology).

The focus of this paper is on logico-semantic induction which is a method for the systematic pattern identification and extraction in linguistic data sequences,

in particular at a semantic and a combined syntactic and logical level of interpretation. It provides a means for the automated analysis of verbal protocols, and it constitutes a method for the automated construction of a logico-semantic parser.

Logico-Semantic Induction and its automated variant Computational Logico-Semantic Induction designate a completely new method from the area of logic programming and natural language processing. In contrast to the majority of other inductive approaches the method here does not deal with induction in a space of possible assertions but instead with induction in a space of possible logico-semantic representations. Here is given a short introduction to the concepts. A more comprehensive discussion by this author may be found elsewhere.

The particular kind of inductive inference that we have in mind may be illustrated by means of a diagram. Along the first axis we shall map all the possible assertions or utterances (in some suitable encoding), and along the second axis we shall map all possible representations within the framework of a particular representational notation (and similarly in a suitable encoding). A semantic theory will then occur in the shape of a mapping from the axis of utterances into the axis of representations (as long as we presuppose unambiguity, otherwise it will be generalised to a relation).

For example, we might from the following facts

crow number 1 is black
crow number 2 is black
crow number 3 is black
etc.

make the attempt to induce the following more general assertion: [2]

all crows are black.

The type of induction advocated here is of a different kind: From the following conventions

text E1 has the logico-semantic representation F1
text E2 has the logico-semantic representation F2
text E3 has the logico-semantic representation F3
etc.

we should like to find a (possibly very limited) linguistic universe L and a (logical) program P such that for each text e in L its corresponding logico-semantic representation f is the result (output) of executing the program P with the given e as input. Here the example texts E1, E2, E3 etc. are all included in the linguistic universe L.

Computational Logico-Semantic Induction may be considered a generalisation of the old concept grammatical inference that may be characterised as a kind of computational syntactic induction [11].

The possibility of automation is discussed in considerable detail. The implementation of computational semantic induction has to do with the construction

of a kind of blackbox to accept a traditional syntactic description of a linguistic universe. Besides the blackbox must accept as input a finite set of pairs $<e_k,f_k>$ where e_k is a text from the linguistic universe, and f_k is the intended semantic representation corresponding to the input e_k. For instance, the f_k may be in the form of a logical formula or a logical code. Output from the blackbox should be a program that translates linguistic input e into logical output f where especially the input e_k gives the output f_k. Here is required a complete match with the given examples.

Some possible principles for such a blackbox are discussed. These principles are clarified by application to a few small sample texts. We conclude that this new concept of computational logico-semantic induction is extraordinarily promising.

This paper contains a brief discussion and sketches a solution. A more comprehensive discussion is in preparation [13, 14, 16].

Here we are concerned exclusively with parsing or textual analysis. Analogous considerations can be made concerning textual synthesis or generation.

This work on computational logico-semantic induction was performed under heavy influence by some of the leading approaches within logic grammars like those of A. Colmerauer [3, 4], V. Dahl [7, 8, 9], F. Pereira [23, 24, 25], P. Saint-Dizier [27, 28], and M. McCord [21, 22].

It may really be seen as an attempt to unify some rather diverging tendencies in the philosophy of language, namely Creswell's lambda-calculatoric theory [5, 6] and some montagovian ones [19, 10], and on the other hand, the first order logical theories from logic grammars [12, 16]. The contribution here seems to support any of these theories.

As an example we may investigate the following English sentence

(1) Mary believes that Peter loved a woman

Within the limits of a modestly extended first order predicate calculus we may assign to the sentence the following two interpretations or logico-semantic representations, respectively:

(2) ∃y[woman(y) & believe(pres,mary,love(past,peter,y))]

(3) believe(pres,mary,∃y[woman(y) & love(past,peter,y)])

An absolutely central problem of semantics (here called the logico-semantic problem) is to assign to each input text from the appropriate linguistic universe one or several formalized semantic representations. As formalizations we will here consider only logical formulae belonging to some particular logical calculus (like definite clauses or Horn clauses, first order predicate logic, some extended first order predicate logics, the lambda calculi, and Montague's intensional logic [19]).

The principles of implementation are quite clear and fairly well developed, as may be seen by studying the example below (another example may be found in [16]). But as far as an actual implementation is concerned, we are working on it albeit in a rather slow pace (due to lack of resources).

A Small Example

Now time is probably ripe to investigate the example mentioned above. This may be seen as a further development of the ideas discussed in [16]. If the syntactic description is the following little grammar

(4) Sent → Np Vp
Np → Det Noun | Prop
Vp → Tv Np | Vp-s that S

(in the last production rule we have used a categorial grammar notation) then we may look for a representative, also called an exhaustive text. Such an exhaustive sample text may be the following:

Mary believes that Peter loved a woman

Within the chosen semantic representational notation (a predicate calculus of arbitrary high order, PC_ω) we may prefer to use a kind of generalised quantifiers for representing some (two) possible interpretations of the sample text in the following way:

(5) a(y,woman(y),believe(pres,mary,love(past,peter,y)))

(6) believe(pres,mary,a(y,woman(y),love(past,peter,y)))

The two interpretations deviate by one having as a presupposition the existence of such a female and the other not having that presupposition. Montague grammars like PTQ would obtain the same distinction.

If we choose to consider the first formula (5) to be the intended representation, the method here will lead in a mechanical fashion to the logic program shown below (7), written in the form of a logic grammar.

The program constitutes just a syntactical description augmented with attributes or decorations as may be seen by ignoring the functional arguments (then quite simply the grammar of (4) occurs).

Let us see what happens more precisely in our method. The intended resulting formula (5) should be represented as a tree structure like that in figure 1. Then the following steps should be performed:

Step 1: Enumerate the boxes in the intended result structure. (In our example this means that the boxes will get the numbers from 1 to 7, as in figure 1).

Step 2: Construct the syntactic structure (by performing parsing or syntactic analysis).

Step 3: Create a match between the result structure and the syntactic structure. More precisely, make a connection from a numbered box in the result structure to the lexical category in the syntax structure to which the word (lexical or syncategorematic) belongs. This is an indication of the vertex in the syntax tree where that fraction of the result structure

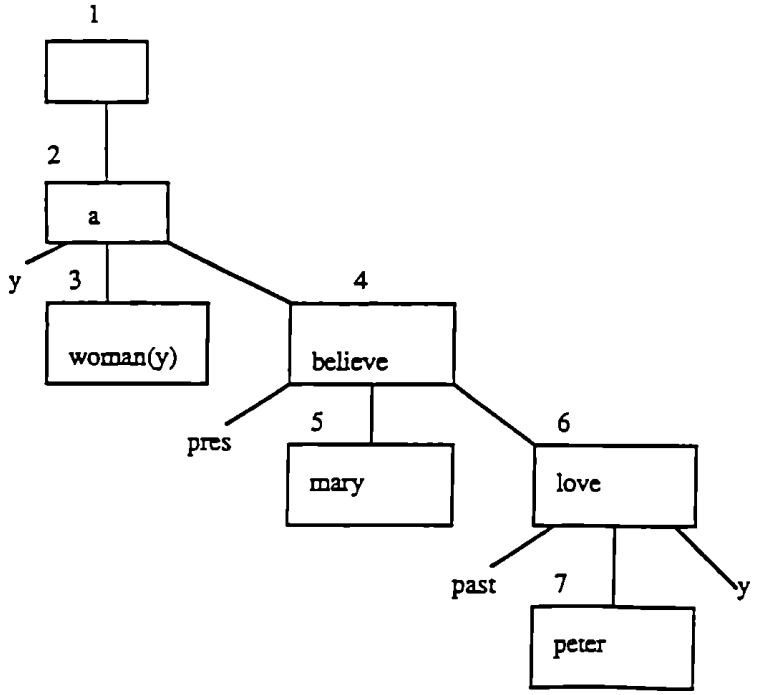


Figure 1:

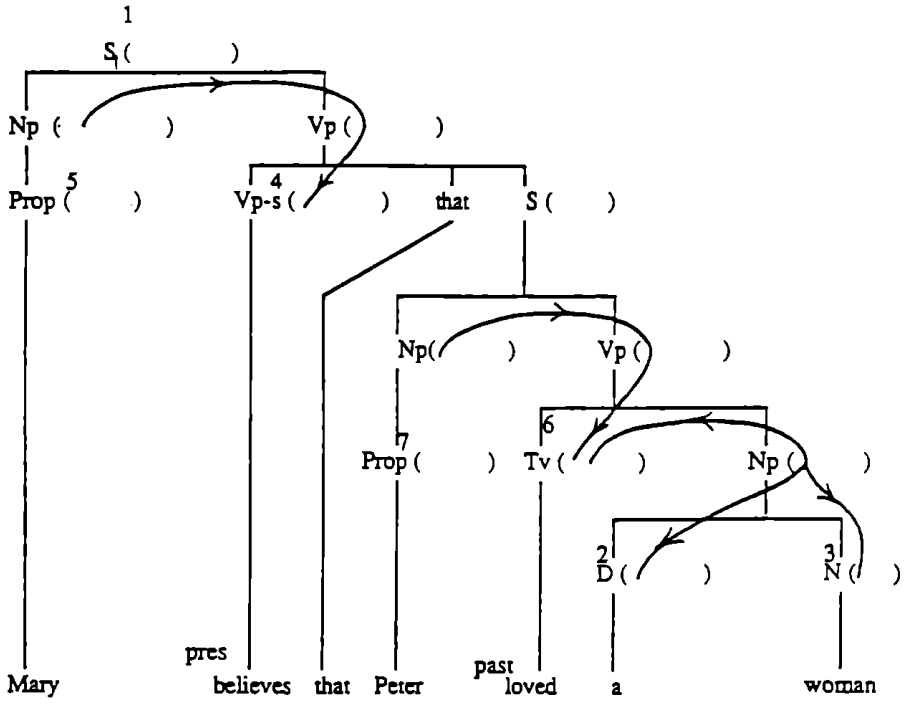


Figure 2:

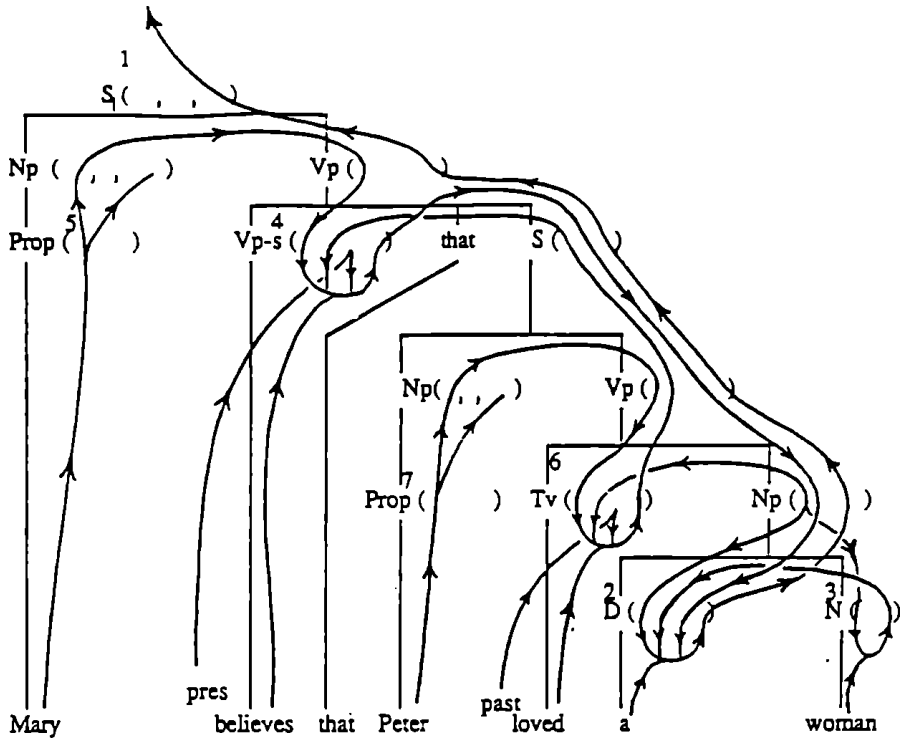


Figure 3:

having the relevant word as its top vertex, is being constructed as the result attribute of the vertex.

Step 4: Construct the flow from the so-called focus variables (form a new variable for each Np phrase, as in figure 2).

Step 5: Construct the flow in the lexical rules.

Step 6: Connect each pair of numbers corresponding to an edge in the result structure (here the tree structure should be respected, as in figure 3 where the following pairs are connected: 7–6, 5–4, 6–4, 3–2, 4–2, 2–1).

Step 7: Check the consistency concerning arity and local flow.

In our example the augmented syntactic structure will be like figure 3.

The resulting logic grammar will be the following:

(7) S(V,W,U) → Np(X,Y,Z),Vp(X,W,V,U)
Np(X,Y,Z) → Prop(X)
Np(X,Z,W) → D(X,Y,Z,W),N(X,Y)
Vp(Y,X1,Y1,V) → Vp-s(X,Y,Z,W),[that],S(W,Z,V)
Vp(Y,W,V,U) → Tv(X,Y,Z,W),Np(Z,V,U)
D(X,Y,Z,a(X,Y,Z)) → [a]
D(X,Y,Z,every(X,Y,Z)) → [every]

Concluding Remarks and Perspectives

As to which representation languages are acceptable with respect to this method, there seems to be a high degree of freedom so that we seem to be near the implementation of a general information theoretical or computer science paradigm like this:

Anyway, there exists a requirement that a kind of homomorphy property, a kind of compositionality should be available in the relationship between input and output. One or another variant of Frege's principle of compositionality should be obtained:

To the extent that our rules are of the form

$$p_0(G(y_1,...,y_n)) \rightarrow p_1(y_1),...,p_n(y_n)$$

we know about the semantic representation function Sem that

$$Sem(P_0)=G(Sem(P_1),...,Sem(P_n))$$

where $P_0=P_1 \hat{} P_2 \hat{} .. \hat{} P_n$

provided that P_k is the fragment of the input text belonging to the syntax category p_k for all $k \in \{0,1,..,n\}$.

And this property is precisely one way of expressing Fregean compositionality.

One perspective of this approach is that it allows a generalisation into what we tend to call computational logico-semantic abstraction [18]. In this context it is profitable to make use of certain results from the modern computer science disciplines of logic programming, attribute grammars, and denotational semantic theories.

Another perspective concerns automated learning. Computational logico-semantic induction has the property that the system will be able to improve its linguistic performance (i.e., handling new information of a semantic nature) by adoption from a single occurrence of a grammatical rule. That must be effective automated learning par excellence!

So, besides concluding that the method of logico-semantic induction is not only new but also promising we are able to discuss AI-problems related to inductive learning from the following perspective: inductive reasoning as a way of managing linguistic information in logical systems. Hence in this case it is not really a question of empirical information, and of course its relationship to AI is always arguable (what is the precise content of AI?), but a surprisingly high degree of automated learning is actually obtainable.

References

[1] J.W. Bresnan [ed.]. 1982. *The Mental Representation of Grammatical Relations.* MIT Press.

[2] E. Charniak & D. McDermott. 1985. *Introduction to Artificial Intelligence.* Addison-Wesley.

[3] A. Colmerauer. 1978. Metamorphosis Grammars. L. Bolc [ed.]. *Natural Language Communication with Computers.* 133–189. Lecture Notes in Computer Science No. 63.

[4] A. Colmerauer. 1982. An Interesting Subset of Natural Language. K.L. Clark & S.-Å. Tärnlund [eds.]. *Logic Programming.* Academic Press.

[5] M.J. Cresswell. 1973. *Logics and Languages.* Methuen.

[6] M.J. Cresswell. 1985. *Structured Meanings, the Semantics of Propositional Attitudes.* MIT Press.

[7] V. Dahl. 1979. *Logical Design of Deductive Natural Language Consultable Data Bases.* Proc. Fifth International Conference on Very Large Data Bases. Rio de Janeiro, Brazil.

[8] V. Dahl. 1979. *Quantification in a Three-Valued Logic for Natural Language Question-Answering Systems.* Proc. IJCAI. Tokyo, Japan.

[9] V. Dahl & M. McCord. 1983. *Treating Coordination in Logic Grammars.* Am. Journ. Comp. Ling.

[10] D.R. Dowty, R.E. Wall & S. Peters. 1981. *Introduction to Montague Semantics.* D. Reidel.

[11] J.J. Horning. 1969. *A Study of Grammatical Inference.* Technical Report No. CS 139. Computer Science Department, Stanford University.

[12] G. Koch. 1985. *Who is a Fallible Greek in Logic Grammars.* 54–77 in [15].

[13] G. Koch. 1986. *Relating Definite Clause Grammars and Montague Grammars.* Institute of Computer Science, Technical University of Denmark. 20 pages.

[14] G. Koch. 1986. *The Application of Prolog for the Translation into a Semantic Representation.* Proc. Nordic Seminar on Machine Translation. EUROTRA-DK, Copenhagen. 175–189.

[15] G. Koch [ed.]. 1985. *Fifth Generation Programming vol. 1: Logic Programming in Natural Language Analysis.* Proceedings of Workshop in Copenhagen. Dec. 1984. DIKU report 85/2.

[16] G. Koch. 1987. *Automating the Semantic Component.* Information Processing Letters 24. 299–305.

[17] G. Koch. 1987. *LFG og Prolog.* Institute for Applied and Mathematical Linguistics, Copenhagen University.

[18] G. Koch. [Forthcoming]. *A Technical Perspective on Expert Systems, Modern Grammars, Semantic Abstraction, and their Implementations.* Proc. Fifth Symposyum on Empirical Foundations of Information and Software Science, Risø, Denmark, Nov. 1987.

[19] R. Montague. 1974. *The Proper Treatment of Quantification in Ordinary English.* In [20].

[20] R. Montague. 1974. *Formal Philosophy.* Yale University Press.

[21] M. McCord. 1982. Using Slots and Modifiers in Logic Grammars for Natural Language. *Artificial Intelligence.* 18,3:327–367.

[22] M. McCord. 1987. Natural Language Processing in Prolog. A. Walker [ed.]. *Knowledge Systems and Prolog.* Addison-Wesley.

[23] F.C.N. Pereira. 1983. *Logic for Natural Language Analysis.* SRI International. Technical Note 275.

[24] F.C.N. Pereira & D.H.D. Warren. 1983. *Parsing as Deduction.* SRI Technical Report 293. Stanford Research Institute.

[25] F. Pereira & D. Warren. 1980. Definite Clause Grammars for Language Analysis — A Survey of the Formalism and a Comparison with Augmented Transition Networks. *Artificial Intelligence.* 13,3:231–278.

[26] U. Reyle & W. Frey. 1983. *A Prolog Implementation of Lexical-Functional Grammar.* Proc. IJCAI, International Joint Conference on Artificial Intelligence. 693–695.

[27] P. Saint-Dizier. 1985. *On Syntax and Semantics of Modifiers in Natural Language Sentences.* In [15].

[28] P. Saint-Dizier. 1986. An Approach to Natural Language Semantics in Logic Programming. *The Journal of Logic Programming.* 3(4):329–356.

Institute of Datalogy
University of Copenhagen
DK 2100 Ø
Copenhagen, Denmark

JORDAN ZLATEV

Criteria for Computational Models of Morphology: The Two-Level Model as an NLP Framework

Abstract

Computational models of morphology are best seen not as morphological models but rather as natural language processing frameworks which can express descriptions in the style of one morphological model or the other, and even go further, but without necessarily being bound by "purely" theoretical considerations. Criteria for their adequacy can be derived by treating them (together with the linguistic descriptions that are expressed in their formalisms) as NLP systems, for which a number of goals can be stated, among which are sufficient coverage, efficiency, augmentability and flexibility. The two-level model (TWOL) of Kimmo Koskenniemi is the main object of attention in this article and examples of its applicability to Bulgarian morphology are presented.

1 Introduction

The criteria for what a "morphological model" should be able to account for, and the manner in which this should be done, have risen high during the past few years in accordance with the situation in the neighbouring linguistic "levels" of syntax and semantics. Apart from the traditional requirements for *linguistic felicity* ("capturing the generalizations"), *rigour*, and *simplicity*, opinions are being expressed that a morphological model should be *general*, (understood as universal), *explanatory* and even *psychologically real*. Now far from doubting the plausibility of these requirements, I feel that they tend to place the models of human language provided by the field of computational linguistics in a rather unfavourable light. This is especially relevant for computational morphology, which only during this decade seems to have "stepped out from the cradle", as for example Lars Borin (p.c.) has implied. And instead of being blindly critical and sceptical towards its potentials, (which "linguists proper" often tend to be

towards computational linguistics in general, as a form of self-defence), isn't it best to watch its first steps carefully, with a helpful hand where it can be lent?

If in the previous paragraph I have suggested the picture of computational morphology, and more concretely of it's best known representative, the *two-level model*, (first presented in (Koskenniemi 1983) and most often abbreviated *TWOL*), as a clumsy, stumbling baby, then I have gone too far in my manner of expression. Nothing can be further from the truth considering the enormous amount of attention and subsequent work that Koskenniemi's dissertation unleashed. Hardly a conference can go by—including this one—without a few contributions pointing out TWOL's achievements—or deficiencies, and in the best case offering improvements or alternatives, e.g. (Sproat and Brunson 1987, Bear 1988, Kataja and Koskenniemi 1988, Calder 1989). But even these can without doubt fall at the hand of the theoretical linguist who will not fail to see the inadequacy of Bear's reintroducing the notion of "negative rule features", or of Calder's "string equations". As the last author himself carefully states: "... one may justifiably have reservations about introducing string equations into linguistic descriptions." (Calder 1989:62).

In this paper I wish to propose what I think is a more "constructive" view of the aims of computational morphology, which is also more or less applicable to the field of (computational) natural language processing in general. I will argue that there are a number of properties, which can help us compare, evaluate and develop models in a more short-term perspective so that one need not necessarily be overwhelmed by the "theoretical argument" from the beginning. In my opinion a computational model that finds the best combination of these properties, has also the best chances of being theoretically significant as well, though this is a somewhat controversial matter. A viewpoint that is at least **less** controvercial is that the goals of computational and theoretical linguistics differ. Shieber (1987), for example, has claimed that these differences, especially concerning "restrictiveness", are so essential that from a computational perspective one is more interested in **what** the linguistic theories say than **how** they say it and that it is meaningful to try to separate theories ("how") from their analyses ("what") and concentrate on the latter in computational models.

I would like to continue on this line of thought with one substantial difference: while Shieber discusses models in their property of being "computer tools for linguistics", I regard them as potential candidates for becoming language theories on their own. This difference is illustrated in the choice of model to exemplify the issues under discussion: in Shieber's case this is the formalism of PATR-II, while I will use the two-level model. I will be presupposing at least **some** previous knowledge of it.

2 Computational Models as NLP Frameworks

I believe that one could say that the aims of computational and theoretical linguistics eventually converge, namely to gain a better understanding of the nature of human language and of its user. Still they differ in their methods.

Computational linguistics (partly because of utilitarian reasons) is much more inclined to use the trial-and-error approach, starting with a fragment and then augmenting it; taking some categories for granted, (phonemes, for example) as "working hypotheses", if they facilitate the overall work of the system. This is so because the short-term goal of computational linguistics is the construction of a *natural language processing system*, no matter if it does or does not model human language processing at a sufficiently theoretical level. On the other hand it is theoretical linguistics that should stand for the "conceptual insights", the new ideas and the quest for linguistic universals. Of course, the closer the connection between computational and theoretical linguistics, the better, but at least to begin with, this is **not** a necessity.

What I'm aiming at is to say that computational and theoretical models should not be considered *on a par*. A computational model is both less and more than a theoretical one. Less, because it is the backbone of a system and thus is subjected to the limitations I mentioned above, i.e. working hypotheses, fragments etc. More, because if it is flexible enough it could permit **several** theoretical models to be implemented (simulated) within it. So the question whether TWOL is a morphological model or not, is not all that relevant. As to whether it is "general" and in what sense, I will come to that later. Right now an important (in my opinion) question arises, namely:

If at least the short term aims of computational and theoretical linguistics split, then what are to be the criteria for, let us say "evaluating", computational models (theories, formalisms—the terminology varies) for morphology in particular, and natural language in general? There is no simple answer to this question. As a half year's survey of the relevant literature, reported in (Zlatev et al. 1989) and (Sågvall-Hein et al. 1989), has managed to convince us—opinions differ. We came to believe that in order to come to more abstract things such as desiderata, requirements etc. for the models, one should start with something more concrete. The key lies in what I mentioned above was one of the first aims of computational linguistics, and definitely the first of its more practically oriented sub-branch, Natural Language Processing (NLP): the creation of an *NLP system.*

Now what kind of animal is that? This need hardly be defined for "insiders", but for someone unfamiliar with the jargon in the field, it should be enough to say that an NLP system can be regarded as a unity of (at least) the following elements: (1) an implementable formalism, (2) a processing mechanism, and (3) linguistic knowledge expressed in the formalism. (1) and (2) together make up the computational model or—using a term more neutral to the computational/theoretical dichotomy which I myself introduced—an *NLP framework.* (3) is the *language description.* The three are as I said interdependent, but to different degrees in different systems.

3 Viewing TWOL as an NLP Framework

The main advantage in viewing computational models of natural language and of morphology in particular as NLP frameworks comes from the fact that it is possible to formulate relatively clearly what goals NLP systems should aim at. Then one could continue "bottom-up" to state "criteria" on how the models should be shaped in order to correspond to these goals. Consequently these are criteria of a practical nature which are not "theoretically bound" to begin with. Most interestingly, however, they have implications which are highly compatible with linguistically motivated considerations. I will come to this in the last section.

A computational model such as TWOL may be seen as providing the framework for an NLP system. It still remains to be "filled" with the concrete linguistic knowledge. Now the first question that arises is: how much knowledge can be expressed in the framework? The first goal for an NLP system is that this knowledge is sufficient for current purposes, or alternatively formulated, that it has sufficient coverage.

3.1 Sufficient Coverage

If a description of a certain fragment of one or several languages can be made so that the system "works" as intended with respect to this fragment, then the framework can be regarded as expressive enough in relation to this fragment. Thus one may say that an NLP framework is *weakly complete* (in Shieber's terminology) if and only if it provides a system with the linguistic coverage necessary for the given purposes.

TWOL has been applied to substantial fragments of the inflectional morphology of a number of languages ranging from Finnish (1983) to Japanese (Alam 1983) and Old Church Slavonic (Lindstedt 1986). Now while this implies that the TWOL-framework is *general* in the sense that it has a **potentially** large coverage, it does not mean that TWOL is "general" in the sense that it can be applied to all of the world's languages and their morphology - inflectional and derivational (where this distinction exists), i.e. that it is a universal morphological model. It is rather a matter of degree: TWOL is "better" than most other models because it has been applied to larger fragments of single languages, e.g. "an (almost) full description (of all the forms of all inflectional types)" (Koskenniemi 1983:125) and because it has been applied to more languages. But then, what more is needed? The fact that the morphology of for example Kubachi (cf. Johannessen, this volume) yields difficulties, doesn't make TWOL a less suitable framework for the description of, let's say, Bulgarian inflection. This only means that the morphologies of the two languages are different—the opposite would be surprising. However, if one by a "general" framework means one that can provide adequate descriptions for all language types: agglutinating, isolating, inflecting, etc. then more is to be desired. This falls, in my opinion, not under the goal of coverage but of flexibility, which will be discussed further on.

Let us be more concrete. In (Zlatev 1988) I have given what I think is a complete description of Bulgarian nominal inflection in terms of the original

TWOL, i.e. as presented in (Koskenniemi 1983). Bulgarian morphology is very well developed and poses some non-trivial problems for any linguistic description, computational or not, such as extensive allomorphy and morphophonemic alternations within the stems. TWOL has proved quite satisfactory in describing both, with its finite-state lexicon and two-level rules, respectively. The demonstrative pronouns, however, dispay an "irregular" internal inflection, which in the original (Pascal) format of the lexicon gives no other opportunity for description than the following, which is far from elegant,

```
t          o-a-ov-e/P      "PRON DEM IDENT"
```

with 't' as the "invariant stem" (I have sticked to the principle: "One entry per Stem" so as to avoid masking some problematical areas through listing) and the continuation class `o-a-ov-e/P` which is the name of a mini-lexicon with the following content:

```
LEXICON o-a-ov-e/P  ozi  #    "MASC SING";
                    azi  #    "FEM SING";
                    ova  #    "NEUTR SING";
                    ezi  #    "PLUR"
```

If this had been the regular pattern for inflection in Bulgarian, then a possible computational description in the form of a system of intersecting lexicons—as those presented in (Kataja and Koskenniemi 1988) for the non-concatenative morphology of Semitic languages—would have been necessary (and probably sufficient). However, since the number of mini-lexicons of the kind shown above is *5* altogether and all other types fall neatly into the finite-state pattern, a compromise seems to be the best solution: I consider TWOL expressive enough, i.e. sufficient for current purposes, and decide to leave the description at that.

What if I decide to treat derivational morphology as well? Five classes of Bulgarian pro-forms seem to be readily describable as a derivational pattern which is something of the sort:

```
                                | INTERROGATIVE | + to = RELATIVE
                                |
INDEFINITE    = nÄ    +  |   kakyv
NEGATIVE      = ni    +  |   koga
GENERALIZING  = vsÄ   +  |   kak
                                |    :
                    A           B              C
```

That is, the interrogative pro-forms (B), act as the "base", which together with the appropriate "prefix", build respectively indefinite, negative and generalizing pro-forms (A), and with the "suffix" 'to' (which is actually the postponed definite article for nouns and adjectives of neuter gender)—relative pro-forms (C). However, if we try to express this simple pattern in a finite-state lexicon then we will also derive ungrammatical word-forms such as **nÃkakto*, **nikakyvto* etc., i.e. overgeneration. The reason is that if a finite-state mechanism allows

AB and BC, then it must also allow ABC, which in this case we want to forbid. Similar problems with the English prefix un-, are discussed in (Karttunen and Wittenburg 1983).

Now does this mean that we have found a point where TWOL is not sufficient in terms of coverage and an argument that it is inapplicable to Bulgarian as well as possibly the derivational morphology of most languages?

To some extent—yes. For practical purposes we may double the entries of type B in the lexicon, so that we have B' and then connect the mini-lexicons, to get AB and B'C (for example). But this is a kind of "solution" that would lead us back to where we started, and it is in some sense even worse than listing the different word-forms—it is absurd that we should have to go all this way only to start duplicating entries. (Here I'm not concerned with matters of efficiency—but these are of course more than relevant as well.)

There are, however, two other much better ways out. One would be to replace the finite-state lexicon component with a phrase-structure one, which furthermore can use a feature-matching (unification) mechanism which would guarantee that only the grammatical forms are generated. For example the problem I mentioned above can be resolved the following way:

```
(1)  PRO(IND)  -->  nÄ  + PRO(INT)
     PRO(NEG)  -->  ni  + PRO(INT)
     PRO(GEN)  -->  vsÄ + PRO(INT)
     PRO(REL)  -->  PRO(INT) + to
```

An alternative—without increasing the expressive power of the formalism—is to use the model's two-level rules in order to block out ungrammaticalities. In the case above one must use at least two "diacritic characters", let us say, @ and # (which must be clearly defined as bearers of morphological features and have nothing to do with phonology) and associate them with the entries of type A and C, respectively. Then a rule can be stated which would prevent their co-occurrence, (the operator /<= means "is disallowed" and what follows is the context which characterises all Bulgarian interrogatives, followed by the "relative sign"):

```
(2)  @   /<=  _ k V C (V) (C) #
```

Both (1) and (2) should have the same effect, and which one would be preferred is largely a matter of how they influence the goals to be discussed below, namely efficiency and augmentability.

3.2 Efficiency

Efficiency is something that concerns not only NLP systems for practical purposes, but theoretical ones as well, since all interesting applications of computational techniques to natural languages involve fragments that go beyond vocabularies of several hundred words and a predetermined number of sentences.

It is not hard to believe that it is just this criterion that has been the main reason for TWOL's popularity rather than its linguistic characteristics. The *restrictiveness* of the formalism gives the opportunity of extremely efficient implementations in which the lexicon has the form of a word tree and the rules—finite-state transducers. This has brought about the possibility of constructing systems with lexicons of tens of thousands of stems which can process text corpora and return analyses with morphological features at a spead of up to 100 word-forms a minute (Fred Karlsson p.c.)

I don't intend to indulge in this matter since I'm no expert. Still, the interdependency of system efficiency and other goals must be pointed out. For example, when it comes to choosing between the two solutions to the coverage problem which I discussed above, one would probably adopt the second alternative—that with the disallowing rule—if abandoning the finite-state format of the lexicon is likely to slow down implementations drastically. And this would be a reasonable move—as long as it doesn't get in the way of the next goal.

3.3 Augmentability

An NLP system is said to be augmentable if it can be improved with regard to each of its subcomponents—formalism, processing mechanism and linguistic description. Even if the first two are far from unchangeable, they are nevertheless more stable than the third, the development of which is by its nature an incremental and interactive process, which goes through loops, dead ends, partial solutions, gradual generalizations etc., until it reaches a provisionally stable level, and then again must be such that it is possible to develop it when the need arises. For this reason it is central that the formalism is *perspicuous*—a quite informal criterion, but nevertheless an important one.

I have already discussed questions pertaining to the TWOL lexicon component so I will take a few examples from the "heart" of the model—the two-level rules. Before a compiler for them existed, it was a cumbersome affair to translate into transducers even the simplest rules. The TWOL compiler (Karttunen et al. 1987) was a great advance in this respect. It gives the opportunity for a quite general morphophonemic rule to be formulated as simply as this, (`FrontV` is defined as `I,E`.):

```
(3)  "Palatalization of Velars"
     Cx:Cy <=> _(V:0) FrontV:
               where Cx in (k g x)
                     Cy in (c z s)
               matched ;
```

The reading is (for one unfamiliar with the notation): "lexical k is realized as surface c; lexical g as surface z and lexical x as surface s, if and only if they are followed by an optional lexical vowel which is realized as nothing (because of another rule), and a lexical front vowel (realized as anything on the surface level)". It will account for example for the following pairs. (The second example shows how productive the rule is!)

```
Lexical representation: r y k a  E   (hand + PL)
Surface representation: r y c    e

Lexical representation: h o t d o g  I   (hotdog + PL)
Surface representation: h o t d o z  i
```

One of the reasons why such rules are more perspicuous than for example rules of generative phonology is that they are purely *declarative* statements which need not take into consideration any requirements of ordering and the complex interactions that go with it. It is first after compilation (into finite-state transducers) that they gain their procedural interpretation.

Another factor that makes it easier for a TWOL system to be augmented is the lexicon/rules distinction itself, which is an example of the positive consequences of *modularity*. For example even if new inflectional and derivational types are eventually discovered, the lexicon can be changed, but if a rule—as the one above—is general enough, then it need not be "tinkered with" at all, but can safely apply on the new lexical information.

Now, the rule as I stated it above is actually different from that in (Zlatev 1988) in that it does not use any diacritic characters which have the function of "morpheme boundaries", "triggers", "blockers" etc. The point is that such characters work against the perspicuity of the formalism and thus against the augmentability of the system. For example in (Zlatev 1988:33) I wrote: "The operational lexicon can be augmented with new lexical stems which only have to be given the appropriate continuation classes (and if necessary to use the diacritics in the right positions)". It is just these "right positions", which would make it so hard for anyone else than myself to develop the system.

3.4 Flexibility

The goal of flexibility is close to that of augmentability discussed above, but concerns the ability of change not only for the sake of improvement, but as a value in itself.

There are different levels of flexibility. One is the hardware dimension: a system should preferably be independent of any particular type of machine. The recent "emancipation" of the TWOL compiler from the demanding environment of Lisp-machines, i.e. the existence of a compiler running on the more powerful models of Apple Macintosh (Kimmo Koskenniemi p.c.) may be considered in this respect as a step in the right direction.

Another aspect of flexibility is with regard to software: a system should not in any major degree depend on a particular programming language. The fact that modern programming languages have equivalent absolute expressive power, i.e. that they are Turing equivalent, doesn't mean that they are functionally and notationally so (cf. Shieber 1987) and it is sometimes easy to fall for the "procedural seduction" of computational linguistics that Kaplan (1987) discusses, e.g. to depend on PROLOG's backtracking mechanism or on Lisp's evaluation procedures. TWOL has been implemented in Pascal (Koskenniemi 1983), Common Lisp (Gajek et al 1983), Interlisp-D (Dalrymple et al 1987) and C (Kimmo

Koskenniemi p.c.) which is a sign that the model is more or less independent of programming environment. Both these aspects of flexibility call for treating *hard- and software independence* as a criterion in itself.

Matters of implementation, however, are not of primary interest for us here. I would actually want to "stretch" the concept of flexibility of an NLP system and interpret is as *linguistic flexibility*, or in other words: the property of allowing different styles of description. It is here that the relevance of theoretical considerations is greatest. Let us look again at TWOL in a little more detail.

As familiar, TWOL can provide morphological descriptions in the style of both of the traditional morphological models "Item and Arrangement" (IA) and "Item and Process" (IP). For example considering rule (3) and the examples in 3.3. one can describe the singular and plural form of the Bulgarian lexeme 'hotdog' in the following way (assuming the feature-value format of the lexicon presented as an option in (Dalrymple et al 1987), though without the facilitating device of "templates"):

```
(4)     hotdo [[semantics: [meaning: 'hotdog']]
               [syntax: [cat: n]
                        [continuation: G/Z]]].
        LEXICON G/Z
            g  [[semantics: [num: sg]]
                [syntax: [continuation: #]]].

            z   [syntax :[continuation: /i]].

        LEXICON /i
            i  [[semantics: [num: pl]]
                [syntax: [continuation: #]].
```

and alternatively:

```
(5)  hotdog [[semantics: [meaning: 'hotdog']]
                 [syntax: [cat: n]
                          [continuation: /I]]].

       LEXICON /I
            I  [[semantics: [num: pl]]
                [syntax: [continuation: #]].
```

While (4) makes use only of the lexicon, (5) relies on the "Palatalization of Velars"-rule as well (plus a default mechanism stating that *num* recieves the value *sg*, if unspecified).

If we have to compare the two alternatives, (5) seems to be better in almost all respects. It is not only more "elegant", it is shorter, simpler and as I argued in the previous section this type of description is more perspicuous and modular, thus a system based on it would be more easily augmented. From a linguistic perspective (4) would simply describe 'hotdog/hotdoz' as allomorphs

in complementary distribution where only the second takes plural which is obligatory while (5) would furthermore incorporate the process of palatalization in the description and thus "explain" the allomorphy.

This could possibly imply that Bulgarian morphology (and probably any morphology with morphophonemic alternations) is more readily describable in terms of IP than IA. This is equivalent to saying that it could be associated with a "typological parameter" which determines the most appropriate style of description (cf. Matthews 1974:163).

However, the fact that this is an option in TWOL, which also permits descriptions of type (4)—supposedly sufficient for purely agglutinating languages—is an obvious advantage in terms of flexibility.

So TWOL can in practice fully "model" both IA and IP. The extent to which this is so is sometimes overlooked because of the fact that all implementations that I know of have used input such as `book + s`, instead of `book + PL` during generation, i.e. neglelected the information in the lexicon. This should not be considered a disadvantage of the model itself, since the only reason why it hasn't been implemented is that up to now TWOL has not been used for word-form production in any larger application. Morphological conditions are furthermore expressible (and are expressed all the time) in the contexts of the rules through the diacritic signs and "morphophonemes"—just as in IP.

What about "Word and Paradigm" (WP)? The fact is that if TWOL would also be flexible enough to permit descriptions of type WP, this would improve the model in terms of perspicuity to a considerable extent. I mentioned in the previous section that for the sake of the latter, diacritic characters are best avoided. However, in eliminating the "morpheme boundary" (e.g. + or -), I had to specify that the front vowels the suffixes start with do not belong to the stem and I did that in usual manner—by using uppercase letters (i.e. `I` and `E`), which under the popular terminology usually go as "morphophonemes". Now as for example Nyman (1988) points out, these are not internal to the model in any theoretical aspect, but simply express a convenient way of encoding morphological information in segments (e.g. `I = i + PL` etc.) and are necessary because the two-level rules, or rather the finite-state transducers they are compiled in, operate only on segments.

Now looking at (5) above, we can furthermore see that this information is redundant, since `I` is specified as a plural suffix in the lexicon as well. Furthermore there must be an "ordinary i" plural suffix for adjectives such as "plax" (frightful), which do not undergo palatalization in plural form (plax/plaxi).

A possible way to preserve the efficiency that processing segmental representation provides, while avoiding inconsistencies and improving clarity would be to incorporate a WP element in the TWOL formalism. This could possibly look the following way:

```
(6)  hotdog {[[semantics: [meaning: 'hotdog']]
                  [syntax: [cat: n]
                           [continuation: /i]]]
                (paradigm: 'HOTDOG')}.

     LEXICON /i
          i  [[semantics: [num: pl]]
              [syntax: [continuation: #]].
```

The idea is that 'HOTDOG' can be defined (for example under the heading PARADIGMS) as a prototypical entry for a paradigm which undergoes an alternation—it need not be specified which, the information for this is contained in the rule—in **plural form**. Now the simplest way to implement this would be something like:

```
(7)  TRANSFORM(entry (continuation))
              IF (num (continuation)) = pl
```

On the other hand the set which the palatalization rule referred to could be redefined, so that it says instead:

```
(8)   FrontV = i + pl, e + pl.
```

Now if these can be compiled together, then one should be able to (though I haven't figured out a general mechanism yet) associate the value of TRANSFORM with the value of respectively `i + pl` and `e + pl`. A trivial way to do this would be to express the first after compilation as `I`, the second as `E` and TRANSFORM would then only have to be a function such as UPPERCASE. In this way the result would be the same as with "morphophonemes" but on a level that more clearly belongs to the "machinese" (cf. Nyman 1988) than the present formalism. The major gain from such a strategy would be that one would not have to worry about using uppercase letters "in the right positions", or to assign different continuation classes for the same suffix—as long as one gives the right "paradigm", which seems intuitively a much easier thing to do and, at the same time, is more linguistically motivated.

If this, or some similar mechanism, existed at least as an option in TWOL (which as yet it does not), then most probably the flexibility of the overall system would be increased in a way which would also permit faster development. This could be described as a matter of economy, which may also be considered as a goal in itself, albeit mainly for practical systems.

3.5 Economy

If by "economy" we mean that both the construction and the running of a system should not be too expensive in terms of time and work then one could say that all the criteria I have discussed in the previous sub-sections, i.e. sufficient generality, perspicuity, modularity etc. can be regarded as relevant. One may further point out that for the sake of economy is is much more practical to have a system

that analyses and generates (if both are required, of course) with the same program, than to use two different ones, and this can only be achieved if the NLP framework is *bidirectional.*

In this respect, too, TWOL "scores a point", because the finite-state transducers are actually **correspondences**, not transformations, and it is as easy to go from lexical to surface form, as in the opposite direction. One drawback in all implementations up to now, as I mentioned above, is that the lexicon has only been equipped for recognition, but this should be amendable.

3.6 Psychological Plausibility

Finally I come to the controversial matter of what it means for a model to be "psychologically plausible", and since so much has been written and said on the matter, (e.g. Linell 1979) I will try to say as little as possible. Expressed with precaution, a model may be regarded as plausible if it adheres—on some feasible level—to the psychological evidence we have of human verbal behaviour.

Obviously psychological plausibility, as a goal for an NLP system, is qualitatively different from the other goals discussed above. Firstly it concerns only systems which explicitly try to model human verbal behaviour. A great many of them—openly or not—do not aspire to do so, be it for practical or theoretical reasons ("competence models" etc.). Then it seems that such a goal would take us back to the vagueness that often characterises approaches to theoretical linguistics and that we indirectly tried to avoid by setting up all "goals" and "criteria" up to now.

This is not quite so. When I said at the beginning that I was sceptical to such maximum goals as explanatory adequacy and psychological reality in the field of computational linguistics, I tried to stress—"to begin with". The point of laying down alternative goals was rather to give an idea of what properties formalisms or models should have in order to go beyond the limitations inherent in most current computational work with natural language. When the NLP frameworks have been developed to the degree that it becomes meaningful to ask questions pertaining matters such as plausibility and explanatory adequacy—then of course one should ask them.

TWOL seems to have been a breakthrough for computational morphology in this respect as well. One can still be critical towards it—for example concerning its stress on efficiency while somewhat neglecting perspicuity, but on the whole—as the discussion of properties 3.1–3.5 indirectly demonstrated—it provides reliable ground to build on.

Probably it has also been the first computational model of morphology to make an interesting hypothesis about human processing of morphological information. The evidence that Koskenniemi (1983) discusses is based on performance errors. It concerns the fact that speakers—mainly children and aphasics—tend to make mistakes in that they produce word-forms of more productive inflectional patterns instead of less productive ones. But errors that correspond to "automatic alternations", i.e. phonologically motivated ones, are extremely rare. So if one describes only this type of alternations with two-level rules, the following

"error model" can be stated: "The performance errors in producing word forms are mostly faulty choices in the construction of the lexical level—there are hardly any errors in the application of the two-level rules" (Koskenniemi 1983:132). The simplicity and directness of the two-level rules in contrast with the intermediate stages and orderings of generative phonology rules make it possible to say that the TWOL framework actually **predicts** this error model. In this sense TWOL is more plausible than e.g. generative phonology.

4 Conclusions

In the preceding I have presented in a somewhat speculative way a number of goals for natural language processing systems. To make this less speculative I have concentrated on computational morphology and most of all on one single representative—the two-level model. The examples of Bulgarian morphology have been provided in the belief that "metatheoretical" studies in linguistics can not be carried on successfully without "anchorings" in particular languages. The choice of Bulgarian has been dictated by the fact that only there can I claim some originality.

The goals which I have brought up are of course not absolute, but I have held that they provide a good starting point for setting "standards" in computational linguistics and more specifically—in computational morphology. For each of these goals I have discussed properties of the frameworks that the systems are based on, which are necessary or simply desirable for the sake of these goals. The following table shows these once more,

GOALS of NLP SYSTEMS	PROPERTIES of NLP FRAMEWORKS
Sufficient coverage	Weak completeness
Efficiency	Restrictiveness
Augmentability	Perspicuity
	Declarativeness
	Modularity
Flexibility	Hard- and software independence
	Linguistic flexibility
Economy	Bidirectionality
Psychological plausibility	?

It is the goals that are the more stable part. Different systems can lay stress on some as opposed to others but still system-goals are less controversial and less contradictory to one another. The properties, however, can interact in complex ways and sometimes work against some other goal than the one which has called for them in the first place. So there is no exact relation between goals and properties, but rather a dynamic process giving priority to some instead of others in particular cases.

Despite that, I believe that these properties can be regarded as guidelines for work in the field of computational morphology and probably in computational

linguistics as a whole, which is even more important in the absense of clear criteria for what should be considered as psychological plausibility and explanatory adequacy in the field of theoretical linguistics and the neigbouring disciplines. The question mark in the table above should signify that. Whatever its answer, I think that one could say that it is not "linguistic felicity". What is meant by this is as controversial as anything in linguistics nowadays. **Unless**, it is equivalent to what I have called (in 3.4.) "linguistic flexibility", i.e. the possibility of expressing language descriptions according to different types of theoretical models. However, I see this as a means for achieving the aim, not as the aim itself.

The whole point of working independently of theory is only to return to theory, but with less sectarianism and greater insight. Only this way will computational linguistics contribute to the understanding of what a psychologically plausible model should be, which is a prerequisite for approaching a model/theory of natural language that is "psychologically real". In doing so, computational linguistics is bound to come closer and probably merge with the broader paradigm of cognitive science.

References

Alam, Yukio Sasaki. 1983. A Two-level Morphological Analysis of Japanese. *Texas Linguistic Forum*, 22:229–252.

Bear, John. 1988. Morphology with Two-Level Rules and Negative Rule Features. *Proceedings of the 12th International Conference on Computational Linguistics*:28–31, COLING, Budapest.

Calder, Jonathan. 1989. Paradigmatic Morphology. *Proceedings of the Forth Conference of the European Chapter of the Association for Computational Linguistics*:58–65, ACL, Manchester.

Dalrymple, Mary, Lauri Karttunen and Sami Shaio. 1987. DKIMMO: A Morphological Analyzer using Two-Level Rules. In R. Kaplan and L. Karttunen: Computational Morphology, Course Script LI283, 1987 Linguistic Institute, Stanford University, June 29–August 7, 1987.

Gajek, Oliver, Hanno T. Beck, Diane Elder and Greg Whittemore. 1983. KIMMO Lisp Implementation. *Texas Linguistic Forum*, 22:187–202.

Kaplan, Ronald. 1987. Three seductions of computational psycholinguistics. In P. Whitelock et al. (Eds.), *Linguistic Theory and Computer Applications*:149–188, Academic Press, New York.

Karttunen, Lauri, Kimmo Koskenniemi and Ronald Kaplan. 1987. TWOL: A Compiler for Two-Level Phonological Rules. In R. Kaplan and L. Karttunen: Computational Morphology, Course Script LI283, 1987 Linguistic Institute, Stanford University, June 29–August 7, 1987.

Karttunen, Lauri and Kent Wittenburg. 1983. A Two-Level Morphological Analysis of English. *Texas Linguistic Forum*, 22:217–228.

Kataja, Laura and Kimmo Koskenniemi. 1988. Finite-state description of Semitic morphology: A case study of ancient Accadian. *Proceedings of the 12th International Conference on Computational Linguistics*:313–315, COLING, Budapest.

Khan, Robert. 1983. A Two-level Morphological Analysis of Rumanian. *Texas Linguistic Forum* 22:253–270.

Koskenniemi, Kimmo. 1983. *Two-level morphology: A general computational model for word-form recognition and production.* Publication 11, Department of General Linguistics, University of Helsinki, Helsinki.

Lindstedt, Jouko. 1984. A Two-level Description of Old Church Slavonic Morphology. *Scando-Slavica*, 30:165–189.

Linell, Per. 1979. *Psychological reality in phonology: A theoretical study.* Cambridge University Press, Cambridge.

Matthews, P. H. 1974. *Morphology.* Cambridge University Press, Cambridge.

Nyman, Martti. 1988. Abstrakta fonem och modeller som førstår bara "modellska". The Linguistic Association of Finland, Helsinki.

Sågvall-Hein, Anna, Mats Dahllöf and Sofia Hörmander. 1989. Studies of Grammars, Formalisms, and Parsing. *Rapporter från Språkdata*, 25. Department of Computational Linguistics, University of Gothenburg.

Shieber, Stuart. 1987. Separating Linguistic Analyses from Linguistic Theories. In U. Reyle and C. Rohrer (Eds.)*Natural Language Processing and Linguistic Theories*, D.Reidel, Holland.

Sproat, Richard and Barbara Brunson. 1987. Constituent-Based Morphological Parsing: A New Approach to the Problem of Word-Recognition. *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*65–72, ACL, Stanford, CA.

Zlatev, Jordan. 1988. Bulgarian Nominal Inflection in the Two-Level Framework. Manuscript, Department of Linguistics, University of Stockholm.

Zlatev, Jordan, Gunnar Eriksson and Gunnel Källgren. 1989. *Linguistic Theories, Formalisms and Natural Language Processing: an Overview.* Institute of Linguistics, University of Stockholm.

Department of Linguistics
University of Stockholm
S-106 91 Stockholm
JORDAN@COM.QZ.SE

Part II

Machine Translation

POUL ANDERSEN

How Close Can We Get to the Ideal of Simple Transfer in Multi-lingual Machine Translation (MT)?

Abstract

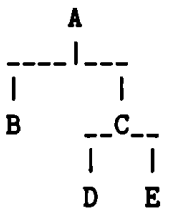
The ideal of simple transfer aims at restricting transfer rules to the exchange of unstructured lexical entities—the terminal leaves in the tree structure that is output from monolingual analysis. All information that is not lexicalised in the source language is represented as features to be transferred unchanged to the target language. In EUROTRA this ideal is approached through a centrally coordinated research within various phenomena which are supposed to be of translational relevance, i.e. having language specific surface manifestations. The outcome of this research ideally is to agree on a uniform treatment of these phenomena across languages, thus leading to simple transfer.

The paper makes a non-exhaustive overview over problems solved, problems under investigation, known but outstanding problems, and on this basis introduces a discussion of what will remain as unsolvable problems within an essentially sentence-based MT-system.

1 Introduction

MT-systems traditionally are classified into transfer-based systems and interlingual systems, as illustrated by figure 1 and figure 2, resp., on the next pages. The Interface Structure (IS) for some language is an annotated tree structure, where information is encoded as structure + features:

<u>structure</u>



<u>features</u>

```
B = {attribute 1 = value X,
     attribute 2 = value Y,
     ...                   }
```

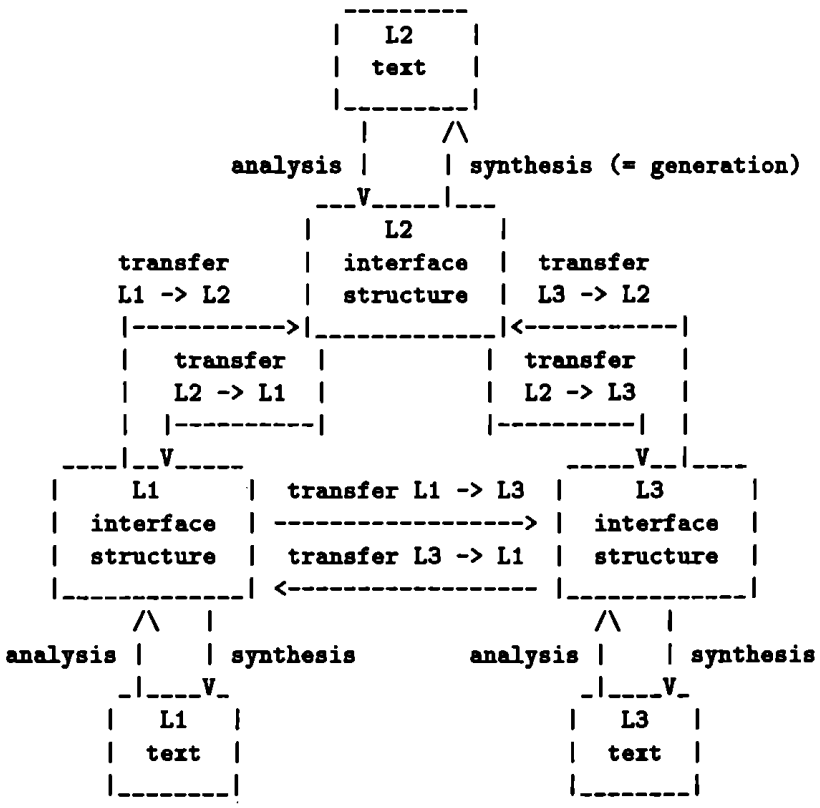


Figure 1: Schematic representation of transfer-based multi-lingual MT

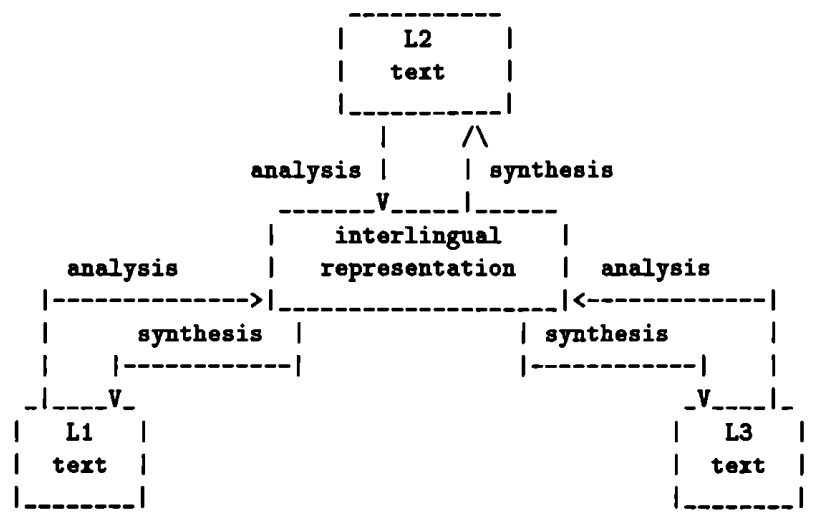


Figure 2: Schematic representation of interlingual multi-lingual MT

EUROTRA is conceived as a transfer-based system, which may seem less appropriate for an MT system comprising 9 languages in all combinations, thus leading to the construction of 72 transfer modules on top of 9 analysis and 9 synthesis modules, instead of just having one interlingua, 9 analysis and 9 synthesis modules.

What we want to show, is that the distinction between transfer- and interlingua-based systems should not be pushed too hard, especially if an interlingua is not perceived as a natural language-like representation but as any kind of information encoding that is neutral with respect to a source language and a target language.

The ideal in transfer is sometimes described as simple lexical transfer, which means that the lexical values are the only information in the interface structure that is not shared by source and target language and which consequently has to be changed by a transfer component, whereas all other information is represented language-independently in an interlingua. Actually, the greater part of lexical transfer may also be dispensed with through the inclusion of a comprehensive terminological component that is treated interlingually.

As the IS representation may be split up into structure information and feature information, we shall treat these independently and distinguish between

1. Transfer of structure

and

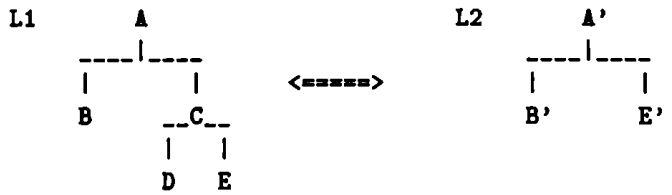
2. Transfer of features

2 Transfer of Structure

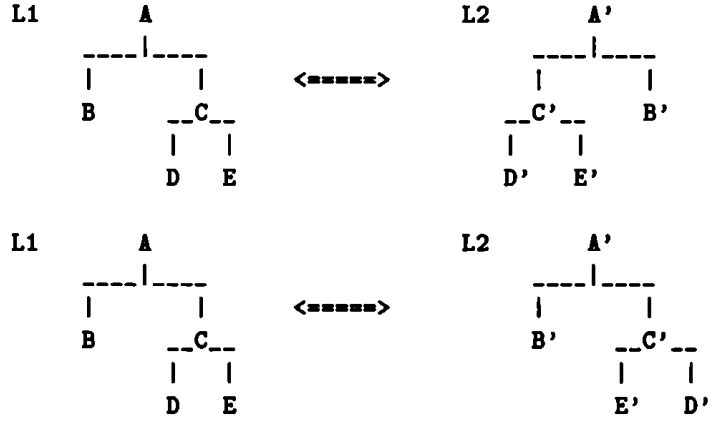
Here we distinguish between three possibilities:

2.1 Simple transfer = interlingua (= no explicit transfer)

2.2 Deletion/insertion of node



2.3 Reordering of elements



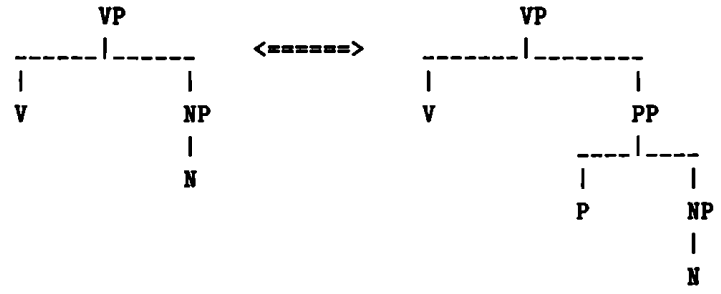
2.1 Simple Transfer

This is the unproblematic case where there is isomorphy between source language and target language or where this isomorphy is achieved between output from source language analysis and input to target language synthesis. How this isomorphy is achieved, is described in 2.2 and 2.3 below.

2.2 Deletion/Insertion of Node

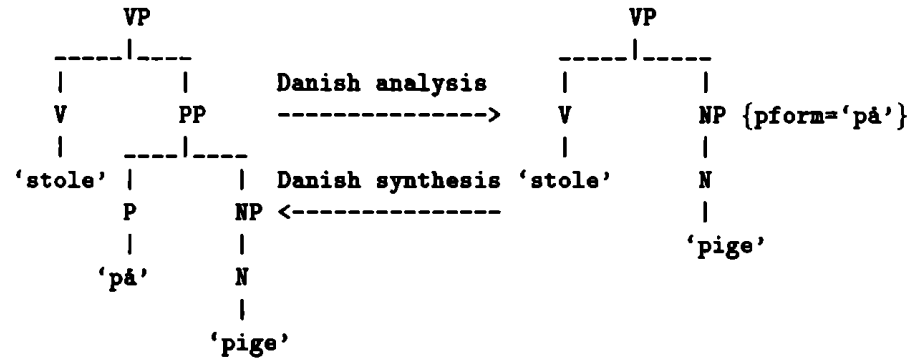
2.2.1 Direct/Indirect Government

We have to delete, respectively insert, a node in cases where we have direct government by a verb of a noun phrase in one language corresponding to indirect government through a preposition in another language, e.g.




```
EN : (He)  trusted    her      DA : (Han) stolede  på    hende
DA : (Han) betragtede hende    EN : (He)  looked   at    her
```

The solution is to featurise all valency bound prepositions, without regard to whether they have a correspondence or not in one or more other languages, and delete the preposition and the PP-node from the IS representation:



This featurisation of the preposition is accompanied by a feature in the IS dictionary entry for the verb:

```
{da_lu = 'stole', valency = subject_object, pform_of_object = 'på'}
```

It should be noted that it is not always without problems to distinguish between valency bound complements, where the preposition is deleted from the structure, and free modifiers, which at present keep their preposition.

Another related problem is indirect government by a verb through an NP, which is described in detail in Susanne Nøhr Pedersen's paper 'The Treatment of Support Verbs and Predicative Nouns in Danish'.

2.2.2 Function Words vs. Inflectional Endings

Another example of deletion/insertion of nodes are function words in one language which correspond to inflectional endings in other languages, e.g. articles with nouns and auxiliaries with verbs. Here again the problem is solved by representing the information contained in the function word as a feature on the content word or its projection, i.e. the NP or the VP.

A problem arises e.g. in country names, which take the definite article in French but go without article in Danish, English and German. In these cases we would prefer to block the automatic transfer of definiteness and leave it to the target language to calculate its surface representation. Modified country names, again, might have their definiteness transferred, as in 'a united Europe' or 'das Europa der Nachkriegszeit', although this is not without problems. Determination and quantification in general is a very complicated subject to be treated contrastively, and at present it is being investigated as a special research topic within EUROTRA.

2.2.3 Featurisation vs. Structural Representation

There are strong advantages in representing as much information as possible as features and thus reducing complex structural transfer,—an approach I personally favour. In EUROTRA there is, however, a certain opposition against removing too much from the structure. The argumentation is that most surface words can be modified, and it is more convenient to represent a modification of

a node in a structure than to modify information that has been featurised. As two examples of surface expressions that might be featurised—and actually were featurised, but now must be present as nodes in the IS representation—we may mention modal verbs and demonstrative pronouns.

2.3 Reordering of Elements

2.3.1 Reordering at NP Level

Reordering of elements occurs at NP level, where a modifier may precede the noun or follow after the noun, and where the ordering in different languages also differ according to the category of the modifier:

```
        Adjective + Noun     <=====>     Noun + Adjective

DA : den  blå     himmel
EN : the  blue    sky            FR : le ciel   bleu
DE : der  blaue   Himmel
but
FR : la   petite  fille

     NP modifier + Noun      <======>     Noun   +   NP/PP modifier

                                    EN : the inhabitants of the country
DA : landets   indbyggere           DE : die Einwohner      des Landes
                                    FR : les habitants   du     pays
```

The solution is to have a common, language-independent ordering (referred to as 'canonical ordering') of the elements in the IS representation, and do the necessary reordering in analysis and synthesis:

```
Canonical ordering
-----------------------------|--------------------------
|                            |                          |
governor          complements (valency-bound)    modifiers (free)
```

Canonical ordering of NPs

```
        NP                          NP
   _____|_____                 _____|_____
   |          |                |          |
   N          AP               N          NP {case=genitive}
              |                           |
             ADJ                         N {case=genitive}
```

```
+ featurisation of prepositions:

        NP                                        NP
   _____|_____  EN/FR analysis            _____|_____
   |          | ==============>          |          |
   N          PP                         N          NP {case=genitive}
          ___|___  EN/FR synthesis                  |
          |     | <==============                  N {case=genitive}
          P     NP
          |     |
    EN : of     N
    FR : de
```

2.4 Reordering at Sentence Level

Two examples of reordering at sentence level:

sentence 1

```
      NP + Vaux + Vmain + NP                NP + Vaux + NP    + Vmain

DA :  Hun   har   spist   brødet
EN :  She   has   eaten   the bread   DE : Sie  hat das Brot gegessen
FR :  Elle  a     mangé   le pain
```

sentence 2

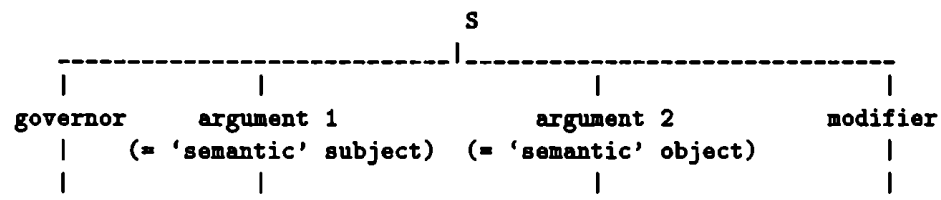
```
      AdvP   +  Vaux +  NP    +   Vmain  +  PP
DA :  I går      blev    forslaget  vedtaget  af Rådet

      AdvP   +  Vaux +  NP       +     PP    +  Vmain
DE :  Gestern    wurde   der Vorschlag  vom Rat  verabschiedet

      AdvP   +  NP        +     Vaux + Vmain +  PP
EN :  Yesterday the proposal     was    adopted  by the Council
FR :  Hier      la  proposition  a été  adoptée  par le Conseil
```

In analysis, articles and auxiliary verbs are featurised and removed from the structure, and the fact that the sentence is in passive voice is marked as a feature at the top node. At present, we do not use a refined set of semantic case roles but restrict ourselves to a numbering of arguments, where i.a. the subject of a sentence in active voice is labelled 'arg1' and the object is labelled 'arg2'. The maximum number of arguments in a sentence is 4.

Somewhat simplified, and without feature information, the IS representation of the two sentences looks like this:



sentence 1

DA : spise	hun	brød	
DE : essen	sie	Brot	
EN : eat	she	bread	
FR : manger	elle	pain	

sentence 2

DA : vedtage	Rådet	forslag	i går
DE : verabschieden	der Rat	Vorschlag	gestern
EN : adopt	the Council	proposal	yesterday
FR : adopter	le Conseil	proposition	hier

The canonical ordering of the elements is in itself fairly straightforward and poses no major problems. What creates problems may be differences between languages and differences between language groups in analysis of some constituent, e.g. as complement or modifier. This is the reason why we are very wary of introducing a too ambitious approach in assigning case roles, as this would give rise to inconsistencies between assigment carried out in different language groups.

3 Transfer of Features

Here again we distinguish between three possibilities:

3.1 Features which are transferred unchanged.

3.2 Features which are not transferred but calculated again in the target language or found in the target dictionary.

3.3 Features with an explicit translation in the transfer component.

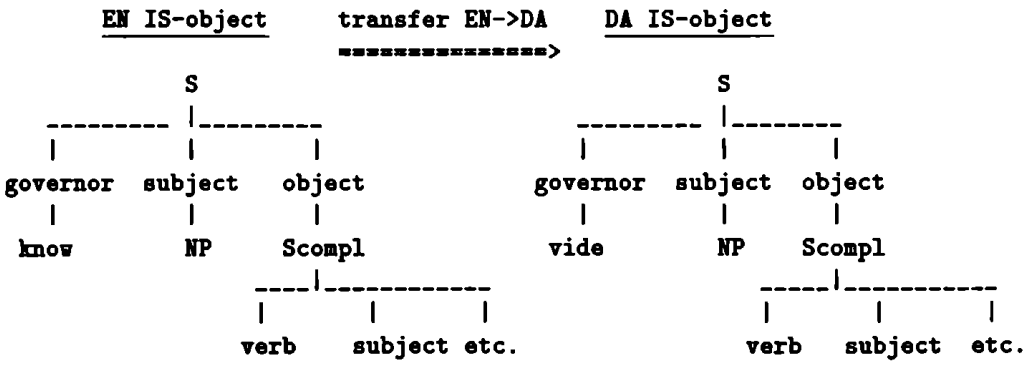
A feature has the form {attribute=value}, and what is transferred unchanged, calculated or translated explicitly is only the value of the feature.

In the first two cases no explicit transfer is needed, which means that we have simple transfer or interlingual treatment. In 2.2.2 above we mentioned definiteness as an example of a surface phenomenon that gives rise to both the first two types of transfer of features. In many cases morpho-syntactic definiteness may express semantic definiteness in a consistent way across languages, and in these cases we may transfer the value for the 'definiteness' attribute unchanged. Some (sub)categories, however, allow only of one of the paradigmatic set of values for definiteness, and this value may not be the same for different languages. In these cases the value is not transferred but found in the target dictionary — or in the target grammar if it is possible to generalise over a class of words, cf. the example

mentioned in 2.2.2 with country names, e.g. 'la France' {definiteness=definite} versus 'Frankrig' {definiteness=absent}.

In general, feature values which are not transferred, are typically bound to a lexical value, e.g. gender and semantic features on nouns and pforms on verbs (i.e. the preposition used in a valency bound PP), as well as other valency frame information, including restrictions on the semantic features of valency bound complements. These values are looked up in the dictionary. The value for 'gender' is then used to generate the correct form of modifying adjectives and determiners, and the valency information in the dictionary entry for a verb is matched with the available information on the complements. It is also used to connect the complement by means of the correct preposition in the target language.

Where there is more than one translation of a verb, the valency information in the target dictionary is used to decide which translation matches the structure of the IS object, which may be transferred unchanged, e.g.



lexical transfer rules

```
{en_lu = know}  =>  {da_lu = kende}
{en_lu = know}  =>  {da_lu = vide}
```

Danish target dictionary

```
{da_lu = kende, da_isframe = np_subject_np_object}
{da_lu = vide, da_isframe = np_subject_scompl_object}
```

We do not need to transfer the valency information of 'know', as only 'vide' matches the transferred IS-object, due to the restriction on 'da_isframe' in the Danish target dictionary (the description here is somewhat simplified).

We want to restrict explicit translation in the transfer component to lexical values in the narrow sense as uninflected wordforms. But this lexical transfer may also be reduced through an interlingual approach to certain categories of words. We have already mentioned function words such as noun determiners and auxiliary verbs, which are featurised and given an interlingual description.

But where we really hope to save a lot of explicit transfer rules is in the treatment of terms. The implementation of terminology is just being started, but we hope to treat the greater part of the planned 20.000 entries dictionaries

for each of the 9 EUROTRA languages as terms without entering them in the 72 transfer dictionaries. Terms are coded centrally with their valency frames and they are assigned a unique term-number to be used as reference (instead of the lexical value) by all language groups. The general frame description specifies how many, and which, valency bound arguments a given term takes, and each language group then complements this with language-specific information about which preposition, surface case etc. is used together with which argument.

4 Conclusion

As conclusion, we shall show by means of an example how far towards an inter-lingual interface structure we in principle have advanced. The example is somewhat simplified, and the actual state of implementation in EUROTRA may differ slightly from this presentation of an 'ideal' implementation.

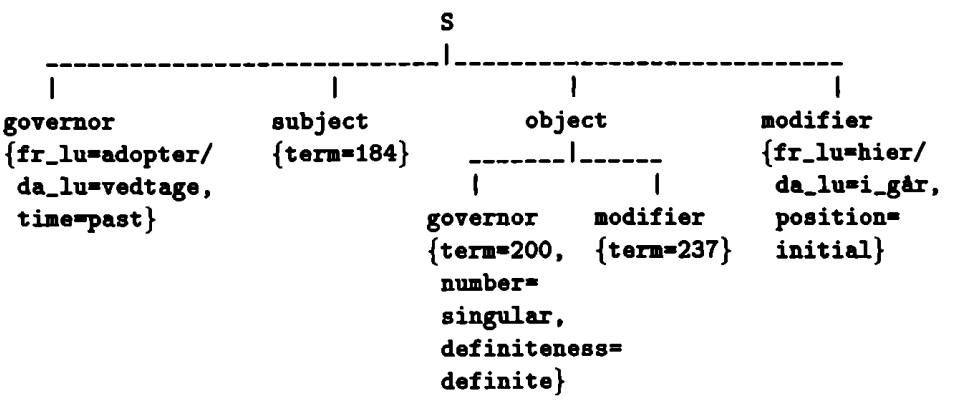
French text
Hier, la France a adopté la proposition du Conseil.

Danish translation
I går vedtog Frankrig Rådets forslag.

Please note that the translation involves i.a. —

- reordering of 'la France/Frankrig' and 'a adopté/vedtog'
- reordering of 'la proposition/forslag' and 'du Conseil/Rådets'
- change of morpho-syntactic tense/aspect from 'parfait simple' to 'imperfektum', — both expressing the same semantic past
- change of morpho-syntactic definiteness of 'la France/Frankrig'
- change of the surface manifestation of the definiteness of 'du Conseil/ Rådet' and 'la proposition/forslag' (in the latter case only being expressed through a preposed genitive)

IS representation



The only difference in the IS representation for the two languages are the lexical values for the two non-terms, so the only explicit transfer rules needed are the following:

FR-DA transfer dictionary

```
{fr_lu = adopter}   ==>  {da_lu = vedtage}
{fr_lu = hier}      ==>  {da_lu = i_går}
```

All other information is contained in the two monolingual dictionaries:

FR dictionary

```
{term = 184,
 fr_lu = 'France',
 fr_definiteness = definite,
 fr_gender = feminin,
 fr_number = singular}

{term = 200,
 fr_lu = 'proposition',
 fr_gender = feminin}

{term = 237,
 fr_lu = 'Conseil',
 fr_definiteness = definite,
 fr_gender = masculin,
 fr_number = singular}

{fr_lu = 'adopter',
 fr_isframe = subject_object}

{fr_lu = 'hier'}
```

DA dictionary

```
{term = 184,
 da_lu = 'Frankrig',
 da_definiteness = absent,
 da_gender = neuter,
 da_number = singular}

{term = 200,
 da_lu = 'forslag',
 da_gender = neuter}

{term = 237,
 da_lu = 'Rådet',
 da_definiteness = definite,
 da_gender = neuter,
 da_number = singular}

{da_lu = 'vedtage',
 da_isframe = subject_object}

{da_lu = 'i_går'}
```

For clarity of exposition, only information relevant to our example is included here.

In this example we distinguish between 'definiteness' and 'fr_definiteness'/ 'da_definiteness'. The idea is that a feature may have a language-independent attribute name in cases when it expresses semantic information to be carried over, and the same attribute name with a language prefix in cases when the value is not semantically significant but concerns monolingual wellformedness. The distinction between universal features and monolingual features is currently made in EUROTRA by means of uniform attribute names + prefixes, which enables/disables matching, but this way of using the same attribute name with or without prefix is not implemented.

EUROTRA-DK
Njalsgade 80
DK-2300 København S.
poul@eurotra.dk

ANNELISE BECH

The Design and Application of a Domain Specific Knowledgebase in the TACITUS Text Understanding System

Abstract

TACITUS is a text understanding system being developed at SRI International. One of the main components in the system is a knowledgebase which contains commonsense and domain specific world knowledge encoded as axioms in a first order predicate calculus language. The prime function of the knowledgebase is to provide extra-linguistic facts to be used in the resolution of a range of ambiguities such as compound nominal constructions, definite reference, and in drawing conclusions on the basis of the implicatures in the text. The paper discusses the methodology used in building a knowledgebase for analyzing news reports about terrorist attacks, and demonstrates how it is used in an application extracting information to be stored in a simulated database.

1 Preamble

During my term as International Fellow at SRI International, California, this past winter, I had the opportunity to familiarize myself with the TACITUS text understanding system. Under the supervision of Jerry Hobbs, who is head of the TACITUS project, I developed a domain specific knowledgebase for the TACITUS system. The present paper is a brief and fairly high-level and non-technical overview of the enterprise.

Section 2 of the paper presents the methodology used in the construction of the knowledgebase for news reports about terrorist attacks; a crude outline of the TACITUS system is given in section 3 as necessary background information before we go on to looking in detail at an example text in sections 4 and 5. We conclude with some final remarks in section 6.

2 The Methodology behind the Construction of the Knowledgebase

Our goal was to build a fairly large knowledgebase for a specific domain, namely terrorist attacks, to be used as a basis for automated understanding of texts falling within this domain, and subsequent automatic extraction of specific information. We decided to work on the basis of a set of sample texts, and we compiled a corpus consisting of several news reports about terrorist events. This corpus then constituted the backbone in our work.

Rather than adopt what might be termed a strict sublanguage approach to the descriptive task (cf. Hirschman 1986, and Hobbs 1984 for more detailed discussions), we employed a methodology of stepwise refinement (cf. Hobbs 1984).

The three steps of our working methodology, which will be elaborated on below, consisted in:

- An (informal) analysis of the corpus texts in order to establish a basic vocabulary, determine and select relevant facts for the domain.
- Breaking up the domain into self-contained and coherent sub-domains.
- Axiomatizing the facts of the subdomains.

2.1 The Analysis of Corpus Texts

Firstly, the corpus texts served the purpose of establishing the basic vocabulary in our system. Secondly, they constituted a picture of the world we intended to model in our knowledgebase, i.e. what are the settings, what are the typical actions, who are the agents, what are the roles and relations between the entities in our 'terrorist' universe, etc. Thus they indicated what linguistic and extra-linguistic information would be needed in our knowledgebase.

Using a full-sentence concordance of the sample texts, we looked at each single lexical item in context, and noted down, in an informal manner, what facts were linguistically presupposed and what general background knowledge would be needed in order to understand a given occurrence of a lexical item in its context. (We will not discuss the meaning of 'understand' here, but we use it in a sense similar to that of Eco's term 'actualisation' (Eco 1979)).

The analysis results in a first breaking down of each item into component parts and explicit statements about the implicatures (Grice 1975) carried by the text.

2.2 Structuring the Domain Information

The aim of the second step was to structure the domain information by sorting facts into sub-domains or 'clusters' (Hayes 1985). The prime reason for imposing a structure on the domain is to enhance conceptual clarity, attain modularity, and to be able to discover gaps and logical dependencies in the knowledgebase.

Sorting facts into sub-domains is generally a straightforward process. The first crude distinction which can be made, is that between facts pertaining to commonsense knowledge and domain specific or specialized knowledge. The former is facts about the world in general and not particularly tied to a specific domain (be it terrorist actions, information technology, or what have you), whereas the latter characterizes the facts which are quite often found to be restricted and highly specialized.

Facts pertaining to for example space, time, and belief are considered commonsense knowledge, whereas various facts about terrorist organizations are clearly domain specific, and essential for the understanding of reports about terrorist events. Geographical facts about the location of cities and countries seem to fall somewhere between the more abstract commonsense notion and the specialized domain knowledge.

On the basis of the results from our fact-finding, i.e. step one above, we defined 30 sub-domains. The overall conceptual structure for the knowledge base, the sub-domains and the relations between them, can be schematically rendered by the illustration in figure 1.

Apart from providing conceptual clarity, the advantage of this modular approach is obviously that it permits you to later enhance or modify the sub-domains in the knowledgebase independently of each other.

2.3 Axiomatization of the Facts

The final step in the construction of the knowledgebase consisted in creating precise ontologies for the individual sub-domains, i.e. what entities exist and what are the relations between them, and axiomatizing the facts.

The main task here was to decide on which predicates to decompose, i.e. characterize by other or new predicates, and which were to be basic predicates, i.e. ground terms for which no further description is provided.

The idea behind the adopted approach is neither to fully define each lexical item in the sense of providing necessary and sufficient conditions, nor to decompose it into a predefined set primitives in the Schankian tradition. Rather, the purpose was to characterize the predicates used in the knowledge-base. Consider as an example the following axioms from the 'organization' sub-domain.

```
organization (o)   ->   E s (Vx. x∈s ->   person (x) &
                                           member (x,o)) &
                        E p,g  plan (p,g,o)

member (x,o)       ->   E e. role (e,x,o)

role (e,x,o)       <-   agent (x,e) & in_service_of (e,g,p) &
                        plan (p,g,o)
```

These axioms give the basic facts about organizations, i.e. that an organization has persons as members, and that they have a plan. Furthermore, a member

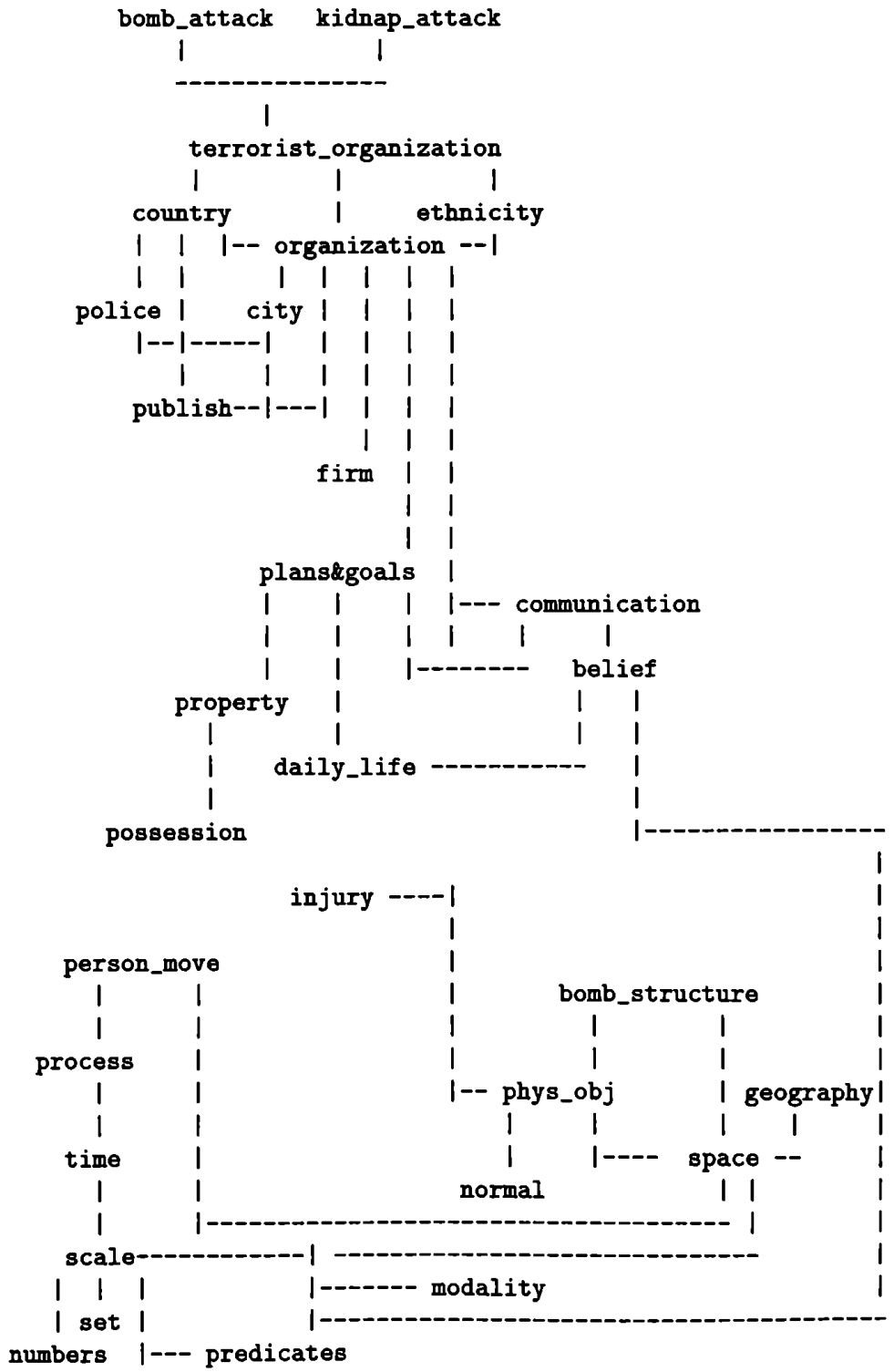

bomb_attack
kidnap_attack
terrorist_organization
country
ethnicity
organization
police
city
publish
firm
plans&goals
communication
belief
property
daily_life
possession
injury
person_move
bomb_structure
process
phys_obj
geography
time
space
normal
scale
modality
set
numbers
predicates

Figure 1:

has a role, which is being the agent of some action which is in service of the plan of the organization.

3 The Knowledgebase and the TACITUS System

To test our knowledgebase, we implemented a subset (app. 100) of the axioms we had defined on the system, and ran different types of sentences. The axioms are stated in the 'ontological promiscuous' notation developed by Hobbs (cf. Hobbs 1985b).

This notation is a first order predicate calculus language with the addition of a nominalization operator, written '!', and an extra argument, informally referred to as the 'self' argument.

To be more concrete and to convey the basic intuition of the notation to the reader, let us consider a simple example:

explode (b) which is to be read as: b explodes

explode!(e1*, b) which is to be read as: the explosion of b.

Where p(x) says that p is true of x, p!(e,x) says that e is the eventuality or possible situation of p being true of x. Consequently, Hobbs' notation can be related to standard first order predicate expressions by the following axiom:

(Vx) p(x) <=> (Ee) p!(e,x) & Rexists(e)

where Rexists(e) says that the eventuality 'e' does in fact really exist.

In sum, the basic idea of the notation is that of splitting a sentence into its propositional content and an assertional/existential claim. Furthermore, the self argument, i.e. the 'e', provides a 'handle' for referring to a predication, i.e. a predicate and its argument, in other predicates.

Before we go on to discussing a sample text, we will give a crude overview of the basic components and their functioning in the TACITUS system. We deliberately ignore some of the more advanced features of TACITUS in order not to get bogged down by too many technical details. Unfortunately, this means that we do not do TACITUS full justice (but for more detailed and comprehensive descriptions of the system, see for example Hobbs 1986c and later).

The system, which is implemented in LISP and runs on Symbolics, comprises an interpretation component and a task specific analysis component. In the interpretation component, there is a parsing and a pragmatics module. The parsing module handles the syntactic and what we will call the basic semantic analysis; this module is a further development of the DIALOGIC system (Grosz et al. 1982) used in TEAM (Grosz et al. 1987). As output, it produces a logical form of the parsed sentence in a first-order predicate calculus language. The logical form is elaborated on, or more precisely, further processed by the second module of the interpretation component, the pragmatics module. The task of

the pragmatics module is to resolve referential expressions and some syntactic ambiguities, to expand metonymies, and to interpret the implicit relations in compound nominals. The pragmatics module works by constructing a logical expression for the basic semantic analysis result, and calling the KADS theorem prover (Stickel 1982) to prove or derive it using a scheme of abductive inferencing in which it is permitted to assume the existence of 'new' facts. The theorem prover draws on the knowledgebase of commonsense and domain knowledge to complete the task.

Abductive inference is, of course, a logically invalid mode of inference, i.e. given $p(X) \rightarrow q(X)$ and $q(a)$ we conclude $p(a)$. However, we may argue, as does Hobbs (cf. Hobbs et al. 1988), that it is a reasonable way of looking at text understanding because abduction is inference to the 'best explanation' in a given context. $q(a)$ can be thought of as the observerable evidence, the implication as the general principle that could explain the occurence of $q(a)$, and the antecedent of the implication as the underlying cause or explanation of $q(a)$.

An interesting feature of the pragmatics module is that it uses a scheme for abductive inferencing in which weights and costs are assigned to the axioms (for further details, see e.g. Stickel 1988). Thus if we cannot prove an antecedent, we assume its existence at some cost. Some basic heuristic principles controlling the weights and assumability costs are hardwired into the system (e.g. it is more expensive to assume a fact than to prove it, and it is less expensive to assume an indefinite entity than a definite one), but the axioms in the knowledgebase may be assigned costs manually (cf. 4.2). The interpretation of a text in this abductive and assumption-based framework, amounts to producing the minimal explanation of why the text would be true (cf. Hobbs et al. 1988 for a detailed discussion).

The analysis component, i.e the component for extracting task specific information from an interpreted text, is basically a specialized call to the theorem prover (see further below). The enhanced logical form, i.e. the result output from the pragmatics module, is abductively proved by back-chaining over the axioms in the knowledgebase.

In the next sections, we will have a look at an example text and show how the knowledgebase is used for disambiguation and computation of implicit information.

4 An Example

Let us now consider the following two sentences as an example text to be treated within our framework:

(1) A bomb exploded at a Renault showroom in Bilbao. A person claiming to represent the ETA-M had warned of the blast in a call to the police.

Linguistically, the sentences present us with problems of resolving a compound nominal construction, 'Renault showroom', and locating a possible antecedent for 'the blast'.

The extra-linguistic knowledge needed in order to achieve some reasonable level of understanding of the text is among other things: Renault is a French firm manufacturing products, i.e. cars, a showroom is a building owned by a firm where the products of that firm are on display, Bilbao is a city in the country Spain, ETA-M is a terrorist organization, and terrorist organizations have members, certain plans and goals and violent methods for reaching their goals, and an explosion generally involves a blast.

The basic facts such as for instance Spain being a country and ETA-M being a terrorist organization, are encoded as existential axioms in the knowledgebase. E.g:

```
(1a) (Defaxiom COUNTRY-SPAIN-1 (terror)
         ''Spain is a country''
         ((SOME ((e1* . ev) (country! e1* spain)))

(1b) (Defaxiom TERORG-ETA-M-1 (terror)
         ''ETA-M is a terrorist organization''
         ((SOME ((e1* . ev) (terorg! e1* eta-m)))
```

The quantified variables in the axioms are marked for their type such that 'ev' denotes event and 'nev' non-event variables.

4.1 Axioms for Disambiguating Compound Nominal Constructions

From the linguistic point of view, the TACITUS framework offers interesting possibilities for disambiguating compound nominal expressions using linguistic as well as extra-linguistic knowledge.

The individual nouns in a compound nominal construction are analyzed as arguments of the generic 'nn'-predicate. That is, the expression 'Renault showroom', would appear as **nn(e1*,Renault,Showroom)** in the initial logical form of the sentence produced as output from the parsing module.

In formulating the axioms for resolving such nn-relations, we adopted a strategy combining the line of analysis for coumpound nominals proposed by Downing (1977), and that advocated by Levi (1978). In summary, Downing argues that the semantic relationship between the elements of a coumpound cannot be characterized in terms of a finite list of appropriate compounding relationships, whereas Levi tries to establish such a list for the most common cases on the basis of the transformational relationship between the elements.

Our combined approach can be seen in the following sample axioms, where the first two axioms encode the possible general relationship as expressed in terms of prepositions, and the subsequent two axioms state further specific constraints.

```
(2a) (Defaxiom NN-1 (terror)
     ''An nn-relation: for''
     (ALL ((e1* . ev) (p . nev) (s . nev))
          (IMPLY (for! e1* s p)
               (SOME ((e2* . ev))
                    (nn! e2* p s)))))

(2b) (Defaxiom NN-2 (terror)
     ''An nn-relation: of''
     (ALL ((e1* . ev) (f . nev) (s . nev))
          (IMPLY (of! e1* s f)
               (SOME ((e2* . ev))
                    (nn! e2* f s)))))

(3a) (Defaxiom FOR-1 (terror)
     ''A showroom is for products''
     (ALL ((e2* . ev) (s . nev) (e3* . ev) (p . nev) (e4* . ev) (f . nev))
          (IMPLY (AND (showroom! e2* s) (product! e3* p) (firm! e4* f))
               (SOME ((e1* . ev))
                    (for! e1* s p)))))

(3b) (Defaxiom OF-1 (terror)
     ''A showroom is owned by a firm''
     (ALL ((e2* . ev) (s . nev) (e3*. ev) (e4* . ev) (f . nev))
          (IMPLY (AND (showroom! e2* s) (own! e3* f s) (firm! e4* f))
             (SOME ((e1* . ev))
                      (of! e1* s f)))))
```

In trying to abductively prove a relevant logical form output from the parsing module and to make implicit information explicit, the pragmatics module has the theorem prover back-chain over the axioms in the knowledgebase. Thus an nn-relation as the above is resolved against 2a and 2b, then the new goals, of!(e1* s f) and for!(e1* s f), are resolved against 3a and 3b respectively, yielding new goals to be resolved.

4.2 Axioms for Resolving Referring Expressions

As mentioned above, one of the basic heuristic assumption hardwired into TACITUS' pragmatics module is that an indefinite noun phrase introduces new information and a definite noun phrase refers to a known entity, i.e. something which is either in the knowledgebase or has been introduced in the previously processed text. Hence the cost of assuming an indefinite noun phrase is cheaper than assuming a definite noun phrase.

In the example sentences given in (1), the noun phrase 'the blast', is related to the event of the explosion mentioned in the preceeding sentence. Simplifying somewhat (cf. further below), we could say that 'the blast' is in a sense a nominalization of 'a bomb exploded'.

In order to establish reference connections of this type, we define the following kind of axiom in our knowledgebase:

```
(4) (Defaxiom EXPLOSION-BLAST-1 (terror)
    ''An explosion generates a blast''
    (ALL ((e1* . ev) (x . nev) (y . nev) (z . nev))
         (IMPLY (AND (ASSUMABLE (etc-expl e1* x y z ) 0.3)
                     (explode! e1* x y z))
                (SOME ((e2* . ev) (b . nev))
                     (AND (blast! e2* b) (genn e1* e2*))))))
```

Essentially, this axiom says that a blast (e2*) implies the occurrence of some explosion event (e1*), and that the latter generates the former, which is stated by way of the primitive predicate 'genn'. The predicate 'etc-expl', which can be seen as 'additional', but not spelled out properties relating to the explode predicate, is introduced because we do not want to state flatly that 'a blast' and 'an explosion' is the same thing.

Since an 'explosion' is known (it was introduced in the previous sentence), it is free of charge to resolve the second predicate in the antecedent of the axiom against this known fact. The first predicate in the antecedent has been assigned such a low assumability cost (0.3), that proving 'blast' by use of the axiom is cheaper than to assume its existence.

5 Extracting Specific Information from the Texts

The logical form encapsulating the interpretation found for a text, i.e. the output from the interpretation component, is the input to the task specific analysis component. The analysis is performed on the basis of the logical form and a 'task schema specification' given to the theorem prover.

5.1 The Schema

Let us here consider a simplified example of the kind of event related specific information we would like the system to compute. For a given text describing a terrorist event, we would like to find answers (if any) to 'questions' such as the following:

```
INCIDENT TYPE:
TARGET TYPE:
TARGET NATIONALITY:
INCIDENT CITY:
INCIDENT COUNTRY:
RESPONSIBLE ORGANIZATION:
.
.
etc.
```

The above actually simulates a database record to be automatically filled in. However, as the system was not yet hooked up to produce actual database

entries, the answers found are printed out on the screen. The slots in the 'record' are filled by the values found for variables when presenting the theorem prover with goals to be abductively proven by using the information from the text interpreted and the facts in the knowledgebase.

The goals of the schema appear as the consequent in what might informally be called the 'linking axioms' in the application task specific part of the knowledgebase. Linking axioms can be thought of as guidelines for how to find answers to the 'questions' posed by way of the schema specification.

The schema itself is a metalogical LISP expression in a first-order predicate calculus form annotated by non-logical operators for search control and resource bounds. The two non-logical operators are 'proving' and 'enumerated-for-all'. Without going into technical details about these two operators (for more details, see Tyson and Hobbs 1988), let us simply present a small excerpt from the schema for the above example 'record', and make some explanatory comments in order to convey the basic intuitions of the process to the reader:

```
(proving
     (enumerated-for-all ((e1 . ev))
          (proving
               (some ((it . nev)) (incident-type e1 it))
               (terror-limits default-time)
               print-incident)
               (and
                    (enumerated-for-all ((it . nev))
                         (proving
                              (incident-type e1 it)
                              (terror-limits default-time)
                              print-incident-type)
                             :true)
                                :
                                :
                    (enumerated-for-all ((ro . nev))
                         (proving
                              (responsible-organization e1 ro)
                              (terror-limits default-time)
                              print-responsible-organization)
                             :true)
               (terror-limits default-time)
          print-sentence-finished)))
```

The linking axiom in the knowledgebase for 'responsible organization' could be the following statement:

```
(5)  (Defaxiom RESP-ORG-1 (terror)
     ''The organization responsible for the attack''
     (ALL ((e1* . ev) (e . ev) (e2* . ev) (o . nev) (e3* . ev))
          (IMPLY (AND (terattack! e1* e) (responsible! e2* o e)
                                         (terorg! e3* o))
                 (responsible-organization e o))))
```

Thus, we find the organization (o) responsible for an attack (e) by proving that e is a terrorist attack, that o is a terrorist organization, and that o is responsible for e.

Contrary to the pragmatics module, no assumptions are made in the task specific analysis phase when trying to prove the goals of the schema; this step is meant to extract information only. However, the process is still back-chaining controlled abductive inferencing. This means that everything has to be proved against the knowledge in the database in conjunction with the interpretation of the text.

Proving the antecedents of the linking axioms may of course involve resolving the new goals with knowledge asserted in the text or in this case, proving further axioms in the knowledgebase.

There may also be different axioms for the same goal, indicating that a goal can be explained, or more correctly proved, in different ways. Actually, this is only a reflection of the fact that a given phenomena can be brought about in different ways. For example, there are actually three different axioms for 'responsible' in our knowledgebase.

5.2 The Information Extracted from the Interpretation Result

Let us now return to our example text. For illustration, we first show an excerpt from the result of the interpretation of the sentences in external format (6) — note the resolved compounding relationship; and then the print-out of the information automatically extracted by the analyze component from the interpretation (7) of the two example sentences.

(6)
```
INTERPRETATION 1 OF SENTENCE:
      Cost: 34
New and Assumed Information:
x1:           bomb!(e2, x1)
y1:           explode!(e4, y1, x1)
x12:          bilbao!(e13, x12)
x8:           renault!(e9, x8)
x6:           showroom!(e7, x6)
              in!(e11, x6, x12)
e4:           at!(e5, e4, x6)
              past!(e15, e4)
Given or Inferred Information:
x8:           renault!(e9, x8)
              nn!(e10, x8, x6)
              own!(e25, x8, x6)
              firm!(e26, x8)
x6:           of!(e29, x6, x8)
```

(7)
```
INCIDENT TYPE: explosion
TARGET TYPE: commercial
TARGET NATIONALITY: french
INCIDENT CITY: bilbao
INCIDENT COUNTRY: spain
PROPERTY DAMAGE: <unknown>
WARNING: yes
METHOD: phone
RESPONSIBLE ORGANIZATION: eta-m
```

6 Final Remarks

TACITUS offers an interesting framework for experimenting with knowledge-based natural language processing, and in fact it is a quite sophisticated system. Previously, the TACITUS team at SRI has been experimenting with implementations of knowledgebases for domains such as the break-down or malfunctioning of mechanical parts in ships (Hobbs 1987). Constructing a knowledgebase for the terrorist attack domain was the first attempt to deal with a slightly less restricted subject field in the TACITUS system. The main conclusion to be drawn from the experiment with the terrorist texts is that very careful axiomatization of the facts is necessary in order to achieve good results, i.e. 'nuts and bolts' have to be carefully fitted together to create 'delusions of grandeur'.

Acknowledgements:
The Danish Carlsberg Foundation provided the financial support for my stay at SRI International. Constructing and testing the domain specific knowledgebase for terrorist texts in the TACITUS system described here, was suggested to me by Jerry Hobbs and carried out under his supervision. I am indebted to Jerry for his guidance and many useful hints. Needless to say, if the present paper contains errors or misconceptions in the presentation of TACITUS, the author alone can be blamed.

References

Eco, U. 1979. *Lector in Fabula.* Milan.

Grice, H.P. 1975. Logic and Conversation. R. Schank and B. Nash-Webber [Eds.], *Theoretical Issues in Natural Language Processing*169–174. Cambridge, Mass.

Grosz, B., N. Haas, G. Hendrix, J. Hobbs, P. Martin, R. Moore, J. Robinson, and S. Rosenschein. 1982. *DIALOGIC: A Core Natural-Language System.* SRI Tech. Note 270. SRI, Menlo Park, California.

Grosz, B., D.E. Appelt, P.A. Martin, and F.N.C. Pereira. 1987. TEAM: An Experiment in the Design of Transportable Natural-Language Interfaces. *Artificial Intelligence*, 32:173–243.

Downing, P. 1977. On the Creation and Use of English Compound Nouns. *Language*, 53:810–842.

Hayes, P.J. 1985. The Second Naive Physics Manifesto. J.R. Hobbs and R.C. Moore [Eds.], *Formal Theories of the Commonsense World*:1–36. Ablex, New Jersey.

Hirschman, L. 1986. Discovering Sublanguage Structures. R. Grishman and R. Kittredge [Eds.], *Analyzing Language in Restricted Domains: Sublanguage Description and Processing*211–234. Erlbaum, New Jersey.

Hobbs, J.R. 1978. Coherence and Coreference. SRI Tech. Note 168. SRI, Menlo Park, California.

Hobbs, J.R. 1984. Sublanguage and Knowledge. SRI Tech. Note 329. SRI, Menlo Park, California.

Hobbs, J.R. 1985a. Granularity. In: *Proceedings of IJCAI-85*:1–4.

Hobbs, J.R. 1985b. Ontological Promiscuity. In: *Proceedings of ACL-85*:61–69. University of Chicago, Illinois.

Hobbs, J.R. 1986a. Commonsense Metaphysics and Lexical Semantics. SRI Tech. Note 392. SRI, Menlo Park, California.

Hobbs, J.R. 1986b. Discourse and Inference. Ms. SRI, Menlo Park, California.

Hobbs, J.R. 1986c. Overview of the TACITUS Project. *Computational Linguistics*, 12:220–222.

Hobbs, J.R. 1987. Local Pragmatics. SRI Tech. Note 429. SRI, Menlo Park, California.

Hobbs, J.R., W. Croft, T. Davies, D. Edwards, and K. Laws. 1988. The TACITUS Commonsense Knowledge Base. Ms. SRI, Menlo Park, California.

Hobbs, J.R., M. Stickel, P. Martin, and D. Edwards. 1989. Interpretation as Abduction. Ms. SRI, Menlo Park, California.

Levi, J. 1978. *The Syntax and Semantics of Complex Nominals.* Academic Press, New York.

Stickel, M.E. 1982. A Nonclausal Connection-Graph Resolution Theorem-Proving Program. *Proceedings of the AAAI-82 National Conference on Artifical Intelligence*: 229–233. Pittsburgh, Pennsylvania.

Stickel, M.E. 1988. A Prolog-like Inference System for Computing Minimum Cost Abductive Explanations in Natural Language Interpretation. *Proceedings of ICCSC 88*:343–350. Hong Kong.

Tyson, M. and J.R. Hobbs. 1988. Domain-Independent Task Specification in the TACITUS Natural Language System. Ms. SRI, Menlo Park, California.

EUROTRA-DK
University of Copenhagen
Njalsgade 80
DK-2300 Copenhagen S.
annelise_bech@eurokom.ie

ANNA BRAASCH

Udnyttelse af maskinlæsbare ordbogsdata til maskinoversættelse

Abstract

The multilingual machine translation project EUROTRA works with a transfer-based system. For the running prototype, which is going to be produced in the last phase of the project, each language group has to reach the goal of approximately 20.000 dictionary entries.

In the Danish EUROTRA group, we have started a small-scale pilot project. The purpose is to investigate the possibilities for reuse of existing machine-readable dictionary data in the extension and improvement of the monolingual Danish dictionary.

We have two different data sets available: The machine readable version of the official Danish Spelling Dictionary (RO2) and a small subset of records from the Danish-French Dictionary Base (DFOB).

The two main points of the pilot project are:

1. To get an idea as to how the dictionary information should be systematized for it to be useable in different applications;
2. How we can exploit existing machine-readable dictionary information in the coding process of the general vocabulary.

The state of the art:

1. We have made a comparison between information types needed for the EUROTRA Translation System (ETS) and the information types represented in the two data sets.
2. We carried out some tests with a subset of available dictionary data to estimate how much of the information needed by the ETS can be deduced and/or converted without too extensive programming effort.

1 Indledning

Kort skitse af projektets formål:

Vi har for nylig indledt et pilotprojekt i den danske **EUROTRA**-gruppe, der har til formål at vurdere hvilke muligheder vi har for automatisk/halvautomatisk hhv. manuel tilpasning af ordbogsdata, der er kodet til andre formål end maskinoversættelse. Pilotprojektet koncentrerer sig først og fremmest om at finde frem til de oplysningstyper, hvis bearbejdelse ikke kræver kompliceret programmering.

Metode:

Vi sammenligner de oplysningstyper, som vort oversættelsessystem har brug for i sine leksikalske regler (morfologiske, syntaktiske og på senere tidspunkt semantiske informationer) med oplysningerne i de valgte ordbøger og finder derved frem til, hvilken delmængde af de oplysninger, vi har behov for, der principielt kan findes i disse ordbøger.

Forventet resultat:

1. overblik over, hvilke krav der kan stilles til systematisering af ordbogsoplysninger, hvis genbrug for forskellige formål skal være mulig;
2. rent konkret: lemmaordbogsindgange til brug i oversættelsessystemet.

2 Pilotprojektets baggrund

Da **EUROTRA** arbejder med et niveaudelt oversættelsessystem, bruges der tilsvarende niveauspecifikke etsprogsordbøger til analyse og generering for hvert sprog, samt et sæt transferordbøger for hvert sprogpar.

I stedet for at lave de etsprogede niveauspecifikke ordbøger hver for sig, ønsker vi at fremstille én niveauuafhængig ordbog ved at samle samtlige oplysninger, der er relevante på hvert enkelt niveau, i én ordbog. Herfra skal systemet så for hvert opslagsord selektere de oplysninger, der er behov for på det pågældende niveau i oversættelsesprocessen. En ordbog, der er opbygget på denne måde kalder vi for **lemmaordbog**.

Den viste ordbogsindgang består af to dele, jf. Figur 1:

1. Selve den kørbare leksikalske regel, der indeholder 'de indre oplysninger' der anvendes af oversættelsessystemet;
2. De ikke kørbare 'ydre oplysninger' til brug for leksikografen, indledt med udkommenteringssekvensen '%%', hvilke omfatter
   (a) administrative oplysninger (koder, dato, kilde, kommentarer) og
   (b) 'pragmatiske' oplysninger (definition, eksempler).

Ordbogsindgangene er kodet efter det samme princip som grammatikkens regler, men i stedet for at indeholde generelle oplysninger om fx sætnings- eller

```
1. 'antal_n3' = [cat=n, scat=specifier, part=no, level=zero,
    dalu='antal', darno=n3,
    ers_frame=comp0, dapform1=no, dapform2=no, dapform3=no, dapform4=no,
    daisframe=arg0, dapargl=no, daparg2=no, daparg3=no, dapparg4=no,
    flex_type=fx3, dagd=neut, dcons=1, oc=no, infl=sub_root, term=xx0].
2. %% Coder: anna   22-May-89
   %% Source: Ph3 corpus
   %% DEF: en m[ngde enheder af ngt. t[lleligt
   %% Comments: NDO: uden plur.
   %% Examples: TV-sSt <nm> er en tysk satellit med 3 TV-kanaler og et ukendt
   %%             antal radiofonikanaler. 362 text4
```

Figur 1: Eksempel på en ordbogsindgang fra den danske lemmaordbog.

frasestruktur, indeholder den specifikke leksikalske oplysninger om de enkelte opslagsord.

3 Ordbogskodning

I oversættelsesprojektets tredje og sidste fase, som vi befinder os i, skal det forventede antal ordbogsindgange nå op på ca. 20.000 enheder ialt (nu: 4600), deraf 6.000 tilhørende det generelle ordforråd (nu: 3.900), 14.000 **EUROTRA**-termer (disse kodes ikke helt efter de samme principper som det generelle ordforråd, men dette vil jeg ikke komme nærmere ind på her).

Der er altså en hel del kodningsarbejde tilbage, især hvis vi også tager med i betragtning, at de allerede kodede indgange skal opdateres i takt med grammatikkernes udbygning. I den afsluttende (indeværende) projektfase opprioriteres ordbogsarbejdet.

Vi har behov for effektivisering af arbejdet, hvilket kan gøres på forskellige måder, dels ved organisationsmæssige ændringer, dels vha. automatisering af kodningsarbejdet på de områder, hvor det praktisk er mulig.

Kodningen kan til en vis grad automatiseres ved brug af **makroer** i inddateringsproceduren. Dette letter leksikografens arbejde ved tastatur og skærm.

Den manuelle ordbogskodning rummer muligheder for både skrivefejl og indholdsmæssige fejl. For at begrænse disse fejl har vi udarbejdet faste ordbogsmakroer i UNIX-editoren **emacs**. Disse små programmer omfatter forskellige faciliteter, såsom konsistens- og validitetscheck, prompt for oplysningtype til inddatering, liste af lovlige værdier for hver informationstype (attribut) osv.

Desuden anvendes ETS (**EUROTRA** Translation System) regelfortolkeren til at kontrollere, at ordbogsindgangen er kodet i overensstemmelse med regelsættet for formel og indholdsmæssig beskrivelse af leksikalske regler.

Der gives fejlmeddelelse om syntaksfejl (manglende separatorer, parenteser etc.), samt hvis et attributs kodede værdi ikke er element i det pågældende attributs værdiliste (dvs. ikke lovlig værdi). Der gives ingen fejlmeddelelse, hvis attributtets værdi er en fri streng (fx opslagsord eller præposition).

En anden metode til at effektivisere ordbogsarbejdet er at udnytte oplysninger om ord fra maskinlæsbare ordbøger. Dette er emnet for pilotprojektet. Formålet er at undersøge, hvilke muligheder vi har for at udnytte maskinlæsbare ordbogsdata ved udbygningen af lemmaordbogen. Vi arbejder jo i forvejen på den måde, at vi definerer opslagsordet og derefter 'slår op' manuelt i forskellige trykte, monolingvale ordbøger, primært i Nudansk Ordbog, Retskrivningsordbogen og Dansk Sprogbrug. Dette betyder at vi henter de relevante informationer fra ordbøger, der er lavet til andre formål end maskinoversættelse.

Arbejdet med disse ordbøger viser, at oplysningerne ikke behøver at være formaliserede, og at de ikke altid er eksplicit til stede. Andre gange følger ordbogen ikke helt den prædefinerede formatbeskrivelse. Der kan fx være afvigelser fra den erklærede forkortelsesliste, eller det kan forekomme, at den søgte oplysning ikke er repræsenteret i den ordbog, man har slået op i.

Dette er oftest ikke noget større problem for ordbogsbrugeren, men et maskinoversættelsessystem skal have entydige, eksplicitte og udtømmende oplysninger til rådighed i sine ordbøger, udtrykt i den valgte formalisme, som i vort system hedder E-(**EUROTRA**)formalismen.

Ordbogskoderens arbejde ved anvendelse af de traditionelle, trykte ordbøgers oplysninger til udbygning af lemmaordbogen består altså af flere trin:

Søgning og udvælgelse, systematisering, validitetscheck, konvertering til E-formalisme og selve indtastningen af ordbogsartiklen.

EUROTRAs ordbøger har naturligvis en række særlige træk, der er afhængige af maskinoversættelsessystemets krav, men de grundlæggende leksikalske og kontekstuelle oplysninger svarer til dem, der også findes i de nævnte trykte ordbøger.

4 Pilotprojekt — indledende overvejelser

Da vi rent faktisk 'genbruger' en del oplysninger fra trykte ordbøger, er spørgsmålet nu, hvordan vi kan inddrage maskinlæsbart ordbogsmateriale i arbejdsgangen for ordbogskodning: Hvilke trin i processen kan automatiseres, i hvor høj grad kan vi inddrage maskinkraft til at udføre opgaver automatisk eller interaktivt, og hvilke oplysninger skal indføres rent manuelt?

Vi har valgt p.t. at inddrage to forskellige datasæt i pilotprojektet, som lægger hovedvægten på forskellige oplysningstyper, svarende til deres leksikografiske koncept og anvendelsesområde:

Vi har erhvervet den maskinlæsbare version af **Retskrivningsordbogen** (herefter forkortet: **RO2**). Dette materiale er med få undtagelser indholdsmæssigt identisk med den trykte udgaves alfabetiske afsnit 'Ordbog a-å', som omfatter opslagsord tilhørende det almindelige danske ordforråd. Materialet indeholder hovedsagelig oplysninger om selve opslagsordet uden (større) kontekst, fx ordklasse, bøjningsformer, sammensætningsformer, sideformer. Der er dog artikler, der også indeholder kommentarer og eksempler, fx ved forholdsordene 'af' og 'ad' på grund af disses komplekse betydningsstruktur.

Den maskinlæsbare version (**RO2**) adskiller sig på to væsentlige punkter fra den trykte udgave:

1. Hver oplysning indledes af en feltkode, der angiver det pågældende felts oplysningstype. (Feltkoden overtager dermed skriftgradsskiftets rolle som adskiller af oplysningstyper.)

2. Der er yderligere nogle få oplysninger i den maskinlæsbare version (fx ordklassebetegnelse for navneord og udsagnsord), der kun implicit er til stede i den trykte version.

Vi har af ordbogsgruppen på Handelshøjskolen i København (HHK) fået stillet et udvalg af ordbogsposter fra **Dansk-fransk ordbase** (herefter forkortet: **DFOB**) til rådighed. Ordbasen indeholder ordbogsindgange fra Blinkenberg-Høybye's Dansk-fransk Ordbog, der er en oversættelsesordbog med kildesproget dansk og målsproget fransk.

Materialet er på HHK blevet optisk indlæst og derefter lagt ind i databaseposter. Det af os udvalgte materiale omfatter kun et mindre antal substantiver begyndende med 'afv-' og transitive verber begyndende med 'af-'.

Ved dette materiale er det tale om kontrastiv behandling af de danske opslagsord; foruden basisoplysninger vedrørende ordklasse og bøjning er der til de fleste ord yderligere materiale, omfattende betydningsopdeling, definitioner, semantiske oplysninger (emneområde), brugsrestriktioner, frekvens, eksempler og naturligvis de tilsvarende franske oversættelser.

Den franske del af materialet er p.t. ikke inddraget i pilotprojektet, men bliver det på længere sigt.

Begge datasæt foreligger som strukturerede ASCII-tekstfiler og vi har fået detaljeret formatbeskrivelse af dem begge.

Det første trin i pilotprojektet var at skaffe overblik over, hvilke af de oplysninger, **EUROTRA**-oversættelsessystemet har brug for, der er repræsenteret

```
'antal_n3' = [cat=n, scat=specifier, part=no, level=zero,
 dalu='antal', darno=n3,
 ers_frame=comp0, dapform1=no, dapform2=no, dapform3=no, dapform4=no,
 daisframe=arg0, dapargl=no, daparg2=no, daparg3=no, dapargl=no,
 flex_type=fx4, dagd=neut, dcons=1, oc=no, infl=sub root, term=xx0].
%% Coder: anna   22-May-09
%% Source: Ph3 corpus
%% DEF: en m[ngde enheder af ngt. t[lleligt
%% Comments: NDO: uden plur.
%% Examples: TV-sSt <nm> er en tysk satellit med 3 TV-kanaler og et ukendt
%%           antal radiofonikanaler. 362 text4
```

```
HORD:   an‖tal          --->        HORD:   tal
HOKL:   no                          HOKL:   no
HTGN:   ,                           HENT:   -let,
HSMS:   antals-,                    HNFL:   tal -lene;
HSMX:   antalsbegr[nsning           HEKS:   1200-tallet
```

Figur 2: Lemmaordbogsindgang og tilsvarende poster fra **RO2**.

i de to datasæt, hvordan de er kodet, og på hvilken måde vi kan udnytte eller overføre disse til lemmaordbogen.

I fremstillingen i Figur 2 vil jeg primært koncentrere mig om **RO2**, og for overskuelighedens skyld viser jeg kun substantivernes kodning som eksempler.

Lemmaordbogsindgangen i Figur 1 er forsynet med markering af de værdier, der indsættes af systemet (ubrudt understregning), samt oplysninger der er automatisk indsat fra **RO2** (markeret med stiplet linje). Figuren viser desuden sammenhængen mellem en ordbogsindgang i E-formalisme og den artikel i datasættet, som oplysningen hentes fra. (I dette tilfælde kræves 2 opslag i **RO2**, da et sammensat opslagsords bøjning som regel findes i ordbogen under det sidste led.)

5 Pilotprojektets indeværende fase

Efter at have lavet en liste over de oplysningstyper for de enkelte ordklasser, som vi skal have adgang til i lemmaordbogen, og efter at have sammenlignet denne liste med de oplysninger, der er til stede i de maskinlæsbare materialer, har vi vurderet at mulighederne for 'genbrug' er følgende:

1. En del oplysninger kan konverteres direkte fra det maskinlæsbare materiale til oplysninger i E-formalisme vha. editor-makroer (emacs), fx ordklassebetegnelse i feltet *HOKL* (='ordklasse'): no → cat (='ordklasse') = n;

2. Oplysninger kan konverteres vha. programmeret opslag i **RO2** og sammenligning af den maskinlæsbare oplysning med listen over de værdier, der kan repræsentere informationstypen (attributtet) i en lemmaordbogsindgang; fx kan de kodede bøjningsendelser for substantiver i feltet *HNFL* (='navneord flertal') sammenholdes med listen over værdierne for attributtet *flex_type*(='bøjningstype').

Første kolonne i tabellen i Figur 3 angiver koden i E-formalisme (udsnit: *flex_type*=fx1 ... fx5 for regelmæssigt bøjede substantiver), anden kolonne: eks. på opslagsord i ental, ikke genitiv. Kolonnerne 3 og 4 viser, hvordan **RO2** anfører de pågældende bøjningsendelser. Kolonne 5 indeholder den automatisk ekspliciterede flertalsendelse i bestemt form (ved regelmæssig dannelse er denne ikke til stede i **RO2**). Kolonne 6 viser attributtet *d_cons* (='fordobling af slutkonsonanten'); denne oplysning bliver ekspliciteret i konverteringsforløbet, udskilt fra felt *HENT* eller *HNFL*.

3. Udledning af en oplysning der kun er implicit til stede i materialet, fx kønnet for et substantiv, udledes af *HENT*-feltets (='navneord ental') sidste tegn og konverteres til eksplicit oplysning i E-formalisme i attributtet *dagd* (='Danish gender').

Foruden automatisk konvertering hhv. udledning af oplysninger har vi andre muligheder:

```
*************************************************************************
      s.ind-nge       s.def           pl.ind-nge        pl.def-nge
FX1:  stol            -en             -e                -ene      (d_cons=no)
      hat             -ten            -te               -tene     (d_cons=t)
      bord            -et             -e                -ene      (d_cons=no)
      blik (1)        -ket            -ke               -kene     (d_cons=k)
*************************************************************************

*************************************************************************
      s.ind-nge       s.def           pl.ind-nge        pl.def-nge
FX2:  rede            -n              -r                 -rne
      vindue          -t              -r                 -rne
      NB! ingen fordobling af konsonant
*************************************************************************

*************************************************************************
      s.ind-nge       s.def           pl.ind-nge        pl.def-nge
FX3:  virksomhed      -en              -er               -erne     (d_cons=no)
      anorak          -ken             -ker              -kerne    (d_cons=k)
      abstrakt        -et              -er               -erne     (d_cons=no)
      stakit          -tet             -ter              -terne    (d_cons=t)
*************************************************************************

*************************************************************************
      s.ind-nge       s.def           pl.ind-nge        pl.def-nge
FX4:  film             -en                -              -ene      (d_cons=no)
      bit              -ten               -              -tene     (d_cons=t)
      forslag          -et                -              -ene      (d_cons=no)
       ar (2)          -ret               -              -rene     (d_cons=r)
*************************************************************************

*************************************************************************
FX5:   s.ind-nge       s.def           pl.ind-nge        pl.def-nge
       bager           -en               -e              -ne
       NB! Ingen fordobling af konsonant
*************************************************************************
```

Figur 3: Udsnit af tabel til sammenligning af koder i E-formalisme og kodningen af bøjningsendelser i **RO2**.

4. Automatisk opslag og selektering af visse oplysninger og automatisk el. interaktiv overførsel af disse;

5. Mulighed for interaktiv kodning: valg af eksempler samt tilpasning af eksempler (tilføjelse, sletning etc).

De sidste to muligheder bliver især aktuelle ved import af **DFOB**'s danske oplysninger: **DFOB** indeholder naturligvis langt flere oplysninger end **RO2**, bl.a. kommentarer og eksempler. Det har dog vist sig, at selv om dette materiale er fyldigt, har vi brug for begge datasæt, da vi har fundet visse tilfælde, hvor selve

```
HORD:   afviser#sten                OPSL      afviser/sten
HOKL:   no                          UNDERK    c,
HENT:   -en,                        HENV      v. afviser  (2).
HNFL:   afvisersten -ene            ADM       #OPSL (afvisersten)
                                              #DATO 1975/10/01
                                              #BEAR (o)
```

Figur 4: Post fra **RO2** *og* **DFOB**.

a)

```
'afvigelse_n1' = [cat=n, scat=no, scat=no, level=zero,
 dalu='afvigelse', darno=n1,
 ers_frame=f30, dapform1=no, dapform2='fra', dapform3=no, dapform4=no,
 dalsframe=arg12, dapargl=no, daparg2='fra', daparg3=no, dapparg4=no,
 flex_type=fx2, dagd=comm, d_cons=no, oc=yes, infl=root, term=xx0].
%% Coder: anna   21-Apr-89
%% Source: makrotest
%% DEF:
%% Comments:
%% Examples: afvigelser fra regelen (NDO)
```

b)

afvigelse -n, -r.

HORD:	afvigelse
HOKL:	no
HENT:	-n,
HNFL:	-r

c)

afvigelse *c* (*-r*) (= *vigen til side, fjernen sig fra*) (1) déviation* (2) écart (*fra* de, d'avec) (3) (*om magnetnål, ogs.*) déclinaison*; (*astron.*) (4) (*i bane*) perturbation* (5) (*stjerners*) aberration*; (6) (*drt.*) dérogation* (*fra* à). II) (= *forskellighed*) (7) divergence* (8) différence* (9) (*i meninger, ogs.*) dissidence*.
– fra betingelser (*fx. i romburs*) ¶ discordance*; *– fra dagsordenen* (6), incident; *danne en – fra n.* faire dérogation à qc.; *– fra emnet* digression*; *en – mellem ... og ...* un écart entre ... et ..., un écart qui sépare ... de ...; *– fra princippet* (*undert.*) tempérament du principe; *projektils –* (1) (2); *– fra regelen* (1) (2), anomalie*, exception*; (*undert.*) tempérament de la règle; *udvise en – fra* présenter un écart avec.

c)

```
OFSL    afvigelse
UNDERK  c
FLEKS   -r
GLOS    = vigen til side, fjernen sig fra
NUM     1
OVERS   déviation *
NUM     2
OVERS   écart
PART    fra  de, d'avec
NUM     3
GLOS    om magnetnåle, ogs.
OVERS   déclinaison * ;
GLOS    astron.
NUM     4
GLOS    i bane
OVERS   perturbation *
NUM     5
GLOS    stjerners
OVERS   aberration * ;
NUM     6
ENNE    drt.
OVERS   dérogation *
PART    fra  à
BET     =
GLOS    = forskellighed
NUM     7
OVERS   divergence *
NUM     8
OVERS   différence *
NUM     9
GLOS    i meninger, ogs.
OVERS   dissidence * .
DAORDF  xiðx fra  betingelser  (fx. i romburs)
ENNEEDO com.
FRORDF  discordance * ;
DAORDF  xiðx fra  dagsordenen
FRORDF  (6), incident;
DAORDF   danne  en xiðx fra n.
FRORDF  faire dxlxrogation x2x qc.;
DAORDF  xiðx fra  emnet
FRORDF  digression * ;
DAORDF  en xiðx  mellem  ... og...
FRORDF  un xlxcart entre... et..., un xlxcart qui sxlxpare
        ... de ...;
DAORDF  xiðx fra  princípp et
GLOSEDO undert.
FRORDF  tempxlxrament du principe;
DAORDF   p roje ktils  xiðx
FRORDF  (1) (2);
DAORDF  xiðx fra  regelen
FRORDF  (1) (2), anomalie * , exception * ;
GLOSFFO undert.
FRORDF  tempxlxrament de la rxlxgle;
DAORDF   udvise  en xiðx fra
FRORDF  prxlxsenter un xlxcart avec.
ADR     $OPEL (afvigelse)
        $DATO 1975/10/01
        $BEAR (o)
```

Figur 5: Overblik: En lemmaordbogsindgang — og det tilsvarende opslagsord fra de valgte ordbøger.

den søgte oplysning (fx et sammensat substantivs flertalsendelse, eksempelvis ved ordet 'afvisersten') mangler i **DFOB** (se Figur 4).

Om vi vil bruge begge datasæt parallelt, eller bruge **RO2**'s ordklasse- og bøjningsangivelser til at verificere hhv. komplettere oplysningerne vi har hentet fra **DFOB**, har vi endnu ikke taget stilling til.

Figur 5 omfatter foruden en autentisk lemmaordbogsindgang (a), den tilsvarende trykte ordbogsartikel fra Retskrivningsordbogen og ordbogspost fra **RO2** (b), samt ordbogsartiklen fra Dansk-fransk Ordbog og databaseposten — uden typografiske styretegn — fra **DFOB** (c). (Bemærk det specielle tegnsæt i **DFOB**-posten!)

Vi har en del uløste grundlæggende spørgsmål i forbindelse med automatisk udfyldning af lemmaordbogsindgangen, som det fremgår af markeringerne i Figur 5:

- Kan et homografnummer hhv. et betydningsnummer fra **RO2** eller **DFOB** udnyttes til at definere opslagsordets læsninger for oversættelsessystemet, svarende til attributtet *darno* (='Danish reading number');

- Kan valensrammerne (*ers_frame* hhv. *daisframe*) til et ord udledes af de danske ordforbindelser der er anført i artiklen som brugseksempler for ordet og udtrykkes i E-formalisme vha. en konverteringsrutine;

- Kan det antages, at alle obligatoriske valenser er repræsenteret, at de valensbundne præpositioner (*daparg/dapform1–4*) er til stede, at eksemplerne er valgt og listet ud fra et bestemt leksikografisk princip, som kan danne grundlag for den påkrævede (sandsynligvis interaktive) formalisering af oplysninger?

I indeværende fase af pilotprojektet har vi undersøgt materialet fra **DFOB** for at skaffe os overblik over, hvilke muligheder vi har for at løse disse spørgsmål. Materialet indeholder kontekstseksempler for næsten hvert kodet dansk opslagsord. Eksemplerne er ofte sætninger, hvor ordet optræder med alle sine valensbundne led (både med og uden præpositioner). Det tosprogede materiale er kodet udfra et kontrastivt synsvinkel med fokus på danske ord og disses betydningsopdeling (og deres franske ækvivalenter).

Ofte er der knyttet definitioner og kommentarer til ordet, især fx vedr. syntaktisk subkategorisering, semantiske selektionsrestriktioner og pragmatiske forhold. I **DFOB** er disse oplysninger ikke formaliserede. Vi mener dog at kunne hente de manglende oplysninger for kodningen af lemmaordbogsindgange efter at have udfoldet ordbogsartiklen helt, dvs. oprettet en ny artikel for hver nummererede betydning af opslagsordet, hvor ordet automatisk er blevet indsat (eller dets korrekt bøjede form) i stedet for tilde osv.

Vi er i gang med at udfolde artiklerne og vælge de oplysninger fra, som vi ikke vil bearbejde på nuværende tidspunkt. Desuden udarbejdes der en liste over de søgekriterier, samt tabeller mm. der bliver brug for når vi henter og konverterer oplysningerne fra DFOB. I første omgang udføres aktionerne interaktivt.

6 Pilotprojektets næste fase

Det andet trin i pilotprojektet skal koncentrere sig om at undersøge forholdet mellem på den ene side den nuværende manuelle del af kodningsarbejdet i forbindelse med udbygning af lemmaordbogen, og på den anden side den mulige effektivisering der kan påregnes ved overførsel af oplysninger fra maskinlæsbare ordbøger. Ved vurderingen skal der naturligvis også tages hensyn til kvalitetsaspekter.

7 Konklusion

Pilotprojektets emne og formål er bestemt af et konkret behov i **EUROTRA** oversættelsessystemet. Det grundlæggende spørgsmål er, om behovet til dels kan dækkes ved at automatisere arbejdet med eksternt ordbogsmateriale, der kan erhverves i maskinlæsbar form.

Det er ikke uproblematisk at forsøge at sammenføre oplysninger fra maskinlæsbart ordbogsmateriale, der dels er kodet til forskellige formål og ud fra forskellige leksikografiske principper, dels fremtræder teknisk forskelligt, fx hvad tegnsæt (jf. posten fra **DFOB**) eller datastruktur/format angår.

Andre problemer er fx, at de indhentede oplysninger kan være inkompatible indbyrdes eller med **EUROTRA**s system, eller at det maskinlæsbare materiale indeholder sammenlægning af forskellige indholdstyper i ét oplysningsfelt (i **RO2**). Selve selektionen af relevante oplysninger fra et meget omfattende materiale er også forbundet med væsentlige principielle beslutninger.

Disse aspekter kan selvsagt ikke glemmes, kun gemmes, da vi først og fremmest ønsker en generel vurdering af mulighederne for udnyttelse af eksisterende maskinlæsbare ordbogsdata i **EUROTRA-DK**'s ordbøger.

Vi har behov for en ny type ordbog, der indeholder udtømmende, systematiserede og formaliserede oplysninger, der kobler morfologiske, syntaktiske, semantiske og evt. pragmatiske oplysninger sammen til en helhed — den fulde beskrivelse af opslagsordet. Dette vil kunne danne basis for en etsprogsordbog, der kan bruges som kildesprogsordbog ved opbygning af to- eller flersprogsordbøger.

Da den ordbog vi har beskrevet her, omfatter veldefinerede oplysningstyper og formaliserede oplysninger, vil den efter automatisk kontrol og konvertering (der kan indbygges i systemet) også kunne blive velegnet til andre formål.

En tak til:

Bodil Nistrup Madsen, HHK, for at venligst have stillet **DFOB**-materialet til rådighed; Bente Maegaard og Henrik Selsøe Sørensen, EUROTRA-DK, for at deltage i drøftelserne af pilotprojektet, samt Ole Norling-Christensen, Gyldendals Ordbøger, for omhyggelig korrekturlæsning og kommentarer til manuskriptet.

Litteratur

Erik Brun. 1980. *Dansk Sprogbrug.* Gyldendal, København.

Dansk-fransk Ordbog. 1975. [Ved] Andreas Blinkenberg og Poul Høybye. Nyt Nordisk Forlag Arnold Busck, København.

DFOB: Et udvalg af poster fra Dansk-fransk Ordbogsbase i maskinlæsbar form, fra Institut for Datalingvistik ved Handelshøjskolen i København.

Lauterbach, Birgitte. 1988. Dansk-fransk Ordbogsbase, manual. Red: Dansk-fransk Leksikografi. Internt papir ved Handelshøjskolen i København..

Nistrup Madsen, Bodil. 1987. Dansk-fransk Ordbogsbase. Ordbøger i Danmark, En oversigt [Udgivet af DANLEXgruppen]:124–131, København.

Nudansk Ordbog. 1987. [Red.:] Becker-Christensen, Christian m.fl. 13. udgave, 2. oplag. Politikens Forlag, København.

Retskrivningsordbogen. 1988. Udgivet af Dansk Sprognævn, 1. udgave 5. oplag. Gyldendal, København.

Retskrivningsordbogen på edb. 1988. Udgivet af Dansk Sprognævn. København.

RO2. 1988. Den maskinlæsbare version af Retskrivningsordbogens alfabetiske del (1. udgave 5. oplag med rettelser). København, Dansk Sprognævn/Klokker & Bro Aps.

EUROTRA-DK
Københavns Universitet
Njalsgade 80
DK-2300 København S.

STEFÁN BRIEM

Maskinoversættelse fra esperanto til islandsk

Abstract

Machine Translation from Esperanto to Icelandic

It has become evident that Esperanto, due to its logicality and regularity, is well suited as an intermediate language in machine translations between ethnic languages. The project presented here is intended to be a contribution to a later inclusion of Icelandic in a larger multilingual system for machine translation, where Esperanto is used as an intermediate language. In fact such a multilingual machine translation system, DLT (Distributed Language Translation), is being developed at BSO/Research in Utrecht (Netherlands).

My project started in the year 1981. It was a spare time activity exclusively on a private basis until two years ago when it received public support in Iceland. This is a machine translation system applicable as an instrument for studying various ideas relating to machine translation from Esperanto to Icelandic.

An effective machine translation system must be capable of coping with unexpected words, i.a. words created by spontaneous need in the daily use of the language. Those new creations follow rules and conventions that can be different from one language to another. This theme, that involves automatic word creation, is discussed with respect to Esperanto and Icelandic. The derivation of words from words of different word classes by changing the endings of the words, is a common feature of Esperanto and Icelandic. In cases where this happens in a similar and systematic way in both languages, automatic word creation is feasible. A noticeable characteristic of Esperanto is the extensive use of affixes, both prefixes and suffixes, in order to modify the meaning of words and to create new words. In Icelandic affixes are used less frequently. This means that when translating between Esperanto and Icelandic sometimes suffixed words in Esperanto correspond to different word constructions in Icelandic.

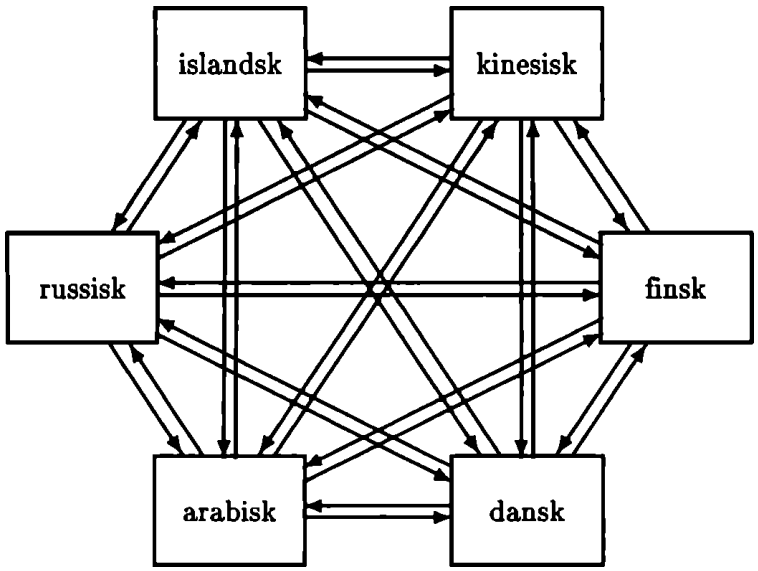
1 Indledning

I de sidste otte år har jeg beskæftiget mig med maskinoversættelse fra esperanto til islandsk, ganske vist ikke uafbrudt. De første 6 år var det udelukkende

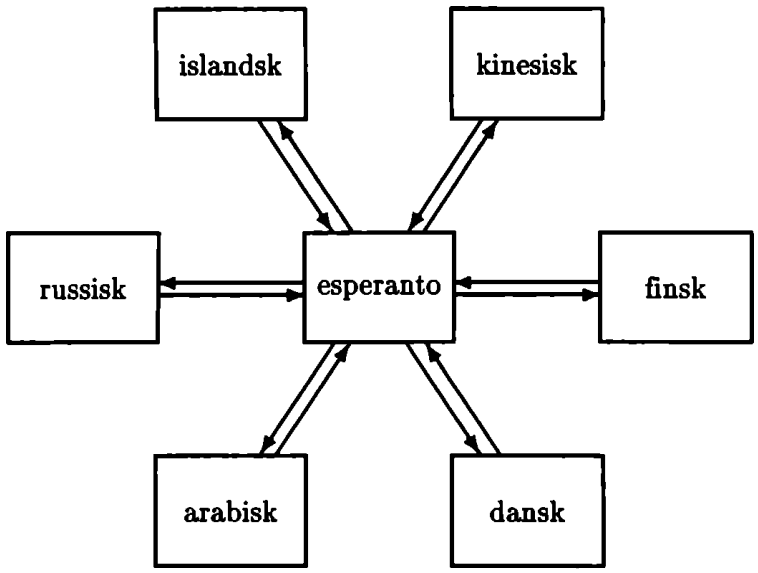
fritidsbeskæftigelse helt på egne vegne. Men i den sidste tid har mit projekt mødt stadig voksende interesse og har fået offentlig støtte.

Jeg vil begynde med at placere mit projekt i større sammenhæng.

Hvis man betragter maskinoversættelse parvis mellem mange sprog så kunne den tænkes at foregå direkte mellem hvert sprogpar.



En anden mulighed er at bruge et fælles mellemsprog, hvor f.eks. esperanto tjener som mellemsprog.



Valget af esperanto som mellemsprog er ikke tilfældigt, men baseres på, at esperanto er et særdeles logisk sprog med regelmæssig grammatik, hvorfor det

er velegnet til datalingvistisk brug. Det er netop denne idé, som jeg har sluttet mig til, da jeg har valgt at beskæftige mig med maskinoversættelse fra esperanto til islandsk. Grunden til at jeg hidtil har begrænset mig til oversættelse i denne retning, men ikke fra islandsk til esperanto, er simpelthen den at parsning af esperanto er meget nemmere end parsning af andre sprog, og der er jo alligevel problemer nok at glæde sig over.

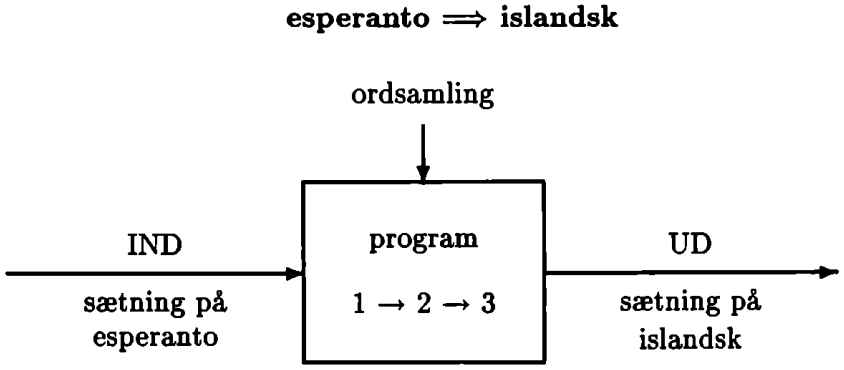
Med henblik på denne grundlæggende idé går mit projekt i takt med det kendte DLT-projekt for multisproglig maskinoversættelse, hvor man også har valgt esperanto som mellemsprog.

I tidens løb har jeg udviklet et oversættelsesprogram sammen med en ordsamling på godt 6.000 ord. Programmet er i stand til at gengive esperanto-tekst på islandsk, således at de islandske ord har den ønskede bøjningsform og sætningernes struktur er ændret i retning af korrekt islandsk syntaks.

Jeg har fundet det nyttigt at have et oversættelsesprogram, som fungerer. Selv om programmet meget lidt berører semantik så kan det bruges til at studere forskellige problemer ved maskinoversættelse, især syntaktiske og morfologiske problemer. Man kan eksperimentere med idéer og mulige løsninger til disse problemer ved at ændre lidt på program og ordsamling og, hvad der kan være meget opmuntrende, hurtigt se et resultat på dataskærm eller på papir.

2 Programmets funktion

Hvordan fungerer oversættelsesprogrammet?



2.1 Etaper:

1. Analyse
2. Oversættelse ord for ord
3. Strukturændringer

Programmet arbejder med én sætning ad gangen og går fra det enkle til det mere komplicerede. Inddata er altså en sætning på esperanto. Programmets første operation er at dele sætningen op i enkelte ord. Den anden operation er at slå op i ordregistre som giver for hvert enkelt esperanto-tekstord det tilsvarende

islandske ord sammen med morfologiske oplysninger. Den tredje operation består af strukturændringer i henhold til forskellen på esperanto og islandsk. Til slut afleverer programmet som uddata en islandsk sætning med næsten islandsk syntaks og forhåbentlig nogenlunde samme mening som den oprindelige sætning på esperanto har.

Det, som jeg især vil diskutere ved denne lejlighed, er hvilke udveje man har når et esperanto-ord ikke findes direkte i ordsamlingen. Løsningen af dette problem indebærer automatisk orddannelse.

3 Ordsamlingens inddeling

På grund af esperantos simple grammatik afdækkes et tekstords ordklasse og bøjningsform på grundlag af ordets endelse alene, uafhængigt af de andre ord i samme sætning. Jeg har derfor fundet det nyttigt at dele ordsamlingen op i fem ordregistre som vist her.

esperanto $\Longrightarrow$ islandsk

		endelser
1.	Substantiver	o *oj* *on* *ojn*
2.	Adjektiver	a *aj* *an* *ajn*
3.	Adverbier	e *en*
4.	Verber	i *as* *is* *os* *us* *u*
5.	Småord	(pronominer, præpositioner, konjunktioner, adverbier, numeralier o.fl.)

Register nummer 5 indeholder en lukket klasse af ca. 200 ord. De andre registre indeholder åbne ordklasser. Hvis et esperanto-ord ikke findes i register 5 så tilhører det med sikkerhed én bestemt af de fire andre registres ordklasser. Såfremt ordet er til stede i ordsamlingen, finder programmet hurtigt frem til det. Men hvad hvis ordet alligevel ikke findes? Og det vil man jo ikke kunne undgå i tilfælde af nydannelser.

4 Automatisk orddannelse

Man kunne selvfølgelig simpelthen give op og lade være med at oversætte ordet. I tilfælde af esperanto kunne man alligevel holde rede på syntaksen, idet ordets ordklasse og bøjningsform er kendt fra endelsen, selv om stammen måske er ukendt. Men der er også den mulighed at ordets stamme eller bestanddele faktisk er til stede selv om ordet ikke findes direkte i registret for den rigtige ordklasse.

Man kan dele de ukendte esperanto-ord i fem tilfælde:

1. Sammensatte ord
2. Ord afledte fra andre ordklasser

3. Ord med affikser
4. Kombinationer af 1–3
5. Andre ord

Af disse tilfælde har jeg især eksperimenteret med 2 til 4 og i den anledning inkluderet nogen orddannelsesregler i mit oversættelsesprogram.

Lad os først betragte de andre tilfælde.

Sammensatte ord er mindre almindelige i esperanto end i islandsk. I de fleste tilfælde kan man nok automatisk oversætte et sammensat ord direkte fra esperanto til islandsk, d.v.s. man finder frem til de enkelte stammers oversættelser og danner af dem et sammensat ord på islandsk.

Hvad 5. tilfælde angår, *andre ord*, så drejer det sig om ord som ikke kan oversættes i de omgivelser. Man kan simpelthen gengive ordet uden oversættelse i den islandske tekst. Man kan jo alligevel benytte sig af kendskabet til ordets ordklasse og bøjningsform til at bestemme den islandske sætnings struktur.

4.1 Afledning fra en anden ordklasse

Jeg vil nu vise et par eksempler på automatisk orddannelse ved afledning fra andre ordklasser.

I esperanto er denne type af orddannelse meget almindelig, meget mere end i islandsk og dansk f.eks.

Eksempel 1. adverbium dannet af adjektiv

adjektiv:	feliĉa	hamingjusamur	lykkelig
adverbium:	feliĉe	hamingjusamlega	lykkeligt

Når programmet skal oversætte adverbiet *feliĉe* til islandsk, så vil dette ord ikke findes i samlingen af adverbier. Programmet vil da søge i samlingen af adjektiver for adjektivet *feliĉa*, som jo findes der. Resultatet bliver det islandske adverbium *hamingjusamlega* afledt af det islandske adjektiv *hamingjusamur*.

Det andet eksempel drejer sig om et adjektiv, eller snarere om et participium, som behandles som et adjektiv.

Eksempel 2. adjektiv (participium) dannet af verbum

verbum:	kuri	hlaupa	løbe
adjektiv:	kuranta	hlaupandi	løbende

Ordet *kuranta* findes ikke i samlingen af adjektiver. Da ordet ender på *anta* er det muligvis præsens participium af et verbum. Programmet vil derfor søge i samlingen af verber for verbet *kuri*. Resultatet er positivt og programmet danner automatisk ordet *hlaupandi* ved afledning af det islandske verbum *hlaupa*.

De to eksempler kan også kombineres og blive til en dobbeltafledning.

Eksempel 2 + 1. adverbium dannet af verbum

adverbium:	kurante	hlaupandi	løbende

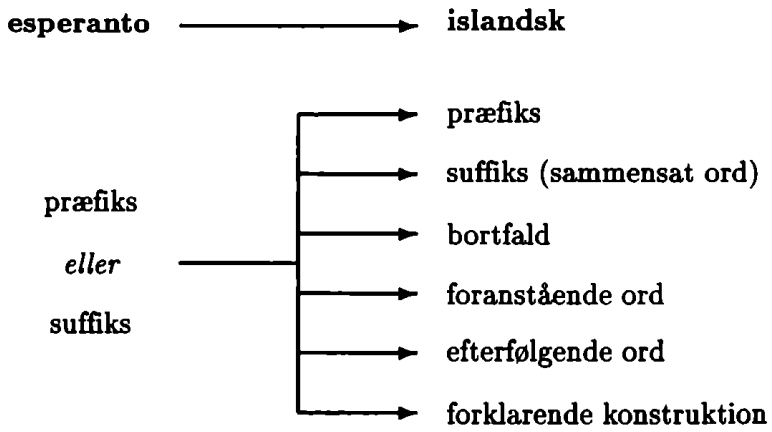
Adverbiet *kurante* oversættes ved at programmet først søger forgæves efter adjektivet *kuranta* og derefter finder frem til verbet *kuri*.

4.2 Esperanto-ord med affikser

En anden type af orddannelser på esperanto går ud på udstrakt brug af affikser, både præfikser og suffikser.

Når esperanto-ord med affikser skal kunne oversættes til islandsk, kan de mest almindelige med fordel inkluderes i ordsamlingerne.

For de andre affiksede ords vedkommende kan man prøve at vælge en af de oversættelsesmetoder som antydes nedenfor.



Både i tilfælde af et præfiks og et suffiks kommer alle disse muligheder i betragtning. Man kan vise eksempler på alle mulighederne, måske med én undtagelse. Om lidt vil vi se nogle eksempler.

Men først skal vi se på en særpræget brug af præfikser på esperanto. Det drejer sig om konstruktionstvillinger, hvor en præposition kan placeres som præfiks på et verbum med ingen eller minimal ændring af mening.

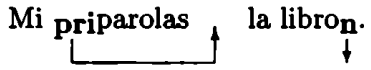
4.2.1 Strukturændring på esperanto

esperanto:	Mi parolas **pri** la libro.
islandsk:	Ég tala um bókina.
dansk:	Jeg taler om bogen.

Den samme mening kan ytres ved sætningen:

esperanto:	Mi **pri**parolas la libron.

Når den sidste sætning skal oversættes til islandsk bliver dens struktur først ændret ved at præfikset **pri** laves om til præpositionen **pri**, der placeres efter verbet, samtidig med at det akkusative **n** fjernes fra substantivet.



4.2.2 Oversættelse af præfikser

Her får vi eksempler på 5 muligheder for oversættelse af ord med præfikser.

1.	**miskompreni**	**misskilja**	**misforstå**
2.	**senlabora**	atvinnulaus	arbejdsløs
3.	kamaradoj	félagar	kammerater
	gekamaradoj	félagar	kammerater (af begge køn)
4.	**eksministro**	**fyrrverandi** ráðherra	**forhenværende** minister
5.	**foriri**	fara **burt**	gå **bort**

4.2.3 Oversættelse af suffikser

Og nu over til 5 eksempler på oversættelse af ord med suffikser.

1.	leono	ljón	løve
	leonino	**kvenljón**	**hunløve**
2.	grupo	hópur	gruppe
	grupestro	**hópstjóri**	gruppe**bestyrer**
3.	paroli	tala	tale
	paroladi	tala	tale (vedvarende)
4.	homo	maður	menneske
	homaĉo	**lítilfjörlegur** maður	**usselt** menneske
5.	skribi	skrifa	skrive
	skribema	**hneigður fyrir að** skrifa	**tilbøjelig til at** skrive

4.3 Kombinationer

Nu har vi betragtet nogle eksempler på tilfælde nr. 2 og 3 af ukendte esperanto-ord. Ifølge tilfælde nr. 4 kan man også have kombinationer af 1, 2 og 3, og den slags kombinationer er endog ret almindelige i esperanto. Når ord af denne art skal oversættes, kan det være afgørende for resultatets kvalitet i hvilken rækkefølge ordets struktur analyseres. Den mest gunstige rækkefølge vil selvfølgelig ikke blive den samme for de forskellige ordklasser. I denne henseende er det min filosofi, at man ved hjælp af et simpelt oversættelsesprogram som mit kan afprøve de forskellige muligheder på en naturlig tekst og derved finde frem til den optimale rækkefølge for hver ordklasse.

5 Forøgelse af ordsamlingen

Den begrænsede størrelse af ordsamling virker ofte utilstrækkelig, når man vil eksperimentere med en naturlig tekst. Derfor arbejder jeg nu på en forøgelse af antal ord fra godt 6.000 til knap 24.000, d.v.s. næsten firedobling. Dette sideprojekt, som nu er i den sidste fase, består i ud fra en temmelig stor islandsk-esperanto ordbog (Skaftfell 1965) i trykt form at fremstille en esperanto-islandsk ordsamling i datamatlæsbar form. En esperanto-islandsk ordbog af tilsvarende omfang ville selvfølgelig have været nemmere at arbejde med, men den eksisterer desværre ikke. Konverteringen er tilvejebragt til dels ad mekanisk vej, idet den består af følgende etaper:

- Optisk karakterrekognition (ved hjælp af Kurzweil-maskine)
- Maskinstøttet konvertering fra islandsk-esperanto til esperanto-islandsk
- Manuel bearbejdelse

Litteratur

Briem, Stefán. 1988. *Vélrænar tungumálaþýðingar.* Projektrapport. Reykjavík.

Skaftfell, Baldvin B. 1965. *Íslenzk-esperanto orðabók.* Samband íslenzkra esperantista. Reykjavík.

BOEL VICTORIA BØGGILD-ANDERSEN

Valence Frames Used for Syntactic Disambiguation in the EUROTRA-DK Model

Abstract

The EEC Machine Translation Programme EUROTRA is a multilingual, transfer-based, module-structured machine translation project. The result of the analysis, the interface structure, is based on a dependency grammar combined with a frame theory. The valency frames, specified in the lexicon, enable the grammar to analyse or generate the sentences. If information about the syntactical structure of the slot fillers is added to the lexicon, certain erroneous analyses may be discarded exclusively on a syntactical basis, and complex transfer may in some cases be avoided. Where semantic and syntactical differences are related, problems of ambiguity may be solved as well. This will be exemplified, and the frame theory will be explained. The paper concentrates on the valency of verbs; according to the EUROTRA theory the verb is the governor of a sentence.

1 The EUROTRA Model

The structure of the system as a whole is as shown in figure 1.
Units at the different levels:

EBL: Signs, codes.

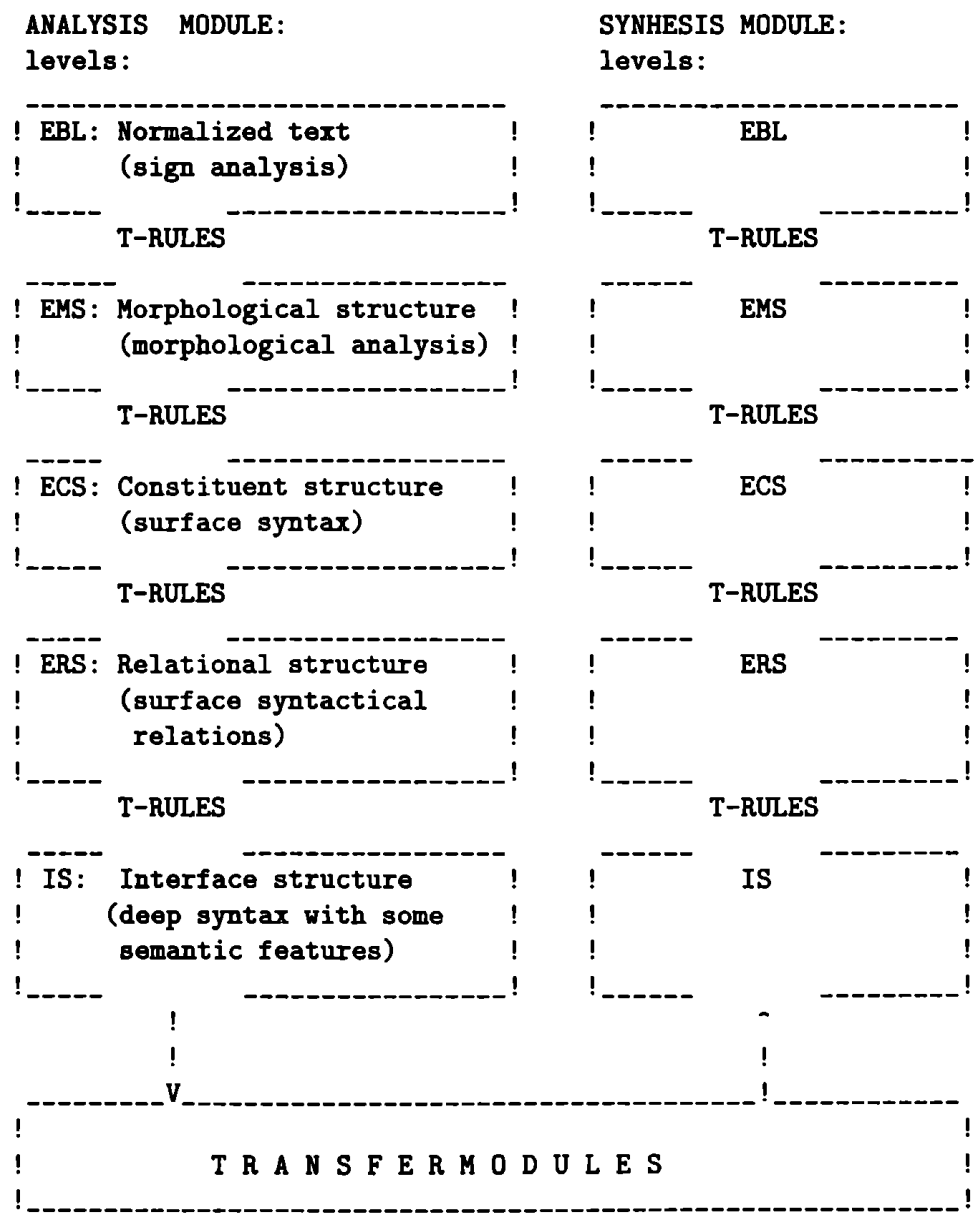
EMS: Morphemes.

ECS: Words, syntactical categories, phrasal categories.

STRUCTURE: Constituent structure: indicates the natural sequence of the sentence constituents by means of syntactical categories and subcategories.

ERS: Syntactical functions (surface syntactical relations): Governor, subject, object etc., modifiers.

Fig.1. The modular structure of the EUROTRA system.



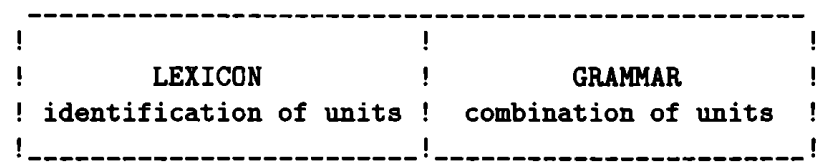
STRUCTURE: Dependence structure: canonic sequence. Indicates the surface syntactical relations, determines agreements and percolates features. Sequence of units: governor, subj, obj/obl_ag/attr_subj/ pobj/obl_go/obl_loc/compl, iobj(obj2)/pobj/compl/obl_go/obl_loc/ attr_obj, modifiers. (Abbreviations are explained in fig.4.; the source is the EUROTRA Reference Manual version 5.0, 1989. A later version was distributed after the symposion.)

IS: Deep syntactical relations; governors, arguments and modifiers with some semantic features.

STRUCTURE: Dependence structure: canonic sequence. Indicates the depth syntactical relations. Elevates certain simple entities by percolating them to the relevant mother node as feature bundles. Sequence: governor, argument1, argument2, argument3, argument4, modifiers.

Each level consists of two components, a lexicon and a grammar. The system identifies its units at a certain level by using the lexicon, where at the same time the information about the various lexical entries, necessary for the grammatical rules at each level to function, is drawn.

Fig.2. The structure of a level.



Between the levels the T-RULES ensure the correct transformation of the output from one level to the form necessary as input for the following level.

As it will appear, the sign analysis occurs at the EBL level, the morphological analysis at the EMS level and the syntactical analysis at the three following levels. Some semantical information is included in the IS level. We shall concentrate on the last two syntactical levels, the ERS and the IS levels, and we shall pose the question: What information must be present at the lexical entries at these levels for the grammar and thus the system to function? In order to determine this, we must look more closely at the task of these two levels in the total analysis process.

2 The ERS Theory

On many points the ERS theory is in agreement with the theory behind the f-structure in LFG (Lexical Functional Grammar). Our grammar at the ERS level is a dependence grammar. A dependence structure consists of a governing lexical unit (GOVERNOR) and (possibly) a number of sentence members (DEPENDENTS), which presuppose the presence of the GOVERNOR. We distinguish between two types of DEPENDENTS:

Fig.3. Dependents at the ERS level.

1: COMPLEMENTS, which fill out a place in the frame of the governor, i.e. are frame bound or valency bound: The GOVERNOR requires their presence.

2: MODIFIERS, which also presuppose the presence of the GOVERNOR but do not fill out a place in its frame, are not required by the GOVERNOR.

At ERS the complements of the main verb are constituted by the following syntactical functions, from which, however, the Danish implementation differs on certain points:

Fig.4. Verbal complements at the ERS level.

SUBJ.
Example: DA: PETER sover.
ENG: PETER sleeps

OBL_AG (the subject in passive clauses).
Example: DA: Suppe spises AF MANGE.
ENG: Soup is eaten BY A LOT OF PEOPLE.

OBJ (the indirect object, if this is not a sentence).
Example: DA: Peter spiser SUPPE.
ENG: Petes eats SOUP.

ATTR_SUBJ (the subject complement if this is not a sentence).
Example: DA: Problemet er ULOESELIGT.
ENG: The problem is INSOLUBLE.

ATTR_OBJ (the object complement if this is not a sentence).
Example: DA: Det kalder jeg EN OVERDRIVELSE
ENG: That's what I call AN OVERSTATEMENT.

POBJ (frame bound prepositional phrase i.e. prepositional phrase governed by the main verb and requiring a particular preposition).
Example: DA: Jeg haaber PAA EN FORANDRING.
ENG: I am hoping FOR A CHANGE.

OBL_LOC (obligatory prepositional phrase denoting place).
Example: DA: Peter bor I SLAGELSE.
ENG: Peter lives IN MANCHESTER.

```
OBL_GO     (obligatory prepositional phrase
denoting direction).
Example: DA:  Peter tog TIL PARIS.
         ENG: Peter went TO PARIS.

COMP (clause-shaped complement).
Example: DA:  Peter har lovet, AT HAN NOK SKAL KOMME.
         DA:  Peter hoerte HANS KOMME.
         ENG: Peter has promised, THAT HE WILL BE
              THERE.
         ENG: Peter heard HANS COMING.

OBJ2 (the direct object, if this is not a
sentence).
Example: DA:  Jeg skylder HAM en tjeneste.
         ENG: I owe HIM a favour.
```

For the sake of clarity I have chosen sentences which do not bear much resemblance to the constructions that we usually work with in texts from the Commission.

From LFG we have also taken over the so-called "Principle of Completeness and Coherence", which can be formulated as follows:

> A structure must contain ALL the complements required by its governor AND NO OTHERS.

From this principle follows:

> Complements can only be described as obligatory. Empty elements (empty nodes) must be inserted in a number of cases, for instance in passive clauses where the logical subject (the agent) is not present, and in infinite constructions without an explicit subject directly connected to the infinite verb form.

The ERS guidelines are of a directive character, and not obligatory; in the Danish implementation we have differed from them for example by not including the category COMP, which, as we saw it, was defined not by its syntactical function, but exclusively by its being clause-shaped. In the Danish implementation we have simply allowed that subjects as well as objects may also consist of substantival sub-clauses or infinitives. This simplifies the mapping between the ERS and IS levels, and besides it agrees more with our linguistic intuition, also because precisely the same applies in Danish to words and phrases governed by a preposition. (for example in a so-called POBJ); in this position we may also find both nouns, noun clauses and infinitives.

3 The IS Theory

The next level, the IS level, is, however, actually legislative. It is here that the decorated tree structures, which constitute the starting point of the synthesis AFTER the transfer process, are formulated. Since, as it is well known, EUROTRA is a multi-lingual translation system, it is necessary to model the IS level in such a way that, using a common feature theory, it describes the linguistic features that are relevant for translation purposes and which are common to the languages in question, in a way that is compatible with all nine EEC languages. To formulate this theory and the common feature theory is a difficult task — some might even say an impossible one — but it is also a challenge, because no theories existed that might be transferred to a multi-lingual, transfer-based machine translation system. The IS theory has been formulated in a cooperation between linguists in EUROTRA with constant feedback from the various language groups, and it rests on the following main principles:

1. IS is primarily a syntactical theory.

2. The starting-point of the description is English, which functions as a kind of meta-language.

3. The IS theory consists of a dependence grammar with a sunken governor, combined with a frame theory.

4. The theory must satisfy the following requirements:

   (a) The description must be adequate; it must, as far as possible, disambiguate polysemantic surface structures.

   (b) The description must be calculable; it must be formalized so as to permit a computer to calculate the relevant phenomena.

As mentioned above, the theory must be able to describe the linguistic features, relevant for translation purposes and common to the nine EEC languages; hence all non-significant differences are neutralized. This applies to the individual language (for example the difference between the active and the passive voice) as well as to differences between the EEC languages. An example of a difference specific to a particular language is the difference between noun clause types (infinitive constructions or that sentences). If one wants to specify this difference in the analysis, it must be done on the underlying level, the ERS level. It is a monolingual matter, whether a verb requires finite or non-finite clausal complements. Some verbs do not take clausal complements at all, others take special types, and some (the support verbs) require deverbal nouns as objects, while the equivalents in other languages may not have the same restrictions. I shall show an example indicating these differences in the next section.

The neutralization of differences, specific to a particular language, is done in the following way:

As mentioned earlier, already at the ERS level a "euroversal", canonic sequence of sentence members is determined. Thus word orders, specific to a particular language, are neutralized. At the IS level this sequence describes a small, defined, number of depth syntactical relations between the members.

Certain sub-systems (tense, aspect, modality, etc.) are removed from the actual structural representation and re-coded, attributed to the overall sentence complex by calculation.

Certain surface phenomena are removed from the tree structure and are represented instead in the overall sentence complex as features, if they are relevant for the translation.

As has been mentioned, the IS structure is also a dependence structure, including a governor and two types of dependence relations. These are:

ARGUMENTS, of which a maximum of four may occur. An argument number may only be indicated, if the preceding one also occurs in the frame of the governor

MODIFIERS, which do not occur in the frame of the governor.

Thus, the maximum completion of a sentence, in which the main verb is always regarded as the governor, is as follows (the Kleene star indicates zero, one, or more occurences of the subsequent member):
S = GOV, ARG1, ARG2, ARG3, ARG4, *MOD.

The relation between the complements of the ERS level and the arguments of the IS level can be schematically described as follows:

```
SUBJ ------------- ARG1
OBL_AG ----------- ARG1

OBJ -------------- ARG2
ATTR_SUBJ -------- ARG2

COMP ------------- ARG2 eller ARG3
POBJ ------------- ARG2 eller ARG3
OBL_GO ----------- ARG2 eller ARG3
OBL_LOC ---------- ARG2 eller ARG3

ATTR_OBJ --------- ARG3
OBJ2 ------------- ARG3

Frame-bound arguments (PP's) not otherwise
indexed ---------- ARG4
```

So, both at the ERS level and at the IS level it is necessary to specify the valence structure of the lexical units in the level specific lexicons. And where can

information about the valence of Danish words be obtained? Generally speaking, possibilities are few; no actual valence dictionaries for Danish exist, as they do for German for example. The Danish EUROTRA group has to work out these dictionaries themselves.

4 The Coding of Verbs in the Lexicon

In the Danish implementation we indicate the valence of the words at the ERS level already in the lexicon for the ECS level. We shall see how this can be done in the case of the verbs, taking a concrete example.

The Danish verb BEMAERKE (notice, remark) is mono-transitive. the lexical entry to this verb may for instance look like this in the ECS dictionary:

```
'bemaerke_v1' = cat=v, ers_frame=f20, ctrl=no, dalu='bemaerke',
reflex=no, vfeat=nstat, auxlu='have', t=no, term=xx0.
```

The formula ers_frame=f20 refers to the sentence rule that describes sentences with mono-transitive verbs. There are, however, different meanings of this verb. If we follow the definitions in Nudansk Ordbog, which, in Denmark, comes closest to the medium-sized, monolingual dictionary used for the project, we can make a division into these entries, where only the definitions differ. The examples are coded in the format used for the Lemma dictionary, where information necessary for the different levels are gathered under the relevant entries:

```
'bemaerke_v1' = cat=v, scat=mainv, level=zero, dalu='bemaerke',
darno=v1, ers_frame=f20, dapform1=no, dapform2=no, dapform3=no,
dapform4=no, daisframe=arg12, reflex=no, daparg1=no, daparg2=no,
daparg3=no, daparg4=no, auxlu='have', vfeat=nstat, flex_type=fx1,
dcons=no, oc=yes, infl=root, term=xx0.
%% Coder: boel  16-Jun-89
%% Source: experiment
%% DEF: iagttage, laegge maerke til
%% Comments:
%% Examples:Ingen bemaerkede hans fravaerelse. NDO.

'bemaerke_v2' = cat=v, scat=mainv, level=zero, dalu='bemaerke',
darno=v2, ers_frame=f20, dapform1=no, dapform2=no, dapform3=no,
dapform4=no, daisframe=arg12, reflex=no, daparg1=no, daparg2=no,
daparg3=no, daparg4=no, auxlu='have', vfeat=nstat, flex_type=fx1,
dcons=no, oc=yes, infl=root, term=xx0.
%% Coder: boel  16-Jun-89
%% Source: experiment
%% DEF: udtale, ytre
%% Comments: Ambiguous example in the NDO.
%% Examples: Han bemaerkede, at han var forhindret. NDO
```

Here, we use f20 in both cases: The verb only takes an object as a complement. Dalu means Danish lexical unit, darno refers to Danish reading number. DEF means definition. Information preceded by %% is not relevant for the grammar.

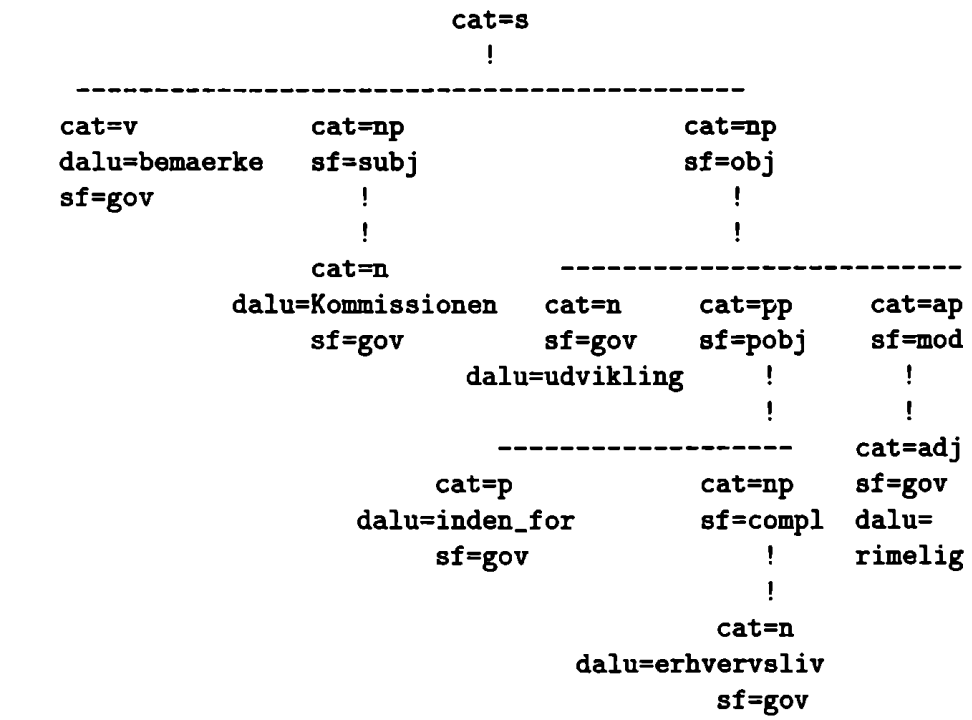
According to this, there must be two different entries in the lexicon, where the coding is identical, but the definitions differ. Hence, an analysis of the sentence:

DA: Kommissionen har bemaerket en rimelig udvikling inden for erhvervslivet.

ENG: The Commision has noticed a reasonable development in industry. (bemaerke=notice, sense 1, bemaerke_v1 above)

produces two identical results at the ERS level, both of this form:

Fig.5. ERS object with no object differentiation.



If, however, we supplement our lexical entries with the information that BEMAERKE in sense 1 may have certain types of sentence objects (an NP, an at-clause or an interrogative clause), while in sense 2 (bemaerke_v2 above) the word only takes at-clauses or pronouns as the object, the entries will look like this:

```
'bemaerke_v1' = cat=v, scat=mainv, level=zero, dalu='bemaerke',
darno=v1, ers_frame=f244, dapform1=no, dapform2=no, dapform3=no,
dapform4=no, daisframe=arg12, reflex=no, daparg1=no, daparg2=no,
daparg3=no, daparg4=no, auxlu='have', vfeat=nstat, flex_type=fx1,
dcons=no, oc=yes, infl=root, term=xx0.
```

```
%% Coder: boel  16-Jun-89
%% Source: experiment
%% DEF: iagttage, laegge maerke til
%% Comments:
%% Examples:Ingen bemaerkede hans fravaerelse. NDO.
```

f244: The verb only takes an object as complement. The object is: an NP OR a nominal at-clause OR an interrogative clause.

```
'bemaerke_v2' = cat=v, scat=mainv, level=zero, dalu='bemaerke',
darno=v2, ers_frame=f262, dapform1=no, dapform2=no, dapform3=no,
dapform4=no, daisframe=arg12, reflex=no, daparg1=no, daparg2=no,
daparg3=no, daparg4=no, auxlu='have', vfeat=nstat, flex_type=fx1,
dcons=no, oc=yes, infl=root, term=xx0.
%% Coder: boel  16-Jun-89
%% Source: experiment
%% DEF: udtale, ytre
%% Comments: Ambiguous example in the NDO.
%% Examples: Han bemaerkede, at han var forhindret. NDO
```

f262: The verb only takes an object as complement. The object is: a nominal at-clause OR a pronoun.

Furthermore, we change and sub-divide our grammar rules on the basis of this information. As a result, only one analysis, using the first entry of BEMAERKE, is possible, and the number of analyses are reduced. We are thus able to discard certain erroneous analyses exclusively on a syntactical basis, because it turns out, that semantic and syntactical differences may be connected. And this must happen at the ERS level, this being the level where a distinction is still made between different object types and sub-clause types. At the IS level this specific distinction is neutralized.

The following transfer rules will ensure the correct translation:

```
1: cat=v, dalu=bemaerke, darno=v1 =>
   cat=v, enlu=notice, enrno=v1.

2: cat=v, dalu=bemaerke, darno=v2 =>
   cat=v, enlu=remark, enrno=v2.
```

5 Conclusion

As claimed above, problems of semantic differences may in some cases be related to syntactical differences. In these cases, more systematic use of the syntactical information may solve some, if certainly not all, semantic problems of machine translation.

We have also made it theoretically possible to GENERATE or produce the correct sentence structure for a sentence translated into Danish, precisely by

including information about sentence structure in the lexicon, contained in the more specific indication of valence. What forms the basis of the translation into Danish is a tree structure, where the individual words make up the leaves on the tree. In the transfer process, the words of the source language are exchanged with the equivalent words of the target language. In case of sentential complements, finite or non-finite, the tree does not specify which sentence type makes up the object of the sentence. The Danish lexicon will contain the information about the syntactical combinations of the different verbal entries, which make possible the establishment of a more specific sentence structure. In these cases we avoid having to work out word-specific rules of complex transfer; the problem can be solved monolingually in a more general way.

The fact remains that we lack dictionaries in Danish that allow us to draw information about the combination possibilities of Danish words to an extent that suits our purpose. One of the many tasks that the Danish EUROTRA group faces is to produce such dictionaries. I hope to get the possibility to experiment with drawing this information from the Gyldendal dictionary: "Dansk Sprogbrug" by Erik Bruun. Here we find examples of the use of Danish words and a typology comparable to a rough valency description. We shall have to complete this information and transform it to a formalism suited for this special purpose.

References

Bresnan, Joan (ed): *The Mental Representation of Grammatical Relations.* MIT press. Cambridge and London 1982.

Boeggild-Andersen, Boel Victoria: *Forslag til udvidelse af verbalkodningen i den monolingvale ordbog.* Intern Rapport. 15. sept. 1988. EUROTRA-DK, Copenhagen.

Boeggild-Andersen, Boel Victoria: *Diderichsens saetningsskema anvendt ved 'parsing' i EUROTRA.* Lecture from the Danish EUROTRA seminar at the University of Copenhagen 1988. In press.

Boeggild-Andersen, Boel Victoria: *Verbale komplementer og argumenter i EUROTRA-teorien og deres adskillelse fra modificerende led.* Lecture from the Danish EUROTRA seminar at the University of Copenhagen 1989.

Boeggild-Andersen, Boel, Hanne Fersoe, Lise D. Johansen and Patrizia Paggio: *Sentential complements and non-finite clauses.* PO-22 report, EUROTRA 1989.

Bruun, Erik: *Dansk Sprogbrug.* Gyldendal. Copenhagen 1978.

Nudansk ordbog. Politikens forlag. Copenhagen 1989.

The EUROTRA reference manual 5.0, draft version. 1989.

Quirk, Randolph, Sidney Greenbaum, Geoffrey Leech and Jan Svartvik: *A Grammar of Contemporary English.* Longman. London 1972, 1986.

HANNE FERSØE

Representational Issues within Eurotra

Abstract

Machine translation of subordinate clauses, whether finite or non-finite, poses a series of problems within representational issues. In **EUROTRA** this subject has recently been investigated for complements as well as for modifiers to verbal, nominal, adjectival, and prepositional heads.

Some of the problems encountered during these investigations will be addressed in this paper, with a particular focus on the representation of finite and non-finite modifiers.

The claim is made that neutralization of surface differences is necessary. This neutralization of the surface structure is shown to be a prerequisite to simple transfer. The concept of simple transfer is briefly introduced.

Various example representations are shown and discussed.

1 Introduction

Machine translation of subordinate or dependent clauses, whether finite or non-finite, poses a series of problems and challenges within representational issues. In **EUROTRA** this subject has recently been investigated for all types of sentential complements and modifiers, finite as well as non-finite, to verbal, nominal, adjectival, and prepositional heads. (PO-22, 1989)

This paper presents some of the results of the subset of the work, which has been carried out by the Danish language group, and within this, in particular the special research topic dealt with by my colleague Patrizia Paggio and myself.

We have been particularly interested in the definition of a representation which neutralizes arbitrary surface variation found in dependent clauses in the nine EEC languages. Of the many possible structures encountered in the above mentioned syntactic functions we have primarily concentrated on finite and non-finite adverbial modifiers to verbs and nouns. The specifications, which we have proposed as a result of our work, are now part of the Eurotra Reference Manual, which has 'legislative' status in the project. (R.M. 6.0).

In the following quite a few examples of such structures will be shown and discussed. The following list of examples serves to clarify, which types of constructions are involved.

Complements to verbal head, non-finite

Industrien prøver AT OVERTAGE FINANSIERINGEN

La Commission veut RENVERSER CETTE TENDANCE

Complements to verbal head, finite

EF ønsker, AT INDUSTRIEN SKAL OVERTAGE FINANSIERINGEN

The Commission wonders, WHETHER INDUSTRY WILL TAKE OVER

Modifiers to verbal head, non-finite

EVERYBODY HAVING AGREED ON THE DATE, the meeting was adjourned

AL SER CONSCIENTE DEL PROBLEMA, la Comisión formuló una propuesta

Modifiers to verbal head, finite

EFTER AT ALLE VAR BLEVET ENIGE OM DATOEN, blev mødet hævet

DA DEN BLEV OPMÆRKSOM PÅ PROBLEMET, formulerede Kommissionen et forslag

ALTHOUGH LABOUR IS EXPENSIVE Europe employs many people

Modifiers to nominal head, non-finite

The actions TAKEN by industry have not reversed the trend

Modifiers to nominal head, finite

de forholdsregler, SOM INDUSTRIEN HAR TAGET, har ikke vendt tendensen

2 Simple Transfer and IS

One very fundamental issue in the **EUROTRA** translation model is the concept of simple transfer. Poul Andersen has given an introduction to this concept, its implications, and provided exemplification in Andersen (1989).

Simple transfer is achieved through a very deep analysis of the input string such that the 'meaning' of e.g. a clause is represented as an annotated tree structure. Simple transfer then consists of substituting the lexical units of the source language tree, also called the **IS** representation, with the appropriate lexical units of the target language. For the translation to be successful, however, many conditions have to be fulfilled, one of them is successful lexical disambiguation, which is a non-trivial problem. The condition in focus here, though, is not lexical, but representational. Simple transfer requires that the annotated tree structure must be transferred unmodified.

The **EUROTRA IS** theory attempts to give a detailed definition of the IS representation from which simple transfer is to take place. One important aspect

of the structures defined at IS is that arbitrary surface differences, which exist between the languages, should be neutralized precisely so that structural transfer needs not take place. Structural transfer means modification of the source language IS structure during transfer to the target language IS structure.

Such surface differences can be observed in the examples listed above. They show examples of both noun and sentence modifiers, where a direct structural mapping from one language to another, i.e. simple transfer, is not possible. Our investigations have shown evidence of great variation w.r.t. surface realization of the so called sentential complements and modifiers across the nine EEC languages.

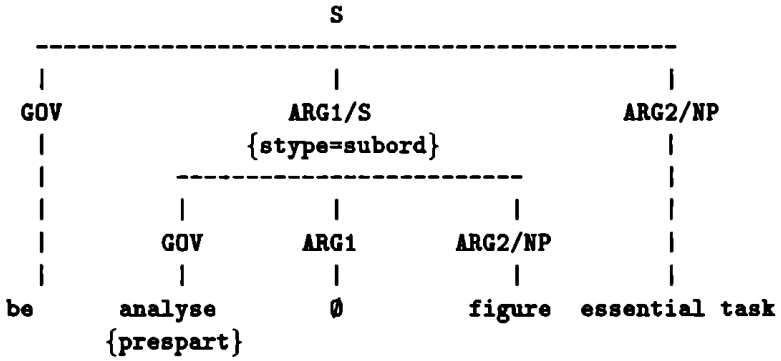
Up until recently the **EUROTRA IS** theory had not been extended to cover all these phenomena. This means that no so called 'legislation' was available for them, except for infinitives and finite clauses in object position to verbs. The filling of such legislative holes, i.e. development, extension, and refinement of the IS theory, is contributed to by the language groups through their experiments with possible representations. Consequently quite different structures have so far been implemented to support these constructions by the different language groups.

3 Example Representations

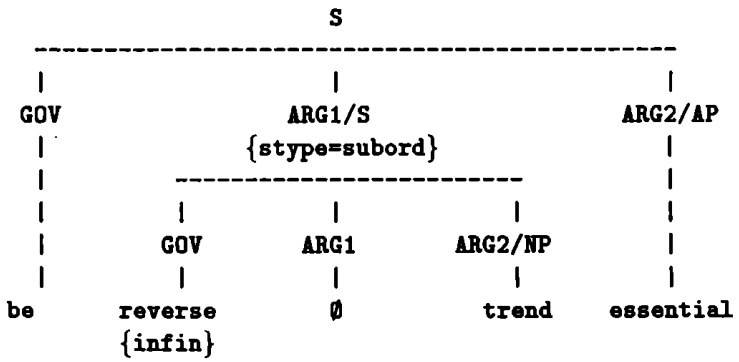
In the following a series of example structures for the clauses in question will be discussed. These examples are authentic, i.e. they have been produced by the different analysis modules developed within **EUROTRA**. The structures, however, have been simplified here for the purpose of highlighting the issue in question. Some of the examples have already been listed above, but here they will be discussed in more detail. The following labels are used in the representations:

```
GOV   - governor, head of a construction
ARG1  - argument 1, deep subject
ARG2  - argument 2, deep object
ARG3  - argument 3, indirect object
MOD   - modifier
TRANC - transconstructional
SBAR  - dependent clause
S     - sentence
NP    - noun phrase
AP    - adjective phrase
PP    - prepositional phrase
∅     - empty element
{..}  - annotations, features, decorations
```

(1) ANALYSING FIGURES was an essential task



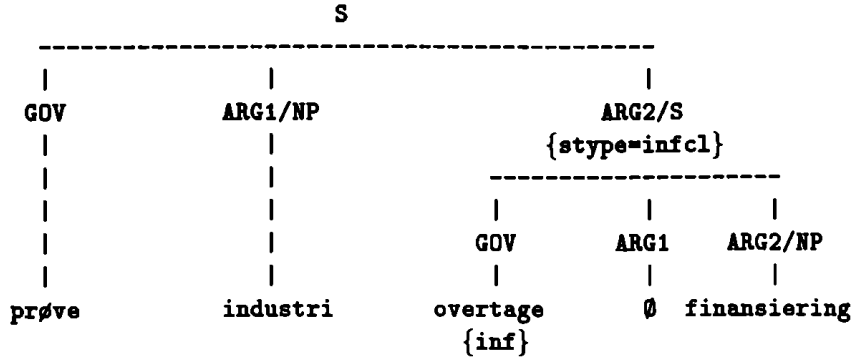
(2) TO REVERSE THE TREND is essential



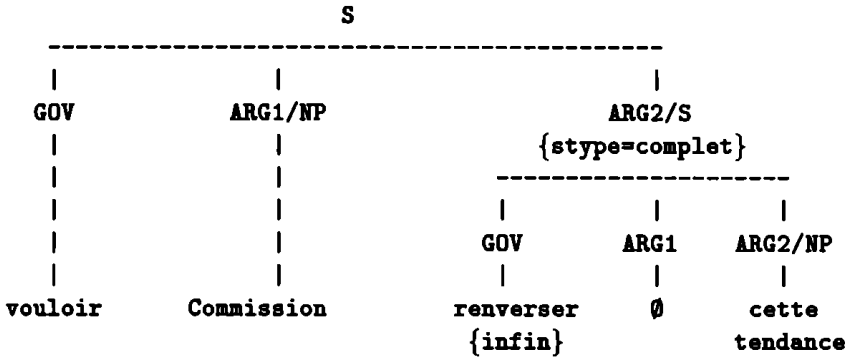
In both (1) and (2) the ARG1's, although they are internally different (-ing clause and infinitive clause), have been assigned the same structure, S, they only differ in morphological typing of the verbal head.

In the following three examples, (3), (4), and (5), the ARG2's are all infinitive constructions, and they are all represented as S structures with an empty element inserted as the logical subject of the infinitive. Again we see, that they differ in features, but not in structure.

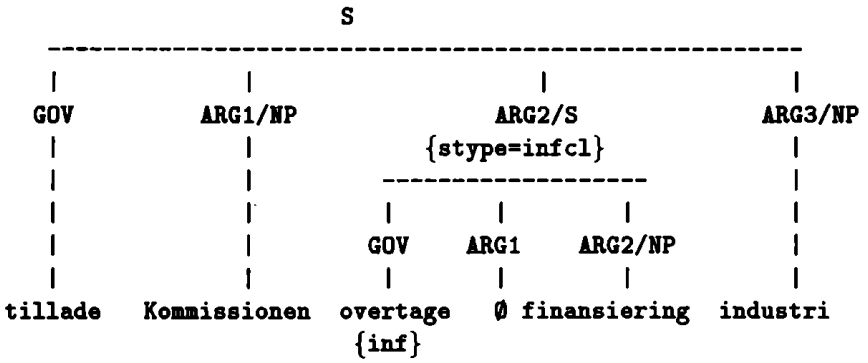
(3) Industrien prøver AT OVERTAGE FINANSIERINGEN



(4) La Commission veut RENVERSER CETTE TENDANCE



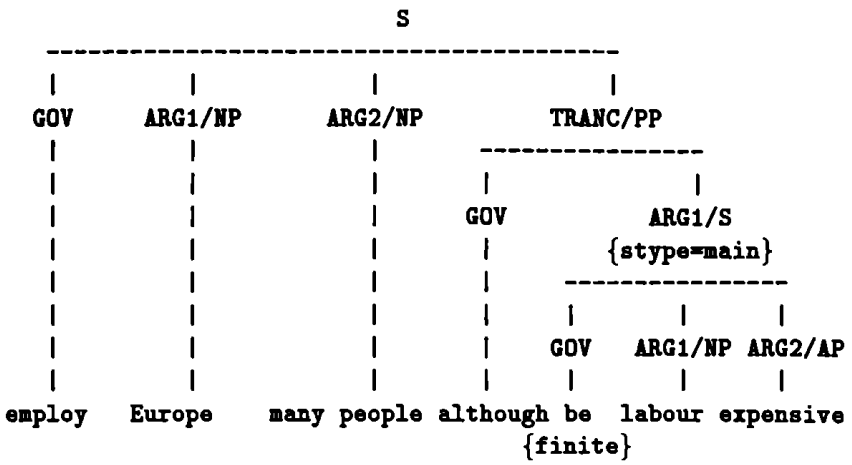
(5) Kommissionen tillader industrien AT OVERTAGE FINANSIERINGEN



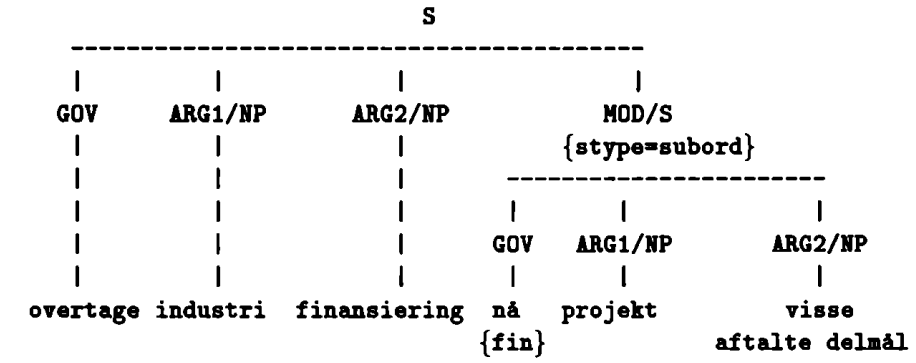
The preceding five examples all show complements—subjects or objects. Their structural representation is in accordance with the IS legislation, which already existed before our investigations started, i.e. S with an inserted empty element in the non-finite clauses.

The three following examples, (6), (7), and (8) show representations of finite adverbial modifiers to a verbal head.

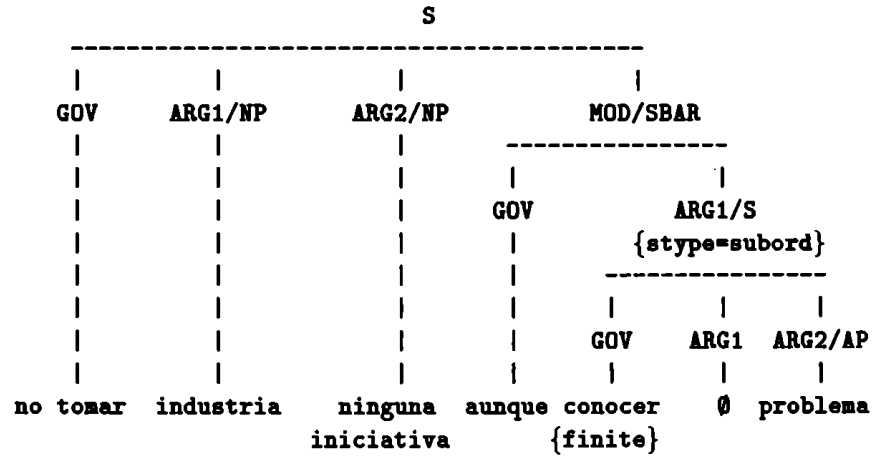
(6) ALTHOUGH LABOUR IS EXPENSIVE Europe employs many people



(7) Industrien overtager finansieringen, EFTER AT PROJEKTET HAR NÅET VISSE AFTALTE DELMÅL



(8) La industria no tomó ninguna iniciativa, AUNQUE EL PROBLEMA ERA CONOCIDO



As the examples show there is great variation w.r.t to the structures and the labelling chosen for these modifiers.

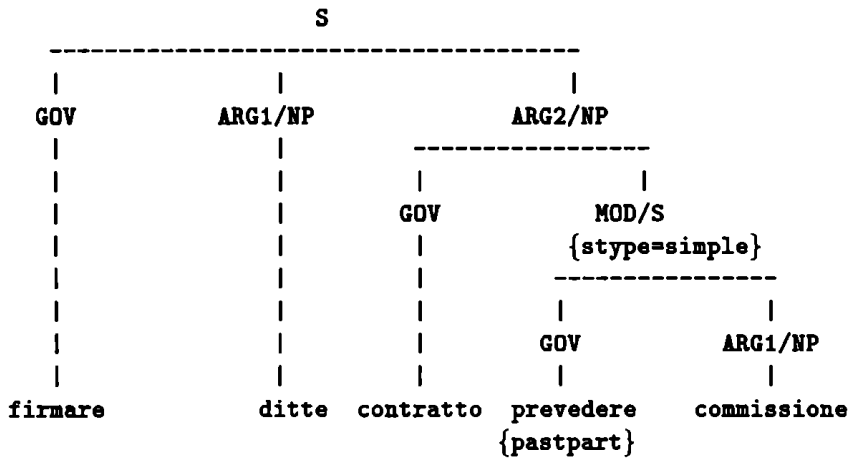
The English analysis, (6), shows the modifier labelled TRANC assigned the category PP. This approach makes the subordinating conjunction, here called a preposition, the head or governor of the entire modifier.

A similar structural approach is chosen in the Spanish analysis (8), but the labelling is different. MOD is chosen to label the modifier, but the category is SBAR with a preposition as governor of the clause. Linguistically it does not seem particularly well motivated to have a preposition as the head of a clause. The head role to clauses is usually reserved for verbs.

The Danish analysis, (7), is more flat, the subordinate clause is attached directly to the main clause, and the subordinating conjunction is elevated, thus leaving the governor position of the sentence to the verb. Here it is not quite clear, how the conjunction should be retrieved.

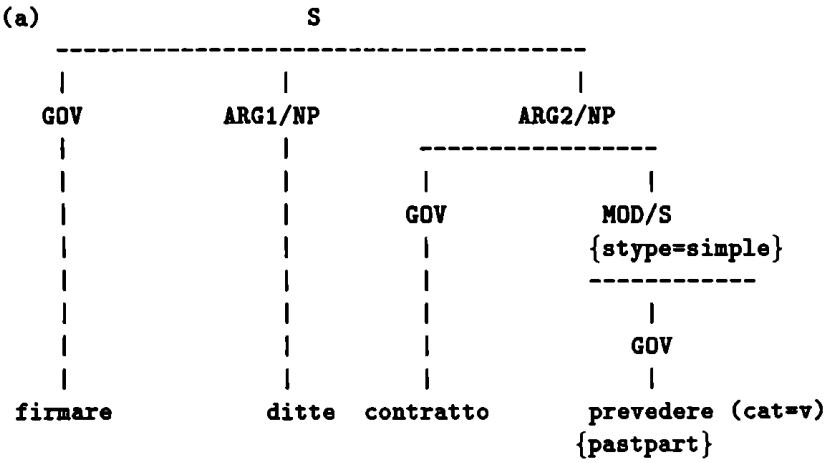
The following five examples show dependents of nominal heads.

(9) Le ditte firmeranno il contratto PREVISTO DALLA COMMISSIONE

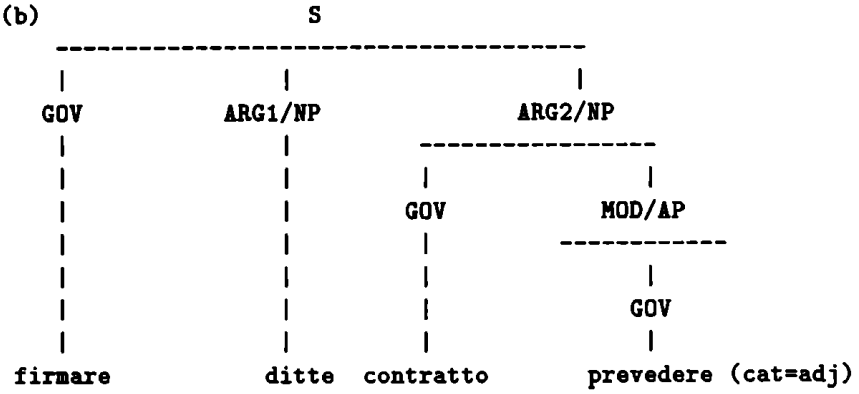


(10) Le ditte firmeranno il PREVISTO contratto

(a)

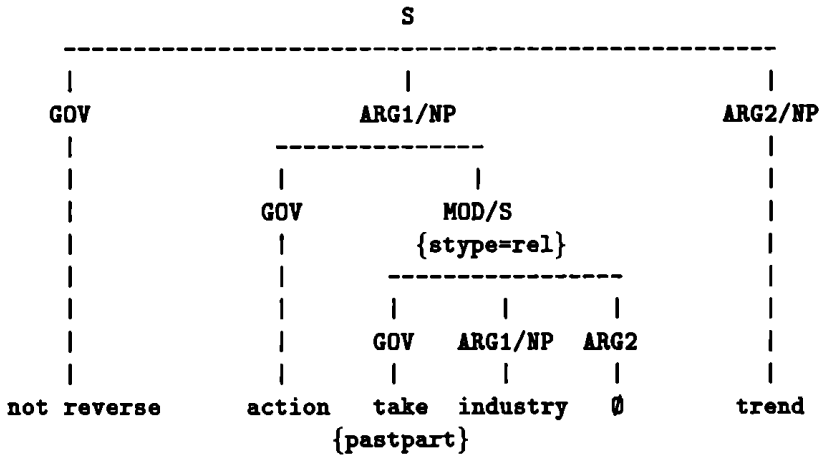


(b)

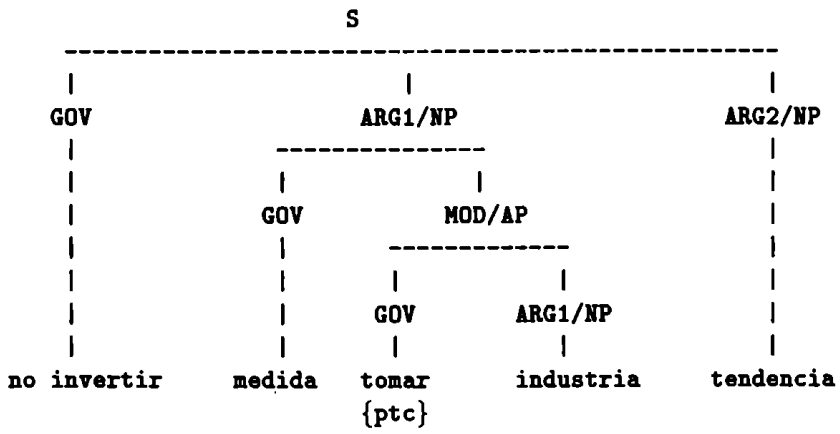


The most interesting observation about these Italian examples is that when the participle has an explicit by-object, as 'dalla commissione' in (9), it is unambiguously considered an S, whereas (10), where no explicit by-object is present, is considered ambiguous between S and AP, and thus the grammar yields two results. This, of course, is unacceptable, since ultimately only one translation should be produced.

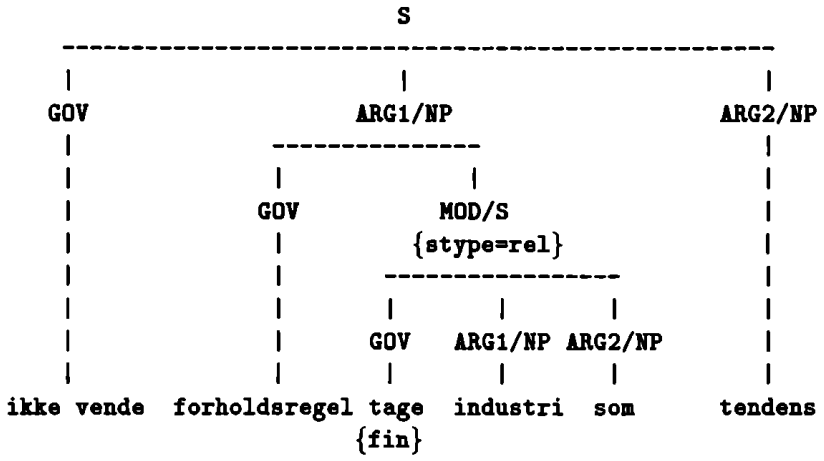
(11) The actions TAKEN BY INDUSTRY have not reversed the trend



(12) Las medidas TOMADAS POR LA INDUSTRIA no invirtieron la tendencia



(13) De forholdsregler, SOM INDUSTRIEN HAR TAGET, har ikke vendt tendensen



(13) as compared to (11) and (12) shows that for Danish a finite relative clause is required to express the meaning conveyed by the non-finite participle in English and Spanish.

A number of observations have been made here concerning the variations in structure and labelling of these representations. One observation, which holds for all of them though, is that the structure assignment is linked to the category assignment to the topmost node of the subordinate clause. The IS theory specifies what constitutes an S, an NP, an AP etc., and therefore the decision taken about the category of the top node implies the structure of the local tree. The following table lists our observations in short form.

3.1 Complements

Category assignment to top node of finites as well as non-finites: **cat=s**.

Stype assignment to top node:

stype=subord, **stype=complet**, **stype=infcl**.

Category assignment to head:

cat=v.

Morphological verbform assignment to head: **prespart**, **infin**, **inf**

Not surprisingly, the category assignment and therefore the structure is identical for all the complements—they follow the IS specifications. The feature differences, however, do cause some problems. Similar problems arise in connection with the modifiers, which will be in focus in the remainder of this paper.

3.2 Modifiers

Category assignment to top node:	
non-finite sentence modifiers:	**cat=s, cat=sbar, cat=pp**
non-finite np modifiers:	**cat=s, cat=ap**
finite modifiers, all:	**cat=s**
Stype assignment to top node:	
sentence modifiers:	**stype=main, stype=subord**
np modifiers:	**stype=simple, stype=rel**
Category assignment to head:	
sentence modifiers:	**cat=v**
np modifiers:	**cat=v, cat=ptc, cat=adj**
Morphological verbform assignment to head:	
sentence modifiers:	**fin, finite, infin**
np modifiers:	**fin, pastpart**

All these representational and classificational differences reflect a purely monolingual analysis underlying the representation provided by the different analysis modules. What we have aimed at in our work is a multilingual analysis as a point of departure for a euroversal definition of the representation.

We have investigated the translational behaviour of a number of these constructions between a number of EEC languages. Not surprisingly, it turned out that a one-to-one mapping of category, morphological verbform and structure is not possible. Therefore a neutralization of structures was necessary, and we proposed cat=S with its implications as the neutral structure. This proposal has now been integrated into the IS 'legislation'.

4 A Neutral IS Structure

4.1 Non-finite Sentence Modifiers

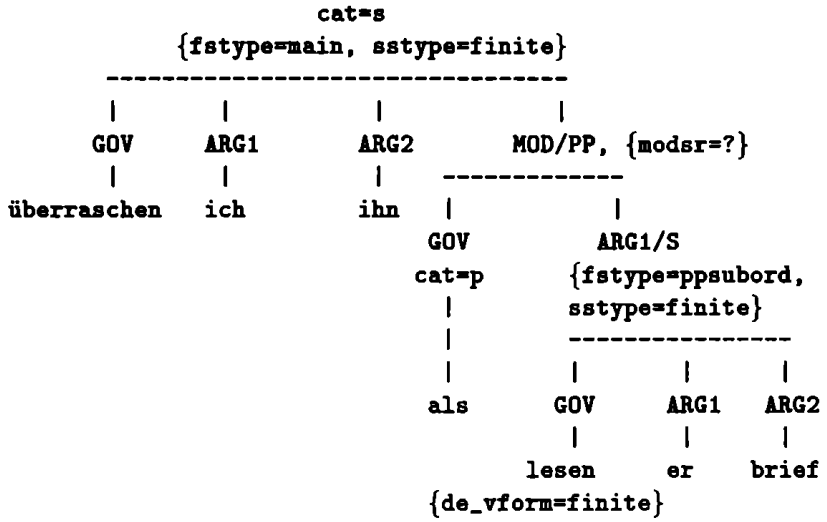
These are not possible in all languages. In the languages which have them, they can be expressed in gerunds, participles or infinitives. In the languages which do not have them, their equivalents are finite clauses preceded by a presentential particle, traditionally classified as either a preposition or a subordinating conjunction.

The annotated tree structures below are a bit more decorated than the structures (1) through (13). Detailed motivation for the presence of this additional information exceeds the aim of this paper. As a basic key to the additional features we can say that **sstype** (surface stype) contains information about the finite- or non-finiteness of the clause, **fstype** (functional stype) contains information about the location in the tree of a given subordinate clause, e.g. the value 'ssubord' is assigned to all ARGi/S and MOD/S daughters to S nodes, **modsr** ('semantic relation of modifier to head') with no values calculated in the

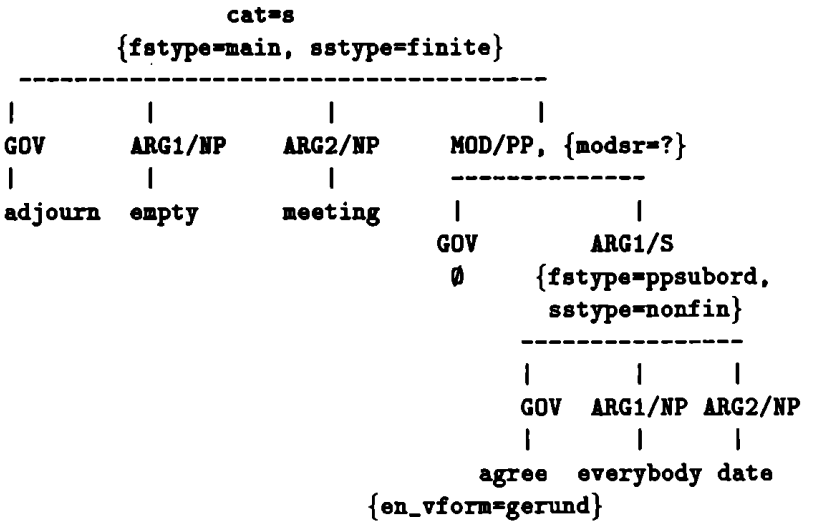
examples, and **V_form** (morphological verb form; the variable, V, is a language prefix, e.g. da) which contains information about verbal inflection, e.g. finite, infin, prespart, etc.

Finite and non-finite sentence modifiers, like in (14) and (15) below, will be represented in the following way:

(14) Ich habe ihn überrascht, ALS ER DEN BRIEF LAS



(15) EVERYBODY HAVING AGREED ON THE DATE, the meeting was adjourned

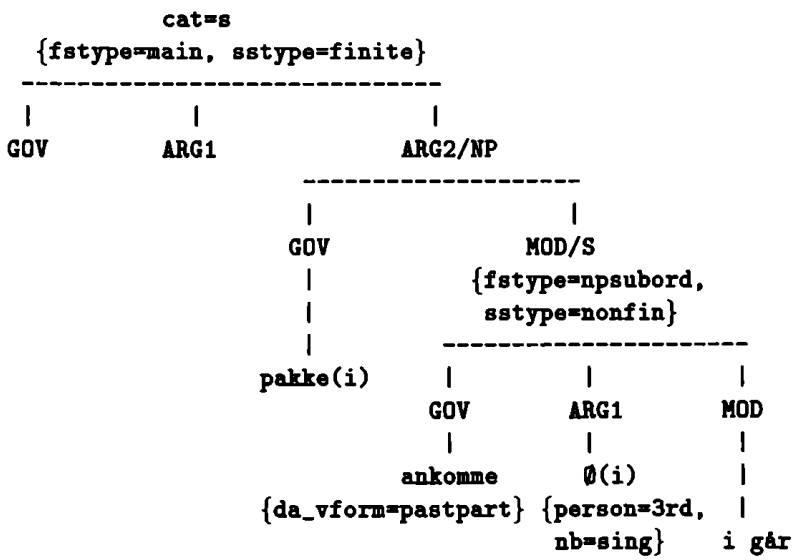


The motivation for this PP structure is translational in that it captures the finite constructions, which in some languages are the only possibility, and which are alternative possibilities in the languages which also have the non-finite constructions. Note in particular the introduction of the 'empty' governor in the structural representation of non-finite sentence modifiers.

4.2 Non-finite NP Modifiers

All the languages included in the investigation can express sentential np modification through non-finite participial constructions or finite relative clauses. But there are restrictions concerning the mappings between languages. Some languages will only accept a relative clause, where other languages will also accept a participle. Relative clauses are always acceptable as a sort of default structure, so the non-finite np modifiers will be represented as an S structure, as in (16) below:

(16) en pakke ANKOMMET i går



The **EUROTRA IS** theory supposes a full GOV - ARG representation for sentences with an obligatory ARG1 (deep subject). This implies insertion of empty nodes in non-finite constructions and a coindexation mechanism as well. The aim of this paper does not permit me to go into details about this interesting aspect of the representation, but we have dealt with it in our research report, and it is an integrated part of our proposal.

5 Concluding Remarks

A surface-neutral structural representation, as shown above, is far from sufficient to ensure successful simple transfer. As shown in the series of trees and the tables earlier, feature assignment problems also have to be solved.

Essentially what is needed to solve these problems is a semantic classification system for subordinate clauses, which is sufficiently fine grained to permit the target language to generate the appropriate surface representation—finite or non-finite—from the input, regardless of the source language. The information, which must be computable from the semantic values, concerns time, tense, mood,

modality, and diathesis for the sentence as such, plus information for the correct translation of presentential particle.

Unfortunately such a system does not exist—yet.

References

Andersen, Poul: How Close Can We Get to the Ideal of Simple Transfer in Multi-lingual Machine Translation (MT)?, This volume:103–113, Reykjavík, 1990.

PO-22: Sentential Complements and Non-finite Constructions. EUROTRA Research Report, April 1989, unpublished. Collaborative effort of EUROTRA-DK, GB, DE, IT, and EL.

The most relevant of the references herein is:

R. Quirk & S. Greenbaum. 1972. A Grammar of Contemporary English, London and New York.

R.M.6.0: *Eurotra Reference Manual version 6.0*, Luxembourg, August 1989, unpublished.

EUROTRA-DK
Københavns Universitet
Njalsgade 80
2300 København S
Danmark

BARBARA GAWROŃSKA-WERNGREN

Identifiering av diskursrefenter vid maskinöversättning från ryska till svenska

Abstract

The problem of tracking discourse referents in the process of machine translation which will be discussed in this paper is a part of my work at SWETRA (Swedish Computer Translation Research) at the Dept of Linguistics, Lund. The SWETRA-programs, implemented in Prolog and based on a GPSG-inspired formalism called Referent Grammar, abbreviated RG (Sigurd 1987, 1988), allow translation of a large repertoire of syntactic constructions between Russian, Swedish and English. Currently, the research is concentrated on translation from Russian into Swedish. One of the difficulties arising in the process of translating from a Slavic language into a Germanic one is inserting correct definiteness values in the noun phrases, as e.g. Russian and Polish do not make use of definiteness as a regular grammatic category. Generating appropriate definiteness values and their morphological representations in the target language is a very complicated task, as the choice between definite and indefinite NPs depends not only on endophoric (textual), but also on many exophoric (external) factors. The most general rules for use of definite noun phrases in Swedish and in English are, nevertheless, accessible for implementation in Prolog.

Our approach to MT requires tracking discourse referents and the paper will also discuss some problems connected with this approach.

1 Begreppen diskursreferent och koreferens

Innan vi presenterar den (preliminära) algoritmen som används i SWETRA-program för att identifiera diskursreferenter och välja nominalfrasernas bestämdhetsvärde, vill vi kortfattat beskriva den diskursmodell, som proceduren bygger på och försöka definiera själva begreppet "diskursreferent". Informellt brukar diskursreferenter karakteriseras som "saker och fakta som man talar om", vilket implicerar en exoforisk (icke-textuell) relation — man refererar till objekt i den icke-språkliga verkligheten. Definitionen av begreppet diskursreferent måste dock samtidigt vara relaterad till texten (det gäller bl a att ägna uppmärksamhet

åt sättet att introducera nya diskursreferenter i texten och till de lingvistiska faktorer som möjliggör associering av två fraser med samma diskursreferent). Den preliminära modellen för identifiering av diskursreferenter som kommer att presenteras nedan innehåller därför både endoforisk och exoforisk komponent; i den senare spelar begreppet "mental verklighet"(dvs en värld av kognitiva enheter och relationer mellan dem) en primär roll. Innan vi övergår till den referent-grammatiska diskursmodellen, vill vi kortfattad kommentera en klassisk definition av sättet att introducera diskursreferenter formulerad av Karttunen (Karttunen 1976):

> the appearance of an indefinite noun phrase establishes a discourse referent just in case it justifies the occurence of a coreferential pronoun or a definite noun phrase later in the text.

Ovanstående definition kan inte tillämpas i SWETRA:s översättningsprocedur, eftersom distinktionen mellan indefinita och indefinita nominalfraser är (vid översättning från ryska eller polska) omarkerad i inputtexten. Man måste dessutom ta hänsyn till det faktum, att en ny diskursreferent kan introduceras inte enbart av en NP, utan också av t ex ett verb eller en hel mening — som i (1):

(1a) Idag förföljde en oidentifierad ubåt en svensk trålare .
(1b) Jakten pågick i ungefär en timme.

eller av ett adjektiv — som i (2):

(2a) Köp inte gula blommor till henne.
(2b) Den färgen tycker hon inte om.

Den diskursmodell, som vi vill föreslå här, tar hänsyn till olika sätt att introducera nya diskursreferenter — bl a till dem som illustrerades i (1) och (2). Den preliminära modellen innehåller fyra nivåer (som i sin tur kan indelas i subnivåer; nedanstående beskrivning är i viss mån förenklad):

nivå 1: Texten.

nivå 2: Mentala koncept som uppstår på basis av lingvistisk kunskap (dvs kunskap om ordens intensioner och extensioner och förmågan att tolka syntaktiska strukturer); dessa koncept vill vi preliminärt indela i tre huvudgrupper — s k nominaliserare (nominalizers), egenskaper och relationer; skillnaden mellan dessa begrepp kan (förenklat) förklaras i predikatskalkylens termer: "relationer" och "egenskaper" kan jämföras med predikat, och "nominaliserare" med argument. Varje enhet på denna nivå kan associeras med flera ord eller ordsekvenser i texten.

nivå 3: Diskursreferenter, dvs kognitiva enheter som konstitueras på basis av de slutsatser som kan dras med utgångspunkt från de egenskaper hos nominaliserare och de relationer mellan nominaliserare vilka representeras på nivå 2. Slutsatserna kan vara baserade antingen på lingvistiska eller icke-lingvistiska kunskaper. En diskursreferent kan associeras med flera enheter på nivå 2 (koreferens).

nivå 4: Objekt och fakta i den "verkliga världen"; denna nivå behöver inte vara representerad i alla diskurstyper, eftersom objektets existens respektive icke-existens i den fysiska verkligheten inte påverkar möjligheten att konstituera diskursreferenter som i ett av exemplen i Frarud (Frarud 1986):

(3a) We don't have a dog.
(3b) He would only fight with the cat.

I enlighet med den ovan skisserade modellen uppstår koreferens om två (eller fler) mentala koncept framkallade av ord eller ordsekvenser kan associeras med samma kognitiva enhet på en högre nivå — en föreställning av t ex ett objekt, ett faktum, en egenskap etc. Förutsättningar för koreferens uppstår således på nivå 2 (begrepp baserade på rent lingvistisk kunskap); i fortsättningen kommer vi dock att — för att förenkla procedurbeskrivningen — ibland använda formuleringar som "koreferenta ord", "koreferenta substantiv" etc — i stället för den mer exakta, men alltför långa frasen: "ord associerade med koreferenta begrepp".

Ett av villkoren för potentiell koreferens (som stora delar av vår komputationella modell bygger på) kan formuleras på följande sätt: två koncept (C1 och C2) framkallade av ord eller ordsekvenser i texten (T1 och T2) kan i regel associeras med samma referent R, om det senare introducerade konceptet (C2) är inte ojämförbar med C1 och om C2 inte är mera specifikt än C1. Med "mera specifikt" menar vi följande relation: C1 är mera specifikt än C2, om föreställningen "att vara C1" implicerar (direkt eller indirekt) "att också vara C2". Resonemanget kan illustreras med exempel (4) och (5):

(4a) Jag träffade min granne.
(4b) Mannen var berusad.

Nominalfraserna *min granne* och *mannen* (eller, mera exakt, de "nominaliserare" som associeras med dem — låt oss kalla dem för N1 och N2) uppfattas som koreferenta eftersom det stereotypa konceptet av "granne" kan representeras som "en människa som bor nära talaren/den tidigare nämnda personen" (vi bortser för eventuell metaforisk användning av ordet i fråga). Att vara någons granne implicerar normallt att vara en manniska, vilket i sin tur implicerar att vara antingen en man eller en kvinna — N2 är alltså mindre specifik än N1. Däremot i en text som:

(5a) Jag träffade en man.
(5b) Min granne var berusad.

upplevs nominalfraser *en man* och *min granne* som icke-koreferenta, eftersom konceptet "min granne" är mera specifikt än "en man".

Distinktionen mer/mindre specifik är naturligtvis inte alltid lika klar; exempel (6) — ett fragment av SWETRA-översättning av en rysk tidningstext — visar ett mera problematiskt fall:

(6a) Israeliska flygplan utförde idag tre bombattacker över libanesiskt territorium.
(6b) Femton personer dödades som resultat av luftpiraternas barbariska aktion.

Det faktum, att begreppen "israeliska flygplan" och "luftpiraterna" interpreteras som koreferenta, kan möjligan förklaras på följande sätt: C1 – som kan representeras som `israeli(N1)`, där N1 motsvarar begreppet "flygplan" (begreppet inkluderar inte bara flygplan som maskiner men också deras "animata delar" – piloter) implicerar — i enlighet med de värderingar som är aktuella i sändarens kultur vissa negativa egenskaper (att vara en israelisk pilot kan innebära — med en hög sannolikhetsgrad — att vara också en luftpirat, en bandit osv). Vid en sådan interpretation skulle det ovan formulerade villkoret för möjlig koreferens vara uppfyllt. Ett försök att med hjälp av en komputationell procedur identifiera koreferens i liknande fall kommer att presenteras i avsnitt 2.1.

2 En preliminär modell för identifiering av diskursreferenter

SWETRA-program för översättning mellan vissa slaviska och germanska språk är — som det har påpekats tidigare — baserade på en GPSG-inspirerad formalism — referentgrammatik (Referent Grammar, RG; Sigurd 1987, 1988). Möjligheter att använda RG för parsing och översättning av vissa polska och ryska syntaktiska konstruktioner har beskrivits i Gawrońska-Werngren (1988) och i Sigurd & Gawrońska-Werngren (1988).

Proceduren som för närvarande används vid översättning från ryska till svenska kan indelas i följande tre huvudstadier:

1. parsing av input-meningen och formulering av en sorts interlinguarepresentation, s k funktionell representation (f-representation), som innehållller information om meningens syntaktiska struktur (uttryckt med hjälp av sådana traditionella termer som subjekt, objekt, predikat, adverbial och satsadverbial), ordens betydelsekoder (formulerade i "maskinengelska") samt koder för vissa grammatiska drag, som numerus, genus, värdet +/- animat, tempus osv. T ex en enkel rysk mening som

mal'čik	bežal	domoj
pojke	sprang	hem

   får efter analysen följande f-representation (förenklat):

   ```
   s(subj(np(r(_,m(boy,sg),D,sg,ani,ma,_,_),H,Rel)),
     pred(run,past),
     sadvl([]),sadvl([]),advl(home),advl([]),advl([])).
   ```

   Symbolen [] (tomma mängden) avspeglar det faktum att meningen inte innehåller några satsadverbial och inga fler adverbial än *domoj* (home). Prolog-atomen med funktorn `r` brukades i några tidigare skrifter om RG kallas för "referentbeskrivning" (referent deskription), vilket naturligtvis måste betraktas som en approximation; denna enhet innehåller snarare

en beskrivning av nominalfrasens huvudord, som under översättningsprocessens nästa etapp möjliggör identifiering av diskursreferenter. I fortsättningen kommer den delen av nominalfrasens representation (ofta betecknad med symbolen R) kallas för "referent nucleus". Variabeln D — bestämdhet — förblir oinstantierad under den första fasen av översättningsproceduren. Variablerna H och Rel används för att lagra information om eventuella attribut: Rel kan innehålla en funktionell representation av en relativsats, medan koder för övriga pre- och postnominala attribut placeras i enheten H.

2. betydelsekoder för syntaktiska konstituenter lagras i listor; därefter börjar sökning efter eventuella koreferenta fraser och analys av den tidigare textuella informationen; procedurens mål är att instantiera variabeln D som "def" (+definit) i de fall, då textuella faktorer implicerar bestämdhet. Meningens funktionella representation lagras också i databasen.

3. generering av inputmeningens ekvivalent i målspråket. På stadiet B sker en omformulering av meningens f-representation till en PROLOG-lista — en enkel transitiv sats får då följande form:

```
[subj(X),pred(Y),obj(Z),sadvl(S1),sadvl(S2),advl(A1),
                    advl(A2),advl(A3)]
```

Därefter jämförs varje nominal konstituent med tidigare översatta substantiv, verb och attribut. De tidigare översatta ordens betydelsekoder är samlade i två separata listor — listan som innehåller substantivens och verbens betydelsekoder kommer i fortsättningen att kallas för R-listan, listan i vilken attributiva bestämningar placeras — för A-listan. Skälet för den sortens indelning är icke-lingvistiskt - sökningsproceduren verkar helt enkelt fungera mera effektivt, om huvudordens koder samlas i en separat lista. Sökning efter koreferenta konstituenter och eventuella bestämdhetsindicer genomförs med hjälp av en rekursiv procedur, som terminerar när den funktionella representationen inte innehåller flera nominella konstituenter. Det ryska lexikonet som inkluderar en viss (mycket förenklad) information om ordens semantiska kategoritillhörighet är tillgängligt under denna del av proceduren.

I början av översättningsprocessen är både R- och A-listan tomma. Proceduren som används för att lagra den information, som senare kan utnyttjas för identifiering av diskursreferenter, kan beskrivas på följande sätt:

1 Är subjektsplatsen i den funktionella representationen tom? (subjektet representeras som en tom mängd t ex vid analys av ryska opersonliga konstruktioner av typen:

ubito pjat' čelovek — 'fem människor dödades'.
döda fem människor
+unpers

Om ja, gå över till f-representationens nästa konstituent (predikatet), lagra dess betydelsekod i R-listan och fortsätt till nästa konstituent

— tills en konstituent som innehåller en NP påträffas. Efter att ha hittat den första NP, gå över till 2. Om subjektsplatsen inte är tom, gäller det också att gå över till 2.

2 Kontrollera först, om enheten H innehåller några attributkoder. Om alla platser i H är tomma, placera huvudordets betydelsekod i R-listan och förse den med numret N1 = N + 1, där N = antalet element i R-listan. Gå över till 5. Om några attribut förekommer, gå över till 3.

3 Kontrollera om platsen avsedd för räkneord innehåller en konstant, t ex 2. Om inte, gå över till 4. Om ja, ändra huvudordets betydelsekod från `m(X,pl)` till `m(X,2)` och placera den i R-listan (med lämpligt indexnummer). Denna del av proceduren används för att möjliggöra generering av bestämd form i texter av typen: *Två pojkar sprang. Den ene var liten. Den andre* ... Gå över till 4.

4 Innehåller den sista platsen i enheten `H` en kod av typen `m(X,prop)`, dvs ett egennamn? (appositionsfall). Om ja — placera huvudordets betydelsekod och egennamnets representation i R-listan och förse bägge med samma nummer (för att hantera fall som t ex *professor Andersson*, där samma referent kan i fortsättningen åberopas antingen med hjälp av enbart egennamnet eller enbart titeln). Instantiera variabeln `D` (bestämdhet) som `def` (+bestämd). Om nominalfrasen innehåller andra attributkoder, placera dem i listan A. Gå till 5. Om enheten `H` inte innehåller något egennamn i apposition, men andra attribut, placera också deras koder i A-listan, placera huvudordets kod i R-listan och gå över till 5.

5 Om den funktionella representationen innehåller fler konstituenter, sök efter nästa NP och — efter att ha funnit den — upprepa proceduren fr o m punkt 2. Om den funktionella representationer inte innehåller flera nominalfraser, omformulera den funktionella representationen till dess ursprungliga form (en PROLOG-atom med funktorn s) och gå över till etapp C (generering av meningen i målspråket).

Den ovan beskrivna proceduren innehåller i praktiken flera stadier: bl a en delprocedur som möjliggör insättning av svenska possessiva pronomen framför beteckningar för släktskap och liknande relationer (typ *morfar*, *granne* etc) och en del andra substantiv som kräver förekomsten av ett possessivt attribut i svenskan, men inte i ryskan och polskan (en inputmening av typen *spotkalem sąsiada* skulle alltså översättas till svenska som *jag träffade min granne*, även om den polska nominalfrasen *sąsiada* (*granne*) inte innehåller något possessivt pronomen).

Delar av koreferenssökningsproceduren kommer att illustreras med några test-exempel (demos).

2.1 Exempel på koreferensidentifiering i SWETRA

Vi antar, att åtminstone en mening har blivit analyserad och översatt. R-listan (substantiv- och verbkoder) och möjligen också A-listan (attributkoder) innehåller således några element. Den första frågan som ställs när en NP påträffas i den aktuella f-representrationen är: kan något av de tidigare översatta orden associeras med samma referent som den aktuella nominalfrasen? Första steget i svarsproceduren är att undersöka, om den lexikala enheten som motsvarar den aktuella NP:s huvudord är subklassificerat som "relation" eller "egenskap". Om den lexikala enheten (rlex) innehåller konstanten "property" med eventuell vidare subklassificering (exempelvis "colour"), består procedurens nästa steg i att undersöka, om A-listan innehåller ett adjektiv som tillhör samma kategori; om så är fallet, kan konstanten "def" placeras i det aktuella substantivets karakteristik. Detta är naturligtvis en approximation; algoritmen borde berikas med en del restriktioner, men den ger i regel korrekta resultat vid översättning av enkla textfragment. En analog princip tillämpas om det aktuella substantivet är i lexikonet subklassificerat som "relation" eller "aktion" — i det här fallet söker programmet i första hand igenom R-listan och letar efter ett verb vars semantiska beskrivning i lexikonet skulle möjliggöra koreferens med den aktuella nominalfrasen. Den delen av proceduren möjliggör korrekt översättning av sekvenser som (1). Den ryska versionen av exempel (1) är följande:

(1a)	Segodnja	neopoznannaja	podvodnaja	lodka	presledovala	švedskij	trauler
	idag	oidentifierad	ubåt		följde	svensk	trålare
(1b)	Presledovanie	prodolžalos'	okolo	časa			
	jakt	pågick	ungefär	timme			

De ryska lexikonenheter som motsvarar det transitiva verbet och det verbala substantivet i texten har följande form (förenklat):

```
rlex(presledovat',m(chase,_),v,vt,inf,_,_,
                  [follow,chase,hunt],_,_,_,_).
rlex(presledovanie,m(hunt,sg),n,sg,ina,ne,nom,
                  [follow,chase,hunt],rel2,_,_,_).
v    --- verb
vt   --- transitive verb
inf  --- infinitive
n    --- noun
sg   --- singular
ina  --- inanimate
ne   --- neutrum
nom  --- nominative
rel2 --- 2-argument-relation
```

Om programmet inte hittar något koreferent adjektiv eller verb, börjar det söka efter en tidigare översatt koreferent NP. Det enklast fallet är naturligtvis koreferens mellan nominalfraser med identiska betydelsekoder. Om R-listan innehåller en kod som inte skiljer sig från den aktuella, återstår bara att undersöka

eventuella motindicier (programmet kontrollerar t ex om det aktuella ordet inte föregås av ett attribut av typen *annan*) och — ifall inga sådana finns — instantiera variabeln D som def. Den delen av algoritmen möjliggör generering av bestämd/obestämd form i exempel som (7) och (8):

Input:

(7a)	Južnee	Sajdy	pojavilsja	izrail'skij	samolet.
	söder om	saida	dök upp	israelisk	flygplan
(7b)	Samolet	prodvigaetsja	na	zapad.	
	flygplan	förflyttar sig	västerut		

Svensk output:

(7c) Ett israeliskt flygplan dök upp söder om Saida.
(7d) Flygplanet förflyttar sig västerut.

Input:

(8a)	Kakoj-to	samolet	pojavilsja	južnee Sajdy.
	något	flygplan	dök upp	söder om saida
(8b)	Potom	pojavilsja	drugoj	samolet.
	senare	dök upp	annan+ma	flygplan

Svensk output:

(8c) Något flygplan dök upp söder om Saida.
(8d) Senare dök det upp ett annat flygplan.

Om R-listan inte innehåller någon kod som är identisk med det aktuella ordets betydelsekod, fortsätter sökningsproceduren; det gäller bl a att ge svar på följande frågor:

- om den aktuella nominalfrasen har värdet +plural: innehåller R-listan åtminstone två enheter med samma betydelsesymbol (den första delen av meningskoden), men med värdet "sg" ? Eller har man tidigare påträffat åtminstone två enheter som bildar en mängd som kan vara koreferent med den aktuella frasen?

- innehåller R-listan en kod, som kan associeras med ett koncept ("nominalizer") som inte är ojämförbart och inte mer specifikt än det aktuella ordets "nominalizer"? Dessa delar av proceduren identifierar koreferens i sekvenser av typen:

Input:

(9a)	Mal'čik	vstretil	devočku.
pojke	träffade	flicka+ack	
(9b)	Rebjata	pobežali	domoj.
	barn	sprang	hem

Output:

(9c) En pojke träffade en flicka.
(9d) Barnen sprang hem.

Substantiven *mal'čik* (pojke) och *devočka* (flicka) är i lexikonet specifierade som subkategorier av "barn"; det PROLOG-predikatet som identifierar koreferens i fall som (9) är formulerat som:

```
coref(m(A,pl),Rlist):- hyponyms(m(A,sg),[H|T],Rlist),
      T/=[].
```

Ovanstående regel kan läsas på följande sätt: ett substantiv med betydelsekod `m(A,pl)` kan associeras med en mängd bestående av åtminstone två tidigare introducerade referenter, om åtminstone två tidigare nämnda substantiv kan tolkas som hyponymer till det aktuella substantivets singularform (symboliserad som `m(A,sg)`). Detta är en av de enklaste varianterna av coref-predikatet (i praktiken används en del ytterligare restriktioner; deras utformning kräver fortsatt arbete).

I ex (9) identifieras alltså substantiven *mal'čik* och *devočka* som hyponymer till den lexikala enheten med betydelsekod `m(child,sg)` och koreferensen upptäckts med hjälp av följande predikat, som rekursivt letar efter möjliga hyponymer till det aktuella ordets (här:*rebjata* — *barn*) singularform:

```
hyponyms(m(A,N),[m(B,N) | Rest],[r(_,m(B,N)) | Rest1]):-
            more_restricted(m(B,N),m(A,N)),
            hyponyms(m(A,N),Rest,Rest1).

hyponyms(m(A,N),Hyponymlist,[H|T]):-
hyponyms(m(A,N),Hyponymlist,T).

hyponyms(m(A,N),[],[]):-!.
```

Variabeln `Hyponymlist` (symboliserad i huvudpredikatet "coref" som `[H|T]`) betecknar naturligtvis en lista, i vilken eventuella hyponymer lagras under sökningsproceduren. Predikatet `more_restricted` är — i dess enklaste version — formulerat som:

```
more_restricted(A,B):- rlex(_,A,_,_,_,_,_,Features1,
                Features2,_,_,_),
                rlex(_,B,_,_,_,_,_,Features2,_,_,_,_).
```

Regeln säger, att begreppet associerat med en lexikal enhet med betydelsekoden A är mera specifik än begreppet associerat med B, om A har de semantiska drag (Features2) som anses vara mest karakteristiska för B, och dessutom åtminstone ett drag, som är mera specifikt. I ex (9) innehåller de lexikala enheterna för "pojke" och "flicka" specifika drag "male" resp "female" och dessutom alla drag som tillåter användningen av enheten "barn" och, följaktigen, tolkas de som hyponymer till samma ord. Formatet för lexikala enheter är i det här fallet följande (förenklad notation):

```
rlex(mal'čik,m(boy,sg),n,sg,ani,ma,nom,[male],
                            [child],_,_,_).
rlex(devočka,m(girl,sg),n,sg,ani,fe,nom,[female],
                            [child],_,_,_).
rlex(rebenok,m(child,sg),n,sg,ani,ma,nom,[child],
                            [human],_,_,_).
```

Det ovan visade fallet av koreferensidentifiering är naturligtvis mycket enkelt; att bygga upp en hierarki av semantiska drag som skulle fungera effektivt vid sökning efter hyponymer i mera komplicerade texter är ingen enkel uppgift; utformning av lexikala enheter i SWETRA är i detta avseende än så länge inte fullständig. Vid översättning av korta textfragment är det dock möjligt att identifiera en gemensam diskursreferent i fall som är mindre "självklara" än ex (9) — som t ex den tidigare diskuterade koreferensen mellan "israeliska flygplan" och "luftpirater". I den sortens fall används följande PROLOG-predikat:

```
possible_coref(A,B):- rlex(_,A,_,_,_,_,_,F1,_,_,_,_),
                      rlex(_,B,_,_,_,_,_,F2,F3,_,_,_),
                           evaluation_in(F2),
                           co_elt(F1,F2).
```

Predikatet `evaluation_in` innebär, att listan F2 (semantisk karakteristik) innehåller ett drag som är klassificerat som "värdering"; det enklaste sättet att i en komputationell modell identifiera koreferens mellan ett substantiv som är markerat i fråga om värderingskomponenten och ett omarkerat sådant är att betrakta värderingskomponenten enbart som ett uttryck för sändarens attityd, och inte som en faktor som påverkar referensrelationen. Eftersom de lexikala enheterna i vårt exempel innehåller gemensamma drag (i listorna F1 och F2) och programmet inte hittar några indicer mot koreferens, associeras både "israeliska flygplan" och "luftpirater" med samma referent. Nedanför visas — i en förenklad notation — de lexikala enheter som i det diskuterade fallet möjliggör identifiering av koreferensrelationen:

```
rlex(samolety,m(airplane,pl),n,pl,ina,ma,nom,
     [airplane,pilot],[machine,human],_,_,_).
rlex([vozdušnyje,piraty],m(air_pirate,pl),n,pl,ani,
     ma,nom,[neg,pilot],[human],_,_,_).
```

Frågan, vilka och hur många gemensamma drag som behövs för att två "nominalizers" ska uppfattas som koreferenta, är naturligtvis komplicerad, och den aktuella versionen av programmet utesluter inte vissa fel och övergeneraliseringar.

Förutom de nämnda predikaten innehåller programmet också en preliminär procedur för identifiering diskursreferenter även när nominalfrasens huvudord är elliderat. Proceduren kräver vidare elaborering, men dess nuvarande utformning ger möjlighet att generera rätt artikel och kontrollera kongruens i sekvenser av typen:

Input:				
(10a)	Dva	samoleta	pojavilis'	južnee Sajdy.
	två	flygplan	dök upp	söder om saida
(10b)	Odin	prodvigaetsja	na zapad	
	en+ma	förflyttar sig	mot väst	
Output:				
(10c)	Två flygplan dök upp söder om Saida			
(10d)	Det ena förflyttar sig mot väster.			

Efter referentidentifiering följer översättningsprocessens sista etapp — generering av outputmeningen. Om variabeln D fortfarande är oinstantierad (som t ex vid översättning av textens första nominalfras eller i de fall, då ingen koreferent fras har upptäckts), genereras den svenska NP:s morfologiska form i enlighet med språkspecifika regler i programmets svenska modul. Den obestämda artikeln genereras alltså inte automatiskt när variabeln D inte har fått värdet def under etapp B — programmet undersöker först, om nominalfrasens huvudord är klassificerad som "unikt föremål" (som *solen*, *jorden* etc) eller som ämnesnamn, om det föregås av indefinita, deiktiska eller possessiva pronomen osv. Ogrammatiska konstruktioner som **en min bil* eller **en ingen bil* kan således inte genereras. Om variabeln D har blivit instantierad, är genereringsproceduren betydligt kortare — man behöver bl a inte undersöka, om substantivet i fråga är en benämning på ett unikt föremål, eller om det föregås av indefinita pronomen.

Enheten R ("referent nucleus") används i genereringsproceduren för att kontrollera kongruens inom nominalfrasen. Efter översättningsprocessen första fas (etapp A) innehåller enheten R följande konstanter:

- betydelsekod — t ex `m(boy,sg)`
- numerusvärde — här, sg
- det genusvärde, som substantivet har i källspråket
- värdet +/- animat

Återstående platser i enheten R kan senare fyllas med symboler för de grammatiska drag, som är nödvändiga för att åstadkomma kongruens i målspråket. Under genereringsprocessen söker programmet efter en svensk lexikal enhet med samma betydelsekod och — om variabeln D (bestämdhet) har fått ett konstant värde under etapp B — med samma bestämdhetskod som den aktuella nominalfrasen i f-representationen. Om t ex en nominalfras med huvudordets betydelsekod `m(boy,sg)` har fått värdet def efter referentidentifieringsproceduren, väljs följande lexikala enhet (om frasen inte innehåller attribut som utesluter användning av efterställd artikel):

```
lex(pojken,m(boy,sg),n,def,sg,re,ma,_,_,_,_,_).
re --- realis
```

och målspråkets genus- och sexusvärden överförs till "referent nucleus", som får följande form:

```
r(_,m(boy,sg),def,sg,ani,ma,re,ma).
```

Om frasen innehåller ett eller flera adjektivattribut, väljs den bestämda artikelns rätta numerus- och genusform enligt följande regel:

```
art(R,D) --> [X],R=r(_,_,_,D,N,_,_,G,_),
                    lex(X,D,art,N,G,_,_,_,_,_,_,_).
```

Regeln innebär, att kategorin "artikel" (art) med bestämdhetsvärde D kan realiseras som en ordform X om X har samma bestämdhets-, numerus- (N) och genusvärden (G) som nominalfrasens huvudord. Analoga regler används för att välja de övriga kongruerande attributens morfologiska former. Kongruenskontroll med hjälp av "referent nucleus" fungerar mycket effektivt, och några fel har i detta avseende inte observerats.

3 Sammanfattning

Identifiering av diskursreferenter i SWETRA:s program bygger på en fyra-nivå diskursmodell (1:ord och ordsekvenser, 2:begrepp baserade på rent lingvistisk kunskap, 3:diskursreferenter — kognitiva enheter som kan associeras med flera begrepp på nivå 2 på basis av analytiska slutsatser och kunskap om den icke-lingvistiska verkligheten, 4: objekt och fakta i den verkliga världen). Den aktuella algoritmen för identifiering av diskursrefenter med hjälp av den textuella informationen befinner sig på ett experimentellt stadium och bygger dels på den textuella informationen (betydelsekoder och meningarnas funktionella representationer lagrade i olika PROLOG-listor), dels på preliminära försök att representera stereotypa begrepp i lexikonet (listor över semantiska drag, semantisk subkategorisering). Programmet möjliggör för närvarande identifiering av koreferens i korta textfragment (5–6 meningar) vid relationer av typen: koreferens mellan identiska begrepp, koreferens mellan ett mera generellt begrepp och dess mera specifika antecedent (typ *granne — människan*), koreferens mellan referentmängder (*pojke* och *flicka — barnen*) samt mellan deras element (*två flygplan — det ena flygplanet*). Referentidentifiering är dessutom möjlig i vissa ellipsfall (*två flygplan — det ena*), vid koreferens mellan ett emotionellt laddat begrepp och dess neutrala motsvarighet (*israeliska flygplan — luftpiraterna*), samt i de fall, då koreferenta begrepp instantieras i texten med hjälp av kategoriellt olika fraser (t ex om diskursreferenten introduceras av ett adjektiv eller en hel mening). Generering av bestämdhetsvärde i målspråket sker primärt på basis av koreferensrelationer, därefter tillämpas språkspecifika regler (nominalfrasens form väljs med hänsyn till attributtyp och substantivens subkategorisering i det svenska lexikonet). Proceduren kräver vidare elaborering — de semantiska representationerna i lexikonet måste utvidgas, dessutom gäller det att åstadkomma ett mera generellt system för utformning av listor med semantiska drag. Databasen måste dessutom berikas med en förenklad representation av stereotypa kunskaper om den icke-lingvistiska världen (detta är en uppgift som kan förverkligas enbart i begränsad utsträckning och leder ur lexikonet in i encyklopedien). Det finns också ett behov av vidare studier kring de faktorer som implicerar valet av bestämd form i svenskan i de fall då ingen koreferensrelation föreligger (i många empiriska svenska texter har t ex den första nominalfrasen bestämd form). Den komputationella modellen som presenterats här har alltså en rad begränsningar, men vid översättning av korta texter med ett begränsat antal diskursreferenter fungerar den tämligen effektivt. Procedurens utveckling är föremål för fortsatt arbete inom SWETRA.

Litteratur

Frarud, Kari. 1986. The introduction och maintenance of discourse referents. *Papers from the Ninth Scandinavian Conference of Linguistics.* Stockholm.

Gawrońska-Werngren, Barbara. 1988. A Referent Grammatical Analysis of Relative Clauses in Polish. *Studia linguistica* 42(1):18-48.

Karttunen, Lauri. 1976. Discourse Referents. *Syntax and Semantics*, vol. 7:383–386. New York: Academic Press.

Sidner, Candace L. 1983. Focusing in the Comprehension of Definite Anaphora. Brady, M. & Berwick, R. C. [eds.] *Computational Model of Discourse*:267–330. MIT Press, Cambridge, Mass.

Sigurd, Bengt. 1987. Referent Grammar (RG). A generalized phrase structure grammar with built-in referents. *Studia linguistica* 41(2):115–135.

Sigurd, Bengt. 1988. Using Referent Grammar (RG) in computer analysis, generation and translation of sentences. *Nordic Journal of Linguistics* 11:129–150.

Sigurd, Bengt, & Gawrońska-Werngren, Barbara. 1988. The potentials of SWETRA, a multilanguage MT-system. *Computers and Translation* 3:238–250.

Lund University
Dept of Linguistics and Phonetics
Helgonabacken 12
Lund, Sweden

NIELS JÆGER

Text Treatment and Morphology in the Analysis of Danish within EUROTRA

1 General Overview

The EUROTRA TRANSLATION SYSTEM consists basically of a C-programme which operates a front-end programme package and a core PROLOG programme, the Engine.

The Engine combines with a series of generators and translators written by the linguist in a user language and compiled into PROLOG code. The generators and the translators contain linguistic information according to the linguistic specifications in the EUROTRA model.

The EUROTRA model consists of three main parts: analysis, transfer and synthesis. The main parts are subdivided into levels. Each level in the EUROTRA model has a grammar (i.e. a generator) and between two levels there is a translator.

The EMS, EUROTRA Morphological structure, is the first level in analysis. The first syntactic level follows immediately after EMS, it is called ECS, EUROTRA constituent structure.

A front-end programme package is being tested in EUROTRA-Denmark, from July to December 1989. It consists of a wordscanner written in C and an SGML parser. SGML is an abbreviation of "Standard Generalized Markup Language".

In the present experimental phase the document which is going to be translated has to be written in "VI" or "Qone", two editors available for UNIX installations. A "VI" document may be formatted by "Nroff". But in principle, any text editor could be used for SGML text entry, just as SGML could deliver an output to any text editor.

SGML provides methods for marking up documents. For instance you can mark up the logical structure of a text (i.e. chapters, sections, paragraphs etc.).

SGML also provides a search-and-replace mechanism. We can make the SGML parser search for layout information and replace it by a marker.

We can also define translations between input character code and output character code.

In the front-end module we intend to use the SGML parser twice:
First the special formatting codes of the word processors will be changed into SGML mark up codes and simultaneously the 8-bit national character sets are turned into a standardised 7-bit Eurotrian character set.
Next the SGML parser creates a translator.

SGML may take in a whole text for the first parsing. But the paragraphs of the text will be forwarded to the second application of the parser and later to the Engine in strict rotation. The paragraphs will be translated one by one and placed in their translated form in the right position in the output file in synthesis.

Between the two applications of the parser it is possible to edit the first SGML document manually.

The last programme in the front-end package is a wordscanner written in C. The wordscanner segments the wordforms on the basis of the lexicon rules of the morphological level EMS.

2 An Example

Please consider the following example which consists of a title and a sentence:

```
Signaler

En kombineret 4 og 6 GHz polarizer konverterer det modtagne
signal.
```

Written in "VI" and supplied with "Nroff" commands the example would look like this:

```
.NH
Signaler
.PP
En kombineret 4 og 6 GHz polarizer konverterer
.UL det
.UL modtagne
.UL signal.
```

`.NH` means: what follows is a title—`.PP` means: here comes a paragraph. `.UL` in front of a word indicates that the word is underlined.

This piece of text is sent to the SGML parser. The "Nroff" commands are changed into SGML markers. The text now looks like this:

```
<DOC   LEVEL="emsda"  SCANLEV="ems"   SURF="dummy"  SCAN=yes
WP="Nroff" NAME=text">
<TEXT>
<CH>
<TI NROFF= "PP">
En kombineret 4 og 6 GHz polarizer konverterer
<STYLE TYPE="UL">
det
<STYLE TYPE="RN">
<SYLE TYPE="UL">
modtagne
<STYLE TYPE="RN">
<STYLE TYPE="UL">
signal.
<STYLE TYPE="RN">
</P></CH></TEXT></DOC>
```

Next the text is forwarded to the second application of the SGML parser which produces a translator:

```
:trans: dummy=>emsda.

:b:

surface0 = ~:{} => {cat=s, io='0'}

<{thcat=wordform, rend=no,upper=first}
%%SCAN signaler
>.

surface1 = ~:{} => {cat=s, io='1'}
<
{thcat=wordform, rend=no, upper=first}
%%SCAN en
,
{string='&bk'},
{thcat=wordform, rend=no, upper=no}
%%SCAN kombineret
,
{string='&bk'},
{string=numerus, lex='4', rend=no},
{string='&bk'},
{thcat=wordform, rend=no, upper=no}
%%SCAN og
,
```

```
{string='&bk'},
{string=numerus, lex='6', rend=no},
{string='&bk'},
{thcat=wordform, rend=no, upper=first}
%%SCAN ghz
,
{string='&bk'},
{thcat=wordform, rend=no, upper=no}
%%SCAN polarizer
,
{string='&bk'},
{thcat=wordform, rend=no, upper=no}
%%SCAN konverterer
,
{string='&bk'},
{thcat=wordform, rend=under, upper=no}
%%SCAN det
,
{string='&bk},
{thcat=wordform, rend=under, upper=no}
%%SCAN modtagne
,
{string='&bk'},
{thcat=wordform, rend=under, upper=no}
%%SCAN signal
,
{string='&fs'}
>.
.
```

&bk stands for blank; **&fs** stands for full stop.

The translator will translate from an empty object (i.e. the empty pair of curly brackets on the left side of the arrow) into the sentences we gave as input (on the right side of the arrow). **{cat=s}** means a sentence. The sentence in its turn contains several wordforms. The lay out information is featurized. 'upper' stands for upper case. The value 'first' means that the first letter in a word is capitalised. 'rend' means rendition and 'rend=upper' means that a given word is underlined. Double percentage signs and the capitalised word 'SCAN' is a signal to the wordscanner that the following word should be segmented.

Now, this translator is sent to the wordscanner. The result will be similar to the translator shown above except for the wordforms which will have been split up into basic strings. The first part of the translator will suffice to illustrate this:

```
surface1 = ~:{} =>
<
{thcat=wordform, rend=no, upper=first}
```

```
<{string=en}>,
{string='&bk'},
{thcat=wordform, rend=no, upper=no}
<{string='kombiner'}, {string='et'};
 {string='kombiner'}, {string='e'}, {string='t'}>,
{string='&bk'},
{string=numerus, lex='4', rend=no},
...
>.
.
```

The substructure is indicated by '<' and '>'.

The wordforms are split up into basic strings which figure as values to the attribute 'string'. A useful device is the possibility of using alternation in the translator. If, for instance, we have three dictionary entries for three different verbal endings '{string=e, ... }', '{string=t, ... }', and '{string=et, ... }' we will get two possible segmentations of 'kombineret' as shown above. The EMS should of course only accept one of the possible segmentations. In case of an Arabic number we have a standard value to the string attribute: 'string=numerus'. This means that we only have to write a single lexicon rule for all possible integers at the morphological level. The actual number is given as value to the attribute 'lex'. Since numbers are the same in all the Eurotrian languages no tranfer rules will be needed.

The wordscanner compares the wordforms with the lexicon rules of the morphological module. A lexicon rule has depth=0—it consists of one feature bundle. Among other things we assign category, a lexeme value (assigned to the attribute 'lu') and a string value—the latter will unify with the string value in the translator rule. We have lexicon rules for inflectional roots (marked by the feature 'infl=root'), we have lexicon rules for invariants (marked by 'infl=full') and we have lexicon rules for inflectional suffixes (marked by 'infl=infl_end'). The approach to inflectional morphology we have adopted is in accordance with the so-called 'word and paradigm model'. Consequently, the lexemes are divided into inflectional classes. Lexemes which can be grouped as an inflectional class receive the same number as value for the attribute 'flex_type'. Examples of lexicon rules (abbreviated):

```
{cat=art, lu=en, infl=full, string=en, ...}.

{cat=v, lu=konvertere, infl=root, string=kombiner, term=xx0,
 ...}.

{cat=v, infl=infl_end, string=e}.
{cat=v, infl=infl_end, string=t}.
{cat=adj, infl=infl_end, string=et}.
{cat=n, infl=infl_end, flex_type=fx3, msdefs=msindef,
 nb=plu, string=er}.
```

```
{cat=n, lu=signal, infl=root, flex_type=fx3, string=signal,
 term=xx00000837, ...}.

{cat=card, lu=numerus, infl=full, string=numerus}.
```

These single feature bundles combine at the morphological level in general constructors. The constructors needed for building past participles (as e.g. 'kombineret') will look like this (slightly abbreviated):

```
b_stem_v = {infl=stem, cat=v, lu=L, vform=inf, ...}
           [ {infl=root, cat=v, lu=L, ...},
             {infl=infl_end, cat=v, string=e} ].

b_full_v = {infl=full, cat=v, vform=pastpart, lu=L, ...}
           [ {infl=stem, cat=v, lu=L, vform=inf, ...},
             {infl=infl_end, cat=v, string=t, ...} ].
```

The constructors needed for building indefinite forms of nouns (as e.g. 'signaler') will have the form:

```
b_stem_n = {infl=stem, cat=n, lu=L, nb=N, msdefs=msindef,
            ...}
           [ {infl=root, cat=n, lu=L, flex_type=F, ...},
             {infl=infl_end, cat=n, flex_type=F, nb=N,
              msdefs=msindef}.


b_full_n = {infl=full, cat=n, lu=L, nb=N, msdefs=M,
            case=nge, ...}
           [ {infl=stem, cat=n, lu=L, nb=N, msdefs=M, ...},
             {infl=full, cat=separator} ].
```

The identifier occurs before the equal-sign to the left. In the constructor the mother node appears above, the daughter nodes appear between sharp brackets below.

The inflectional root may become an inflectional stem by addition of an inflectional ending (e.g. 'e' in the constructor for a verbal stem) which in that case only becomes a final wordform when another inflectional ending is added (e.g. 't' in the constructor which turns an infinitive stem into a past participle above). The finished wordform is marked 'infl=full'.

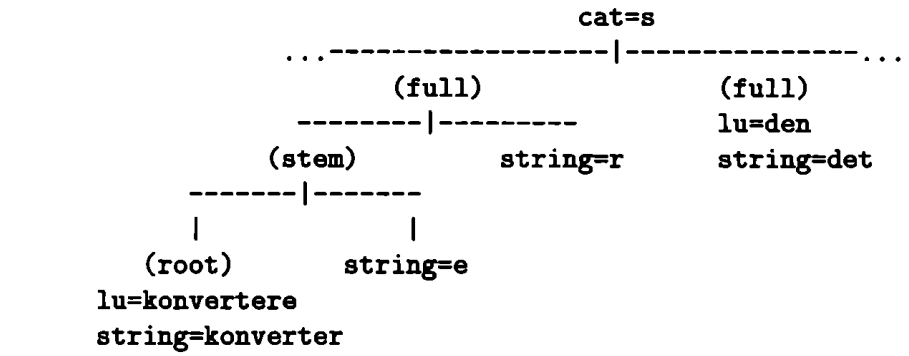
Variables (in the form of capital letters) are used a) to percolate values from daugthers to the mother node (cf 'lu=L') and b) to ensure that values of a certain attribute are the same in the feature bundles of two daugthers (cf 'flex_type=F').

By means of the segmentation and the subsequent unification of the basic strings we turn a wordform into a grammatical word—e.g. we change 'signaler' into 'signal, plural, indefinite, neuter'.

The sentence rule accepts one or more wordforms:

```
b_sentence = {cat=s}
             [{infl=full}].
```

The final representation at EMS has the form of a tree with a sentence node at the top and a series of binary subtrees (one for each wordform) which are hanging on the same level beneath:



Adjectives, verbs and nouns are marked for termhood at EMS. 'term=xx0' means that a given word is not a term. A value different from 'xx0' means that the word is a term. Multi-word terms are identified at EMS in the Danish implementation. As an example of a constructor which builds a multi-word term I will give the one for term number 'x000000003':

```
b_000000003 = {cat=n, lu=kombineret_N-og_N_GHz_polarizer,
               term=xx000000003, gd=G, nb=N, msdefs=M,
               case=C, infl=term}
              [ {infl=full, cat=pastpart, lu=kombinere},
                {infl=full, cat=card, lu=numerus},
                {infl=full, cat=coord, lu=og},
                {infl=full, cat=card, lu=numerus},
                {infl=full, cat=n, lu=GHz},
                {infl=full, cat=n, lu=polarizer,
                 gd=G, nb=N, msdefs=M, case=C} ].
```

'polarizer' is the most important word in this multi-word term. We therefore percolate all relevant grammatical information from the main word to the top node, we assign a number to the attribute 'term' and a value to the 'lu' attribute. The number is assigned in order to avoid translation rules between two languages. A given term has the same number in all nine Eurotrian languages. The lexical unit value is for the benefit of grammarians and lexicographers who develop and maintain the system.

In the translator between the morphological level, EMS, and the constituent level, ECS, all substructure beneath the very top nodes is deleted in general translator rules.

When these general translator rules have applied the input object to ECS will therefore consist of a series of grammatical words and multi-word units hanging beneath the sentence nodes:

```
{cat=s}
  {cat=n, lu=signal, ...}
{cat=s}
  {cat=art, lu=en, ...}
  {cat=n, lu=kombineret_N_og_N_GHz_Polarizer, ...}
  {cat=v, lu=konvertere, ...}
  {cat=art, lu=den, ...}
  {cat=n, lu=modtaget_signal, ...}
  {cat=separator, lu=stop, ...}
```

EUROTRA-DK,
Njalsgade 80,
DK-2300 Copenhagen S
Denmark

SABINE KIRCHMEIER-ANDERSEN

Coordination in Eurotra

Abstract

The treatment of coordination in a multilingual MT-system as EUROTRA poses two major problems:

1. to provide an algorithm for the monolingual analysis of coordinate structures in accordance with the general linguistic theory for EUROTRA which can treat basic coordination and the coordination of incomplete constituents in the surface analysis and perform the mapping of these constructions onto the deeper levels.
2. to establish a semantic feature system, that captures the meaning of the conjunctions in order to facilitate their translation.

Since research in this area within Eurotra on a multilingual basis is still ongoing, what will be presented is the present state of the art, which is a fairly complete theory for basic coordination as well as some ideas for the analysis of the more complex cases of coordination involving gapping and movement.

1 Introduction

It is well known that coordination is one of the most prevalent and complex constructions in European languages which causes great difficulties in all syntactic formalisms. At present in Eurotra, not all aspects of coordination are covered. The current implementations of the grammars of the monolingual modules handle basic coordination of alike constituents, but not special cases like coordination of unlike or incomplete constructions. Coordination is still a topic of ongoing research both from a monolingual and a contrastive point of view.

In the first part of this paper, I would like to illustrate how coordination is handled in Eurotra in analysis, in transfer and in generation and I will comment on some of the universal mechanisms or constraints, which we exploit in order to avoid overgeneration and to facilitate the translation of coordinated structures. (For information on the Eurotra linguistic theory as well as the overall objectives of the project see Perschke 1986, Jaspaert 1986, Arnold 1986 and Raw et al. 1989). In the second part I will go into two of the difficult areas, that we are working on at the moment.

2 Basic Coordination—an Example

Basic coordination is defined as coordination of constituents with the same category, i.e. "John and Mary", "happy or sad" etc.

The following section gives a short description of the translation of a sentence containing a coordinated structure from Danish into English. The following example is used:

(1) Både USA, Japan og Kina viser interesse for EF

both usa, japan and china show interest for eec
(literal translation)

2.1 Analysis

The figures 1 and 2 show the structural representation of the sentence (1) at two of the syntactic levels in the Eurotra theory: ECS—the constituent structure,

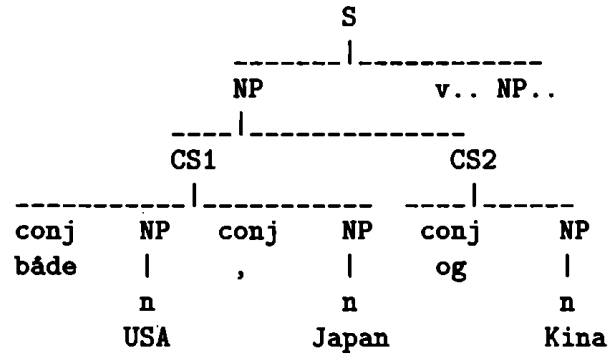


Figure 1: ECS

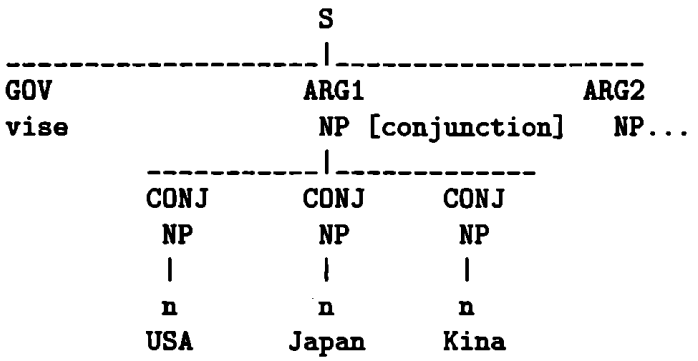


Figure 2: IS

and IS—the deep syntactic structure. Between these levels we have a third level, ERS—the surface syntactic structure, but with respect to coordination the ERS and IS representations are quite similar so ERS is left out in these examples.

Comparing the two structural representations in figure 1 and figure 2, we can note that the intermediate nodes at ECS, "cs1" and "cs2", have been deleted at IS. Their main function is to group the conjuncts and the conjunctions together in the constituent analysis at ECS and to prevent other categories from entering the coordinate structure during the parsing process. The representation at ECS is in some respects similar to the one presented by Gazdar et al. (1985). GPSG uses a constituent similar to the cs2 node for all parts of the coordinated structure. We have introduced a more complex constituent, cs1, for the first two conjuncts in order to speed up the parsing process.

The IS level, where the constituents are defined as dependency structures consisting of a governor and a number of arguments and modifiers, is characterized by the canonical ordering of these constituents. Since coordination is not a dependency relation, the surface order of the constituents in coordinate structures is not changed.

The conjunction, the comma and the preconjunction "både" are deleted at IS. Their semantic content is preserved as a feature "[conjunction]" on the top node of the coordinated structure.

2.2 Transfer

In the transfer phase, the Danish-English transfer module translates the lexical values of the conjuncts into English. This is practically all that happens. The rest of the structure is translated by a default mechanism, that simply tranfers it into the target language IS structure without any changes (figure 3).

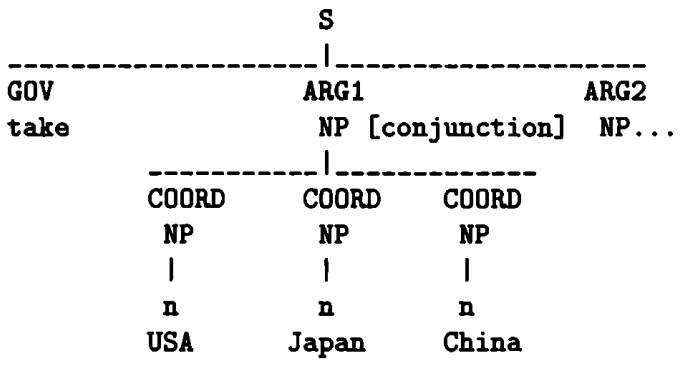


Figure 3: IS

2.3 English Synthesis

The last step of the translation process which will concern us here, is performed during generation in the ECS grammar of the target language, where the English conjunctions are inserted on the basis of the semantic features computed in the analysis. This gives us a structural representation like the one in figure 4.

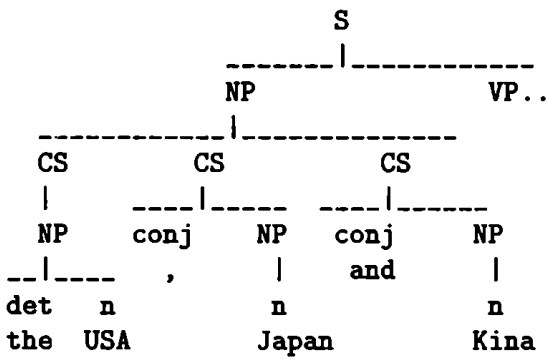


Figure 4: ECS

The final result is the following sentence:

(2) The USA, Japan and China take interest in the EEC

At ECS in generation the "cs" nodes have reappeared. They make it easier to insert the conjunctions in the right places. In synthesis we do not need to distinguish between "cs1" and "cs2", because we know from the analysis, exactly which constituents are to be conjoined.

An equivalent to the preconjunction "både" has not been created and this is part of the justification for the featurization of the conjunctions, since the occurrence of these preconjunctions show monolingual variations. In English "both" is considered a strictly binary conjunction, probably because of the homonymy with the quantifier wich corresponds to "begge" in Danish. In Danish "både" may occur with any number of conjuncts.

We can also note, that the comma has been translated as a comma and not as a lexical conjunction as in (3):

(3) Japan and the USA and China take interest in the EEC

This is a matter of style and taste. Like a joker in a game of cards, the comma can take the place of any conjunction in a coordinate structure, provided that there are at least three conjuncts and that the comma does not appear with the first or the last conjuncts (except in enumerations). If we allowed such variations we would get a multiple output, and since the output of a machine translation system perferably should be one solution, we have chosen the one in (2).

3 Universal Constraints

In order to perform the translation described above, we have relied on a number of general properties, that seem to be the same for all coordinated structures.

3.1 Bar Level Constraints

In order to be able to coordinate all categories, we want to underspecify the category value. The advantage is of course that we can use only one basic set of rules to handle all cases of coordination of alike constituents:

```
1.      X -> cs1[X]  *cs2[X]

2.      cs1[X] -> (preconj) X conj X
3.      cs2[X] ->  conj X
```

X is a variable which is instantiated with the category value of the conjunct and percolated to the top node of the coordinated structure. Using unification we can ensure, that X in every rule contains the same category. These very general rules, however, tend to coordinate categories at every bar level thus performing an anlysis which is obviously wrong. Consider cases as (4):

```
    *  the man   and    boy
(4)      NP             N
        bar=1          bar=0
```

Consequently, we allow only coordination of constituents at the same bar level and not at bar level zero as in (5). A similar approach has been developed by Nirenburg (1989).

```
(5)      *   NP  bar=1
         ____|__________
         det           n   bar=0
         the           |
                      CS1
                 _____|________
                 n    conj    n  bar=0
                 USA  and     China
```

3.2 Binary and Iterative Coordination

For the insertion of the correct conjunctions in synthesis it is important to calculate whether the coordinate construction consists of two or more than two elements, as demonstrated for "both" in (1) and (2). A coordinate structure is binary if it consists of only two conjuncts as in (6):

(6) Both X and Y

Iterative coordinate structures consist of more than two conjuncts as in (7):

(7) X and Y and Z and .....

The occurrence of preconjunctions in binary and iterative coordination is language specific as shown in the translation example above.

3.3 Hierarchy

If we want to introduce the concept of hierarchy for coordinated structures, it entails that our rules have to be recursive. This means that we must be able to create deep representations as in figure 5 as well as the flat ones presented in figure 1 and figure 4.

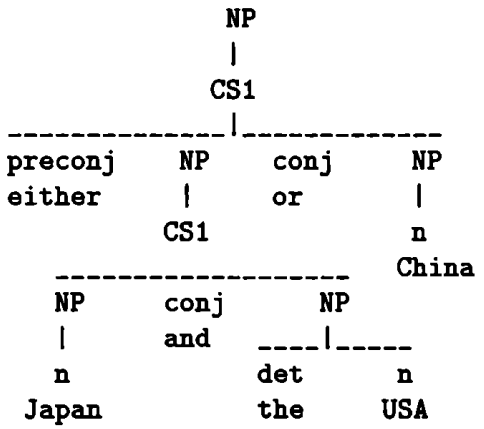


Figure 5: ECS

This structure is hierarchical in the sense that one of the conjuncts is a coordinated structure itself. Allowing this type of recursion without restrictions would cause heavy overgeneration at ECS. The number of possible structures grows fast with the number of conjuncts. 2 conjuncts = 1, 3 conjuncts = 3, 4 conjuncts = 11 etc. Again we prefer only one solution. The constraint that we pose on these structures is simply that coordinate structures may not immediately dominate coordinate structures of the same type (see 3.4 below). In the Eurotra formalism this is expressed in terms of filter or killer rules which delete the undesired structure.

3.4 Cooccurrence Restrictions

Finally, we have to pose cooccurrence restrictions on the conjunctions both with regard to their semantic type and their paradigmatic distribution. The way in which cooccurrence restrictions are expressed is a monolingual matter. For European languages, we can distinguish 5 basic universal types of coordination.

Coordination	Type
både – og, og, samt	conjunction
enten – eller	disjunction
hverken – eller	negation
men	adversion
,	enumeration

The semantic restrictions imply that we cannot have a coordination like (8) on the same hierarchical level.

(8) * både X eller Y

The positional restrictions for the conjunctions operate with 3 positions: initial, non-initial and final. In binary coordination only the initial and final positions are used. In iterative coordination a theoretically infinite number of conjunctions may additionally appear in non-intial position.

Initial	non-initial	final	Example
enten	,/eller	eller	enten A,B eller C
Ø	,/eller	eller	A,B eller C
både	,/og	og/samt	både A,B og C
Ø	,/og	og/samt	A,B og C
hverken	,/eller	eller	hverken A,B eller C

Since coordinate structures consisting of alike constituents are the most frequent ones, the rules and constraints discussed above can handle many of the cases of coordination occurring in the text types we are working with. However, there are still a number of cases where this basic treatment is not sufficient, some of which will be treated in the second part of this paper.

4 Complex Cases of Coordination

4.1 Coordination of Unlike Constituents

The question is now how to integrate the coordination of unlike constituents into the system sketched above.

By coordination of unlike constituents we mean coordinated structures consisting of different categories as in (1)–(5):

(1) Hun sang smukt og med høj stemme
adv prep + sub

(2) Han var glad og i godt humør
adj prep + sub

(3) Hun er bager og stolt af det
sub adj

(4) De spurgte her og på bjerget
adv prep + sub

(5) Han lovede bedring og at det ikke skulle gentage sig
sub sætn

The examples show some of the combinations of constituents that may occur in coordinated structures in Danish texts. However, there are certain restrictions with regard to which unlike constituents can be conjoined, but the rules for these restrictions are not easy to determine from a surfaceoriented constituent analysis.

(6) * Hun sang smukt og en arie fra Aida
adv sub

(7) * Han var bager og på bjerget
sub prep + sub

(8) * De spurgte hende og på kontoret
sub prep + sub

The underlying pattern emerges if we look at the syntactic functions of the constituents instead of their grammatical category.

(1): subj v modifier + modifier

(2): subj v attr.subj + attr.subj

(3): subj v attr.subj + attr.subj

(4): subj v modifier + modifier

(5): subj v obj + obj

(6): * subj v modifier + internal.obj

(7): * subj v attr.subj + modifier

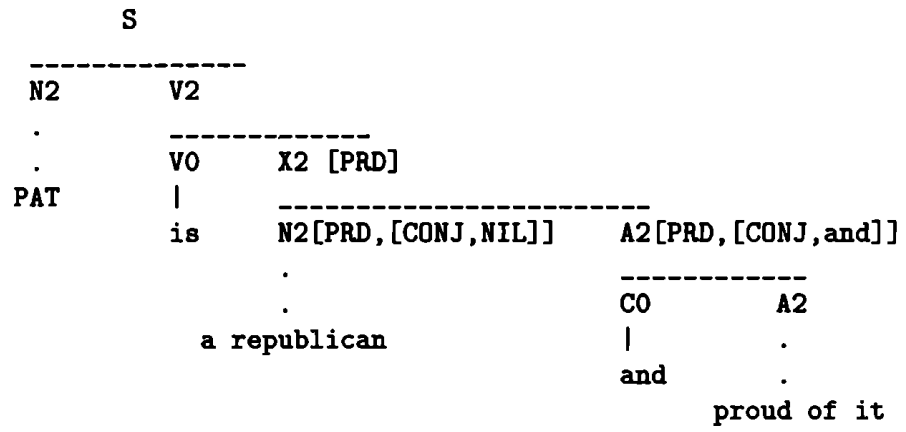
(8): * subj v obj + modifier

Diderichsen (1946) notes that:

> Leddene i en sideordnet Forbindelse staar hvert for sig i samme Forhold til et tredje Led som Helheden

and in this way he introduces the syntactic functions of the constituents as a criterion for a correct coordination. Gazdar et al. (1985) also base their analysis

of coordinated structures on this observation. In GPSG, constituent structure information as well as the syntactic information can be accessed at the same time, and consequently in GPSG the coordination of unlike constituents is treated quite elegantly:



Unfortunately, we cannot just transfer this analysis directly to the Eurotrian model, because of the stratificational design of the system which implies that different information is computed at different levels. This means that we cannot access the information about the syntactic function of the constituents at ECS because it is not computed before the next level. We have to write rules for all possible combinations of constituents at ECS. The result is a provisional overgeneration at ECS before the validation of the constructions can take place at ERS and IS.

4.2 Incomplete Constituents

The most well known account for the coordination of incomplete constituents is probably the generative one advocated by Chomsky in 1965. According to Chomsky, a coordinate surface structure is derived by means of conjunction reduction from two parallel sentences in the deep structure.

Surface structure (1) Han elskede huset og haven

=>

Deep structure (2) Han elskede huset
(3) Han elskede haven

Simon Dik (1972) led this theory ad absurdum with an example like the following, which he claims leads to 81 different sentences in the deep structure.

Surface structure	(4)	John og Karl og Kurt sælger æbler og pærer og bananer i København, Odense og Århus mandag, tirsdag og onsdag
=>		
Deep structure	4.1.	John sælger æbler i København mandag.
	4.2.	Karl sælger pærer i Odense tirsdag.
	4.3.	Kurt ....
	4.4.	etc. til 81.

The question is now, whether conjunction reduction really is an irrelevant rule or whether the concept or some instance of it could be useful. From the point of view of machine translation one of the first things to be investigated is the translational relevance. Consider the following examples:

(5) I know the woman who painted _ and you met the man who stole the picture that Harry was so fond of _

(5a) jeg kender den kvinde som malede _ og du mødte den mand, som stjal det billede, som harry var så glad for _

(5b) * ich kenne die Frau, die _ malte und du trafst den Mann, der daß Bild stahl, das Harry _ so gern mochte

(5c) ich kenne die Frau, die das Bild, daß Harry so gern mochte, malte, und du trafst den Mann, der es stahl.

In the English and Danish sentences the rules for 'Across-the-board extraction' seem to be the same and we can produce a similar surface structure. In the translation into German (5b) it is obvious that the same mechanism does not work and that the sentence is ungrammatical. In German, different operations have to take place if we want to create an adequate translation. In order to do this, the German generation module must have access to the complete information.

(6) John offered _ and Harry gave Sally a Cadillac

(6a) John tilbød _ og Harry gav Sally en cadillac

(6b) *John bot _ (an) und Harry gab Sally einen Cadillac

In (6)–(6b) the same problem arises, here two constituents are extracted from the first conjunct and again an equivalent surface realization is impossible in German because of the detached preposition "an". (7) shows that this type of extraction is possible if the verb does not have a detached prefix.

(7) John verkaufte und Harry gab Sally einen Cadillac

In (8)–(8b) an equivalent structure cannot be build neither in Danish nor in German.

(8) john put the book away and _ the glass on the table

(8a) *john legte das buch weg und _ das glas auf den tisch

(8b) *john lagde bogen væk og _ glasset på bordet

(8aa) John legte das Buch weg und stellte das Glas auf den Tisch

The verb "put" is translated differently depending on the nature of the object it takes—either to "legen"/"lægge" or to "stellen"/"stille" in German and Danish. A new verb must be introduced in the two target languages to ensure the correct translation of the second part of the clause to match the semantic features of the object.

In these cases a completion of the incomplete constituents by filling the gaps would make the translation much easier.

5 Some Solutions

From the examples in section 4 it is obvious that in some cases gaps have to be filled at IS. We cannot be sure that the incomplete constituents have equivalents in the target language. The process of reduction and extraction is monolingually determined and heavily influenced by phenomena as surface syntax, homonymy and selectional properties of lexical items. This means that we have to leave it up to the target level generator to produce the correct degree of reduction on the basis of the maximal structure at IS. However, there is no need to treat every type of coordination as a reduction. This would lead to ridiculous multiplications of the structures as already noted by Dik.

Therefore, as a working strategy we are pursuing the following approach to these problems:

We distinguish two types of coordinate structures:

1. Coordination of arguments/modifiers
2. Coordination of governors

5.1 Coordination of Arguments/Modifiers

In case of coordination of arguments/modifiers the reduced structure is not rebuilt at IS, but simply kept as the coordination of two or more constituents as in figure 6, which illustrates the following sentence:

(9) John and Peter left

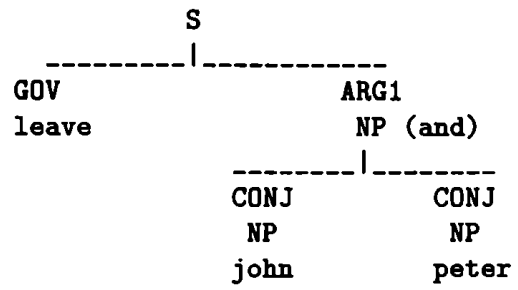


Figure 6:

5.2 Coordination of Governors

In case of coordination of governors (10) the missing arguments are inserted and coindexed with the corresponding constituents in the first conjunct in order to create complete structures as in figure 7, where the coordination of the two verbs is transformed into the coordination of two sentences. Only the valency bound arguments are copied to ensure the completeness and coherence of the structure according to the definition of the IS structure.

(10) We gathered and marched for several hours

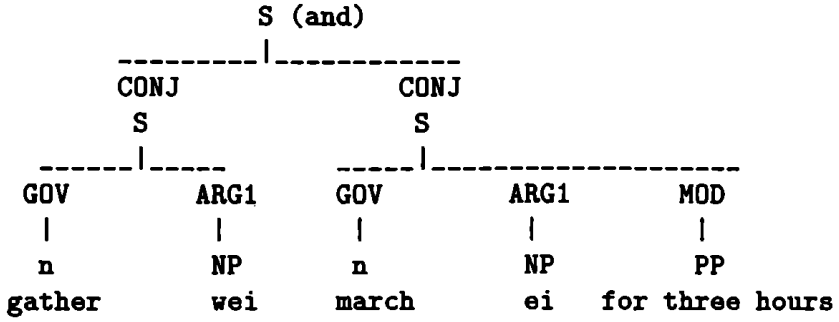


Figure 7:

The completion of the coordinated structure also takes place if more than one gap is found in the coordinated structure as in (8) above, where the second part of the structure consists only of the object (the glas) and the modifier (on the table). In this case both the subject and the verb are inserted and coindexed.

Thus, incomplete structures are only made complete if we are dealing with the coordination of governors or small clauses. We believe that a large amount of problem cases can be solved in this way.

6 Conclusion

Although the theory for basic coordination is well developed and well functioning in the EUROTRA system, there are still a number of problems that we have not mentioned here i.e. the role of negation in coordinate structures, the calculation of features, the determination of the categorial status of a coordinate structure that consists of unlike constituents etc. Some of these have been solved whereas others still are the topic of ongoing research.

For complex coordination the picture is more unclear since we have to deal with gapping both from a monolingual and a translational point of view. However, we believe that the comparative research will give fruitful input to the monolingual analysis.

References

Arnold, D. 1986. General view of the design methodology. *Multilingua* 5–3/1986.

Bresnan, J. [ed]. 1982. *The Mental Representation of Grammatical Relations.* MIT Press, Cambridge, Mass.

Chomsky, Noam. 1965. *Aspects of the theory of syntax.* MIT Press, Cambridge, Mass.

Diderichsen, Paul. 1946. *Elementær Dansk Grammatik.* Gyldendal, København.

Dik, Simon. 1972. *Coordination.* North-Holland, Amsterdam.

Gazdar, G. et. al. 1985. *Generalized Phrase Structure Grammar.* Harvard University Press.

Hansen, A. 1967. *Moderne dansk.* Grafisk Forlag, København.

Jaspaert, L. 1986. The levels of representation. *Multilingua* 5–3/1986.

Jørgensen, P. 1966. *Tysk Grammatik* I–III. Gad, København.

Lang, E. 1984. The Semantics of Coordination, *SLCS* 9. John Benjamins, Amsterdam.

Nirenburg, S. 1989. Knowledge based Machine Translation. *Machine Translation* vol. 3 and 4.

Perschke, S. 1986. Eurotra: General Overview. *Multilingua* 5–3/1986.

Raw, T. et al. 1989. An Introduction to the Eurotra Machine Translation System. *Working Papers in Natural Language Processing* vol. 1, Eurotra—Leuven.

EUROTRA-DK
Københavns Universitet
Njalsgade 80
2300 København S.
Danmark

GUÐRÚN MAGNÚSDÓTTIR

Collocations in Knowledge Based Machine Translation

Abstract

In order to generate colloquial language within the computational linguistics paradigm the problems of co-occurrence must be solved. As of yet research on co-occurrence has mostly focused on problems of syntax and selectional restrictions to describe the contextual relation within the sentence. Collocations and idioms have been neatly put aside as a unified problem to be dealt with in the lexicon or not at all.

In this paper collocations are defined according to the principles of semantics and a suggestion as to how to work on the retrieval of collocations, focussing on adjective noun constructions, from a text corpus will be made.

The research was carried out at the Center for Machine Translation, Carnegie Mellon University, together with Sergei Nirenburg and heavily inspired by Professor Allén's (Allén et al's 1975) work on collocations.

1 Defining Collocations

Idioms and collocations are two very different problems on a semantic level. The early definition of collocations (Firth 1957, Benson 1985) as being anything that frequently co-occurs can no longer be accepted. This definition discards the principle of syntactic atoms and would thus include such frequent patterns as 'it is' etc. Adding the constraint of atomicity would eliminate such patterns but would not be sufficient to distinguish between idioms and collocations.

Collocations are a string of words that co-occur under restrictions not definable by syntax nor selectional restrictions alone. These restrictions can be referred to as lexical restrictions since the selection of the lexical unit is not conceptual, thus synonyms cannot replace the collocate. The meaning of a collocation is compositional whereas the meaning of an idiom is not.

Collocations are compositional with hierarchical relations among the lexical units. The previous structural surface definition including a head, as the main word in the construction, and a collocate, as the supporting word is acceptable as is.

2 Detecting Collocations in a Text

A multi-word idiom often violates selectional restrictions due to metaphorical use of words whereas a collocation will not. Thus an idiom may be detected in failing parses where a collocation will be parsed undetected. Collocations are 'permitted patterns' in contrast to idioms that are often 'prohibited patterns' within the selectional restriction frame.

Permitted pattern: 'large coke'

Prohibited pattern: 'as large as life'

The borderline between the two is difficult to draw. Such questions as is 'shining truth' a collocation or an idiom are indeed not easily answered. Objects that 'shine' have the property TO_REFLECT_LIGHT and are +CONCRETE. The property list of 'Truth' does not include these and should they be added it would result in extreme over-generation together with the possibility of faulty parses. Thus 'Truth' cannot 'shine' except in a idiomatic sense and will therefore be treated as an idiom.

Ambiguities may arise between an idiomatic meaning and non-idiomatic. Similarly, ambiguities may arise between a collocational meaning and non collocational meaning as in the example:

Decide on a boat.

Where 'on' is a preposition or a collocate. Out of context the sentence has two meanings and there is no way of deciding which is the right one. In such cases contextual information is the only disambiguating factor. Thus the manual labor involved when working on collocations will involve going through the corpus to detect the collocations as well as systematically entering them in a lexicon.

Retrieving collocations is not an easy task for a native speaker, simply due to the fact that a collocation is the natural way of expression that is more easily detected through violations in generation, in the output from a natural language system or a non-native speaker. It would be futile to provide a human user with the same interactive knowledge acquisition tool to work on both collocations and idioms.

Consider the sentences

There is a little light in the window.
There is a small light in the window.

The lemma 'light' has three different lexemes in Longman's, lexeme one, the noun, has sixteen senses. Of these the first five are most likely to appear in technical texts.

Light (cat NOUN)	
sense 1: natural force U	property: QUANTITY
sense 2: source of light C	property: SIZE
sense 3: supply of light U	property: STRENGTH, QUALITY
sense 4: light (as time) U	property: QUANTITY
sense 5: set burning C	property: QUANTITY

The abbreviation 'U' stands for uncountable and 'C' for countable. The adjective little has four senses, linked to three scales.

Little—(cat ADJ)	
sense 1: small	scale: SIZE
sense 2: short - time	scale: TIME
sense 3: young	scale: AGE
sense 4: (idiomatic)	

The scale QUANTITY, which is probably the most frequently used meaning of 'little', is not present in the dictionary definition. It matches senses one, four, and five of 'light'.

The adjective 'small' has seven senses in Longman's. The last four have no relevance as to scaler meanings.

Small (cat ADJ)		
sense 1:	little in size,	scale: SIZE
	weight,	scale: WEIGHT
	force,	scale: STRENGTH
	importance	scale: IMPORTANCE
sense 2:	doing only a limited amount of X	scale: ACTIVITY
sense 3:	very little, slight (with U nouns)	scale: QUANTITY

Sense one of 'small' is very heavily compiled, for whatever reason. The scales of the two adjectives can be linked to the properties of the two concepts in a) and b).

a) Light ($IS-TOKEN-OF LIGHT)
b) Light ($IS-TOKEN-OF LIGHT-BULB)

Where a) represents sense one with the property QUANTITY, that given a modifier with a quantitative meaning, will give access to a certain position on the scale QUANTITY. In this case the position is represented by 'small' and 'little' as equivalent synonyms.

Sense two is represented by b) with the property SIZE that similarly gives the position on the scale SIZE, representing the equivalents 'small' and 'little'.

As for analysis this seems futile, since input texts are regarded as correct and the adjective is already present. However it is not possible to access an unambiguous result. Parsing at its best, and lexical mapping i.e. mapping surface expressions to concepts, will give two possible concepts, in a case where there is no ambiguity. Thus lexical restrictions would disambiguate between the concepts.

In generation the wrong choice of adjectives will lead to a wrong interpretation by the reader.

In order to be able to generate any information regarding these links the sentences need to be analyzed conceptually, that sort of analysis is only provided by knowledge based formalisms using ontologies and human interaction to ensure unambiguous results from the source language analysis.

3 Knowledge Based Machine Translation

At the Center for Machine Translation, Carnegie Mellon University, research has been carried out for some years on knowledge based machine translation. The most recent result is a prototype system (KBMT-89) delivered to the financier, IBM Japan, in february 1989.

The prototype system consists of different modules for natural language analysis, knowledge acquisition, and natural language generation. The system is an interlingua system relying on human interaction for disambiguation of multiple parsing results. The user is aided in the disambiguation process by the Augmentor, a specially built interaction component that allows the user to choose between the ambiguities at hand. The result is a meaning representation (interlingua) that is then the input of the generation component.

The analysis is based on a LFG like grammar together with the semantics present in the knowledge base or the ontology, the surface expressions are mapped on to the concepts in the ontology giving optimal grounds for knowledge acquisition.

Due to the fact that the result from the analysis and human interaction is unambiguous with links to the correct concepts, a filter for generation and disambiguation for the analyzed language can be generated. How this is to be systemized will be published in a forthcoming paper.

References

Allén, S. et al. 1975. *Nusvensk frekvensordbok baserad på tidningstext. 3. Ordförbindelser.* (Frequency Dictionary of Present-Day Swedish. 3. Collocations.) Stockholm.

Firth, J.R. 1957. *Papers in Linguistics 1934–1951.* Oxford: Oxford University Press.

Chomsky, N. 1965. *Aspects of the Theory of Syntax.* Cambridge, Mass.: MIT Press.

Cumming, S. 1987. *The Lexicon in Text Generation.* Dupl. 1987 Linguistic Institute Workshop: The Lexicon in Theoretical and Computational Perspective. Stanford University 1987.

Longman Dictionary of Contemporary English. 1987. Longman Group Ltd.

Magnúsdóttir, G. 1987. Fastcat Pilot-Study Report: Translation Systems and Translator Interviews. Språkdata.

Magnúsdóttir, G. 1988. Problems of Lexical Access in Machine Translation. In *Studies in Computer-Aided Lexicology.* Data Linguistica 18. Stockholm: Almqvist & Wiksell International.

Nirenburg et al. 1989. KBMT-89, project report. Center for Machine Translation. Carnegie Mellon University.

Department of Computational Linguistics,
University of Gothenburg,
412 98 Göteborg,
Sweden.
gudrun@hum.gu.se

SUSANNE NØHR PEDERSEN

The Treatment of Support Verbs and Predicative Nouns in Danish

Abstract

This paper will be a short examination of the notions of support verbs (e.g. 'make') and predicative nouns (e.g. 'estimation'). A support verb and a predicative noun form a support verb construction (e.g. 'Peter made an estimation of the damage').

The governing element in a support verb construction is the predicative noun whereas in a phrase like 'Peter wanted an estimation of the damage', the verb is the governing element. The main criteria for identifying support verb constructions as well as some of their properties will be mentioned.

From a translational point of view problems are likely to arise if support verb constructions are treated compositionally as the choice of support verb for a given predicative noun in the target language is not predictable from the source language support verb.

1 Introduction

One of the secondary goals of the Eurotra project has been to encourage collaboration between academics with different scientific backgrounds, with a consequent need to understand both alternative frameworks and more specific notions in those frameworks. This secondary goal of Eurotra has found a realisation in the work on support verb constructions.

The notion of support verb constructions originates from Harris (1962) and has later been used within the linguistic paradigm defined in Gross (1968, 1975, 1983, 1988). In Eurotra the notion has been presented by Laurence Danlos — Eurotra France who has been working with Maurice Gross. Since early spring this year language groups in France, Germany and Denmark have participated in an examination of the notion in their proper language.

The focus in this paper will be on cases where a support verb construction in the source language corresponds to a support verb construction in the target language.

In the following the translational relevance of support verb constructions will be treated in section 2, a definition and some criteria for support verb constructions will be given in respectively section 3 and 4. In section 5 a few properties will be mentioned and then finally in section 6 it will be discussed how support verb constructions may be represented in the Eurotra Interface Structure (IS).

The Danish examples given below are followed by literal English translations.

2 Translational Relevance

Apart from being 'full verbs' as e.g. 'Peter has a girl friend', a limited group of verbs can also be support verbs and thus be part of a support verb construction.

Typical support verbs are 'HAVE/HAVE' (indflydelse/influence), 'LAVE/MAKE' (en undersøgelse/an examination), 'FORETAGE/MAKE' (en undersøgelse/an examination), 'GØRE/MAKE'(brug/use), 'BEGÅ/COMMIT' (en forbrydelse/a crime), 'TAGE/TAKE' (forholdsregler/measures). What is common to these verbs is the fact that they are rather empty of meaning.

This means that these verbs are very likely to cause problems in machine translation. To illustrate this, let us consider a few French sentences and their corresponding translation into Danish:

(1) Jean FAIT UNE ESTIMATION DES DEGATS =>

Jean FORETAGER EN VURDERING af ødelæggelserne
(Jean makes an estimation of the damage)

(2) Jean FAIT UN RESUME du livre =>

Jean LAVER ET RESUME af bogen
(Jean makes a summary of the book)

(3) Jean FAIT UN CRIME contre Marie =>

Jean BEGÅR EN FORBRYDELSE mod Marie
(Jean commits a crime against Marie)

(4) Jean FAIT UNE CONVERSATION avec Marie =>

Jean FØRER EN SAMTALE med Marie
(Jean has a talk with Marie)

What these sentences share is that they all contain the French verb 'faire'. From the sentences it can be seen (not very surprisingly) that they are all translated into different verbs in Danish. This would cause a lot of problems if we

chose to translate compositionally (i.e. verb to verb and noun to noun). It would be extremely difficult — if not impossible — to make rules for how 'faire' in a given sentence should be translated.

The reasons why we do not solve the translational problem simply by making complex lexicon entries saying that e.g. the French 'faire_estimation' has to be translated into the Danish 'foretage_vurdering' are abundant. One reason is that this solution will lead to an enormous number of ad hoc lexicon entries. Another reason is that this treatment will only be a listing of problem cases and not a theoretically well-founded solution.

Thus as we reject the idea of complex lexicon entries and cannot predict the translation of the source language verb on the basis of the verb itself, we have tried to take the noun as our starting point.

3 Definition of a Support Verb Construction

A support verb construction (SVC) as e.g.:

(5) Peter foretager en vurdering af ødelæggelserne
(Peter makes an estimation of the damage)

consists of a support verb (foretage/make) and a predicative noun (vurdering/estimation).

The predicative noun is a noun that has a valency frame and thus can have complements like verbs. A support verb has no frame of its own. Instead a support verb 'inherits' the frame of the predicative noun. Consequently, it is the predicative noun that is the frame-bearing element and NOT the support verb 'foretage/make'.

4 Criteria for Identification of Support Verb Constructions

A) To a given support verb construction:

(5) Peter foretager en vurdering af ødelæggelserne
(Peter makes an estimation of the damage)

there must correspond a predicative noun group, e.g.

(6) Peters vurdering af ødelæggelserne
(Peter's estimation of the damage)

whose head is a predicative noun. In such a predicative noun group the 'agent' (called arg1) is in genitive and the element which corresponds to the object for the action 'vurdering/estimation' (called arg2) is normally realized as a 'pp' with the preposition 'af' (of).

The content of a sentence with a full verb as e.g. 'omtale' (mention):

(7) Peter omtaler en vurdering af ødelæggelserne
(Peter mentions an estimation of the damage)

cannot be fully rendered by a predicative noun group as (6):

(6) ≠ Peters vurdering af ødelæggelserne
(Peter's estimation of the damage)

An ordinary verb has more 'meaning' than can possibly be rendered by a predicative noun group.

So the first criteria is that there must be a reference identity between a predicative noun group and a SVC-construction in contrast to a construction with a 'full' verb.

B) In some sense B) follows from A) in that a support verb construction does NOT accept an insertion of a logical subject for the predicative noun that is different from the grammatical subject of the sentence:

(8) *Peter foretager Johns vurdering af ødelæggelsen
(Peter makes John's estimation of the damage)

Thus a support verb construction such as:

(9) Peter foretager en vurdering af ødelæggelserne
(Peter makes an estimation of the damage)

differs from a construction with an 'ordinary' full verb as in:

(10) Peter omtaler en vurdering af ødelæggelserne
(Peter mentions an estimation of the damage)

where a subject for the predicative noun can be added without changing the acceptability of the sentence:

(11) Peter omtaler Johns vurdering af ødelæggelserne
(Peter mentions John's estimation of the damage)

So the second criteria is that it must not be possible to insert a logical subject for the predicative noun which is different from the subject of the verb.

5 Properties

Apart from the criteria just mentioned, support verb constructions in Danish are characterized by a number of properties. The following properties are NOT criteria used for identification of support verb constructions — however they reveal a clear tendency with respect to behavior of support verb constructions. But as we can only speak of tendencies they cannot be used as criteria.

5.1 Clefting

5.1.1 Constructions with the Preposition 'af' (of)

Let's once again go back to sentence (1) and (3) to see what happens if the sentences are cleft:

(12) Det er en vurdering af ødelæggelserne, Peter foretager

[Npred prep N]

(It is an estimation of the damage, Peter makes)

(13) Det er en vurdering af ødelæggelserne, Peter omtaler.

[Npred prep N]

(It is an estimation of the damage, Peter mentions).

This shows that the sequence Npred prep N is one constituent (one complex nominal group). However, it is also possible to place 'vurdering' (estimation) and 'ødelæggelse' (damage) separately in focus:

(14) Det er en vurdering, Peter foretager af ødelæggelserne.

(It is an estimation Peter makes of the damage).

(15) Det er en vurdering, Peter omtaler af ødelæggelserne.

(It is an estimation Peter mentions of the damage).

(16) Det er af ødelæggelserne, Peter foretager en vurdering.

(It is of the damage Peter makes an estimation).

(17) Det er af ødelæggelserne, Peter omtaler en vurdering.

(It is of the damage Peter mentions an estimation).

The examples in (12)–(17) are perhaps not interesting observations in themselves as they are all considered to be correct sentences in Danish. But they ought to be compared with the examples given below.

5.1.2 Constructions where the Preposition is Different from 'af' (of)

In cleft sentences where the preposition is different from 'af' (of) a different result can be observed:

(18) Det er et overfald på Marie, Luc har begået.

(It is an attack on Marie, Luc has committed)

(19) Det er et overfald på Marie, Luc har omtalt.

(It is an attack on Marie, Luc has mentioned).

The difference can be observed when 'overfald' (attack) and 'på Marie' (on Marie) are placed separately in focus:

(20) Det er et overfald, Luc har begået på Marie.
(It is an attack, Luc has committed on Marie).

(21) *Det er et overfald, Luc har omtalt på Marie.
(It is an attack, Luc has mentioned on Marie).

(22) Det er Marie, Luc har begået et overfald på.
(It is Marie, Luc has committed an attack on)

(23) *Det er Marie, Luc har omtalt et overfald på.
(It is Marie, Luc has mentioned an attack on)

In contrast to ordinary full verbs, support verb constructions accept a splitting when the preposition is different from 'af' (of). However it should be examined more carefully before we conclude that all prepositions different from 'af' (of) behave in this way.

5.2 WH-questions

It is not possible to ask wh-questions to a support verb construction:

(24) *Hvad har Peter taget i dag? Et initiativ.
(What has Peter taken today? An initiative).

(Unless the question is asked in a conversation by the person listening where he/she did not grasp the content at first.)

This is however fully acceptable in connection with an ordinary full verb:

(25) Hvad spiser du i dag? Kager fra kantinen.
(What do you eat today? Cakes from the cantine).

5.3 Modification of SVCs by Adjective/Adverb

Often in connection with adverbs of manner the following possibility for making a paraphrase can be observed:

(26) Peter foretog en hurtig/omhyggelig vurdering af ødelæggelserne.
(Peter made a quick/careful estimation of the damage)

=>

(27) Peter foretog hurtigt/omhyggeligt vurdering af ødelæggelserne.
(Peter made quickly/carefully an estimation of the damage)

In rare cases modification by means of adjectives and adverbs is also found in connection with ordinary full verbs as e.g.:

(28)a. Peter drak en hurtig kop kaffe.
(Peter drank a quick cup of coffee)

b. Peter drak hurtigt en kop kaffe.
(Peter drank quickly a cup of coffee)

but is not possible without changing the meaning when the full verb is followed by a predicative noun:

(29)a. Peter omtalte et hurtigt overfald på Marie.
(Peter mentioned a quick attack on Marie)
≠

b. Peter omtalte hurtigt et overfald på Marie.
(Peter mentioned quickly an attack on Marie)

6 Aspectual Variants

A given predicative noun often has the possibility of being 'supported' by different verbs as e.g.:

(30) Peter har ansvaret for rapporten
(Peter has the responsibility for the report)

(31) Peter tager ansvaret for rapporten
(Peter takes the responsibility for the report)

(32) Peter beholder ansvaret for rapporten
(Peter keeps the responsibility for the report)

The different sentences above express differences with respect to aspectual values. Aspectual variants are called Vasp. In (30) the Vasp is 'neutral', in (31) 'inchoative', in (32) 'durative'. Typically a predicative noun does not have all aspectual combinations. Thus a predicative noun such as 'angreb' (attack) is not likely to have an aspectual variant expressing an 'iterative' aspect or a 'terminative' aspect.

If the sentences (30) to (32) are transformed into relative clauses where the predicative noun is the antecedent of the relative pronoun as in:

(33) Ansvaret, som Peter har for rapporten
(The responsibility that Peter has for the report)

(34) Ansvaret, som Peter får for rapporten
(The responsibility that Peter gets for the report)

(35) Ansvaret, som Peter beholder for rapporten
(The responsibility that Peter keeps for the report)

a difference appears. Only (33) renders fully the meaning expressed by the active nominal group, 'Peters ansvar for rapporten' (Peter's responsibility for the report). And this is what distinguishes a Vsup with Vasp=neutral from a Vsup with Vasp different from 'neutral'.

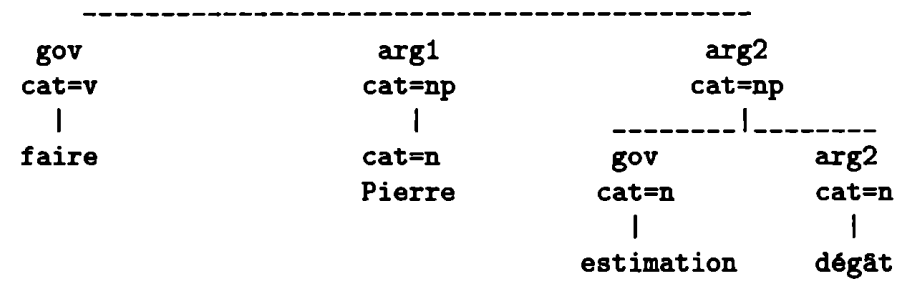
6.1 Which Factors Decide the Choice of Vasp?

On the one hand it is the predicative noun that selects its support verb. On the other hand the support verb with Vasp = neutral decides which aspectual variants a given support verb construction can have.

7 How to Represent Support Verb Constructions?

As pointed out in section 2, the following IS (interface structure) representation of e.g. 'Jean fait une estimation des dégâts' is likely to cause severe problems:

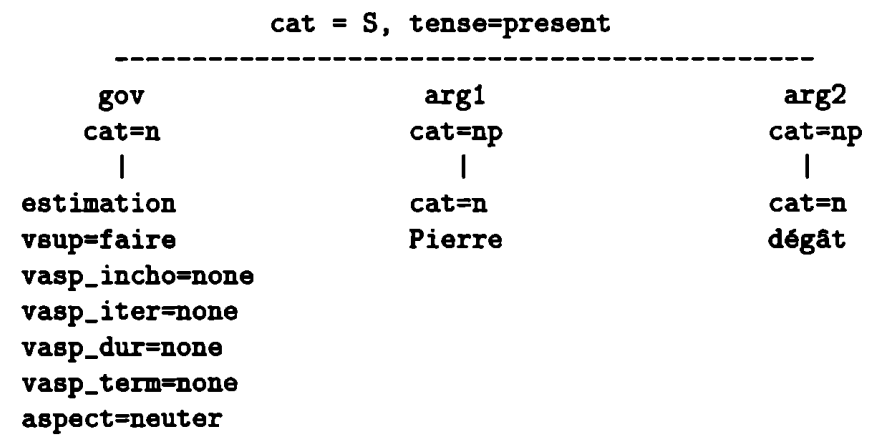
(36)



as there is not only one possible translation into Danish of the French verb 'faire'. Let us assume that we have four different translation possibilities (see the examples (1)–(4)) of 'faire' in the transfer lexicon between French and Danish. The consequence will be that we get four results in Danish that are not all correct translations of the input sentence.

If instead as in (37) the predicative noun (the frame bearing element) becomes governor of the sentence:

(37)



a simple translation — from FR:estimation to DA:vurdering, from FR:Pierre to DA:Peter, etc. — can take place in the transfer module between the two languages.

Next, on IS synthesis the noun 'vurdering' is looked up in the Danish monolingual lexicon where the dictionary entry for 'vurdering' contains information about which support verb(s) the predicative noun can be constructed with. The Danish IS representation will be the following:

(38)

```
                cat = S, tense=present
        ------------------------------------------
          gov                arg1                arg2
        cat=n              cat=np              cat=np
          |                  |                   |
      vurdering            cat=n               cat=n
      vsup=foretage        Peter           ødelæggelse
      vasp_incho=none
      vasp_iter=none
      vasp_dur=none
      vasp_term=none
      aspect=neuter
```

Finally between IS synthesis and ERS (i.e. Eurotra Relational Structure) an insertion of the Danish support verb as governor of the sentence takes place with the result that on ERS and ECS (i.e. Eurotra Constituent Structure) the structure of the support verb construction is the same as the structure of a sentence with any other verb.

8 Final Remarks

We have now demonstrated that the notion of support verb constructions is also applicable to Danish. Moreover we have shown that the notion is also useful in machine translation because it can be left to the target language to generate and insert the correct support verb in a sentence.

References

Danlos, L. 1988. *Les noms prédicatifs et les phrases à verbe support.* Eurotra-France.

Danlos, L. 1985. *Génération automatique de textes en langue naturelle.* Masson, Paris.

Gross, M. 1968. *Grammaire transformationnelle du français: Syntaxe du verbe.* Larousse, Paris.

Gross, M. 1975. *Méthodes en syntaxe.* Hermann, Paris.

Gross, M. 1983. A linguistic environment for comparative Romance syntax. In P. Baldi (ed.): *Papers from the XIIth Linguistic Symposium on Romance Language.* John Benjamins, Amsterdam.

Gross, M. 1988. *Constructing lexicon-grammars.* Laboratoire d'Automatique Documentaire et Linguistique.

Harris, Z. S. 1962. *Structural linguistics.* Phoenix, USA.

EUROTRA-DK
University of Copenhagen
Njalsgade 80
DK-2300 Copenhagen S.
Denmark

KLAUS SCHUBERT

Kunskap om världen eller kunskap om texten?

En metod för korpusstödd maskinöversättning

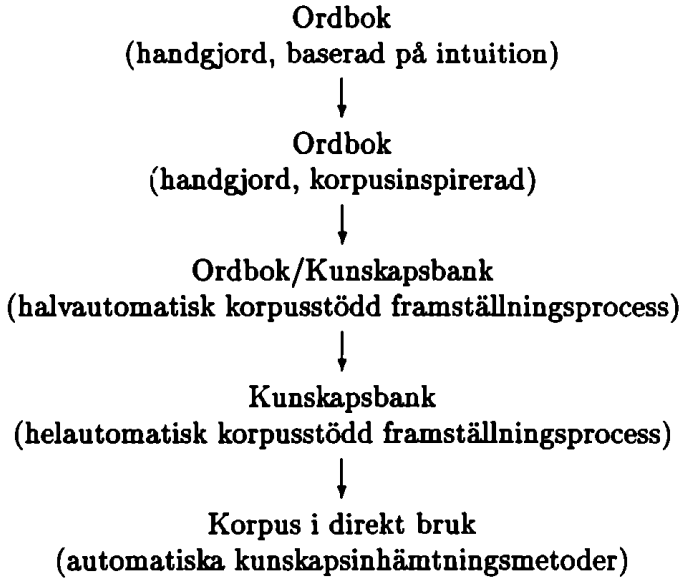
Abstract

World Knowledge or Text Knowledge?
A Method for Corpus-based Machine Translation

As the scope and the quality demands on machine translation systems increase, the developers tend to direct their efforts not only on automating the translation process itself but also on automating the more labour-intensive subprocesses of the system development. This is the reason for a tendency towards more and more automated techniques of knowledge acquisition. Whereas commercial systems at present normally have dictionaries or knowledge banks which were generated in at best semi-automatic corpus-based or corpus-inspired ways, some of the more advanced research projects attempt to approach fully automatically generated knowledge banks. From this idea it is a logical step to using a corpus directly as a knowledge source for the machine translation process. For the DLT system of BSO in Utrecht a method is developed in which the translation process relies entirely on a Bilingual Knowledge Bank as its one and only source of translation-relevant knowledge. The Bilingual Knowledge Bank consists of parallel corpora of texts in a given source/target language and DLT's intermediate language Esperanto. The texts are represented as dependency trees with the sentence trees being linked up by text-grammatical pointers (e.g., for deixis, reference, event chains). Corresponding elements in the parallel versions of the text are combined to form translation units. The translation process is carried out by means of various sets of generalization rules which apply the specimen translation units to a given text to be translated. Such generalizations are made at the levels of monolingual syntax, metataxis (syntactic transfer) and semantics/pragmatics. Since the knowledge acquisition process is not carried out before the translation function requires a certain bit of information, it can be dynamically steered by the information contained in the part of the text already translated.

1 Kunskapskällor för avancerad maskinöversättning

Att översätta är en krävande intellektuell handling. Försöken att bygga maskinöversättningssystem kan betraktas som ett strävande efter en automatisering av denna ytterst komplexa verksamhet. Ett par decenniers erfarenheter med denna strävan har visat att det inte räcker att automatisera bara själva översättningsprocessen. Redan i uppbyggningen av ett maskinöversättningssystem ingår så komplexa arbetssteg att även dessa måste utföras i stor utsträckning automatiskt eller åtminstone på ett avancerat datorstött sätt. Givetvis riktar sig detta sekundära automatiseringsintresse i första hand på de mest arbetsintensiva stegen i systemutvecklingsprocessen. Detta är för de flesta systemens vidkommande de lexikografiska (samt terminografiska) arbetsmomenten. Mera allmänt sagt gäller det i dessa moment att genom lexikografi eller på annat sätt inhämta den kunskap som behövs för översättningsprocessen och att göra kunskapen tillgänglig för datorsystemet. Jämför man de maskinöversättningssystem som har byggts eller projekterats sedan ett fyrtiotal år tillbaka, så kan man iaktta en utveckling i kunskapskällorna som står i direkt samband med nödvändigheten att automatisera själva kunskapsinhämtningsprocessen. Utvecklingen gäller både kunskapskällans innehåll och kunskapsinhämtningssättet. Tar man med även tilltänkta framtida innovationer så förlöper utvecklingen (något förenklat) så här:



De första två utvecklingsstegen var mycket vanliga i maskinöversättningens första decennier. Dagens system har för det mesta uppnått ett sådant omfång och sådana kvalitetskrav att rent handarbete har blivit ogörligt. Fullständigt handgjorda ordböcker förekommer däremot även i dag i maskinöversättningssystem som är relativt små, dvs system som antingen är avsedda enbart för experimentellt bruk eller som är inskränkta till en mycket snäv ämnesdomän.

Större system med friare textsort som är beräknade för praktiskt bruk har ofta en kunskapsinhämtningsmetod som motsvarar det tredje utvecklingssteget: Ett programsystem analyserar en korpus som vanligtvis redan är försedd med tillagd disambigueringsinformation och genererar ur korpusmaterialet lexikoningångar i maskinöversättningssystemets speciella format. Ingångarna granskas sedan av en människa och godtas eller korrigeras och kompletteras. I stället för (eller vid sidan om) en korpus av löpande text används ibland även vanliga ordböcker i bokform som görs tillgängliga för databehandling genom optisk inläsning eller konvertering av datafiler från sätt- och tryckmaskiner.

2 Automatisera systemutvecklingen?

Mig veterligen har det hittills inte marknadsförts något maskinöversättningssystem som bygger på en mera framskriden teknik än den jag här beskriver som det tredje steget. Däremot försöker man i somliga av de mest avancerade nu pågående forsknings- och utvecklingsprojekten att uppnå det fjärde eller till och med det femte steget. Resonemanget är enkelt: Granskningen av automatiskt genererade lexikoningångar är ett tidsödande och därmed dyrt arbete. Det ligger därför nära till hands att rikta automatiseringsintresset igen på det mest arbetsintensiva momentet och försöka att göra granskningsarbetet överflödigt. Om detta vore möjligt utan att ge avkall på översättningskvaliteten, skulle mycket vara vunnet.

Sådana automatiskt framställda lexikoningångar utgör det fjärde utvecklingssteget. Innan man satsar på detta, lönar det sig att föra tankeexperimentet vidare. Om det skulle visa sig vara möjligt att helautomatiskt generera lexikoningångar med utgångspunkt i en korpus, så betyder detta att den information man behöver för att kunna översätta finns i korpustexten och kan hämtas därifrån på ett helautomatiskt sätt. "Helautomatiskt" betyder i detta sammanhang framför allt att ingen kunskap behöver läggas till av människan. Om detta är så, då kan man eventuellt lika gärna låta bli att framställa ingångar och i stället anlita korpusen direkt som kunskapskälla. Detta är det femte steget i kunskapskällornas utveckling.

Tabellen ovan presenterar det femte steget som en fortsättning eller vidareutveckling av det fjärde, men med tanke på programsystemens storlek och snabbhet undrar man kanske om det inte snarare är ett steg tillbaka. Om det fjärde och det femte steget är likvärdiga, kan det då överhuvudtaget ha någon mening att diskutera det femte där man är tvungen att lagra en hel korpus i stället för några redundansfria ingångar? Är inte den kompaktare lösningen utan vidare att föredra? Jag beskriver nedan en lösning som siktar på det femte steget, så att det är på sin plats att skaffa sig klarhet om det precisa förhållandet mellan helautomatisk ingångsgenerering och direkt korpusbruk. Om det fjärde steget är möjligt, så betyder det att man kan generera för översättningsbehov tillräckliga lexikoningångar ur en korpus med hjälp av en på förhand fastställd uppsättning regler. Tankeexperimentet går ut på att man ur en given korpus får fram samma information, oavsett tidpunkten på vilken man tillämpar reglerna.

Det spelar alltså i detta avseende ingen roll om reglerna används innan eller medan det föreligger en konkret översättningsuppgift. Med andra ord, om man överhuvudtaget kan inhämta den nödvändiga informationen helautomatiskt, så har man friheten att välja om man vill förlägga inhämtningsprocessen till den förberedande systemutvecklingen eller till själva översättningsprocessen.

Det lönar sig inte att föra detta tankeexperiment vidare om inte det femte steget erbjuder väsentliga fördelar jämfört med det som är möjligt redan på det fjärde. Man måste ha mycket övertygande argument när man vill avstå från möjligheten att undangöra en så svår delprocess som korpusstödd kunskapsinhämtning onekligen är redan i utvecklingsfasen och i stället uppskjuta den till själva översättningsprocessen i runtime. Det enda giltiga kan vara ett argument som bygger på viktig tillagd information som blir tillgänglig först när översättningsprocessen har kommit igång. Bara om kunskapsinhämtningsprocessen kan styras eller avsevärt förbättras genom kunskap eller villkor hämtade ur den text som är under bearbetning, då kan det löna sig att tänka på en lösning på femte steget.

I den lösning jag skisserar i avsnitt 3 t o m 5 är kunskapskällan och den redan översatta delen av texten representerade i samma ytnära format. Bl a detta gör det möjligt att genomföra frekvensberäkningar, probabilistiska kontextjämförelser och liknande delprocesser på ett specifikt sätt som är anpassat till den konkreta kontexten och som dynamiskt tar med i beräkningen den kunskap som kan inhämtas ur den redan behandlade textdelen. På detta sätt blir den kunskapsbehandlingsprocess som stöder översättningsfunktionen i hög grad styrd av ett välavvägt samspel mellan den allmänna och den för tillfället mest relevanta speciella kunskapskällan.

Utöver fördelar som kan uppnås genom en kunskapsinhämtningsprocess i runtime finns det ytterligare en anledning att intressera sig för det femte steget. Denna anledning har i beskrivningen ovan någorlunda dolts av framställningssättet i den femstegiga utvecklingen. Jag har hittills bara diskuterat det femte steget under förutsättning att det fjärde är genomförbart, och jag har i tämligen allmänna ordalag talat om den information man kan inhämta med de två antydda metoderna: På det femte steget är denna information minst likvärdig med den man får på det fjärde, och det finns anledning att anta att det därutöver är möjligt att inhämta tillagd information som bara är tillgängligt på det femte steget. Detta resonemang får emellertid inte dölja den kvalitativa skillnad som ändå består mellan det fjärde och det femte steget. Skillnaden blir tydlig när man går närmare in på i vilken form informationen lagras. På fjärde steget utvärderas korpusen för att generera lexikoningångar. Processens utdata är alltså en fastlagd representation för den inhämtade kunskapen. Kunskapen representeras sålunda på ett explicit sätt. En lexikoningång skall vara tillämplig på vilka som helst förekomster av uppslagsordet. (I stället för ett uppslagsord kan det givetvis vara fråga om en annan enhet, t ex ett morfem, ett syntagm osv.) När man genererar lexikoningångar, är det meningen att uppnå en så allmängiltig och täckande beskrivning av uppslagsordet som möjligt (eller några få sådana). På femte steget däremot används korpusen direkt som kunskapskälla, och en korpus är av en kvalitativt annorlunda karaktär än en lexikoningång. Medan en lexikon-

ingång skall vara allmängiltig i den mån detta är möjligt, innehåller en korpus enbart exempel. Medan alltså ett system av fjärde steget går från exemplen i den underliggande korpusen genom härledningsregler till en i denna speciella bemärkelse allmängiltig lexikoningång och därifrån genom tillämpningsregler till den konkreta översättningsuppgiften, så läggs på femte steget ett omedelbart förband mellan exemplen och uppgiften. Det allmängiltiga mellansteget kan falla bort.

Detta är en väsentlig iakttagelse. För att bevisa tillämpligheten av det femte steget behöver man alltså inte förutsätta att det fjärde är möjligt. Det räcker att bevisa att man ur korpusexempel direkt kan härleda den information som behövs för översättningsprocessen.

3 DLT:s tvåspråkiga kunskapsbank

För maskinöversättningssystemet DLT projekteras numera en korpusstödd kunskapsbehandlingsmetod som strävar efter att närma sig skalans femte steg.

Innan jag tar upp metoden något mera i detalj kan ett par inledande ord över DLT vara nödvändiga. *Distributed Language Translation* (DLT) är namnet på ett maskinöversättningsprojekt som bedrivs av det nederländska mjukvaruföretaget Buro voor Systeemontwikkeling (BSO/Research) i Utrecht, delvis med statligt anslag. Efter en förstudie (Witkam 1983) inträdde DLT år 1985 i implementeringsfasen. Den första prototypen blev färdig 1987, den andra 1988. DLT skall bli ett mångspråkigt system, bl a för tillämpningar i datakommunikationsnät. Under utgångsspråksanalysen förs en systeminitierad disambigueringsdialog med användaren. Dialogfrågorna ställs på utgångsspråket och det krävs ingen posteditering, så att användaren inte behöver känna till målspråken. Förbindelselänken mellan utgångs- och målspråken är mellanspråket esperanto.

De första prototypversionerna översätter från engelska genom esperanto till franska. Som kunskapskällor anlitar de tre morfosyntaktiska ordböcker (engelska, esperanto, franska), två tvåspråkiga metataxordböcker (engelska-esperanto, esperanto-franska; om termen *metatax* jfr Schubert 1987) och en enspråkig lexikal kunskapsbank (esperanto). De olika framställningsprocesserna låg mellan det första och det tredje steget på skalan (jfr om prototypens arkitektur: Schubert 1986; om kunskapsbanken: Papegaaij 1986).

Utvärderingen av erfarenheterna med prototyperna har lett fram till en vidareutveckling av kunskapskällorna som betyder en ingripande förändring i systemet DLT:s sätt att fungera. DLT är (redan i prototypversionerna) ett modulärt system. Det består av språkparsmoduler som alltid har esperanto på den ena sidan. En text som översätts till ett enda målspråk passerar på så sätt två sådana språkparsmoduler. Det är bl a på grund av denna arkitektur som systemet kan betraktas som ett dubbelt direkt översättningssystem (Schubert 1988). I DLT:s tredje systemversion, som befinner sig i planerings- och modellimplementeringsfasen, har alla kunskapskällor som ingår i samma språkparsmodul sammanfattats till ett enda system. Detta system har fått namnet *Tvåspråkig kunskapsbank.*

DLT:s Tvåspråkiga kunskapsbank består av en parallell korpus, dvs en och samma text parallellt på två språk (original och översättning, även två översättningar från ett tredje språk). I det pågående provimplementeringsarbetet ingår paren engelska-esperanto och franska-esperanto. I den slutgiltiga implementeringen av den tredje systemversionen skall minst två språk komma till; senare versioner projekteras för två paket av sex språk var, varefter flera språk kan läggas till efter behov tack vare DLT:s modulära mellanspråksarkitektur. Jag illustrerar den Tvåspråkiga kunskapsbanken nedan med språkparet danska-esperanto.

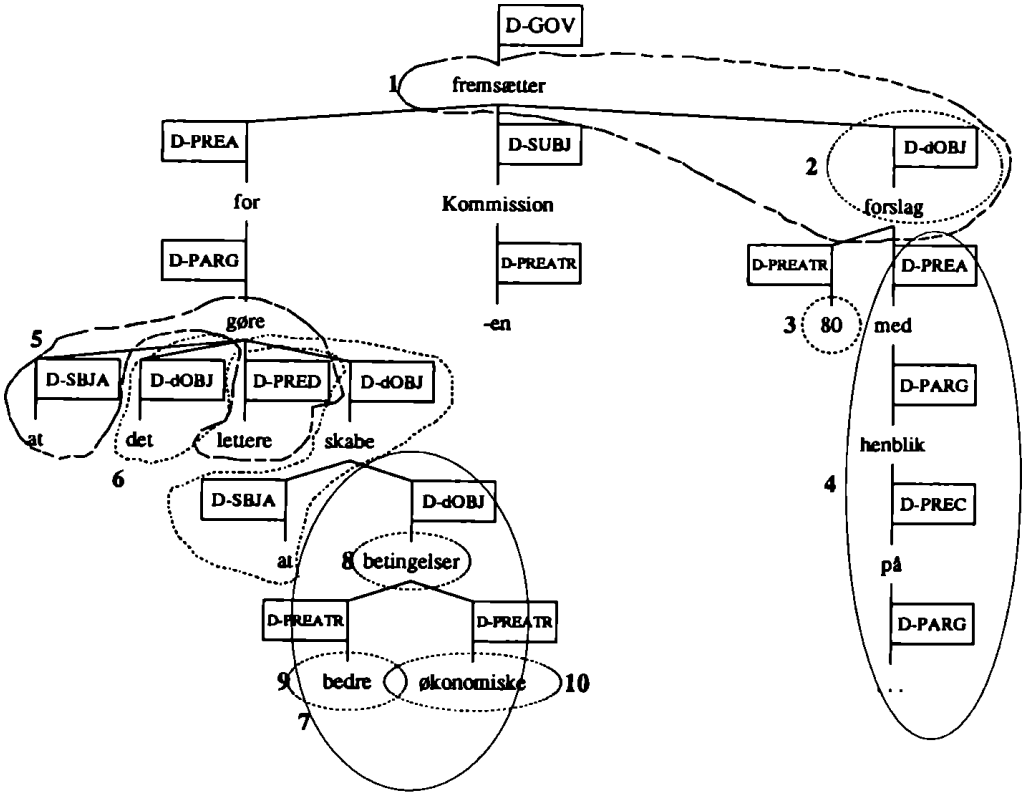
Korpustexterna lagras i den Tvåspråkiga kunskapsbanken i disambiguerad form. Den grundläggande representationsformen är dependenssyntaktiska träddiagram, utdata av en parser (jfr Schubert 1987: 28–129). Dessa är syntaktiskt oambiguösa. Textgrammatiska pekare (deixis, referens, skeendekedjor m m) förbinder satserna och meningarna till sammanhängande texter. Vid sidan om dessa enspråkiga markörer är texterna försedda med speciella tvåspråkiga pekare som bygger upp översättningsenheter. En översättningsenhet är ett ord, en ordgrupp eller bara ett morfem med dess motsvarighet i det andra språket. Markörerna för de syntaktiska relationerna som är utsatta i dependensträdet ingår även i översättningsenheten. En större översättningsenhet kan innehålla mindre enheter. Enheterna är dock inte mindre än att den översättningsmotsvarighet de innehåller kan användas även i andra kontexter.

4 En illustration

Som illustration använder jag följande mening på danska och esperanto. Träddiagrammen bygger på dependenssyntaxerna som har utarbetats enligt DLT-modell för danska (Ingrid Schubert 1989) och esperanto (Schubert 1989) där de här använda etiketterna för dependensrelationer förklaras i detalj. Av utrymmesskäl tar jag bara en enda mening och visar bara den övre delen av träddiagrammen. Trädavsnitt med samma nummer i båda träden utgör översättningsenheter. För översiktlighetens skull markerar jag långt ifrån alla översättningsenheterna. Textgrammatiska pekare utelämnas helt.

> For at gøre det lettere at skabe bedre økonomiske betingelser, fremsætter Kommissionen 80 forslag med henblik på at nedbryde markedsskrankerne og sætte virksomhederne i stand til fuldt ud at drage fordel af den europæiske dimension.
>
> Por faciligi la kreadon de pli bonaj ekonomiaj kondiĉoj la Komisiono faras 80 proponojn, kiuj celas faligi la barilojn de la merkato kaj ebligi al la entreprenoj maksimume eluzi la avantaĝojn de la eŭropa dimensio.

(Med hänsyn till entydighet i ordstrukturen markeras morfemgränserna i DLT:s mellanspråk som är fullständigt agglutinerande. Morfemtecknen ses i esperantoträdet med har utelämnats här.)



Figur 1:

I denna mening förekommer både enkla och komplexa översättningsenheter. Till de enklare hör ettordsenheterna:

[10] økonomiske — ekonomiaj

Översättningsenheten

[2] forslag — proponojn

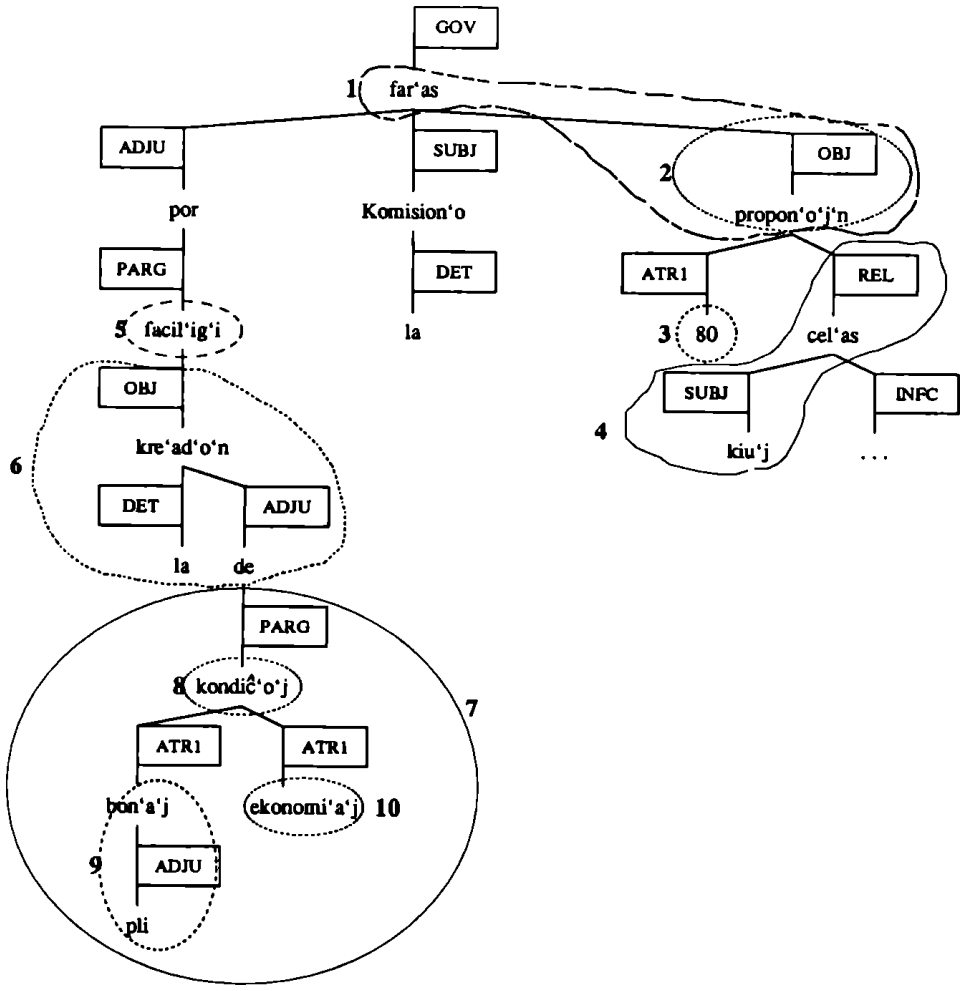
innehåller på esperantosidan en ackusativ (*-n*), så att en giltig motsvarighet bara föreligger när objektetiketten tas med i enheten.

Observera att *forslag* motsvarar *proponojn* men att *fremsætter* inte motsvarar *faras*. Verbet *fari* betyder 'göra' och kan inte betraktas som brukbar översättning av *fremsætte* i andra kontext. Därför omfattar enheten flera ord:

[1] fremsætter forslag — faras proponojn

En mera komplex motsvarighet är

[5] at gøre lettere -- faciligi



Figur 2:

Även mera ingripande syntaktiska förskjutningar kan iakttas i denna mening:

[4] med henblik på — kiuj celas

Detta är en övergång från ett prepositionellt tilläggsled (D-PREA) i danska till en relativ bisats (REL) i esperanto (*kiuj* 'vilka', *celas* 'avser').

Översättningsenheten

[6] det at skabe — la kreadon de

illustrerar bra att korpusinformationen har exempelkaraktär. Givetvis översätter man inte alltid infinitivkonstruktionen *det at skabe* med en substantiv (*la kreado* 'skapandet'), men formen kan mycket väl tänkas förekomma i vissa andra kontexter. Informationen är alltså inte allmängiltig, men illustrerar bara en av många möjliga översättningar för detta syntagm.

5 Översättarens kunnande i maskinöversättningssystemet

Det esperanto-danska meningsparet är godtyckligt valt. Att illustrationen utöver enkla fall som [10] eller [3] även innehåller så många strukturellt och lexikalt intressanta motsvarigheter är ett tecken på att en korpusstödd översättningsmetod har tillgång till en mängd konstruktioner och motsvarigheter som i denna form aldrig upptas i vanliga ordböcker. Här återspeglas i implicit form översättarens kunnande.

I DLT:s kunskapsbank ingår en stor mängd exempel på hur vissa textbitar i konkreta tillfällen har översatts av fackkunniga översättare. "Uppfinnaren" av den (patentanmälda) Tvåspråkiga kunskapsbanken, Victor Sadler, beskriver i detalj hur man kan översätta med hjälp av denna kunskap (Sadler 1989). Hans framställning, som inte kan återges här, vill jag i korthet komplettera med två aspekter som kan belysa denna korpusstödda översättningsmetod med utgångspunkt i resonemanget om den femstegiga skalan i maskinöversättningssystemens utveckling. Den första aspekten beträffar uppbyggningen av en Tvåspråkig kunskapsbank och den andra tillämpningsreglerna.

I diskussionen om DLT:s förnyade systemarkitektur bör två funktioner hållas klart åtskilda: å ena sidan kunskapsbehandlingen under översättningsprocessen i runtime och å andra sidan kunskapsinhämtningen. Den senare processen, i vilken uppbyggningen av den Tvåspråkiga kunskapsbanken ingår, är förlagd till DLT-"fabriken", medan själva översättningsprocessen givetvis äger rum hos användaren. Även om det naturligtvis är önskvärt att automatisera kunskapsinhämtningsprocessen i hög grad, är det dock inte uteslutet att utföra manuella ingripanden och att anlita specialisthjälp som inte är tillgänglig under översättningsprocessen. På ett liknande sätt kan inhämtningen också utnyttja större och annorlunda datorsystem än användarmodulerna. Under översättningsprocessen är den enda mänskliga hjälp systemet kan få de svar som ges i den interaktiva disambigueringsdialogen. Dialogfrågorna ställs på utgångsspråket och måste kunna besvaras av språk- och datavetenskapliga lekmän. Denna skillnad förklarar hur det är möjligt att den Tvåspråkiga kunskapsbanken i systemutvecklingsfasen förses med all den utomspråkliga information jag ovan har nämnt: oambiguösa syntaktiska träd, textgrammatiska pekare, översättningsmotsvarigheter m m. I DLT-fabriken genereras dessa strukturer automatiskt, t ex genom parsning, i den mån detta är möjligt och sedan granskas och kompletteras de av människor. Mänskligt arbete behövs framför allt för identifieringen av översättningsmotsvarigheterna. Medan DLT-användaren kan vara en enspråkig person som bara förstår utgångstexten, kräver kunskapsinhämtningsprocessen specialistarbete, bl a av yrkesöversättare. Den Tvåspråkiga kunskapsbanken är alltså inte inskränkt till den kvalitet i betydelseanalysen som i dagens läge kan uppnås med helautomatiska kunskapsbehandlingsprocesser, utan den har tillgång till mänsklig fackkunskap.

Den andra aspekt som jag vill nämna här gäller generaliseringsfunktionen. Som ett speciellt slags korpus innehåller den Tvåspråkiga kunskapsbanken ex-

empel på faktiskt bruk av ord, uttryck och översättningsmotsvarigheter. Men varje korpus är nödvändigtvis "för liten", dvs en exempelsamling, hur stor den än blir, kan aldrig innehålla varje användningsmöjlighet av varje enstaka ord och varje enstaka konstruktion (jfr Lehrberger/Bourbeau 1988: 129). Statistiken visar att ungefär hälften av alla lemman i en korpus förekommer bara en enda gång i löptexten. De regler med vars hjälp korpusinformationen tillämpas på konkreta översättningsuppgifter (parsning, syntaktisk transfer, lexikal transfer, semantisk-pragmatisk disambiguering osv) bör därför generalisera med utgångspunkt i korpusexemplen. Sådana generaliseringsfunktioner behövs på minst tre plan:

1. enspråkig syntax (parsning [jfr Zuijlen 1989], morfosyntaktisk syntes);

2. metatax (syntaktisk transfer);

3. semantik-pragmatik (mätning av betydelseavstånd m m).

I överensstämmelse med DLT:s arkitekturprinciper följer dessa generaliseringsregler implicitetsidén, vilket bl a innebär att de undviker att anlita en explicit allmängiltig mellanrepresentation som den som skulle behövas i ett system på skalans fjärde steg.

I tankeexperimentet ovan antog jag att en väsentlig drivkraft för att låta utvecklingen gå vidare från det tredje till det fjärde och femte steget är intresset i att ersätta det mänskliga arbete som är nödvändigt i kunskapskoderingsprocessen med automatiska processer. Den lösning jag skisserar ovan realiserar denna vidareutveckling, men den saknar ändå inte arbetsmoment där information läggs till av människor. Människan, systemutvecklaren, har alltså inte rationaliserats bort. Men DLT:s korpusstödda översättningsmetod innebär att det bidrag specialisterna lämnar till kunskapsinhämtningen motsvarar i mycket högre grad än vid tredje utvecklingssteget ett vanligt sätt att resonera över språk. Ber man en specialist att ge exempel på språkbruk i sitt ämnesområde eller att granska föreslagna formuleringar så är uppgiften enklare och svaren pålitligare än när man är tvungen att be om en allmängiltig metaspråklig redogörelse. De som medarbetar i DLT:s systemutvecklingsfas kan därför i högre utsträckning vara specialiserade i ämnesområdet och i översättning och behöver i mindre grad koncentrera sig på teoretisk grammatik eller lexikografi.

6 Kunskap om världen eller kunskap om texten?

Sålunda tillåter utvecklingen av maskinöversättningssystemet DLT med sin nya systemstruktur ett nyartat svar på frågan huruvida grundvalet för maskinöversättningsändamålet bör vara (utomspråklig) kunskap om världen eller (inomspråklig) kunskap om texten. DLT:s svar är att det är kunskap ur texter.

Litteratur

Lehrberger, John, Laurent Bourbeau. 1988. *Machine translation.* Amsterdam/Philadelphia, Benjamins.

Papegaaij, B. C. 1986. *Word expert semantics.* Dordrecht/Riverton, Foris.

Sadler, Victor. 1989. *Working with analogical semantics. Disambiguation techniques in DLT.* Dordrecht/Providence, Foris.

Schubert, Ingrid 1989. A dependency syntax of Danish. Dan Maxwell, Klaus Schubert [utg.]. *Metataxis in practice. Dependency syntax for multilingual machine translation*:39–67. Dordrecht/Providence, Foris.

Schubert, Klaus. 1986. Linguistic and extra-linguistic knowledge. *Computers and translation*, 1:125–152.

Schubert, Klaus. 1987. *Metataxis. Contrastive dependency syntax for machine translation.* Dordrecht/Providence, Foris.

Schubert, Klaus. 1988. The architecture of DLT—interlingual or double direct? Dan Maxwell, Klaus Schubert, Toon Witkam [utg.]. *New directions in machine translation*:131–144. Dordrecht/Providence, Foris.

Schubert, Klaus 1989. A dependency syntax of Esperanto. Dan Maxwell, Klaus Schubert [utg.]. *Metataxis in practice. Dependency syntax for multilingual machine translation*:207–232. Dordrecht/Providence, Foris.

Witkam, A. P. M. 1983. *Distributed Language Translation.* Utrecht, BSO.

Zuijlen, Job M. van 1989. Probabilistic methods in dependency grammar parsing. *International Workshop on Parsing Technologies*:142–151. Pittsburgh, Carnegie-Mellon University.

Klaus Schubert
BSO/Research
Postbus 8348
NL-3503 RH Utrecht
Nederländerna
schubert@dlt1.uucp

BENGT SIGURD

Erfarenheter av Swetra—ett svenskt MT-experiment

Abstract

Swetra is primarily a research project, but its computer programs and reports may result in commercial products. The research group at Lund consists of a few persons only, which has advantages and disadvantages. The main languages studied are Russian, English and Swedish. The grammatical model is Referent Grammar (RG) and the project is a spin-off of the development of that grammar. Referent Grammar is inspired by generalized phrase structure grammar (GPSG) and written directly in Prolog (DCG), which is why the grammars can run directly on computers. Arity Prolog and PC's are used. The work within Swetra consists mainly of writing grammar modules and lexicons for different languages. These modules are connected in automatic translation. Referent grammars are bidirectional and can be used both for analysis and synthesis (generation). A certain number of transfer rules are also needed in translation, however, in order to make necessary changes of functional representations, add features such as definiteness when translating from Russian into English etc. The paper gives a survey of the experience from Swetra research so far.

1 Inledning

Swetra (Swedish Computer Translation Research) har pågått cirka två år, och det finns en del erfarenheter att förmedla—erfarenheter som andra som ger sig ut på maskinöversättningens gungfly kan ha glädje av att känna till. Swetra, som stöds av Svenska Forskningsrådet för Samhällvetenskap och Humaniora, är ett minimalt projekt ifråga om ekonomiska, personella och maskinella resurser. I projektet arbetar—förutom undertecknad—Mats Eeg-Olofsson (halvtid), Lars Gustafsson (kvartstid), Barbara Gawrońska-Werngren (halvtid). Programmeringen har främst skett på PC med användande av Arity Prolog.

Jag skall meddela mina erfarenheter av projektet under följande huvudrubriker: Organisation, Vägval (texttyp, språk, inställning till syntax, grammatikmodell), Vad Swetra kan, Publiktrycket.

2 Organization

Jag tror att en liten forskargrupp lokaliserad på en plats har stora fördelar. Man vet vad de andra gör och behöver inte resa runt eller samlas till stora seminarier. Ledningen förenklas. Det räcker ofta med några ord till de andra för att de skall vara med på noterna. Uppenbarligen säger jag detta mer eller mindre omedvetet med tanke på Eurotra, där medarbetarna synes ha tvingats använda mycket tid på att resa för att informera varandra och komma överens om vilka lingvistiska teorier, programmeringsspråk, format och datorer man skall använda. Den lilla tätt sammansvetsade gruppen har säkert stora fördelar när det gäller att utveckla den grundläggande teorien och prototypen, men när det sedan gäller att implementera teorien och gå från prototyp till produkt, att skriva stora lexikon, att förbättra programmen och att testköra systemet då har större arbetsgrupper säkert fördelar.

Vi hade i själva verket inget val i Lund, när vi började pröva på maskinöversättning. Jag hade länge talat och skrivit om hur svårt maskinöversättning är utan att själv ha någon riktig datorerfarenhet av det—bara pappersspekulationer. Swetra är en biprodukt av grammatikforskning och uppstod sedan jag insett att det gick att använda referentgrammatik för översättning. Swetra är snarast ett försök att se hur långt man kan komma med små medel. Vi ägnar föga tid åt att visa vad man inte kan klara eller bevisa att maskinöversättning är omöjligt, mest tid åt att visa vad man kan klara.

Anpassningen till PC bestämde vi oss tidigt för. Den underlättar flyttning och demonstration av programmen. Vi tänker nu också använda snabba Macintoshdatorer som också kan läsa DOS-disketter och möjliggör flyttning mellan olika miljöer.

3 Vägval

3.1 Texttyp

Det är nödvändigt att göra ett antal vägval när man går in i maskinöversättningsbranschen. Flera av dessa val hör ihop. Man måste välja mellan att försöka sig på att översätta godtycklig text (fritext) eller text inom något specialområde som teknik (t.ex. bilar, datorer), medicin, affärskontrakt, väderleksrapporter, etc. Inledningsvis träffade vi inte något sådant val—man var glada att det gick att översätta några meningar överhuvudtaget. De meningar vi först översatte var typiska lingvistmeningar, typ "En flicka kom", "Pojken slog hunden som sprang", "Hunden, som pojken, som flickan kände slog sprang". Efter att ha satt upp vad vi tyckte var syntaktiska regler som borde klara grundläggande strukturer försökte vi pröva dem på empiriska texter, bl.a. en liten Greenpeace-text om kärnkraftverk i Sellafield som fortfarande släpper ut plutonium och sedan några notiser från Pravda. Mötet med empiriska texter var omtumlande; man inser att teoretiska lingvister lever i ett reservat (med ett svenskt uttryck: en skyddad verkstad). Varenda mening ledde till att vi fick revidera grammatiken, ändra eller lägga till regler och inte bara utöka lexikonet som man ju hoppas.

Man kan tänka sig att rita diagram som visar ökningen av de grammatiska reglerna och lexikon för varje ny mening som skall översättas. I början måste mötet med en ny mening nödvändigtvis leda till stor ökning både av grammatiken och lexikonet, men kurvan bör så småningom plana ut, när nästan alla meningar klaras av utan att någon förbättring behöver göras. Ett färdigt system bör klara allt och inte kräva mera utbyggnad. Om det är ett interaktivt system bör det inte fråga operatören alltför ofta. Jag kan inte säga att vi kommit så långt i Swetra att vi känner att kurvan håller på att plana ut; varenda liten text vi försöker oss på kräver flera komplettringar—och det är inte alltid bara fråga om att sätta in fler ord i lexikon.

Det val i fråga om texter vi avser att träffa innebär att vi säger att Swetra inte kan översätta godtycklig text, bara speciell text, men vi vill specificera vilken speciell text Swetra kan översätta på ett lingvistiskt sätt. Vi vill inte specificera dessa texter genom hänvisning till en genre som t.ex. tekniska manualer, väderleksrapporter, utan ge specifikationen mera lingvistiskt och säga att meningarna inte får innehålla samordnade relativsatser, inte mer än två satsadverbial och tre andra adverbial, inte inbäddade genitiver som "pojkens hunds svans" etc. Lexikalt kan specifikationen göras genom att man t.ex. säger att orden måste tillhöra den marina sfären och röra sig om olika typer av båtar som rör sig i Östersjön.

I själva verket borde nog många MT-system ha specificerat sina begränsningar på detta sätt. Så vitt jag förstått av de demonstrationer Eurotra företagit i Brüssel har Eurotra egentligen också gått på denna linje: Deltagarna fick bara föreslå meningar som hade högst 7 ord, bara ett adjektiv i nominalfrasen, endast subjektiva relativsatser, inga samordningar etc. Det subspråk som Swetra vill kunna översätta definieras alltså i första hand genom vissa syntaktiska begränsningar. Sedan är det en självklarhet att orden måste finnas i lexikonet för att översättningen skall fungera. Jag skall sedan visa typiska Swetra texter som vi brukar köra i vår Demo (se också Sigurd & Gawrońska-Werngren 1988) .

3.2 Språk

Självklart måste man välja vilka språk man vill översätta mellan, men märkligt nog har vi tvekat länge på denna punkt i Swetra. Det beror naturligtvis på att vi åtagit oss att forska i maskinöversättning, inte att leverera ett färdigt program. Det är i själva verket intressant att pröva hur bra den grammatiska modellen referentgrammatik fungerar på olika språk. Vi har erfarenhet av översättning till och från franska, polska, georgiska, samoanska, svenska, engelska, ryska. På senare tid har vi alltmer koncentrerat oss på ryska, engelska, svenska och främst på översättning från ryska till svenska—sedan Barbara Gawrońska-Werngren ställde sina språkkunskaper till förfogande. De olika referentgrammatiska modulerna som programmeras direkt i Prolog (DCG=Definite Clause Grammar) är i princip bidirektionella och kan köras både i analys och syntes (generering). Men vill man göra mera avancerad översättning, använda specifika transferregler, anknyta diskurssemantiska procedurer m.m. och göra ett effektivt program är det bäst att bestämma sig för en riktning.

Vi har bedömt översättning mellan ryska och engelska som intressant därför att detta par länge intresserat MT-forskarna. Ett framgångrikt system som kan översätta mellan ryska och engelska har uppenbara praktiska tillämpningar. Språkens typologiska skillnader gör översättning också lingvistiskt intressant. Eftersom många personer inte kan ryska blir demonstrationer dessutom mera imponerande. Åskådarna blir mera förbryllade av att se hur en obegriplig text översättes till en begriplig än att se t.ex. engelska, som de flesta kan, översättas till svenska. (Å andra sidan blir åskådarnas möjligheter att bedöma översättningens kvalitet mera begränsade).

3.3 Inställning til syntax

Ett annat vägval rör inställningen till syntax. Många—i synnerhet tidigare—system betraktar ordöversättning som den primära operationen vid översättning. Man försöker sedan sekundärt se till att ordens kongruensböjning stämmer och att ordföljden är rätt genom att försöka identifiera vilka ord som hör ihop i fraser och vilka ord (fraser) som är subjekt, predikat, objekt och adverbial—något som ofta är ett minimikrav för att klara böjning och ordföljd. Swetra har inte ord som primära enheter, utan satser och fraser i olika funktioner (såsom subjekt, objekt, predikat, satsadverbial, andra adverbial, attribut). Swetra är på detta sätt en typisk produkt av sin tid—den tid i lingvistiken då syntax skriven såsom generativ grammatik står i fokus. Jag skall förklara detta närmare senare när jag beskriver den grammatiska modell (Referentgrammatik) som Swetra hela tiden arbetat med.

Ett system som prioriterar ordöversättning kan ofta översätta många sorters texter snabbt givet ett stort lexikon, men kvaliteten blir lidande om man inte håller reda på satsstrukturen i detalj. Förvånande många tvetydiga ord blir entydiga om man tar ut satsdelarna—standardexemplet är: Var var det var? Böjning av orden i målspråket ger sig liksom ordföljden om man har detaljerade upplysningar om ordens funktioner i fraser och satsdelar. Å andra sidan är satslösning (parsning) tidsödande, och alla problem kan inte lösas med syntax—ökända är prepositionsfraserna vars anknytning till verb eller nominalfras ofta endast kan avgöras på semantisk väg. När Swetra går grammatikvägen, innebär det att man endast accepterar och översätter meningar vars grammatiska struktur systemet har regler för att identifiera. Ett lexikonorienterat system kan alltid föreslå en översättning även om systemet inte har gjort någon detaljerad satslösning.

3.4 Grammatisk modell

En modern lingvist har till synes flera väl genomdiskuterade formella grammatiska modeller att välja på, t.ex. Transformationell generativ grammatik med frasstruktur och transformationer (EST, Government and Binding), Lexical-Functional Grammar (LFG), Generaliserad frastrukturgrammatik (GPSG), Dependensgrammatik, Funktionell Grammatik av Hallidays typ. Det är påfallande att dessa grammatiker nästan bara har prövats på typiska lingvistmeningar

av ovannämnda typ ("En flicka sprang", "Hunden, som, pojken, som flickan kände slog sprang"). Man ser t.ex. aldrig en GB-grammatiker visa hur alla meningarna på en viss sida text skall analyseras, vilka träddiagram man bör rita och vilka problem man möter. Gängse teoretiska grammatikmodeller visar främst hur vissa formella idéer som t.ex. frasstruktur, transformationer, dependens, subjacency, c-command fungerar. De har inte det primära syftet att visa hur vanliga meningar skall analyseras så att beskrivningen förklarar varför orden har den form de har och står i den ordning de gör. Allra minst förklarar de varför den föreliggande meningen betyder vad den gör.

Såsom framgått var det inte så att medarbetarna i Swetra bestämde att de skulle översätta med dator mellan två bestämda språk och sedan började leta efter en lämplig grammatik. I stället var det så att jag höll på med de pilregler som DCG erbjuder och insåg att man som resultat av analysen (satslösningen, parsningen) kunde få en sorts universell semantisk representation: en predikatslogisk formel eller en funktionell representation, en representation som talar om vad som är subjekt, predikat, objekt, adverbial, etc. och vad orden betyder. (Chomskys klassiska generativa grammatikregler ger endast ett S som resultat av analysen, vilket bara säger att satsen var grammatisk). Sedan jag skrivit en sådan grammatik för svenska och en för engelska insåg jag att man måste kunna översätta via den gemensamma funktionella representationen om man standardiserade den. Det är i princip det vi hållit på med sedan i Swetra. Den funktionella representationen fungerar som ett mellanspråk, *interlingua*.

Den grammatiska modellen för Swetra var alltså given, men samtidigt som vi arbetat med översättningsprogrammen har vi utvecklat referentgrammatik och tagit beslut som standardiserat den och framförallt dess funktionella representation och lexikonformat. Referentgrammatik (RG, se referenser i litteraturlistan) är en sorts generaliserad frasstrukturgrammatik (GPSG), men i motsats till GPSG arbetar RG i de generativa reglerna med två representationer: den ytliga kategorirepresentationen betecknad o-representationen, och den funktionella representationen betecknad f-representationen. Dessa beteckningar påminner om dem som användes i lexical-functional grammar (LFG), en grammatik som nog också kan användas vid automatisk översättning.

Referentgrammatik är bidirektionell, dvs kan användas både i analys och syntes (generering). Det betyder att den ställer samma hårda krav på sitt input och sitt output. En korrekt skriven RG-modul genererar bara korrekta meningar och accepterar bara korrekta meningar. För att en grammatik skall vara användbar för MT krävs att den kan specificera korrekthet på alla nivåer, genererar korrekta former av artiklar, räkneord, pronomen, adjektiv, substantiv, verb, sätta orden i rätt ordning, etc. RG gör det—den genererar t.ex. också korrekta former av relativpronomen i ryska och polska en uppgift som kräver att flera faktorer tas hänsyn till. RG är dessutom lätt att skriva för en lingvist och vi har inte funnit något som den inte kan hantera. Den klarar också s.k. *unbounded dependencies*, dvs flyttningar som kan vara hur långa som helst.

RG använder liksom GPSG defekta syntaktiska kategorier, men de formaliseras inte som "slash-kategorier" i RG utan benämns `sdsent` = subjektsdefekt sats, `odpp` = objektdefekt prepositionsfras etc. I programmet flyttas den saknade

konstituenten till rätt position, vilket gör att man kan säga att RG har dolda transformationer, precis som GPSG.

För att en grammatikmodell skall kunna användas i maskinöversättning måste den också ha tagit ställning till hur ordbetydelser skall beskrivas och vilket format de skall ha i lexikonet. Många teoretiska modeller lämnar den frågan öppen, men det duger inte i ett MT-system där alla komponenter måste finnas och kunna samverka. RG har ett bestämt lexikonformat (se artiklar i litteraturlistan) och lexikonet är en särskild fil på vilken man kan låta diverse operationer arbeta. Ordbetydelsen beskrivs f.n. i en engelskliknande form (*machinese*), men lexikonet har utrymme för många grammatiska och semantiska uppgifter om orden. RG har också utvecklat vissa procedurer för morfologi (ordböjning och ordbildning) och tillämpat s.k. implikationell morfologi, en modell där den morfologiska kunskapen ses som kunskap om vissa ordformer kompletterad med kunskap om hur existensen av en viss form implicerar existensen av en viss annan form. Om man kan en form i språket och vet vad den betyder, kan man dra slutsatser om hur många andra former bör se ut och vad de betyder.

Den grammatiska diskussionen de senaste decennierna har p.g.a. Chomskys starka inflytande mycket berört universella syntaktiska principer för naturliga språk, t.ex. begränsningar på flyttning av konstituenter. Flyttningar och hoppande på complementizers (i engelska) har t.ex. diskuterats åtskilligt, och relativsatser har t.ex. beskrivits som en flyttning av den relativiserade konstituenten till komplementizern = relativmarkören efterlämnande ett osynligt spår. Men det finns mycket annat som är viktigt att beskriva om man skall kunna översätta relativsatser. Man finner emellertid inom EST/GB och andra teoretiska grammatikmodeller inte t.ex. några konkreta regler för val av rätt form av relativpronomen, något som är nödvändigt om man vill implementera MT. Det talas ofta vid presentationen av dessa moderna grammatikmodeller som om böjning vore en trivial uppgift — men det är den inte ens i engelskan. En grammatik som skall användas för MT måste ha regler som genererar "who", respektive "whom" (för att inte tala om "whose"). I ett språk som georgiska är ordböjning huvuduppgiften: om man lyckas få verbformen rätt är det mesta av satsen avklarat. I både den ryska och den svenska grammatikmodulen upptas en ansenlig mängd av reglerna med att böja orden rätt.

4 Vad Swetra kan

Swetra kan analysera, syntetisera (generera) och översätta typiska lingvistmeningar med intransitiva, transitiva och dubbelt transitiva verb och upp till två satsadverbial och tre andra adverbial. Som adverbial kan även förekomma prepositionsfraser och underordnade konjunktionsbisatser. Swetra accepterar kopulativa satser (i ryska utan kopula) och satser med passivum, hjälpverb ("ha", modala hjälpverb som "kan" och aspektuella verb som "börja"). Swetra har omfattande komplex av regler för att hantera nominalfraser med bestämningar före och efter huvudet och komplicerad kongruens. Särskild uppmärksamhet har ägnats åt relativsatser; det referentbegrepp som givit RG dess namn växte i

själva verket fram ur analysen av relativsatser (se uppsatser om relativsatser i litteraturlistan).

Stora likvärdiga grammatikmoduler finns för engelska, ryska och svenska och tillhörande lexikon är för närvarande på några tusen ord. Demonstrationsprogram visar analys av satsen i källspråket, och generering av motsvarande sats i målspråket. Översättningen tar ofta bara några sekunder. Diverse uppsnabbningsknep användes (som vi här inte tänker avslöja). Grovöversättning görs via gemensam funktionell representation, men vissa transferregler har dessutom utarbetats för översättning i en viss riktning (särskilt ryska till engelska).

Typiska empiriska prov är deskriptiva texter, typ nyhetsnotiser. Swetra har bl.a. använt notiser ur Pravda, och det är troligt att Swetra specialiserar sig på notis- eller bulletintexter (se Sigurd & Gawrońska-Werngren 1988). Sådana texter liknar de texter som det textgenererande datorprogrammet Commentator kunde generera och samverkan mellan Commentator och Swetra för att generera och översätta sådana texter till några språk har diskuterats som ett framtida projekt. Ett exempel på en sådan text är följande: "Ett okänt flygplan närmade sig Öland från öster igår. Det girade söderut innan det kom in på svenskt område och försvann sedan söderut. Inget svenskt flygplan fanns då inom området. Flygplanet observerades av radar." Ytterligare exempel på texter som Swetra riktar in sig på finns i artiklar omnämnda i referenslistan.

5 Publiktrykket

När man säger att man håller på med automatisk översättning får man ofelbart frågan: "Hur många år dröjer det innan det går, tror Du?" Svaret: "Aldrig" accepteras inte och inte heller svar av typen: "Vi kan översätta texter med 80% kvalitet om de ligger inom en viss domän, t.ex. det marina området och grammatiken är så begränsad att den inte tillåter samordnade relativsatser och inte mer än två prepositionsfraser i en nominalfras etc."

Det finns ett publiktryck som de som sysslar med maskinöversättning alltid måste ha känt. Det är det tryck som lockar en att säga: "Om 5 år, eller om 10 år" och kanske mumla något ohörbart om textuella begränsningar. Det är det tryck som fått många forskare att lova för mycket och därigenom tidvis förstöra marknaden för forskning inom automatisk översättning. I ansökningarna för Swetra har vi aldrig lovat att leverera ett fungerande översättningssystem, bara att forska inom automatisk översättning.

Ofta sägs också: "Men poesi och skönlitteratur kommer väl aldrig att kunna översättas?" Och några tokroliga vandringexempel ges. Det finns ett tryck att då säga: "Nej, det kommer aldrig att gå." Men här känner jag mig ofta lockad att också gå på tvären och säga att: "Jo, poesi går lika bra, fast det blir kanske inte riktigt samma poesi utan ofta djärvare vändningar och fräschare metaforer." Men åhörarna vill inte heller höra på det örat. MT är ett svårt område—men det är jag inte den förste som konstaterat.

Litteratur

Gawrońska-Werngren, B. 1988. A referent grammatical analysis of Polish relative clauses. *Studia Linguistica* 42(1):18–48

Sigurd, B. 1987. Referent grammar (RG). A generalized phrase structure grammar with built-in referents. *Studia Linguistica* 41(2):115–135

— 1988. Using Referent Grammar (RG) in computer analysis, generation and translation of sentences. *Nordic Journal of Linguistics* 11(1–2):129–150.

—1989. A referent grammatical analysis of relative clauses. *Acta Linguistica Hafniensia* 21(2):95–115

Sigurd, B. & Gawrońska-Werngren, B. 1988. The potentials of Swetra, a multilanguage MT-system. *Computers and Translation* 3:238–250

Inst för Lingvistik
Helgonabacken 12
Lunds Universitet
Sverige

OLE TOGEBY

Translation of Prepositions by Neural Networks

Abstract

Translation of prepositions poses a very serious problem to machine translation because prepositions are highly ambigous. In theory prepositions can be disambiguated by a filter that excludes already generated representational objects with no selection restriction match between preposition and np, but it takes too long time in practice. A neural network makes the disambiguation in fractions of a second, because it is fast, robust and very powerful.

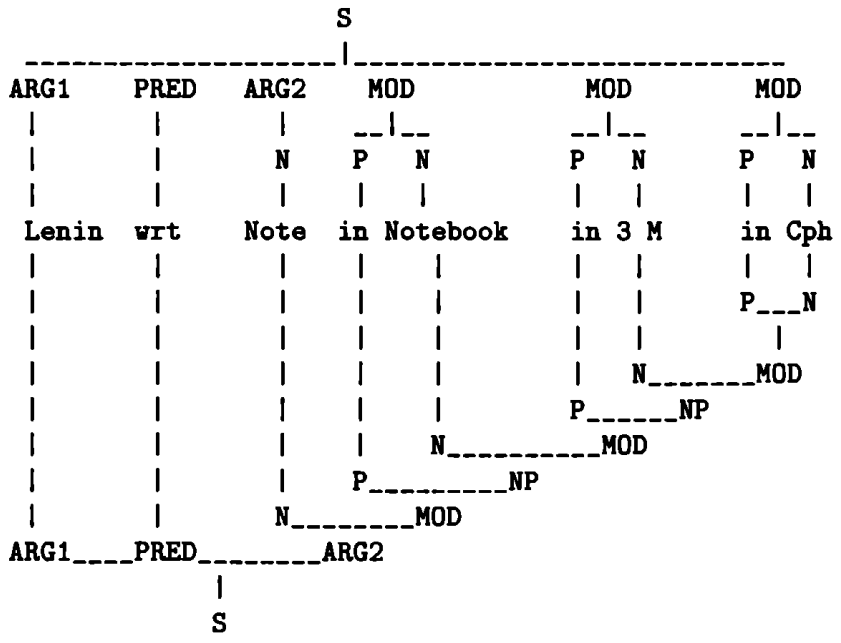
1 The Problem

The translation of prepositions poses a very serious problem to machine translation because prepositions are highly ambigious—each of the most 10 frequent prepositions in one of the 9 EUROTRA languages is translated into 10 different prepositions in each of the 8 other languages—and because prepositions always will generate many attachment patterns.

Take the example:

Lenin wrote this note in his notebook in 3 minutes in Copenhagen

There are the flat structure and the deep structure and 6 attachment patterns in between:



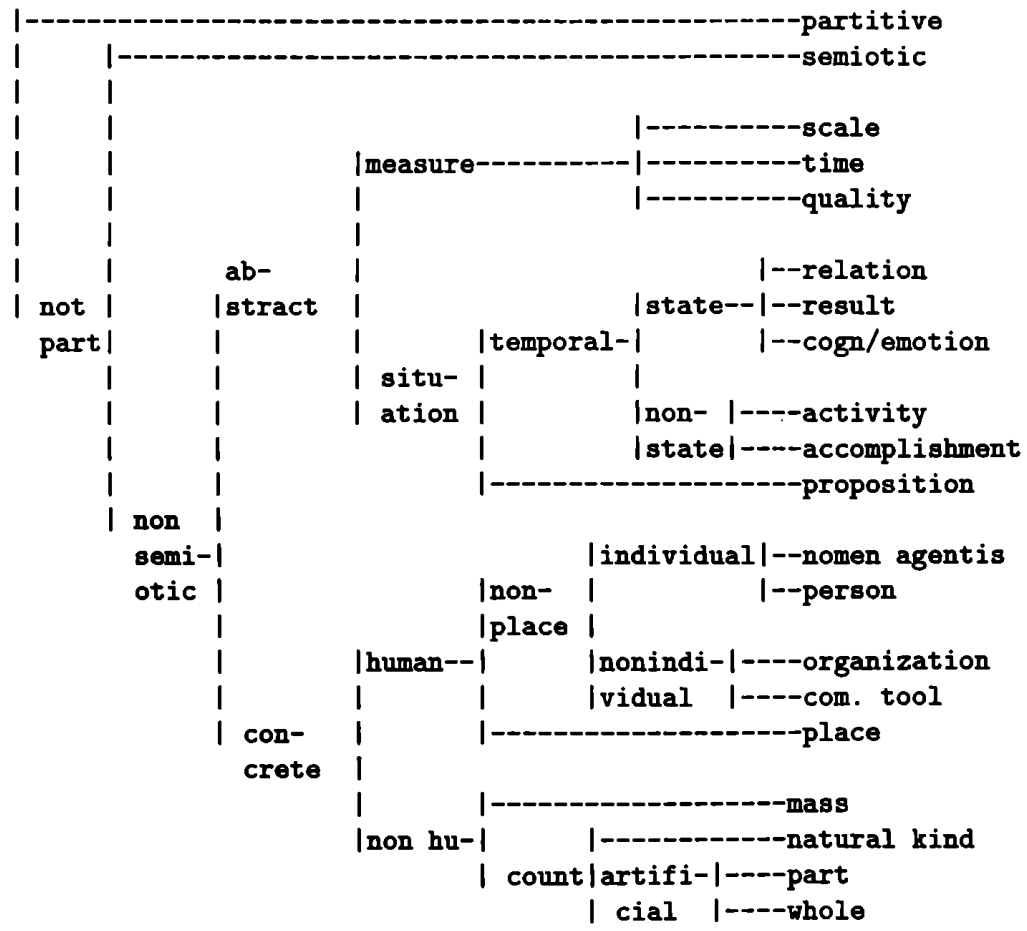
It is obvious in this example that *in* does not mean the same thing in the three clauses: *in his notebook*, *in 3 minutes* and *in Copenhagen*. The first one means DIRECTION, the second one: TIME: HOW LONG, and the third one: PLACE WHERE. In other words, we need (at least) three lexical entries for the word *in*. Other entries are needed for *in 1897*, *in anger*, *in danger*, *in various colours*, with the manings: TIME WHEN, IN A MOOD OF, DURING AN ACTIVITY OF, OF A QUALITY OF. So in Danish there are, in my opinion, at least 12 different meanings of the corresponding preposition *i* (and furthermore all the fixed phrases, e.g. *in all*, *to be in for*).

The distinction between DIRECTION and PLACE WHERE is not made intuitively, but with the so called 'nonsensical conjunction reduction test', which says: if the conjunction of two contexts to the same word does not make sense, there are two readings of the lexical item, one for each context. So it does not make sense to say: *Lenin wrote this note in his notebook and three minutes*, or *Lenin wrote this note in his notebook and Copenhagen*, or *Lenin wrote this note in three minutes and Copenhagen* or *Lenin wrote this note in three minutes and 1897*. But note that contexts of the same type, i.e. contexts of the same reading of a given word form can be coordinated and make sense: *Lenin wrote this note in anger and disappointment.*

If we parse the sentence using a dictionary with 12 different entries for the word *in*, and 8 different attachment patterns, the machine will generate $8 \times 12 \times 12 \times 12 = 13.824$ different representational objects i.e. readings of the sentence. So the problem is how to disambiguate the sentence and make the machine find the correct reading among the 13.824 syntactically possible readings.

2 The Linguistic Solution

In theory, that means from a linguistic point of view, this can be done in the following way: in the dictionary the 12 entries for *in* are differentiated wrt. their **selection restrictions**: in_1 (PLACE WHERE) takes as argument1 nouns of the type CONCRETE, in_2 (TIME WHEN) takes nouns of the type YEAR or DATE, in_3 (MOOD) takes nouns of the type COGNITION OR EMOTION, in_4 (TIME HOW LONG) takes nouns of the type SCALE. That means that all nouns in the dictionary are coded with a semantic type label. I have suggested the following set of semantic types for nouns:



Danish examples: PARTITIVE: *sektor, side, halvdel*, SEMIOTIC: *afsnit, forslag, aftale*, SCALE: *meter, decibel, grad, minut, dag*, TIME: *efterkrigstiden, fremtiden*, QUALITY: *identitet, størrelse, længde*, RELATION: *afhængighed, faktor, position*, RESULT: *produktion_2, undtagelse, investering_2*, COGN/EMOTION: *interesse, frygt, glæde*, ACTIVITY: *databehandling, anvendelse, produktion_1*, ACCOMPLISHMENT: *revolution, investering_1, udforskning*, PROPOSITION nouns: *fordel, mulighed, problem*, NOMINA AGENTIS: *fabrikant af, tilskuer til, herre over*, ORGANIZATION: *hjemmemarked, industri, regering*, COMMUNICATION TOOL: *persondatamat, videobåndoptager, radio*, PLACE: *Europa,*

København, ildlinjen, MASS: *vand, luft, sand*, NATURAL KIND: *blomst, træ, sten*, PART: *kredsløb, svinghjul, taster*, WHOLE: *dataanlæg, elektronik, informationsteknologi.*

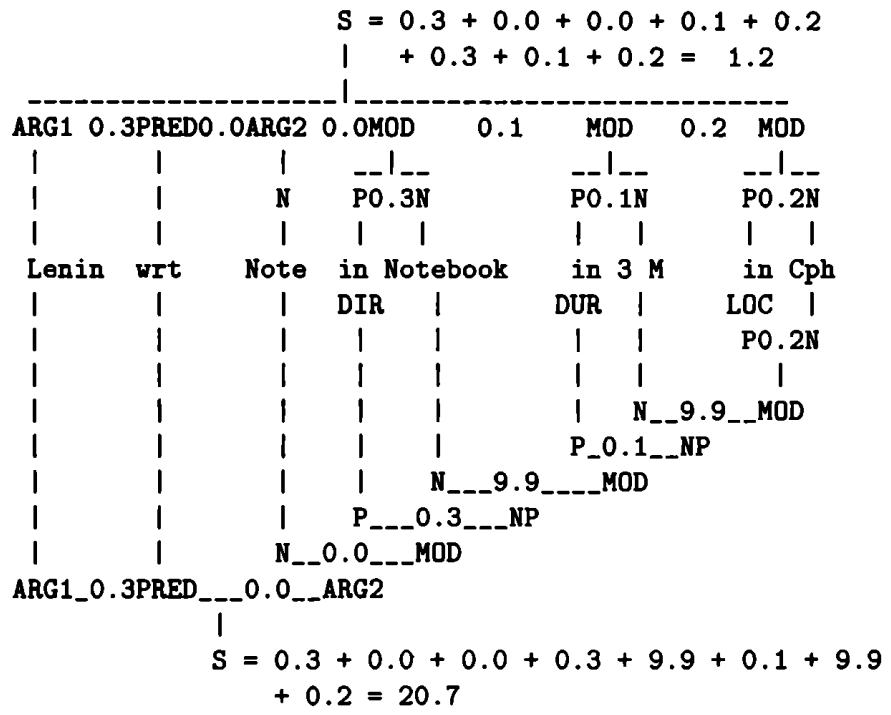
The selection restriction could work as a filter with a 'killer rule', i.e. a rule that would exclude ('kill') all objects with no match between the type which is asked for in the semantic frame specification of the head, in this case the preposition, and the type of the noun that fills the slot. There will be no match in the created object with in_1 (PLACE WHERE) in the clause *in three minutes*, because *minutes* is a noun of the type SCALE, and *in_1* only selects nouns of the type CONCRETE.

This type of rule would exclude most of the not wanted objects among the 13.824 generated representational objects. But the rule is too strong, because it is not uncommon in natural texts to find metaphorical or slightly metaphorical sentences, e.g.: *The situation threatens to become worse.* In this case the selection rule saying that the verb *threaten* only takes nouns of type HUMAN as argument1 will 'kill' all the generated objects so that no analysis or translation will be produced at all.

3 Preference Rules

Instead it is necessary to use a **preference** rule that compares all representational objects generated from the same surface structure, ranks them wrt. *internal semantic fitness*, and selects the fittest. As shown in the first paragraph the simple example *Lenin wrote this note in his notebook in three minutes in Copenhagen* will generate 8 attachment patterns which then can have 12 different readings of each of the three prepositions. What is compared by a preference rule is not two clauses containing the same two or three words, but the sum of the semantic distances between all the pairs in the sentence of 1) a selection restriction bearing head and 2) the corresponding slot filler, added up at the top node.

The concept of semantic distance and semantic fitness can be operationalized in the tree of semantic types. You walk in the tree from the type which is asked for in the selection restriction, step by step, to the type of the slotfiller, counting 1.0 for every step to the left, and 0.1 for every step to the right. The distance from CONCRETE (which is selected by *in_1*, PLACE WHERE) to SCALE (the type of *minutes*) is 1.3, while the distance from SCALE (which is selected by *in_4* TIME HOW LONG) and SCALE is 0.0. Consequently reading *in_4*, TIME HOW LONG is selected in the clause *in three minutes.* Two representational objects, two tree structures representing two whole sentences, can then be compared in the following way:



By such a preference rule the flat structure with the DIRECTION reading, the HOW LONG reading and the WHERE reading, respectively, will be selected—and that is exactly the correct one among the 13.824 possible readings.

But this machinery will only work in theory. The comparison among the objects will be made in pairs, so there will be made $13.824/2 \times 13.825$ comparisons and that will take approximately $6\frac{1}{2}$ hours with a fast machine and a fast program.

4 The Neural Network Design

So in theory it can be done, and the human brain must follow a rule like the one described when it calculates the correct reading in fractions of a second, but it must do it in a smarter way than by comparison in pairs of already genereted objects.

This smarter way must be something like what is called a neural network, which is a strategy for programming the preference rule so that the machine can compute the best solution of the problem in fractions of a second, like the human brain does.

The semantic network is designed in the following way: It consists of three layers, an input layer with 117 neurons, a hidden layer with 65 neurons, and an output layer with 12 neurons. All input neurons are connected with all the hidden layer neurons, and all the hidden layer neurons are connected with all the output layer neurons. That means that there are 7670 connexions between layer 1 and 2, and 792 connexions between layer 2 and 3.

Each of the output neurons represents one of the possible readings of the preposition *in*. The 12 readings of *in* are: ARG1 (deep subject), ARG2 (deep object), LOC (place where), DIR (direction), TIME, DUR (time how long), MEA (measure), STA (state), ACT (activity), EMO (emotion/cognition), QUAL (quality), CLOT (clothes).

The input is a pattern of the syntactic and semantic structure of a sentence containing the word *in*. 4 words to the left and 4 to the right of the preposition are represented in the pattern as syntactic-semantic categories. A given word belongs to one and only one of the following 56 categories, which include the semantic features described in the first paragraph:

```
NOUNS: NONHUMAN              VERBS: AUXILIARY OR MODAL
       PLACE                        INTRANS |  STATE
       HUMAN                        OR PAS- |  PROCESS
       NOMEN AGENTIS                SIVE    |  EVENT
       SEMIOTIC                     TRANSITIVE + noun
       PART                         TRANSITIVE + SENTENCE
       MEASURE OR BARE FORM         TRIVALENT VERB
       TIME                         VERB  with prepositional ob-
       QUALITY OR RELATION           jects and the preposition
       RESULT                        i, til, fra, over, under,
       EMOTION OR COGNITION          for, af, ved
       ACTIVITY
       ACCOMPLISHMENT
       PROPOSITION

PREPOSITIONS: I, PÅ, TIL, FRA, OM, FOR, AF, MED, UDEN, OVER,
              UNDER, MELLEM
PRONOUN                      CONJUNCTION          NUMBER
PUNCTUATION MARK             AND/OR/BUT           TIME ADVERB
THAT                         ARTICLE              PLACE ADVERB
ADJECTIVE                    DIRECTIONAL ADVERB   OTHER ADVERBS
ADJECTIVE + PREPOSITIONAL OBJECT
```

The natural way to represent the 9 word input pattern would be an array with 504 neurons ordered in 9 rows and 56 columns. But that would be a very redundant representation, because only 9 of the 504 neurons would be activated in each sentence.

The input pattern information can be represented by only 117 neurons organized in an array with 9 rows, one for each word in the sentence window, and 13 columns, in which each of the 56 categories is represented by 3 X in accordance with the following coding key (n = noun, v = verb, p = preposition, a = other, o = zero, i = 1):

```
__no_____ni_____vo______vi______po____pi______ao______ai_____
1 nonhum time   aux/mod VPO-i   i     med     pronom  adj    1
2 place  rel    state   VPO-på  på    uden    konj    a-pob  2
3 hum    result process VPOtil  til   over    punkt   tal    3
4 agent  emo    event   VPOov/u fra   under   og      a-tid  4
5 sem    activi t-vb+n  VPOfor  om    mellem  at      a-sted5
6 part   accomp t-bv+s  VPOaf   for   ved     article a-ret  6
7 scale__prop___tri-vb__VPOloc__af____før_____efter___a.adv_7
```

That means that the category INTRANSITIVE VERB OF THE STATE TYPE is represented by *v o 2*. As an example the sentence *Det sker i 1992* ('it happens in 1992') has the representation shown below:

```
SENTENCE: XXX . Det sker  - i - 1992 . XXX XXX
                INPUT PATTERN
                nvpaoi1234567
              -4 .............4-
              -3 ...XX...X....3-
              -2 ...XX.X......2-
              -1 .X..X....X...1-
               0 ..X.X.X......0
              +1 X....X..X....1+
              +2 ...XX...X....2+
              +3 .............3+
              +4 .............4+
                nvpaoi1234567
```

The output is represented by 12 'thermometers' which show how much a given neuron, representing one reading of the preposition *i*, is activated:

```
arg1:     ........
arg2:     ........
 loc:     XXXX....
 dir:     XX......
time:     ........
 dur:     XXXX....
 mea:     ........
 sta:     X.......
 act:     ........
 emo:     ........
qual:     ........
clot:     ........
```

When a pattern of input neurons is activated the neurons 'fire', i e. they activate all the hidden neurons they are connected with, with the weight or strength which is assigned to the specific connexion.

Each of the hidden neurons is now activated by the sum of their input values, which is depending on both the pattern of the firing input neurons and the weight

of the their connexions. The hidden neurons only fire if their activation value exceeds a certain threshold level, and the output neurons are activated in the same way.

5 Rules in Neural Networks

The neural network is 'trained' with examples of input patterns and correct answers. When the training starts all the connexions are randomized, and the output of the network will in the beginning be rather incorrect. Then the correct answer is typed as a second input, and by a process called back propagation all the connexion weights activated by the input sentence are changed. The connexion weights yielding correct output are increased and the connexion weights yielding incorrect output are decreased with a certain rate.

Below I mention some of the 100 Danish input sentences—or rather strings of 9 words, the central word *i* and 4 words to the left and to the right—and the correct answer, i.e. the best reading of the preposition *i* in the context.

20. sikre at hver deltager -i- samme projekt i hele = ARG2
21. deltager i samme projekt -i- hele projektets løbetid til = DUR
22. dominere dette marked og -i- stigende omfang eksportere fra = MEA
23. nu er under overvejelse -i- nogle af de større medlemsstater = LOC
24. til et sådant nyt program -i- stor målestok er kommet = MEA
25. Esprit velkommen til mødet -i- juni 1992 og godkendte = TIME
26. XXX . Det sker -i- 1992 . XXX XXX = TIME
27. som anvendes i operationer -i- mange versioner og varianter = MEA
28. og afprøvning af VLSI/systemer -i- cilisium eller andre halvledere = LOC
29. beslutning end en afgørelse -i- rådet = LOC
30. fuldt kan støtte brugeren i kommunikationsprocessen, og som = ACT
31. ; de vil resultere i nye produkter, processer = ARG2
32. anvendelse foregår meget Langsommere i Europa end i Japan = LOC
33. af alle varer fremstillet i fællesskabet er i små = ARG2

When the connexion strengths have been adjusted a number of times with a number of input sentences the pattern of the connexion strengths will represent a rule wich will yield the correct output to each of the input patterns in the training set.

It is essential that the input sentences are authentic and not grammar book sentences, because all regularities in the input material, even the number of words from the word *i* to the punctuation mark, will be made into a rule by the network.

It is essential too that the number of input sentences is so high that all non-linguistic regularities of any kind are excluded. 100 input sentences are certainly not enough to make sure that all nonimportant word types have been placed in all 8 positions in the input picture.

I am not sure that a window of 9 words is enough, but in the first 100 authentic 9 word input sentences the rule triggering word has been present.

The training can be seen from two screen pictures, the first one showing input pattern, aswer and output of run no. 2 of sentence no. 26. The output is not even slightly in the right direction.

```
------------------------------------------------------------
fact no. 26                                        run no. 2
SENTENCE: XXX . Det sker  - i - 1992 . XXX XXX
   INPUT PATTERN              ANSWER        OUTPUT
   nvpaoi1234567      arg1:   ........      ........
-4 ............4-     arg2:   ........      ........
-3 ...XX...X....3-      loc:   ........      ........
-2 ...XX.X......2-      dir:   ........      XX......
-1 .X..X....X...1-     time:   XXXXXXXX      ........
 0 ..X.X.X......0       dur:   ........      ........
+1 X....X..X....1+      mea:   ........      ........
+2 ...XX...X....2+      sta:   ........      ........
+3 ............3+      act:   ........      X.......
+4 ............4+      emo:   ........      ........
   nvpaoi1234567       qual:   ........      ........
                       clot:   ........      ........
------------------------------------------------------------
```

But in run nr. 15 the network has 'learned' a rule completely, and gives the correct output to all the training sentences.

```
------------------------------------------------------------
fact no. 26                                       run no.15
SENTENCE: XXX . Det sker  - i - 1992 . XXX XXX
   INPUT PATTERN              ANSWER        OUTPUT
   nvpaoi1234567      arg1:   ........      ........
-4 ............4-     arg2:   ........      ........
-3 ...XX...X....3-      loc:   ........      ........
-2 ...XX.X......2-      dir:   ........      ........
-1 .X..X....X...1-     time:   XXXXXXXX      XXXXXXXX
 0 ..X.X.X......0       dur:   ........      ........
+1 X....X..X....1+      mea:   ........      ........
+2 ...XX...X....2+      sta:   ........      ........
+3 ............3+      act:   ........      ........
+4 ............4+      emo:   ........      ........
   nvpaoi1234567       qual:   ........      ........
                       clot:   ........      ........
------------------------------------------------------------
```

It is interesting that the established rule will give the correct answer to new sentences too, i.e. sentences which have never been given as input pattern before. In a way the network has 'learned' a linguistic rule inductively although it has not been formulated explicitly. It can be seen in the following three examples.

```
------------------------------------------------------------
new fact
SENTENCE: Mødet i Strassbourg varede - i - 3 uger
   INPUT PATTERN                ANSWER          OUTPUT
   nvpaoi1234567        arg1:   ........        ........
-4 X....X....X..4-      arg2:   ........        ........
-3 ..X.X.X......3-       loc:   ........        ........
-2 X...X..X.....2-       dir:   ........        XX......
-1 .X..X...X....1-      time:   ........        ........
 0 ..X.X.X......0        dur:   ........        XXXXXXXX
+1 ...X.X..X....1+       mea:   ........        ........
+2 X...X.......X2+       sta:   ........        ........
+3 .............3+       act:   ........        ........
+4 .............4+       emo:   ........        ........
   nvpaoi1234567        qual:   ........        ........
                        clot:   ........        ........
------------------------------------------------------------
```

```
------------------------------------------------------------
new fact
SENTENCE: fordi den fortsatte deltagelse - i - forhandlingerne
          med de implicerede
   INPUT PATTERN                ANSWER          OUTPUT
   nvpaoi1234567        arg1:   ........        ........
-4 ...XX..X..X..4-      arg2:   ........        XXXXXXX.
-3 ...XX.X......3-       loc:   ........        ........
-2 ...X.XX......2-       dir:   ........        ........
-1 X....X....X..1-      time:   ........        ........
 0 ..X.X.X......0        dur:   ........        ........
+1 X....X....X..1+       mea:   ........        ........
+2 .....XX.....X2+       sta:   ........        ........
+3 ...XX.X......3+       act:   ........        ........
+4 ...X.XX......4+       mea:   ........        ........
   nvpaoi1234567        qual:   ........        ........
                        clot:   ........        ........

------------------------------------------------------------
```

```
------------------------------------------------------------
new fact
SENTENCE: for en tredje rekvirent i samarbejde med en virksom
          hed
   INPUT PATTERN                ANSWER          OUTPUT
   nvpaoi1234567       arg1:    ........        ........
-4 ..X.X......X.4-     arg2:    ........        XX......
-3 ...XX......X.3-      loc:    ........        X.......
-2 ...X.XX......2-      dir:    ........        X.......
-1 X...X....X...1-     time:    ........        ........
 0 ..X.X.X......0       dur:    ........        ........
+1 X....X....X..1+      mea:    ........        ........
+2 ..X..XX......2+      sta:    ........        ........
+3 ...XX......X.3+      act:    ........        XXXXXXX.
+4 X...X...X....4+      emo:    ........        ........
   nvpaoi1234567       qual:    ........        ........
                       clot:    ........        ........
------------------------------------------------------------
```

I have not yet—efter 100 input sentences—statistics about how many percent of correct 'guesses' the network will make about new sentences, but it is already clear that it is possible to make a network which can solve the problem of disambiguation of prepositions without the enormous overgeneration which is made by filter rules in serial programming.

It should according to the theories be possible to train the same network to make the disambiguation of all the prepositions (or all the most frequent and ambiguous prepositions). The network I have described is in fact designed to compute 15 different prepositions. But I have not yet trained it with other prepositions than *i*.

6 The Power of Neural Networks

I imagine that the neural network in the translation process will be placed before the parser. The network is fed with the lexical words of the input sentence, and the relevant information about the semantic type of each word taken from the dictionary. All the prepositions in the sentence are then disambiguated by the network and the reading number asigned to them before they are parsed by the grammar parser. The product of the network would in the example from the beginning of this article be:

> *Lenin wrote this note in(DIR) his notebook in(DUR) 3 minutes in(LOC) Copenhagen in(TIME) 1897 in(EMO) anger.*

The enormous disambiguation power of the neural network results from three factors: the parallel distribution, which makes it fast, the nonlocal representation

of the rule, which makes it robust, and the statistical analysis, which makes it powerful.

The machine does not in fact compute the rule in parallel, but in a serial machine the program simulates the parallel processing, and that is enough to compute the disambiguation of a preposition in fractions of a second. 8462 calculations do not take more than a fraction of a second.

The rules which are used for disambiguation of the preposition *i*, one of which could be that *i* followed by a noun of the semantic type PLACE will normally be a *i(LOC)*, are not located in some of the connexions, but in the whole pattern of connexions both from input layer to hidden layer, and from hidden layer to output layer. So irregularities in the input, metaphors or syntactic errors, will not totally disable the rule, but only make minor changes in the output. The network will always find the 'best' solution, i.e. recognize the reading with most semantic fitness regardless how good or bad it is—exactly as we do even when we read the famous sentence: *Colorless green ideas sleep furiously.*

The nonlocal representation offers a solution of the problem of the so called hermeneutic circle, the problem that the whole can not be understood before the parts are understood, and the parts can not be understood before the whole is understood. The meaning of the sentence consists of, but is at the same time more than the sum of the senses of the words.

With nonlocal representations the meaning of the whole is represented, not as the sum of the meaning of the parts, but as a pattern or 'meaning' of something which is subsymbolic, subsignificant or with no meaning at all, but with a differentiating function, viz. the neurons of the hidden layer. So the network computes or recognizes the meaning of the whole by computing, not the sum of the parts, but the pattern of the subsymbolic parts (the hidden layer neurons) of the symbolic parts (the words) of the sentence.

That is exactly the function of letters or phonemes, which have no meaning but only a differentiation function, and nevertheless make it possible to transmit word senses and sentence meaning from sender to receiver in the communication process between humans.

But most important, the neural network will utilize information which can not be used in normal grammar rules, viz. probabilistic infomation. It is a linguistic rule that only *in(DUR)* will be followed by a noun of the type SCALE: *in 3 minutes.* Let us assume that it is a statistical rule that *in(DUR)* is followed by a cardinal number 1.000 times more often than *in(LOC)* is. It is not possible to formulate this regularity as a linguistic rule, not even as a preference rule, because of the possibility of the sentence: *she worked in two rooms.* The semantic network will utilize the probabilistic information but not make errors in this crucial example, because the pattern of connexion weights has learned the rule for the combination of cardinal numbers and measure nouns, not for cardinal numbers only.

References

Rumelhart, David E., James L. McClelland and the PDP sesearch Group. 1986. *Parallel distributed processing. Explorations in the Microstrucure of Cognition.* Vols. 1–3. MIT Press, Cambridge, Mass.

EUROTRA-DK
Njalsgade 80
DK-2300 Copnehagen S
Denmark

IVAR UTNE

Machine Aided Translation between the Two Norwegian Languages Norwegian-Bokmål and Norwegian-Nynorsk

Abstract

The article describes essential parts of a prototype system for machine aided translation from Norwegian-Bokmål to Norwegian-Nynorsk. The central parts of the system are a bilingual word list, inflection paradigms, phrases, and routines to deal with compound words. There are also syntactic and semantic rules, but they can be considered as preliminary. The article also includes a simple comparison of the two languages in question. The program is written in TurboPascal and runs on IBM PC compatible machines.

1 Introduction

The main goal of the project *Machine aided translation from Norwegian-Bokmål to Norwegian-Nynorsk* is to automatize translation between the two official Norwegian languages, both for proposing translated text, and for looking up words, phrases and text parts. There is also an important objective to gain knowledge about machine translation in general. The system will be based on comprehensive bilingual word lists and linguistic rules implemented as data for a computer program.

The development has so far partly been supported by the Norwegian Research Council for Science and the Humanities (NAVF) for 8 months in 1987–88. The author of this presentation has been responsible for the linguistic and computational work. Formal cooperators are also The Noregian Computing Centre for the Humanities and Department of Phonetics and Linguistics at the University of Bergen.

1.1 The Soft- and Hardware

The translation program and other data development programs (especially for word lists) are written in TurboPascal without usage of graphics and special database procedures, and is compatible with versions 3.0, 4.0 and 5.0. The source code takes about 50 kb, and the compiled version about 34 kb without the data tables. The program is run on IBM compatible machines with the operating system DOS 4.0 and lower. Among the machines is a Toshiba T1000 with 512 kb RAM and one floppy disc with 720 kb.

2 The Two Official Norwegian Languages, General Remarks

The two official written Norwegian languages are the majority language Norwegian-Bokmål (NB), with an unofficial English name Dano-Norwegian, and the minority language Norwegian-Nynorsk (NN), with an unofficial English name New-Norwegian. These languages can, for foreigners, be regarded as relatively similar both in spelling, inflection, vocabulary and syntax, which will be exemplified below. To Norwegians these two languages are felt as different, both because of the linguistic differences and because of the political impact of the languages. The two languages have by political decision been stated as official Norwegian languages with equal status.

NN is based on Norwegian dialects, while NB is based on an older version of written Danish which has later been improved with elements from Norwegian dialects. There is nowadays no complete agreement as to which of the two written languages are nearest to Norwegian dialects. That depends on which linguistic categories are considered and their relative importance. There is also no complete agreement to what extent a written language shall be based on dialects in opposition to written tradition.

This language situation implies that there will exist two Norwegian languages in the future. And this will motivate translation tools between the two languages, especially from the majority language towards the minority language.

3 The Two Official Norwegian Languages, Simplified Comparison

To give some impression of what the system has to work with, I will present a simplified comparison of the differences between the two languages, and consider some differences in spelling, inflection, suffixes, vocabulary, and syntax.

First it must be stated that the two Norwegian languages have more similar than different expressions, and that the differences are very often small and cover lots of expressions in a systematic manner (e.g. diphtong versus monophtong). This reflects a common history for the languages.

Norwegian-Bokmål				Comments
Mod. NB	Rad. NB	Rad. NN	Mod. NN	
spelling:				
grøt	graut		graut	(porridge)
høst			haust	(autumn)
løk	lauk	løk	lauk	(onion)
ren	rein		rein	(clean)
fler			fleir	(more)
sette		sette	setje	(put, set)
linje		linje	line	(line)
skap		skap	skåp	(cupboard)
hjem	heim		heim	(home)
inflection:				
gutter			gutar	(boys, m pl)
jenter			jenter	(girls, f pl)
epler			eple	(apples, n pl)
hus			hus	(houses, n pl)
problemer	problem		problem	(problems, n pl)
boken	boka	boka	[boki]	(the book, f sg def)
kastet	kasta		kasta	(threw, pret)
svømte			svømte	(svam, pret)
kommer		[kjemer]	kjem	(come, irreg pres)
skriver		[skriver]	skriv	(write, irreg pres)
suffixes:				
utdannelse,-ing			utdanning	(education)
kjærlighet			kjærleik	(love)
lærer			lærar	(teacher)
elektriker			elektrikar	(electrician)
vocabulary:				
tillatelse,løyve			løyve	(permission)
erfaring			røynsle	(experience)

Table 1: Spelling, inflection, suffixes and vocabulary.

The comparison is presented in two tables. Table 1 contains examples of spelling, inflection, suffixes and vocabulary. Table 2 contains examples of syntactic constructions.

The examples in Table 1 are grouped into four colums, two for each language. The leftmost column below NB and the rightmost column below NN contain the forms that are most different from the other language, and are usually called *moderate forms*. The rightmost column below NB and the leftmost column below NN represent expressions which are regarded as approaching forms, usually called *radical forms*. The square brackets marks forms which are allowed in writing except in documents from the central government and text books for the primary and secondary school, high school, and some other institutions. Two of

genitive:

guttens bil
=> bilen til gutten
(the boy's car; NN-phrase word by word: the car belonging_to the boy)

lederens forslag
=> forslaget/framlegget frå/til leiaren
(the leader's proposal; NN-phrase word by word: the proposal from/belonging to the leader)

passive voice:

bøkene kastes
=> bøkene vert/blir kasta
(the books are thrown (away))

boka blir lest
=> boka vert/blir lesen
(the book is read; common gender (i.e. masc. and fem.) singular)

congruence (occurs in passive and in predicative):

skriftet blir lest
=> skriftet vert/blir lese
(the publication is read; neuter singular)

bøkene blir lest
=> bøkene vert/blir lesne
(the books are read; plural)

nominalization:

forsamlingen gjorde vedtak om nye skatteregler
=> forsamlinga vedtok nye skatteregler
(the meeting/assembly approved new tax rules;
NB-phrase word by word: the meeting/assembly did/passed a resolution on new tax rules)

fyrbøteren har behov for mer kull
=> forbøtaren treng meir kol
(the stoker/fireman needs more coal; NB-phrase word by word: the stoker/fireman has/is_in need for more coal)

Table 2: Syntactic phrases

the expressions in the table are similar for the two languages, e.g. the words for *girls* and *houses*. This means that feminine plural inflection is the same (with some exceptions), and that neuter plural in some cases is similar.

In Table 2 there are examples of genitive, passive voice of verbs, congruence and nominalization. Each of the examples contains: NB phrase, NN phrase (prefixed by an arrow: =>) and an English translation often followed by some grammatical information. When the word order in one of the Norwegian phrases differs from the ordinary English translation, I have added a word by word translation of the phrase in that language. See for instance Myking 1989 for an instructive comparison of the two languages.

A system for machine aided translation will be based on these facts. The most significant differences are found in spelling and inflection of words. As well, we have to perform a syntactic and semantic analysis (parsing) to identify the current form among homonyms (in source language which may imply different expressions in target language), to ensure congruence (which exist in NN; see examples above) and to ensure rewriting of syntactic constructions.

4 The Structure of the System

At present the system is a bilingual translation system with transfer routines, i.e. target language data is inserted in the result structures from the analysis of the source language. A more sophisticated solution, not implemented yet, could be an interlingua system, where the source text would be translated to a language independent representation before generation of the target text.

The system is based on syntax analysis, and a semantic analysis guiding the syntax analysis. The development is performed bottom-up, i.e. it has been of importance to establish a skeleton of a total system, and then refine the parts. Because of this, the linguistic models that are used have to be considered as preliminary.

The system consists of a computer program and a collection of data files. The rules and the reference data is stored in text files that can easily be edited. During the development phase parts of the linguistic rules are part of the program code because of preliminary design issues.

The data files consist of linguistic rules for controlling the translation process (i.e. analysis and generation) and reference data for the rules. For the time being there is one rule file, with syntax rules, and seven reference files, which are the bilingual word list, regular inflection patterns, irregular inflected words, phrases, word formation patterns (for compound words), semantic classification, and internal code conversion (which will not be explained further).

5 Syntax Rules

The syntax rules are binary unification rules with context constraints. The system is working bottom-up, i.e. grouping related words in larger and larger units until the result is complete sentences. At the time being the rules are not based

on a specific linguistic model, and the rules must be regarded as a basis for establishing a skeleton of a syntax parser. Typical rules are `ADNOM + NOM -> NP`, `V + NP -> VP`, `NP + VP -> S`, `PREP + NOM -> PP`. The rules are at the time being based on the sequence of `SUBJECT + VERBAL + OBJECT` in main clauses, and existence of inversion in certain types of subordinate clauses. The syntax rules can to some extent handle relative clauses, conditional clauses and adverbial clauses.

The context constraints concern both syntax and semantics. The syntactic constraints concern the existence of grammatical categories in the context, e.g. `SUBJECT`, `OBJECT`, `OBJECT2`, `THAT`-clauses and `INFINITIVES`. The semantic constraints concern what semantic features may characterize the grammatical categories (see above) which is tied up to a kernel word or phrase (usually a verb). It is quite a big project to work out a complete classification of semantic specification for all words. Therefore our principle is to start with classifying words which otherwise may cause wrong analyses and translations. The parser uses these specifications to reject analyses where the semantic features are in conflict. If one or none of the two phrases (words) have semantic specifications, a semantic check will not be performed. For further details, please see the description of the data file containing the semantic classification.

6 The Bilingual Word List

In the bilingual word list, a stem and an inflection code for both NB and NN are required. The follwing optional categories are present: context constraints (for the moment) for verbs (see syntax rules below) and semantic classification. The entries are stored in alphabetical order in a file with fixed record length to facilitate quick searches. The data is updated in a text file and converted to fixed file format.

The representation of stems are carried out according to a format that enables the program to search for the part of a word that is common to all forms, e.g. *mut* are common for *mutter* (indef. sing.) and *mutre* (indef. plur.), which are the NB word for the English word *nut*. Instances of double consonants which are single in inflected forms are represented with a hyphen in front of the second consonant, as for instance *-t*. And suffixes which are reduced in certain inflected forms are prefixed with an asterisk (*), as for instance **er*. According to this the entry for 'mutter' is *mut -t *er*.

7 Inflection

The system contains two data files consisting of inflection information, a table of regular inflection patterns and a list of irregular inflected words.

The presentation of the regular inflection paradigms has the following format for each pattern: pattern code + inflection expressions. For one of the inflection

patterns for verbs, e.g. one of the two patterns for NB for the verb *kaste* (throw, cast), this means:

```
v6 e er a a
```

The reading of this is that according to pattern v6, these endings should be appended to the stem *kast* to produce the forms *kaste* (infinitive), *kaster* (present tense), *kasta* (past tense) and *kasta* (past perfect). There is a module in the program which uses this pattern information in combination with the stem form and its inflection code found in the bilingual word list. This routine will be exemplified in more detail below.

The presentation of irregular inflection is divided between the bilingual word list and a list containing the irregular forms in NN. In the bilingual word list the most relevant forms (four of nouns and verbs) of NB are listed according to the alphabetical sequence. Each of the NB forms are marked for a form category in the bilingual word list, e.g. present of verb or definite singular of a noun. In the list of irregular NN forms we usually find four forms organized according to a sequence, i.e. infinitive—present—preteritum—past perfect, that expresses what form it is.

8 Phrases

Phrases are stored in a text file to which the program has access. Words that usually occur in the phrase in only one form are presented in that form in the file. Words that may be inflected as part of the phrase are presented in inflected forms (at the time being in different entries, but this will soon be concentrated into one entry). A wild card (*) has been inserted in which an additional word can be inserted (this will be expanded to phrases), e.g. adverbials as for instance *not*. Each phrase (record) is prefixed with one of the words in the phrase. That word acts as a key to the phrase. This means that the keyword is also listed in the bilingual word list with a code that tells the parser to look into the phrase list. This keyword is usually chosen among words that is regarded as having low frequency in text, to prevent too many superflous searches. To the right of the phrase we find the NN synonym, prefixed with a dot ('.'). An entry may look like this (v0 is the inflection code for irregular verbs):

```
behov    ha v0 * behov for          .trenge v0
```

9 Word Formation, Compound Words

A module has been implemented for the analysis and translation of compound words. The main principle is that compound words that do not exist in the bilingual word list may be analysed as being composed of words in the list. The processing of word formation has two important types of restrictions.

One type of restriction is that words that are part of a compound word must usually be the so-called form 1, i.e. infinitive of verbs, indefinite singular of

nouns and common gender (i.e. masculine and feminine) of adjectives. But the rightmost (last) part of the word may of course be inflected. At the time being these restrictions are included in the program code.

The other type of restriction concerns what part of speech the different word parts can come from. An adjective and a noun may be combined, like *rødvin* (red wine). But we cannot combine a noun and an article, like *gutten* (the boy) read as *gutt* + *en* (boy + a) and will be translated to *guttein*. In the system these types of restrictions are written into a data file where patterns express illegal sequences of parts of speech.

10 Semantic Classification

As mentioned above, a module for semantic classification to guide the parser has been included. The system is implemented as a thesaurus in a tree structure. The raw data is written into a text file with expression for the subordinate concept in the left column and the expression for the superior concept in the right. From these data the program builds the thesaurus structure.

The parser utilizes this information to decide if two words or phrases can be unified according to the semantic classification. This is of importance when the two words/phrases have semantic representation at different levels in such a way that one of the representations is subordinated to the other. That means that they are compatible.

11 The Morphological Analysis for Single Stem Words

The main component of the word analysis is recognition of words in the bilingual word list and the related inflection endings in the inflection tables. In the analysis of compound words this information is combined in repeated word-recognition during the whole compound word.

The general strategy for word recognition works from right to left. The program first searches in the word list for the whole word. If it is not found, a new search is performed with the rightmost character deleted. The search process is repeated with deletion of the rightmost character until there will be a match or there will be no search string left. If the input word is *kaster* the program will search for

kaster,
kaste, and then
kast,

which is a verb stem in the list. Kast has two different inflection codes, v6 and v7.

The program will expand the stem with endings from both the inflection paradigms in order to find identity between the complete input word and one or more inflected forms of the word list entry. The paradigms are:

v6	e	er	a	a
v7	e	er	et	et

According to this the stem *kast* and inflection endings produce these word forms:

v6:	kaste	kaster	kasta	kasta
v7:	kaste	kaster	kastet	kastet

The second word forms, i.e. the present tense in both the paradigms match. That means that the program has identified *kaster* as present tense of the stem kast.

12 The Morphological Analysis of Compound Words

The analysis of compound words follows the same main strategy as for single stem words. The main difference is repeated searches for stems, endings and joining characters. If the input word is *lærerhøyskolestudenter* (Teachers' Training College Students, read stem by stem as: 'teacher + high + shool + student + s') the program will delete the rightmost letters until it matches *lær*. That means that searches will be done for the following character sequences:

lærerhøyskolestudenter

lærerhøyskolestudente

lærerhøyskolestudent

.

lærerh

lærer

lære

lær

The matched stems in the word list are the noun stem *lær* with the suffix *-er* and the two masculine inflection codes m3 and m4, and the verb stem *lær* with the inflection code v1. The program first expands these two stems with the three paradigms, like this:

m3:	lærer	læreren	lærere	lærerne
m4:	lærer	læreren	lærerer	lærerne
v1:	lære	lærer	lærte	lært

None of these word forms match the whole (compound) word *lærerhøyskolestudenter*. Then we have to activate the module for analysis of compound words. As stated above, according to our rules, words which are part of compound words can only occur in form 1. In this case there are three candidates that matches the beginning of the compound word, of which two word forms are unique: *lærer* and *lære*. This means that the rest of the word is either *høyskolestudenter* or *rhøyskolestudenter*. The program tries both strings. It fails for the last one, but will succeed for the first. The identified wordforms as parts of the compound word are *høy*, *skole* and *studenter*. The program will also propose the noun *student* and the copula verb *er* instead of studenter. This solution will be excluded because of the rules which imply that part of speech sequences are not allowed in compound words.

This word will be translated to NN *lærarhøgskulestudentar*.

13 Parsing Strategy

The parsing strategy is bottom-up. It is based on binary rules like those listed under "Syntax rules". The rules consists of two conjoining (a pair) syntactic labels and restrictions for their syntactic features (gender, definitness, number, tense etc). The parsing module per June 1989 is a preliminary one. The details of the syntactical and semantical description and procedures will be radically revised and accomodated to a methodology based on a Lexical Functional Grammar (LFG).

References

Myking, Johan. 1989: Term Harmonization, 'Selective Purism' and Language Autonomy. An "intra-national" case. To be printed in the proceedings from The 7th European Symposium on LSP, Budapest, 21.–26. August 1989.

Strømgaten 53
N-5007 BERGEN
Norway

Part III

Computational Lexicography

HENRIK HOLMBOE

Dansk Radiærordbog

Abstract

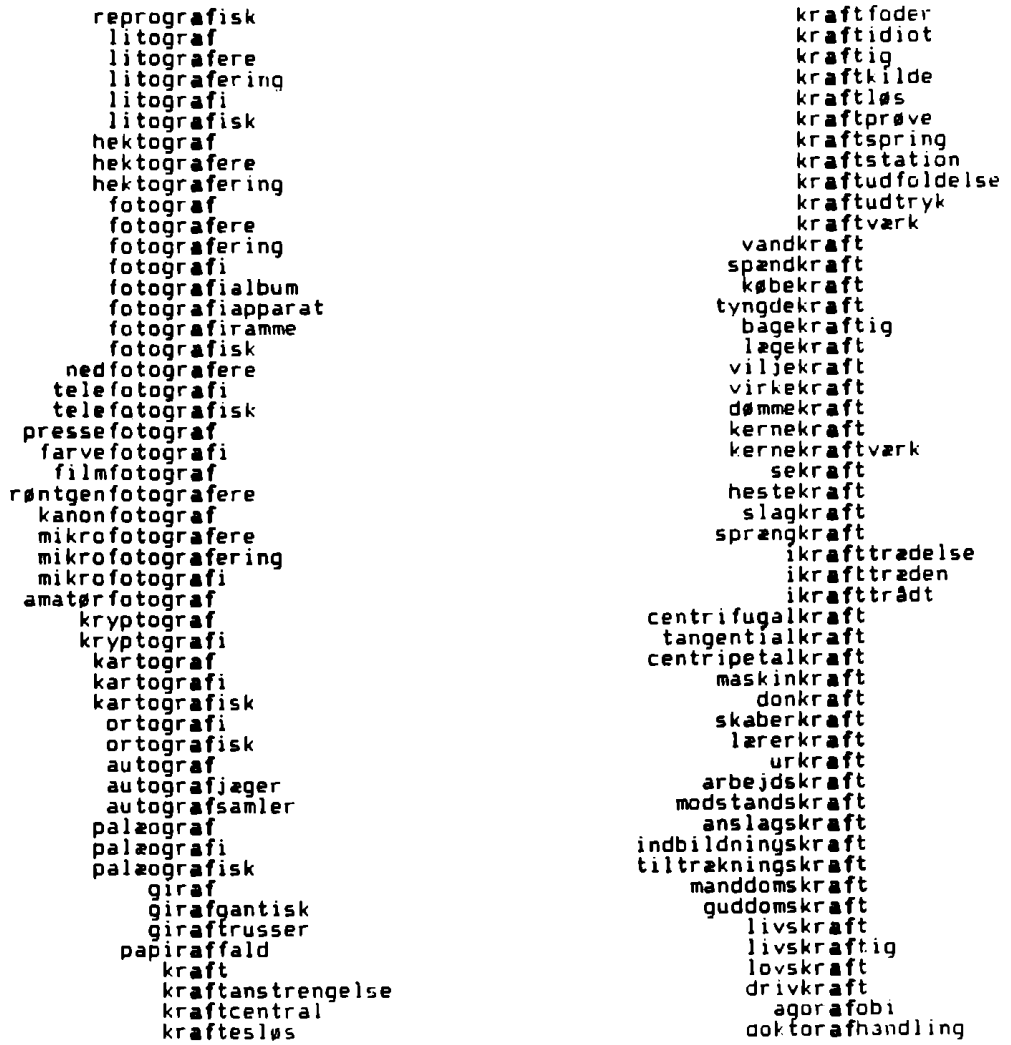
A Radial Dictionary of Danish.

A Radial Dictionary is a KWIC-concordance, where instead of using key words key letters are used. Any pair of letters can be used as a key to identify and find all words containing a specific substring of two or more letters. All words containing the substring in question will be found together in the dictionary whether the substring is word initial, medial or word final. Thus the radial dictionary can be used as a morpheme dictionary as well. For people interested in derivation and composition, which are very frequent phenomena in Danish, the dictionary gives easy access to examples and material difficult to find otherwise.

Jeg er glad for denne lejlighed til at fremlægge resultaterne af mit arbejde med Dansk Radiærordbog ved de nordiske datalingvistikdage i Reykjavik. I 1986 kunne jeg på Säby Säteri præsentere planer for udarbejdelse af radiærordbogen for et mindre nordisk forum af kolleger. Ringen sluttes således med denne rapport.

Både ordbogen og ordbogens koncept er nyt for danskens vedkommende; derfor vil jeg indledningsvis redegøre for ordbogens opbygning: En radiærordbog gør det muligt at finde alle ord, der rummer en valgt substreng på to eller flere bogstaver, cf. fig 1. Denne substreng kan være en konsonantgruppe, en stavelse, et morfem, en del af et kompositum eller en derivationsendelse. Alle de ord, der rummer den valgte substreng, vil findes samlet på ét sted i ordbogen, hvadenten substrengen står initialt, medialt eller finalt. I en almindelig ordbog, dvs. en ordbog sorteret alfabetisk fra ordenes begyndelse (progressivt alfabetiseret), kan man nemt finde alle ord, der begynder med et bestemt præfix eller en bestemt stavelse. I en retrograd ordbog (finalalfabetiseret ordbog) er det tilsvarende let at finde alle ord, der slutter på et bestemt suffix eller en bestemt stavelse eller på en bestemt gruppe af bogstaver. Radiærordbogen kombinerer disse to muligheder med muligheden for også at finde ordmediale segmenter.

Denne specielle måde at opstille ordene på tilvejebringes af en algoritme, som placerer ordet lige så mange forskellige steder i ordbogen, som det rummer bogstavpar. Hvis man har en bestemt ordmængde og lader denne ordmængde gennemløbe den radiære sortering, vil det føles som en eksplosion af ordmængden, i det mindste for et sprog som dansk, som er relativt rigt på komposita



Figur 1.

og derivativer. Mit udgangspunkt var en ordbog på ca. 50.000 ord, og resultatet blev en radiærordbog på ca. 400.000 ord. En sådan sortering lader sig naturligvis i praksis kun gennemføre ved hjælp af en datamat.

Ordbogen anvendes på følgende måde: Først vælger man den substreng, som man vil finde eksempler på. Derefter anvender man substrengens sidste bogstavpar som opslagsnøgle. Alle ord i ordbogen er ordnet efter bogstavpar, som er trykt med fed skrift. Den nævnte nøgle vil altså altid passe til et bogstavpar i ordbogen. Med nøglen som udgangspunkt skal man nu søge retrogradt alfabetisk mod ordets begyndelse, og når man har fundet overensstemmelse med den ønskede substreng, vil hele materialet, dvs. alle eksempler, være samlet i de følgende linier, alfabetisk sorteret, først retrogradt, derefter—dvs. alt andet lige—progressivt. Denne bevægelse fra bogstavparret først til venstre, så til højre, har fremkaldt associationen om radier, og derfor betegnelserne radiærordbog og radiær sortering.

Sorteringsalgoritmen er implementeret på følgende måde, cf. fig.2: Ethvert ord, man ønsker at sortere, kopieres over i lige så mange ordfiler, som ordet rummer bogstavpar. Et ord på fem bogstaver kopieres således over i fire forskellige filer, da der jo er fire bogstavpar i en streng på fem bogstaver. Der findes en specifik fil for hvert enkelt bogstavpar; denne fil vil rumme alle eksempler på netop dette bogstavpar. Ethvert ord transformeres nu til en ny streng, der består af spejlbilledet af den del af ordet, der går forud for det bogstavpar, der er karakteristisk for filen, efterfulgt af resten af ordet i dets uændrede form. Disse nye strenge alfabetiseres nu normalt i overensstemmelse med det pågældende sprogs alfabetiseringsregler, og derefter spejles første del af strengene tilbage igen til deres oprindelige form, og hermed er selve sorteringen gennemført. For at gøre resultatet mere brugervenligt har jeg centreret bogstavparret og trykt det med fed skrift. Denne sidste operation gør det også iøjnefaldende, at radiærordbogen kan betragtes som en bogstavparkonkordans ordnet og opstillet efter KWIC-princippet.

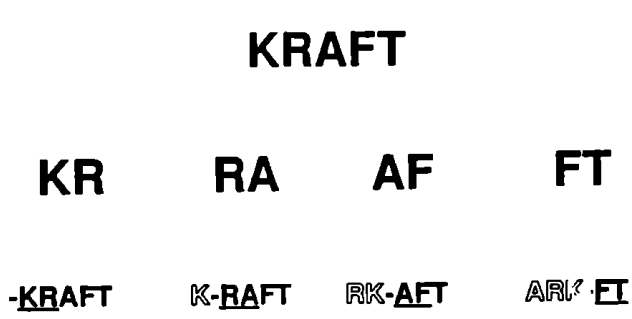
Som nævnt kan man i radiærordbogen finde eksempler på enhver forekommende substreng i sproget. Hvis det, man søger, er eksempler på forskellige bogstavkombinationer, vil en radiærordbog formentlig være til hjælp i ethvert sprog. Hvis man vil bruge radiærordbogen som morfemordbog, vil den være mere eller mindre nyttig afhængigt af nogle typologiske karakteristika ved det sprog, man studerer. Hvis sproget er som dansk derved,

- at det har mange derivativer og komposita,
- at det har en kreativ og innovativ orddannelse,
- at dets stammer og derivativer er relativt stabile og
- at derivation og komposition iværksættes ved agglutination,

så kan radiærordbogen skaffe sin bruger nem adgang til materiale, som det ellers ville være vanskeligt at samle sammen.

Hér skal man formentlig også søge en del af forklaringen på, at fænomener som komposition og derivation i dansk er så sparsomt udforskede og beskrevet. Fænomenerne er efter min mening vigtige, specielt set i et oversættelsesperspektiv. Et sprog som dansk anvender jo orddannelsen til at løse en række problemer, som på f.eks. engelsk løses af syntaks og ordstilling. Med andre ord: Hvis man ønsker samme information om dansk, som man får om engelsk ved at iagttage svarene på spørgsmål vedrørende syntaks, hvilke spørgsmål skal man da stille? Mit svar vil være: Naturligvis syntaktiske spørgsmål, men tillige spørgsmål vedrørende orddannelse.

Som ovenfor anført er ordningskriteriet for radiærordbogen alene bogstavstrenge, altså et rent overfladekriterium eller—anderledes udtrykt—et kriterium, der alene tager hensyn til sprogets udtryksside. I hvilket omfang vil en sådan ordbog kunne anvendes som morfemordbog? Vil ordbogen ikke give forkerte eller i det mindste upræcise oplysninger, når man søger efter størrelser på tegnniveau og ikke kun efter udtrykssidens inventar? Naturligvis er betænkeligheder af denne type berettigede.



A:

aften
aftræk
drivkraft
fyraften
giraftrusser
ikrafttræden
kernekraftværk
kraft
kraftanstrengelse
kraftig
livskraftig
saftig
skaft
vandkraft

B:

aften
aftræk
arkvirdft
aryfften
arigftrusser
arkifttræden
arkenrektværk
arkft
arkftanstrengelse
arkftig
arksvilftig
asftig
aksft
arkdnavft

C:

a-ften
a-ftræk
aks-ft
arig-ftrusser
ark-ft
ark-ftanstrengelse
ark-ftig
arkdnav-ft
arkenrek-ftværk
arki-fttræden
arksvil-ftig
arkvird-ft
aryf-ften
as-ftig

D:

aften
aftræk
skaft
giraftrusser
kraft
kraftanstrengelse
kraftig
vandkraft
kernekraftværk
ikrafttræden
livskraftig
drivkraft
fyraften
saftig

E:

aften
aftræk
skaft
giraftrusser
kraft
kraftanstrengelse
kraftig
vandkraft
kernekraftværk
ikrafttræden
livskraftig
drivkraft
fyraften
saftig

Figur 2.

Fordi en substreng er identisk med et morfems udtryksside, er denne substreng ikke nødvendigvis et eksempel på dette morfem; så set fra morfemanalysens synspunkt vil der være „støj" i et opslag; algoritmen overgenererer, men ikke meget. Betragter man en række ord med et bestemt morfem, vil man meget let kunne se bort fra de irrelevante og distraherende eksempler, cf. fig. 3. Fordelen ved at sortere efter bogstavstrenge er, at alle mulige eksempler er samlet. Også hvis man var interesseret ikke i morfemer, men f.eks. i konsonantgrupper, vil alle eksempler være til stede samlet, og ikke kun de eksempler, som ville slippe igennem det filter, som en hvilken som helst morfemteori vil etablere.

```
  sammenklappelig
    stænklap
 skulderklap
  tænderklaprende
     skyklap
      kollaps
diskusprolaps
       volapyk
        plapre
   efterplaprer
     overlapning
     overlappe
    papirlap
        slap
        slap-af-
        slappe
        slappe-af-
        slappelse
        slapsvans
      afslappe
      afslappelse
   papirslap
     æskulapstav
```

Figur 3.

Til sidst vil jeg nævne, at ordbogen allerede har bevist sin praktiske anvendelighed på Hovedstadens Ordblindeskole i København. Denne skoles klienter har det specifikke sproghandicap, at de vanskeligt kan isolere og genkende ordenes bogstaver enkeltvis, og derfor opbygger de deres læsefærdighed ved at lære at genkende større grupper af bogstaver, hele ord eller dele af ord. Radiærordbogen har gjort det lettere for skolens lærere at sammensætte relevant undervisningsmateriale for klienterne.

Afdeling for Datalingvistik
Handelshøjskolen i Århus
Fuglesangs Allé 4
8210 Århus V

JÓN HILMAR JÓNSSON

A Standardized Dictionary of Icelandic Verbs

Abstract

Traditionally, entries in dictionaries, including historical dictionaries, are organized on semantic principles, and syntactic features only play a secondary role. This holds for all wordclasses, even for verbs where syntactic information is both varied and of great importance. In this paper it is maintained that the characteristics of each wordclass should be given more attention. This is especially important in the description of Icelandic verbs, which vary structurally to a great extent, due to the complexity of the case-system and other related factors. These complexities must take a prominent place in the description given in any dictionary of Icelandic.

The Achilles' heel of large scale dictionaries, where the aim is completeness of description and authentic citations are used, is the difficulty in the retrieval of information. By treating the verbs as an independent project, and making full use of their characteristic traits, the perspective taken in SDIV differs from the traditional one. Syntax is superordinated to semantics in the description of the verb, which greatly facilitates the ease of access to information contained in each entry, as a fixed format will then suffice for all verbs.

The text of the entries in the proposed dictionary is produced by stages, based on an analysis of individual citations for each verb contained in the database. The end result, produced by the database itself, connected to the layout-program TEX is a finished dictionary entry, needing only minor revisions.

1 The Diversity of Historical Dictionaries

A large scale historical dictionary contains a great variety of diverse information on the vocabulary represented. A reader browsing through a work of this type will often notice changes taking place in the language; new words will appear and become common and others will become anachronic and be forgotten. The reader can, given sufficient perseverance in the study of individual entries, obtain a wealth of information on changes in the use of individual words through time, both in regard to meaning and structure. If the reader is willing to immerse

himself more fully in a dictionary of this kind, he will also find that it contains additional information of a different type, e.g. information of a cultural, ethnic, stylistic, or colloquial nature.

The totality of all this information on individual entries adds up to a massive collection of data on the lexicon and its characteristic features; in itself an important contribution to linguistic and cultural history.

1.1 The All-Importance of Semantics

If truth be told it is the enormous quantity of material in itself which gives the large historical dictionaries their edge. However, the very same quality is also their Achilles' heel. Access to the wealth of information contained in these dictionaries is severely restricted and hampered as the only way of access to individual entries is through their alphabetical order. It is therefore almost impossible to search for information on the common features of various entries (cf. Jörgen Pind 1986).[1]

The ease of access to information contained in the text of a dictionary is, of course, dependent on the structure of the text itself. Without overgeneralizing too greatly, it can be said that semantic principles are, traditionally, the preferred criteria used in the structuring of historical dictionaries. Information on individual features not of a semantic nature can therefore only be found under headings defined by the meaning of the word in question. The all-importance of semantics is especially apparent in the entries for nouns and adjectives, but it is also obvious in the entries of verbs, which are typically also structured according to syntactic criteria.

1.2 The Uniqueness of Verbs

The verbs, to an even greater extent than the other wordclasses, make the limiting aspects of the traditional entry of the historical dictionary immediately apparent. Search for information is made all the more difficult by the sheer mass of material contained in each individual entry. The editors are faced with the problem of describing the structural richness which characterizes this wordclass. The use of verbs is characteristically determined by syntactic features, and in every instance of use features such as transitivity and case-marking have to be taken into account, as do phenomena such as periphrastic constructions, phrasal verbs, and in the case of Icelandic, the distinction between active and middle voice.

A proper balance between syntactic and semantic characteristics is vital in any well-organized description of verbs, and it is of the utmost importance in editing the material to make full use of the inherent syntactic structure. It is worth bearing in mind that these features may eventually help the user of the proposed dictionary to find what he is looking for in the text of the dictionary.

[1] The Oxford English Dictionary has now appeared in a computerized version in which there is direct access to various features of single entries and across entries (cf. Weiner 1989).

In fact it holds for all wordclasses that work is being done on the possibilities of describing each one of them in such a way that its characteristic features are reflected in the entries.

2 Data for a Historical Dictionary of Icelandic

Work on the editing of the collection of data amassed in the previous 40 years for a historical dictionary of modern Icelandic[2] was started at *The Institute of Lexicography (Orðabók Háskólans)*, in the beginning of the eighties. At the time there was no firm outline of how such a dictionary should be organized, and there was no existing Icelandic dictionary on which to pattern the new one. This, however, also entailed not being tied to an older model, thus gaining a fair amount of freedom in developing the methods used in the editing of the material, and in deciding what the format of the proposed dictionary should be.

The collection referred to above consists of three parts:

- The Written Language Archive: Approx. 2,600,000 citations.
- The Spoken Language Archive: Approx. 190,000 citations. (See Gunnlaugur Ingólfsson 1988 for a closer description.)
- Excerpts from unpublished dictionaries in manuscript: Approx. 90,000 citations, mainly from the 17th, 18th and 19th centuries. (See Guðrún Kvaran 1988 for a closer description.)

2.1 Two Ways of Processing

As can be seen from these figures, the sheer mass included in the collections makes it a tremendous task to edit them to a satisfactory standard, especially if information from all available dictionaries and neccessary additional material from other sources is included. Therefore, bearing in mind the wish that the material should be computerized to as high a degree as possible, it was decided to proceed on the processing of the data in two different ways at the outset. The first part of the processing is making an index of The Written Language Archive in its entirety. This index, The General Index, serves as a survey of all the material in this archive, (cf. Björn Þ. Svavarsson and Jörgen Pind 1989, and Jörgen Pind 1989, both in this volume). The second part of the processing entails making a detailed description of a chosen part of the vocabulary, including data from all three archives.

I do not think it necessary for me to justify why the verbs were chosen as the first subject of a detailed description of our archives. The verbs are very centrally placed, both in a lexical sense, i.e. in the lexicon itself, and in a lexicographical sense, as any dictionary will show. No other wordclass gets similar attention from lexicographers; there are even special dictionaries on verbs and verbal idioms.[3]

[2]The period after 1540, i.e. from the start of the age of printing in Iceland.

[3]These are, roughly, of two types; valency-based dictionaries (cf. Helbig and Schenkel 1969), and dictionaries of phrasal verbs (cf. Courtney 1983).

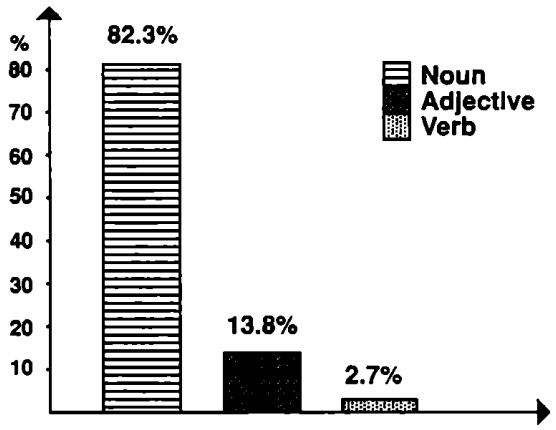


Figure 1: The Written Language Archive: Relative Size of Word-Classes

2.2 The Status of Verbs in the Lexicon

Let us take a look at a few diagrams showing the special status of the verbs in the lexicon. The information comes from the General Index mentioned earlier.

The comparative size of the wordclasses can be seen in Figure 1. Note the low percentage of verbs, i.e. only 2.7% of the lexicon. In real numbers this comes to about 16,000 words. For comparizon it may be added that the percentage for nouns is 82.3%, or over 500,000 words.

The uniqueness of the verbs among the major wordclasses is also shown in the number of citations per verb in the archive, compared to the number of citations per word in the other wordclasses, as illustrated in Figure 2. Words which occur in only *one example* are the least common in the verbs, whereas

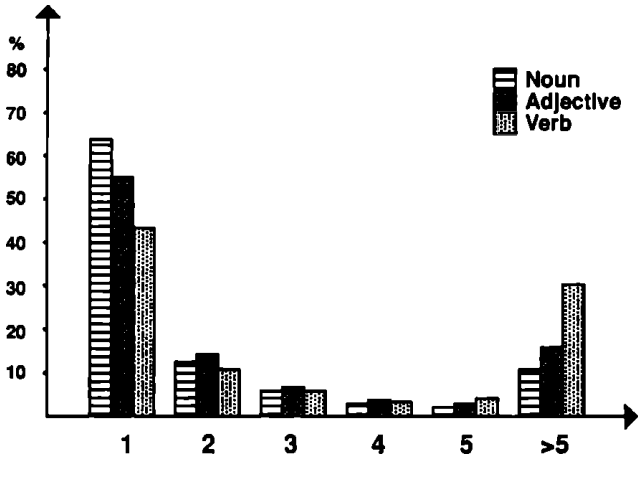


Figure 2: The Written Language Archive: Number of Citations per Lexeme

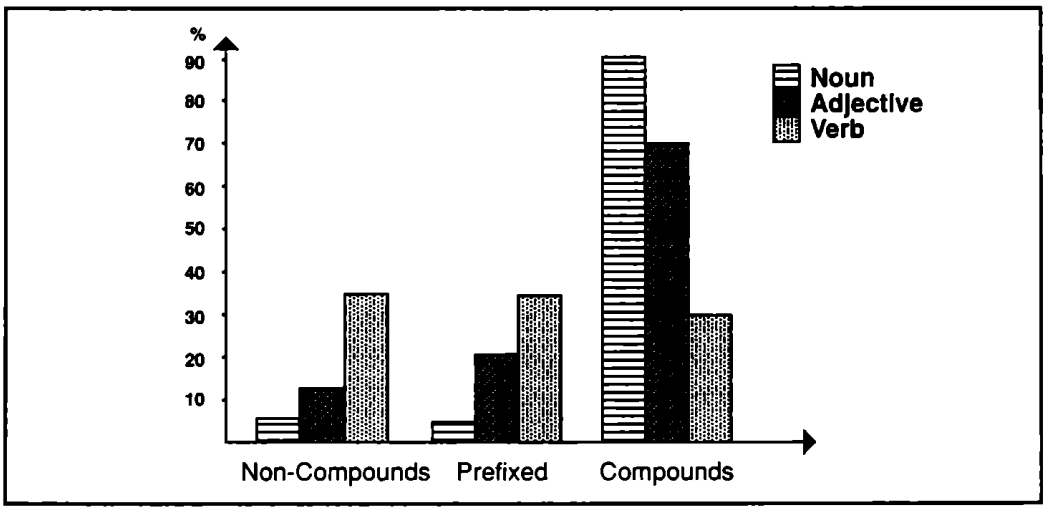


Figure 3: The Written Language Archive: Types of Word-Formation

words occurring in *more than five examples* are more common in verbs than in the other wordclasses.

This is also connected to a different distribution of types of *word-formation*. The verbs as a wordclass consist to a large extent of non-compounds, the percentage of *compounds* being lowest in the verbs, compared to other wordclasses, as shown in Figure 3.

The verbs are also relatively stable in comparizon to the other wordclasses when it comes to the *age distribution* of the examples. This is shown in Figure 4. A relatively high percentage of the verbs is attested both with citations from the 16th century and from the 20th. On the other hand, a relatively low percentage of the verbs is only attested in the 19th century or in the 20th.

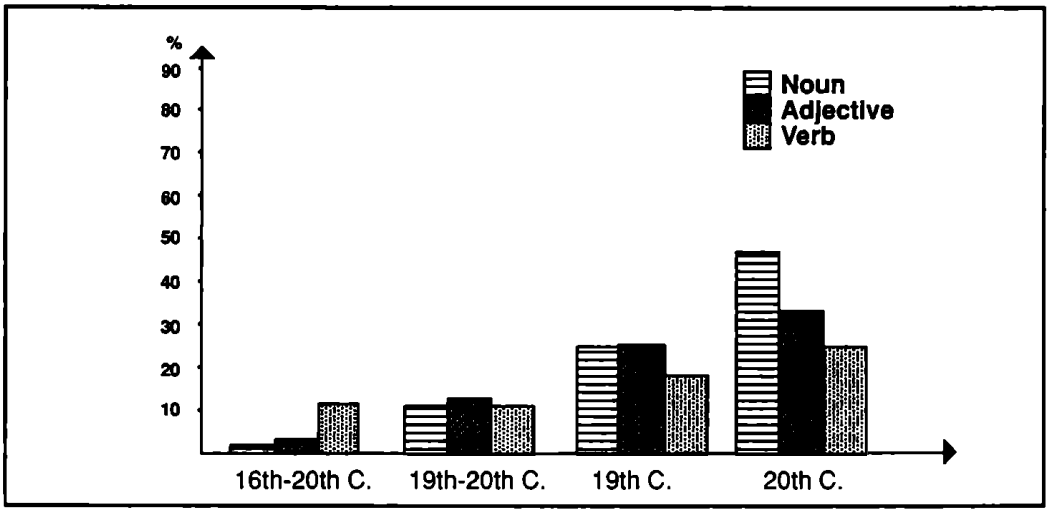


Figure 4: The Written Language Archive: Age Distribution

2.3 Analysing the Data

Right from the beginning the intention was to use archives themselves as a direct source of the structure and format of the actual entries in the dictionary. There were many reasons for this. The production of a fully-fledged dictionary of a preconceived type was not the only purpose of the project. In reality the purpose was threefold:

- Describing a number of lexicological and lexicographical features of each individual verb.
- Producing a dictionary text of a standardized format as end result of the analysis.
- Making it possible to search for different lexicological features across a number of verbs.

By starting with a detailed analysis which in itself was not dependent on the actual format the entries of the dictionary were proposed to have, there was ample time to decide what that format should finally be.

I want to stress the following points to show the importance of using the actual citations as the base of the analysis:[4]

- The context is vital for the understanding of the word-forms and their use.
- The citations show historical development.
- Both the analysis and the description must be based on actual language use, i.e. attested examples.

2.4 The Data in the Three Archives

Before coming to the analysis itself, I would like to stress the main characteristics of the different archives in the collection.

The citations in the Written Language Archive characteristically show words in context, thus showing the actual use of the word, inflections, syntactic structure, etc. As a rule, the meaning has to be inferred from the context.

The material in the other archives is of a different kind. The words are usually not given in context, and consequently information on language use is missing. Characteristically the meaning of the word is given, often taking the form of a definition, but information on cultural and historical aspects is sometimes included.

[4]This method is gaining ground among lexicographers working on modern languages, see Fox 1987.

3 The Analysing Process

Now I will return to the analysis itself, and its different stages. Information on the computerization can be found in the papers by my colleagues, Jörgen Pind and Björn Þ. Svavarsson (this volume). Let me just mention that the database being used is *Revelation*, in the new and improved version from 1987, *Advanced Revelation*. Initially, the database was only running on a single terminal, but in 1986 three more were added. The lay-out of the text of the dictionary entries is made by using the lay-out programme TEX.[5] We are now in the process of reworking the programs used for the analysis in order to switch operating systems from MS-DOS to UNIX.

3.1 The Seven Stages of Analysis

The analysis itself proceeds in stages, taking the Written Language Archive as a starting point. The order of the fields (or slots where different items of the analysis are placed) is structured in such a way that the analysis is done by steps, from the initial processing up to the production of the completed dictionary entry. This step-by-step method has many advantages:

- Each step provides additional data which can be used in sorting and analysing in the following stage.
- At each stage of the analysis the data from the previous stage is used (and corrected).
- A substantial amount of information is available for use at an early stage of the analysis.
- Division of labour between personnel working on the project is relatively easy.

There are seven basic steps in the analysis, as illustrated in Figure 5 on the following page. The first step entails registering the actual citations and their source. The second step concentrates on the morphological structure of the word. The third step handles syntactic structure and contextual information. In the fourth step the citations are analysed semantically and definitions are formulated. The fifth phase involves the sorting of the semantic information in the preceding stage, and ordering of definitions. The sixth stage contains cross-references between related variants. This applies to two fields, *idioms* and *meaning*. In the seventh and last stage the actual lexicographical relevance of individual citations is estimated, i.e. whether the citation should be included in the completed entry or not. The first three stages set themselves apart by the fact that each citation can basically be analysed by itself, without recourse to other citations. In the subsequent steps the analysis is, to a large extent, based on comparizon between citations.

[5] Cf. Jörgen Pind, this volume.

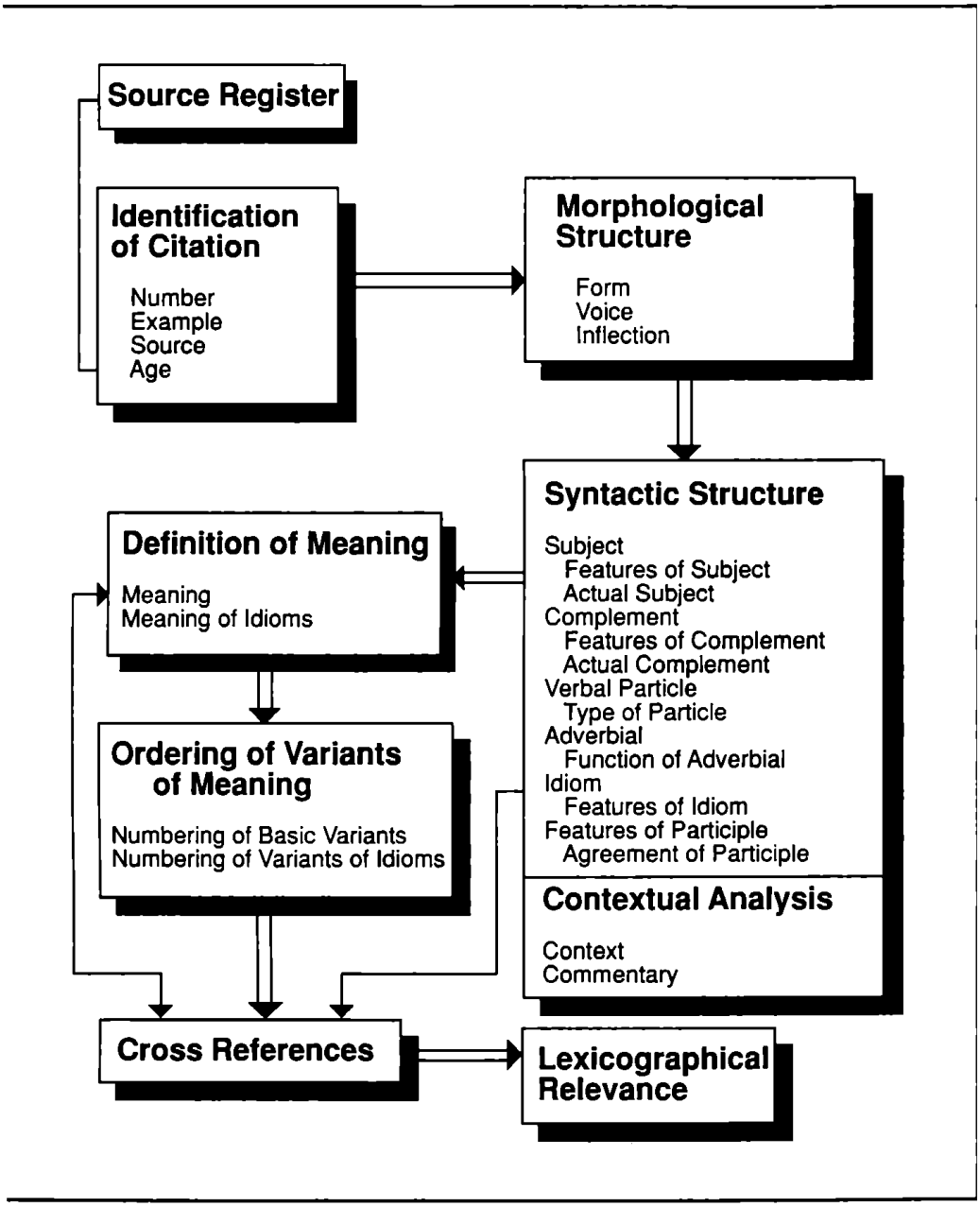


Figure 5: Process of Analysis

The data for the first step of the analysis, and the normalized form of the verb in the second step, is not keyed directly into the database, but entered by using an editing program and then transferred mechanically to the database. After that the analysis is done on two separate screens, the *structural analysis* on one, and the *semantic analysis* on another.

3.2 The Different Fields in the Analysis

A few comments on the information included in different fields. (I will return to the most important fields in connection with the structure of the database later.)

Form. The actual form of the verb in the citation is given with normalized spelling.

Age. Each century is divided into three parts (e.g. early, middle & late 16th C: *f16*, *m16*, *s16*). In case of uncertainty in dating, longer periods can be given (e.g. *ms16*, *18*, *s19f20*). Most of the dating is done mechanically by running a program which contains the short forms of sources used in the database, combined with the age of each source.

Voice. Three structural features of importance in the organizing of the entry are distinguished. These are the *active* and *middle voice*, and also *non-structural participles.*

Inflection. Marking of morphological categories, i.e. *active/passive*, *mood*, *tense*, *number*, and *person.*

Subject. Description of the part of speech in the subject position. *Personal* and *impersonal* subjects are distinguished, and the former are classified according to the case in which they appear. (Icelandic has oblique subjects, as well as nominative ones.)

Classifying features of subject. Used for the distinction *+/- alive.* This field is also used to specify the instances of *complement clauses* or *infinitive constructions* in subject position.

Content of subject. The actual word in subject position is named.

Complement. Description of the syntactic item in complement position.

Classifying features of complement. *Clausal complements* or *infinitive constructions* in object position are specified.

Content of complement. The actual word in complement position is named.

Verbal particles. Adverbs or prepositional phrases forming phrasal verbs.

Type of verbal particle. The particles are divided into *adverbs* and *prepositional phrases.*

Content of prepositional phrase. The actual prepositional phrase following the verb is named (if the verb is considered to be a phrasal verb).

Adverbial. Adverbials not of a phrasal nature.

Function of adverbial. Distinction of adverbials of place, time, etc.

Idioms. Idioms in which the verb in question occurs.

Features of idioms. Proverbs are specially marked.

Features of participles. Non-structural participles are marked as having verbal, adjectival, or adverbial characteristics.

Agreement of participle. This field contains the word with which the participle is in agreement.

Context. Citations of *poetry* or *proverbs* or containing *synonyms* are marked for these features.

Commentary. Commentary on language use or other matters of grammatical or general interest (e.g. explanations, folklore, etc.).

Ordering of variants of meaning. Variants of meaning within the same structural heading are numbered.

4 Organization of the Dictionary Entry

We now come to the *dictionary text* and *the organization of the entries* themselves. In the following part of this paper I will refer to the sample text in the appendix.

4.1 The Content of the Entry

We can begin by sketching the material which must be contained in the text of each entry, independent of the organization of the entry as such. These are the most important points:

- A description of meaning, based on an ordered list of variants of meaning.
- A description of the syntactic structure of the verb.
- A description of inflection.
- Occurrence, i.e. attested examples of (and commentary on) the use of individual forms.
- Age, i.e. dating of individual variants (and forms).

4.2 Principles of Organization

Bearing in mind the wish to organize the material in a user-friendly manner, we feel the following principles should be adhered to:

- The description should give a clear picture of the most important characteristic features of each word.

- The reader should be able to find what he is searching for without too much effort.
- If related information occurs in different places in the text, cross-references must be provided.
- The organization of the entry should make use of the average reader's knowledge of grammatical descriptions and prior experience of dictionaries.
- Obscure and complicated systems of symbols should be avoided.

The purpose of the third principle is providing a bridge between the first two. In case of the last two principles, a great importance is placed on the fact that all headings and classificatory features should appeal directly to a common knowledge of grammar and be in accordance with previous experience of using dictionaries which one has to assume the average reader to have. Going beyond surface structure in analysing the examples is quite risky, and can easily make retrieval more difficult for the average reader. (See Zöfgen 1985 and Sinclair 1987.)

4.3 Syntactic vs. Semantic Organization of Entry

In a comprehensive dictionary of the kind we are talking about, both semantic and syntactic characteristics of the material being described must be accounted for. The prerequisite for doing this successfully is making the relation of meaning and form clearly apparent in the text. Traditionally the dictionary entries are semantically organized.[6] Only rarely are the entries organized according to syntactic factors, but an interesting example of this kind, with a commentary, can be found in Bergenholtz and Mugdan 1984, and in Mugdan 1985. See also Carstensen 1985.

4.4 Superordination of Syntactic Structure

In this present project, we have chosen to use syntactic structure as the framework on which the description is based, consequently placing the semantic description in a subordinate position. This kind of organization is fully in accordance with the most natural way of step-by-step analysis, where the semantic analysis does appear right at the end and is based on the foregoing syntactic analysis. There are, however, other reasons for choosing this type of organization of the entry:

- A general commentary on syntactic structure in the introduction or head of each entry is insufficient.
- Syntactic features are shared by groups of verbs, i.e. the verbs can be classified according to the structures in which they appear.

[6]See Atkins, Kegl, and Levin 1988 for a treatment of the relation of meaning and form in the organizing of verbal entries in in dictionaries. See also Jón Hilmar Jónsson 1985 and 1988:138–140 for information on these factors in Icelandic dictionaries.

- A structurally based framework makes searching for individual constructions or occurrances easier.
- Semantic descriptions are to a large extent based on syntactic structure.
- The semantic descriptions are made more explicit when based on given syntactic structures.
- Semantic descriptions tend to be subjective to a greater degree than syntactic ones. They are, consequently, more liable to be idiosyncratic or inconsequent.

4.5 The Hierarchy of Organization

Superordinating the syntactic features to the semantic ones does not, however, by itself complete the decision-making on how to organize the material. One still has to decide which elements are to be included in the description and how these are to be ordered. There are, all told, nine fields which serve to define the organization of the text of the entry, giving the following hierarchy:

The Hierarchy of Classifying Fields:

1. Lexicographical Relevance
2. Voice
3. Subject
4. Complement
5. Verbal Particles
6. Variants of Meaning
7. Idioms
8. Variants of Idioms
9. Age

4.5.1 Syntactic Fields

Let us take a closer look at the superordinate syntactic fields:

Within the field marked VOICE the order in which the features are listed is *active*, *middle voice*, *perf. participle*, *pres. participle*. In the field marked SUBJECT the *personal subject* (in the order *nominative*, *accusative*, *dative*, *genitive*) precedes the *impersonal subject*.

The field marked COMPLEMENT mainly provides information on transitivity and case-assignment. In this field there is a great amount of variation which produces a corresponding profusion in the structure of the text of the entry. The same applies to the field marked VERBAL PARTICLES:

O = Object
a = Accusative
b = Dative
c = Genitive

1 = -Alive
2 = +Alive
3 = Reflexive

Complement	Particle	Examples
Ø	Ø	gráta
Ø	á	vinna á
Ø	á a1	minna á e-ð
Ø	á a2	ganga á e-n
Ø	á a3	reyna á sig
Ø	á b1	vinna á e-u
Ø	á b2	vinna á e-m
Ø	á b3	sitja á sér
Oa1	Ø	lesa e-ð
Oa1	til	tína e-ð til
Oa1	til c3	taka e-ð til sín
Oa1	við a1	bera e-ð við e-ð
Oa1	við a2	tengja e-ð við e-n
Oa1	við a3	skilja e-ð við sig
Oa1 Ob1	Ø	svipta e-ð e-u
Oa2	Ø	berja e-n
Oa2 Ob1	Ø	ræna e-n e-u
Oa3	Ø	raka sig
Ob1	Ø	kasta e-u
Ob2	Ø	hrósa e-m
Ob2 Oa1	Ø	gefa e-m e-ð
Ob3	Ø	skemmta sér
Oc1	Ø	iðrast e-rs
Oc2	Ø	gæta e-s
Oc3	Ø	skammast sín

Table 1: Structural Types (Complement + Particle)

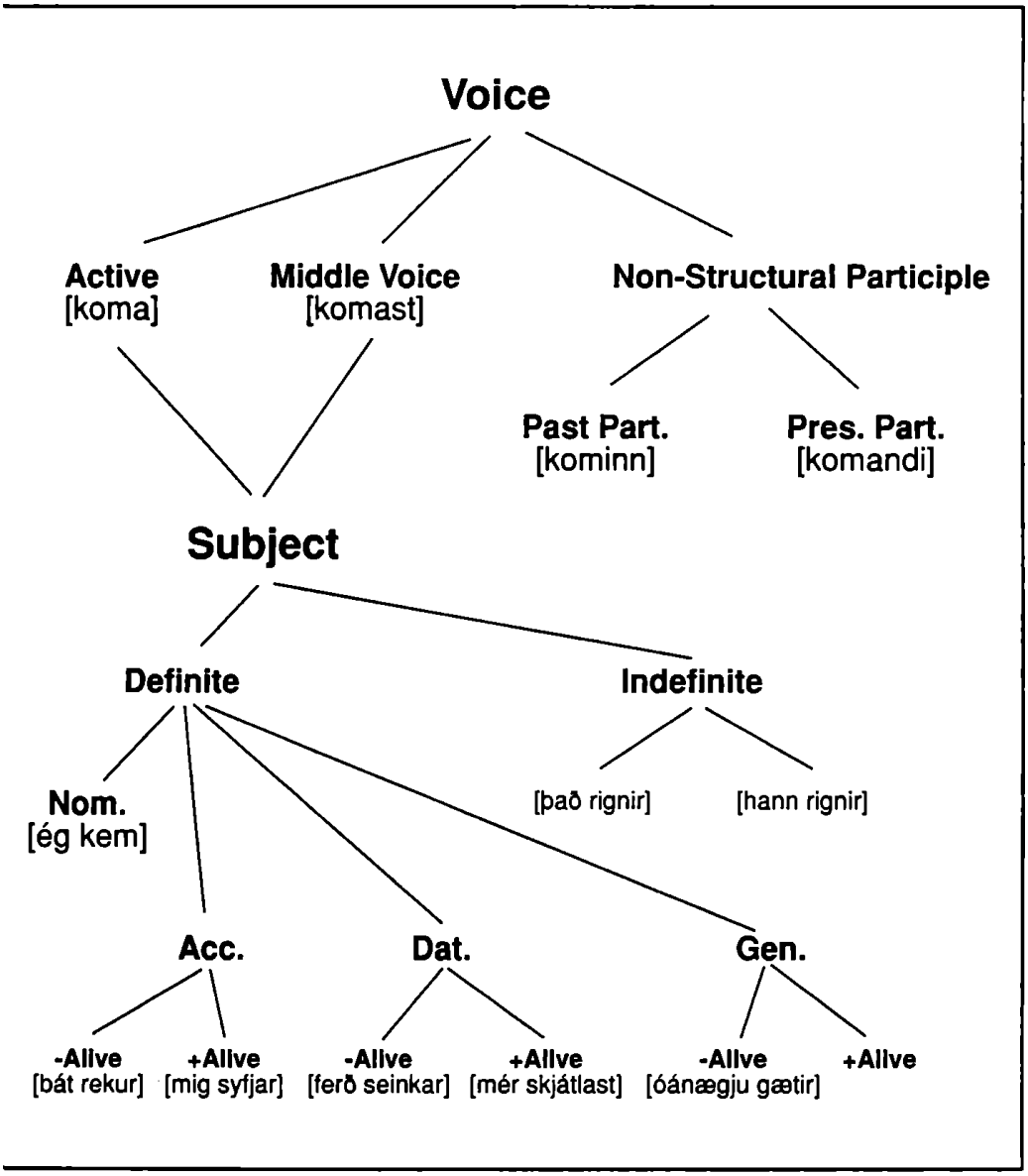


Figure 6: Basic Structuring of Dictionary Entry

4.5.2 Semantic Fields

After the text has been organized according to these four syntactic fields, the semantic descriptions follow. The internal ordering of semantic variants is specified in the field VARIANTS OF MEANING. This field specifies numbers of variants, which are directly followed by definitions of meaning, placed in the field MEANING. The definitions are always construed so that they correspond to the structural heading under which they appear. Square brackets are used to supply additional details on meaning not indicated by the structural heading or to specify variants of meaning with limited reference. The variants of meaning are

always numbered independently within each part of the superordinate syntactic structure.

4.5.3 Idioms

After the fields dealing with the meaning are completed, we come to IDIOMS. When dealing with idioms it is necessary to start by deciding whether the idiom should be placed under one of the numbered variants of meaning described above or placed directly under the structural heading. If more than one idiom occurs within one numbered variant of meaning, these are placed in alphabetical order.

4.5.4 Meaning of Idioms

After entering the idioms themselves, these are ordered by the use of the field VARIANTS OF IDIOMS, where different variants of meaning within each idiom are ordered. The next step is giving definitions of the idioms. A large number of idioms which are placed under basic definitions of meaning is given without separate definition of meaning of the idiom itself. The last part of the classification is sorting the citations within each category according to age. The citations are listed chronologically under each heading.

4.6 Heads

In most dictionaries the most characteristic features of each entry are given as an introduction after the headword itself, before coming to the actual description. In Icelandic dictionaries it is customary to devote this space to information on inflections, but it is also quite common to give some syntactic information at the outset.

The introductory description, which I will call the *verbal head*, is placed within a square frame. The main feature of the verbal head is a description of inflection. Firstly, information on the paradigm of the verb is given, i.e. strong vs. weak verbs, and stem allomorphy. Secondly, deviations in inflections occurring in the citations are given, at the editor's discretion.

In addition to this morphological information, the head also contains the number of citations for the verb in question, both the complete number in the database, and the actual number used as examples in the edited entry.

4.7 Cross-References in the Margins

The organization of the entry as such is not difficult to grasp and make use of, but searching for material not inherently classifiable under the fields described above does, of course, cause some problems. Therefore, definite measures have been taken to make this kind of information easily visible in the text. This has been done by placing notes in the margins, providing additional information. The use of the keys can be divided into three parts:

1. Comments on specific context or use.
    - Language use or register.
    - Proverb.
    - Grammar.
    - Folklore.
    - Explanation.
    - Synonym.
2. Interesting inflectional variants, giving period of occurrance.
3. Cross-references
    - of variants of idioms (lower-case letters).
    - of similarity of meaning (upper-case letters).

The cross-references between related variants are made by marking the variants with the same letter. An arrow, which can take any one of three forms, is placed next to this letter. An arrow pointing downwards refers the reader to one or more related variants occurring later in the description of the verb. An arrow pointing upwards refers to variants occurring above. The third type of arrow points upwards and downwards at once, indicating multiple occurrences of variants.

5 The Project in Progress

As I said in the beginning, the motives behind this project are both of a lexicological and lexicographical nature. In reality, though, the actual text of the proposed dictionary has gradually come to play a more and more central role as the work on the actual format of the entries has been developed and a standard for a historical dictionary of Icelandic verbs has been decided upon.

Let us conclude by taking a look at the main features of the type of entry we have developed, paying special attention to the points which differ from the traditional approach in historical dictionaries:

- The description of each lexeme is directly based on authentic examples.
- Full and emphatic use is made of the unique characteristics of the verbs as a wordclass.
- The organization of entries is fully standardized (to fit a given structure).
- There are multiple ways of accessing the information contained in the text.

Approximately 200 verbs have been analysed so far, to different degrees of completion. Among these are most of the really complex (and large scale) verbs.[7] According to our plan, the next stage of the project is producing a trial leaflet containing the entries of a few verbs (ca. 10) and a detailed description of our working model and the principles on which the work is based. The idea is to produce this next year. The long-term plan is producing and publishing a complete dictionary of Icelandic verbs in the next ten to fifteen years.

Concurrently with the editing of the historical dictionary, we hope to be able to make the information contained in the database directly available for research on the lexicologic and lexicographic features registered in the process of analysis. This is of great importance in the editing itself, i.e. in maintaining cohesion between and comparizon of individual entries. A fascinating by-product, however, is a multitude of new possibilities of research in the field of lexicology.

References

Atkins, Beryl T., Judy Kegl, and Beth Levin. 1988. Anatomy of a Verb Entry: from Linguistic Theory to Lexicographic Practice. *International Journal of Lexicography* 1:84–126. Oxford University Press, Oxford.

Bergenholtz, Henning, and Joachim Mugdan. 1984. Grammatik im Wörterbuch: von *Ja* bis *Jux*. *Germanistische Linguistik* 3–6:47–102. Georg Olms Verlag, Hildesheim.

Björn Þ. Svavarsson and Jörgen Pind. 1989. Database Systems for Lexicographic Work. (This volume.)

Carstensen, Broder. 1985. Von *Ja* bis *Jux* ohne *Tollerei*: Bergenholtz/Mugdans grammatisches Wörterbuch. H. Bergenholtz and J. Mugdan [eds.]. *Lexikographie und Grammatik. Akten des Essener Kolloquiums zur Grammatik im Wörterbuch 28.–30.6. 1984*:175–186. Max Niemeyer Verlag, Tübingen.

Courtney, Rosemary. 1983. *Longman Dictionary of Phrasal Verbs*. Longman Group, Harlow, Essex.

Fox, Gwyneth. 1987. The Case for Examples. J. Sinclair [ed.]. *Looking up. An Account of the* COBUILD *Project*:137–149. Collins, London.

Guðrún Kvaran. 1988. Sérsöfn Orðabókar Háskólans. *Orð og tunga* 1:51–64.

Gunnlaugur Ingólfsson. 1988. Söfnun Orðabókar Háskólans úr mæltu máli. *Orð og tunga* 1:65–72.

Helbig, Gerhard, and Wolfgang Schenkel. 1969. *Wörterbuch zur Valenz und Distribution deutscher Verben*. Bibliographisches Institut, Leipzig.

Jón Hilmar Jónsson. 1985. Íslensk orðabók handa skólum og almenningi. Ritstjóri: Árni Böðvarsson. Önnur útgáfa, aukin og bætt. Reykjavík, 1983. [Review.] *Íslenskt mál* 7:188–207.

Jón Hilmar Jónsson. 1988. Sagnorðagreining Orðabókar Háskólans. *Orð og tunga* 1: 123–174.

[7]The most voluminous verbs in The Written Language Archive are *taka* (4,099 citations), *ganga* (3,550 citations), *koma* (3,373 citatations), *leggja* (3,281 citations), *bera* (2,675 citations), and *standa* (2,418).

Jörgen Pind. 1986. The computer meets the historical dictionary. *Konferensdokumentation NordDATA 86.* Band 2. 83–88. Stockholm.

Jörgen Pind. 1989. Computers, Typesetting, and Lexicography. (This volume.)

Mugdan, Joachim. 1985. Pläne für ein grammatisches Wörterbuch. H. Bergenholtz and J. Mugdan [eds.]. *Lexikographie und Grammatik. Akten des Essener Kolloquiums zur Grammatik im Wörterbuch 28.–30.6. 1984*:187–224. Max Niemeyer Verlag, Tübingen.

Sinclair, John. 1987. Grammar in the Dictionary. J. Sinclair [ed.]. *Looking up. An Account of the* COBUILD *Project*:104–115. Collins, London.

Weiner, S.C. 1989. Editing the OED in the Electronic Age. *Dictionaries in the Electronic Age*:23–31. Fifth Annual Conference of the UW Centre for the New Oxford English Dictionary. Proceedings of the Conference. Oxford University Press, Oxford.

Zöfgen, Ekkehard. 1985. Definitionswörterbuch kontra Valenzwörterbuch. Zur lexikographischen Darstellung der Verbsyntax aus pragmatischer Sicht. H. Bergenholtz and J. Mugdan [eds.]. *Lexikographie und Grammatik. Akten des Essener Kolloquiums zur Grammatik im Wörterbuch 28.–30.6. 1984*:130–158. Max Niemeyer Verlag, Tübingen.

Institute of Lexicography
University of Iceland
101 Reykjavík
Iceland

Appendix

On the following pages is a sample entry from the *Standardized Dictionary of Icelandic Verbs.*

berja v., barði – barinn (barður);
655 dæmi alls;
dæmi í texta: 428

BERJA: ■ **1.** ***slá, veita högg*** s17 *Hesturinn barde og hesturinn slo. (ÍFk. IV, 184); „Ekki mátti berja kringum sig með svipu eða öðru keyri. Það fældi frá manni allar góðar verur." (Tms. (Árn.)); „Lundi kemur á skot. Þá slær hann [ɔ: veiðimaður] háfnum á eftir honum. Þetta er kallað að berja, lemja eða slá." (Tms. (Vm.eyjar)); **berja hæl og hnakka** ***brjótast ákaft um*** f18 Þott hun [ɔ: veröldin] velltest umm / og berie Hæl og Hnacka. (VídPost. I, 199); **berja í nestið** ***vera í andarslitrunum, vera að deyða kominn*** (Tms. (Suðvesturl.)); „Talað var um að berja nestið, en oftar var þó talað um að berja í nestið. Ég hugsa að þetta hafi verið dregið af því að harðfiskur var mikið notaður þegar farið var í langferð. Við skildum þetta svo að hér væri verið að berja í síðustu langferðina." (Tms. (Grindavík)). ■ **2.** ***mylja [það til dæmis] á túni]*** s18 er og [sleggja] naudsynleg til ad beria aa tunum med. (OOlUrt., 9); s18f19 *adallinn kvasast ef hann fer / út á tún ad berja. (MStLjóðm., 102); ms19 Gudda ráðskona var úti á túni að berja. (BGröndRit. II, 44); s19 Þar sem túnin eru sléttari, aka menn venjulega mykjunni á máli hverju út á túnin (til þess betra verði að berja segja menn). (Skuld. 1880 nr. 102, 19); s19 er það víða siðr að klína blautri mykjunni, undir eins og hún kemur úr fjósinu, út yfir þúfnakollana, og láta hana liggja þannig, þangað til farið er að berja. (Skuld. 1880 nr. 102, 18); m20 haltu áfram að berja og vertu ekki að glápa úti loftið, sagði eldri bróðirinn, þeir stóðu á túninu og börðu. (HKLSjfólk., 343); s20 Svo var á vorin unnið á með kláru, barið og strokið jafnt úr mylsnunni, sem vannst vel. (GÞLundStarfsh., 35); **berja á túni** m19 berja á túni glebas subigere. (Lbs307 4to.); m20 var farið að berja á túni, þ.e.: taðið var mulið með klárum, og mylsnan breidd yfir túnið. (Breiðdæla., 89); **berja á velli** 18 *Alla vorsins úti stund, / er eg á velli ad berja. (Gamkv., 73); m18 Eg Ber á Velle (tune) occo. (JÁLbs224 4to., 93); m20 Byrjað var að berja á velli vorið 1871. (JSigSig., 73). ■ **3.** ***knýja á ([dyr] með höggum)*** s16 Eg sef / enn mitt Hiarta vaker / Þad er mijns Vinar Raust / sem þar ber. (Ljóð. 5, 2 (GÞ)); f17 med þui ad þu sialfur aminner mig / þa vil eg leita og bidia og beria. (MollMed. K, 7r); m19 geingu þeir að dyrum og börðu. (JÁÞj. II, 181); m19 heyrði hún að það var barið. (JÁÞj. I, 281); m19 sagðist [hann] ekki hafa þorað að fara til dyranna, en víst hefði einhver barið. (JÁÞj. II, 476); m19 Þegar þeir eru búnir að berja kemur nokkuð roskinn maður til dyra. (JÁÞj2. V, 320); m20 *Það var eitt kvöld að mér heyrðist hálfvegis barið. (JHelgLand., 25); **berja að dyrum** m19 Þar barði eg að dyrum. (JÁÞj. II, 38); fm20 Ég var búinn að klappa á hurðina fyrir æðitíma. Nú vildi eg reyna hitt, að berja að dyrum. (GFrRit. II, 250); s20 Það er ósvinna að berja að dyrum í Færeyjum. (JÁrnVeturnóttak., 28); **berja á dyrum** m19 Einar barði á dyrum. (JÁÞj2. IV, 156); m19 Hann ber á dyrum. (JÁÞj2. IV, 425); m19 Um nóttina er barið á dyrum og er sagt að sýslumaður sé þar kominn. (JÁÞj2. III, 502); **berja til dyra** s19f20 óðara en varði var hann kominn að garðshliðinu og barði til dyra. (MJSherl. I, 290). ■ **4.** ***róa (kröftuglega) á móti veðri/vindi*** s19 er veðrið lítið lægði, börðu þeir úr eyjunni. (FrEggFylg. II, 116); f20 Ætlaði ég þá að láta karlana berja og tókum við saman seglin. (SvbEgFerð. II, 464); m20 Varð hann þó að 'berja' klukkutímum saman ... þar til að hann komst 'undir vind', þ.e. þangað til að hægt var að nota seglin. (JóhBárðÁr., 92); m20 svo að ekkert var annað að gera en að fara að berja í land aftur. (ÞJForm., 80); m20 Hann [ɔ: formaðurinn] taldi ómögulegt að berja til baka á móti veðrinu. (ÁrsrÍsf. 1959, 156); m20 við höfðum barið hvíldarlaust utan úr Bolungarvík allan daginn. (ÁrsrÍsf. 1959, 160); m20 barði ég á bátnum ... inn á Þingeyri. (ÁrsrÍsf. 1961, 189); m20 Þótt sjór væri tipplóttur ... fóru flest róðraskip að berja fram í djúp. (Suðurn., 319); ms20 Svo börðu feðgarnir af alefli gegn veðurofsanum í bátskelinni. (KristmGMorg.,

A↓ þjóðtrú
málv.
samh.
a↓
b↓
skýring
B↓
þjóðh.
þjóðh.
skýring
þjóðh.
C↓
c↓
þjóðh.
c↓
c↑
D↓

180); s20 Nú berjum við saman á móti veðrinu heim. (JóhHSkált. 19); s20 Það gat tekið sex klukkustundir að berja inn fjörðinn. (JóhHSkál. 37); **berja að landi** m19 hann ... fór svo að berja að landi, gekk þá mjög lítið. (Ísl. 3, 119); **berja til lands** m20 Tóku þeir nú að berja til lands, en sáu skjótt, að hríðrak. (EGuðmNýttSagn., 63); m20 var þá orðið svo hvasst, að þeim þótti ekki ráðlegt að sleppa landinu, felldu því seglið og börðu til lands. (JHermBsjóm. I, 47); m20 Engin leið var að berja til lands með árum. (ÁrsrÍsf. 1962, 11); **berja til lendingar** m20 hefur mörgum sjómanninum orðið erfið raun að berja til lendingar. (JHermBsjóm. I, 17); m20 Gengu þeir síðan allir á skip með Tómasi og börðu til lendingar í Skeri um kvöldið. (JHermBsjóm. II, 80). ■ **5.** ***slá [gras við óhagstæð skilyrði]*** E‖ f18 Engjar eru öngvar, nema hvað barið er úr forræða flóum. (Jarðab. VII, 229); f18 Engjar öngvar, nema hvað barið er á fastalandi í peningabeit. (Jarðab. V, 46); s19 Sláttumenn voru að reyna að berja úr hagbeitarmóum ... hingað og þangað. (Ísaf. 1886, 146). ■ **6.** ***[vél:] hafa höggkenndan gang*** m20 Vélin í gamla Fordinum var farin að berja. (WilsonGrá., 144).

berja á: ■ **1.** ***mylja [það til áburðar á túni]*** B‖ s17f18 að berja á og ausa tún. (Ann. V, 166); f20 var barið á með verkfæri, er kallað var klára. (Blanda. IV, 213). ■ **2.** ***knýja á [dyr]*** f17 beried a / og fyrer ydur skal verda C‖ vpploked. (SpangenbCat. Ee, Iv).

berja á e–ð: ***slá á e–ð, láta högg dynja á e–u*** m16 bardi hann a sitt briost og sagdi A‖ Gud vertu mier syndugum liknsamr. (Lúk. 18, 13 (OG)); s16 Saunguararner þeir ganga fyrst framm vndan / þar næst Spilmennerner medal Meyanna / sem a Bumburnar beria. (Sálm. 68, 26 (GÞ)); **það er eins og að berja á bjargið** ***[e–að] er algerlega árangurslaust*** fm20 En það d‖ fór allt á eina leið og var eins og að berja á bjargið — steinhljóð og dauðaþögn. (GFrRit. VI, 218).

berja á e–m: ***lumbra á e–m, jafna um e–n, veita e–m harða ráðningu*** m19 Sendir nú A‖ Justinianus keisari Belisarius til Ítalíu þess erindis að berja á Austgotum. (MelMið., 7); m19 Hann bardi á Filisteum allt til Gesa. (2Kong. 18, 8 (1841)); s19 hann hafði meðal annars barið á manni og leikið hann illa. (Skírn. 1870, 71); s19 og ekki tók hann nærri sér á Hafnarárum sínum, þegar hann var við öl, að berja á Danskinum, sem löndum þá var sumum títt. (Sunnf. V, 20); f20 og gerðu ýmist, að þær greiddu óvinunum atlögur eða þá börðu á mönnum sínum, er illa gengu fram. (TacGerm., 40).

berja í e–ð: ***beina höggi að e–u, láta e–ð verða fyrir höggi*** **berja í borðið** ***mótmæla kröftuglega*** fm20 Við getum nú haft völdin samt, ef við höldum hópinn og berjum í þorðið. (GFrRit. III, 206); **berja í bresti (e–rs)** ***afsaka e. breiða yfir ágalla (e–rs)*** m19 og er ekki til neins að berja í þá bresti. (Þjóð. 1853, 156); fm20 Og þótt Arnljótur Ólafsson berji í bresti þjóðkirkjunnar með Kaupahéðni og Jóni Helgasyni. (GFrRit. VI, 89); **berja í brestina** ***a. slá í brestina til að dylja þá*** s19f20 og hái rennismiðurinn ... fór nú að berja í brestina og sýna hvernig fara ætti með kvistina, umflýja sprungur og jafna ávalann. (MJSherl. III, 55); s20 Fljótlega ... fóru fjárhúsin þar að segja af sér sakir fúa í viðum. Sveinn reyndi auðvitað að berja í brestina og tefja fyrir algeru hruni. (RGSnæSkáld., 61); ***b. afsaka e. breiða yfir ágalla e–rs*** s18f19 þá gæti eg kannske barið eitthvað í brestina. (GVíðBr., 101); fm19 Rask bardi aptur í Brestina og forsvaradi þad sem ecki lét sig forsvara. (SafnF. XIII, 197); m19 Englendingar hafa með öllu móti reynt að berja í brestina fyrir þeim [þ: Tyrkjum]. (Skírn. 1864, 91); ms19 en hann er með okkur svo, að hann er alltaf að berja í brestina. (JSBréf., 178); s19 Herra Tryggi vill ... berja í brestina að því, er Gránufjelagið snertir. (Þjóð. 40, 101); f20 Væntum vér, að góðir menn taki viljann fyrir verkið og berji í brestina. (ÁrsrÍsf. 1966, 50 (1923)); f20 að láta mistökin ekki endurtakast, heldur reyna að berja í brestina. (ThArnHest., 340); m20 Þú tekur svari hans og reynir að berja í brestina. (GuðrLDal. I, 145); **berja í brestina á e–u** ***afsaka e. breiða yfir ágalla e–(r)s*** m19 Hinn virðuglegi konúngkjörni varaþíngmaður andlegu stéttarinnar hefir haft mikla fyrirhöfn fyrir að berja í brestina á

konúngsfrumvarpinu. (Alþ. 1849, 863); s19 Er mikið drengilegra að játa það og kannast við það, enn að vera að berja í brestina á gjörðum sínum. (Fróði. 1883, 40); **berja í vænginn** *hafa afsakanir í frammi* f↓ s19 Í sama streng tóku þeir Jón ... og Benedikt ... en Halldór ... barði heldur í vænginn. (Ísaf. 1885, 115); **það er eins og að berja í stein(inn)** *[e-að] er algerlega árangurslaust* d↑ s19 en það var eins og barið væri í stein, og sögðu að enginn hefði þekkt það illa frá því góða. (EirÓl., 226); s19 ekkert var sparað til að frelsa hann, ... en það var sama sem að berja í steininn. (2Ið. I, 54).

berja ofan af fyrir e–u: *vinna (hörðum höndum) fyrir e–u* F↓ ms19 Enn er eg að reyna að berja ofan af sjálfur fyrir mínu vesæla lífi. (BóluHj. V, 239).

berja ofan af fyrir e–m/sér: *vinna (hörðum höndum) fyrir lífsframfæri e–s/sínu* F↑ s19 móðirin barði ofan af fyrir hinu [ɔ: barninu] um sumarið. (FrEggFylg. I, 214); s19 ráðalaus var hann með öllu að berja ofan af fyrir sér. (FrEggFylg. II, 294); s19f20 fremur til að berja ofan af fyrir sér en til bóknáms. (SGStBR. IV, 203).

berja um e–ð: berja um bresti(na) *a. slá í bresti(na) til að dylja þá* e↑ skýring ms18 at berja umm bresti, propriè fracturas in ferro, vel aliqvo alio duro metallo, malleo complanare. (JÓGrvOb.); *b. afsaka e. breiða yfir ágalla e–rs* skýring ms18 at berja umm bresti, ... nævos alicujus emendare (JÓGrvOb.).

berja upp: *knýja dyra* C↕ m20 barði upp á Bessastöðum í vökulok. (JBjörnJómf., 54).

berja upp á: *knýja dyra* C↑ m19 munud þér, sem fyrir utan standid, taka ad berja uppá og segja: lúk þú upp fyrir oss, Herra. (Lúk. 13, 25 (1841)); s19 Maður ber upp á, ef ekki með kröfum, þá samt alténd með eptirvæntingu, og hvað sér maður þá fyrst, þegar dyrnar ljúkast upp? (Þjóð. 30, 61); s19 kemur John Brown inn þegjandi og ber eigi upp á. (Ísaf. 1894, 83); s20 Þegar tveir ribbaldar börðu upp á hjá okkur. (Vísir. 12/6 1972, 16).

BERJA E–Ð: ■ 1. *slá á e–ð, beina höggum að e–u* A↕ f17 Og þijn ohrein Klæde verda laaten j Vatn / þueigen j skarpre Lwt / baren og sleigen og þuætt. (NicSpeg., 590); 18 og þótti þá sem veðrið berði skipið á allar hliðar. (JÞorkÞjs., 156 (18. öld)); f19 og koddar séu bornir út og bardir vel á sumrum. (SpurnHeil., 48); m19 Þegar þú hefir barid ávextina af þínu olíutré, þá skaltú ei gjøra þar eptirleit. (5Mós. 24, 20 (1841)); m19 *nú veiztu, hvernig hjartað brjóstið ber, / er blóðið logar þar í djúpum sárum. (JHall. I, 134); s19f20 *Þau [ɔ: skipin] börðu löðrið, langt og skammt, / en lending engri náðu samt. (SGStAnd. II, 432); f20 Börðu hásetar þá [ɔ: lifrarpokana] öðru hvoru, og var það sú vörn, að skipið fékk aldrei áfall meðan dreif. (Blanda. V, 97); **berja ána/hylinn** *„reyna að veiða á stöng“* (Tms.); **berja barrið** málv. „Berja barrið = að berja lóminn var algengt og þýddi að barma sér.“ (Tms. (Fljótsdalsh.)); **berja bumbu(r)** *slá á bumbu(r) (í fagnaðarskyni)* g↓ m16 Horn var þeytt, en bumba börð. (Pont. börð m16 I, 68); 18 þá voru barðar bumburnar, / og slegin symfónin. (Ífk. V, 138); s19 bumbur voru barðar, turnklukkum hríngt. (MelNs. II 1, 64); f20 Verksmiðjustofnun þessi hefir fæðst ofboð hljóðalaust og engar bumbur verið barðar við fæðingu hennar. (Fjallk. 1903, 157); **berja bumbur sínar fyrir e–u** *halda e–u ákaft fram* g↑ m20 hann hóf ... að flytja þá kenningu, sem hún [ɔ: kirkjan] hafði sjálf barið bumbur sínar fyrir um nítján alda skeið. (SnæJVörð., 82); **berja e–ð augum** *sjá e–ð, virða e–ð fyrir sér* h↓ m20 Þá biblíu hef ég sjálfur barið augum hér í Dómkirkjunni. (HKLBrekk., 30); m20 munu að vísu nokkrir yðar heyrt hafa getið þess lands, þó að fæstir yðar hafi barið það augum. (HKLGerpla., 484); **berja fótastokk(inn)** *dingla fótunum á hestbaki (til að knýja hestinn áfram)* þjóðh. s19 Eg hefi heyrt frá barnæsku það lastað sem ljótan óvana, að ‘berja fótastokk’. (Bún. 1894, 132); f20 Hættið þeim ljóta vana, að berja ‘fótastokk’ og hnýta hestum í taglið. (Alm. 1901, 91); f20 að telja þann með reiðmönnum ... sem patar, ber fótastokkinn og baðar út öllum öngum. (Skírn. 1915, 191); s20 Við héldum áfram að berja fótastokkinn til að flýta förinni. (GuðmHUndljá., 40); **berja gadd(inn)** *[hestur e. annar fénaður:] stappa í freðna jörð (til að ná e–u til að bíta)* s18 helldr beria [hrossin]

gadd, er menn kalla, edr lemia upp jørdina med hófunum. (LFR. VI, 61); s19 að sjá horað fje ... berja gaddinn. (Ísaf. 1880, 8); s19 ekki eiga hrossin betra. Þau verða að berja gaddinn, allan veturinn. (Dagskrá. 1897, 331); m20 Sýndist munurinn stundum helzt til mikill, sem var á æfi reiðhestanna og hinna, sem máttu berja gaddinn líknarlausir. (KrÞorstBgf. II, 148); **berja hóstann** „Að lemja hóstann og berja hóstann var sagt um rækilegan hósta." (Tms. (N.-Þing.)); **berja lóm(inn)** ***kvarta (sífellt), bera sig illa*** s19 þegar hann var búinn að ... berja lóminn út af sjer og þessari ónæðisstöðu. (Heimd. 1884, 170); s19f20 *Fólkið svalt með sinateygjum, / sífellt barði lóm. (MJLj. I, 108); f20 lómurinn er barinn látlaust. (SGStBR. III, 43); f20 Mér dettur ekki í hug að fara að berja lóminn eða betla fyrir Jón. (Ægir. 1924, 51); f20 að þýzka togaraútgerðin er sem stendur í hinum mesta vanda ... enda ber hún óspart lóminn og hrópar á hjálp. (Ægir. 1932, 283); m20 enda er það eigi siður Íslendinga að berja lóminn framan í erlenda menn. (Grímaný. II, 217); **berja nestið** ***vera í andarslitrunum, vera að dauða kominn*** s19 orðtækið að berja nestið, sem haft er enn í dag um þá, er berjast í andarslitrunum. (SGuðmForngr. I, 50); s19 því eftir útliti og athöfnum manna erlendis sýnist skipaútgerð vor eiga skamt eftir til að berja nestið. (Bjarki. 1898, 22); fm20 eftir þann tíma fer ég að berja nestið. (Rauðsk. IV, 42); m20 Um þann sem lá banaleguna var sagt: já hann er nú að berja nestið auminginn. (HKLBrekk., 66); m20 Ég á gott að vera gamall og eiga ekki eftir nema berja nestið. (EyGuðmPabbi., 254); m20 Í dymbilvikunni flaug það fyrir að gamli maðurinn í Gljúfrum mundi nú vera að berja nestið. (HKLHeimsl. II, 319); m20 Þannig lá hún mánuðum saman og barði nestið, og var því líkast sem hún gæti ekki skilið við. (GJÞjs. I, 68); „Að berja nestið. Ég heyrði þetta oft sagt þegar ég var að alast upp fyrir norðan. Það var algengt að hafa harðfisk í nesti, og oft var það síðasta verk þess sem lagði af stað í ferðalag, t.d. göngur, að berja harðfisk í nesti. Ég held allir hafi skilið orðasambandið út frá þessu." (Tms. (Norðurl.)); **berja nestið** ***vera í andarslitrunum, vera að dauða kominn*** „Austur í Landbroti var alltaf sagt: berja nestið sitt, en ekki berja nestið eða berja (sér) í nestið." (Tms. (V.-Skaft.)). ■ **2.** ***beina e-u harkalega [í e-a stefnu]*** s19f20 * Ið enska gull skal fúna fyrr / en frelsisþrá sé börð á dyr. (SGStAnd. I, 546); f20 Heljarvegurinn og vígða moldin eru uppdiktur tilað berja svolítil dramatísk áhrif inní kvæðið. (ÞórbÞEdda., 112); fm20 Sigrar Sverris voru sem hamarshögg á hamarshögg ofan og börðu þá trú inn í hugskot almennings, að hann væri réttborinn til ríkis. (ÁPálsVíð., 320); m20 Já; hann hafði barið óvissuna og kvíðann úr líkama hennar og sál. (HKLSjfólk., 362). ■ **3.** ***mylja e.*** ***mýkja e-ð með barefli*** f18 Þeir forsorga sinn kvikfjenað mest af beinum, sem þeir með sleggju berja um veturinn, en gefa ei meir af heyi en vel til jórturs. (Jarðab. X, 316); s18 [taðið] vard ei tækt fyrrenn brunned var under því, þá var þad bared. (MKetHest., 33); m19 þegar barin var mykja á velli fannst hnífurinn í einu hlassinu. (JÁÞj2. III, 373); s19 Í öðrum löndum er aldrei tíðkað að berja áburðinn á vorin. (NF. XXX, 73); s19 barinn [ɔ: áburðurinn] á vorum og rutt áburðinum. (Austurl. I, 111 (1874)); s19 Ef tveir berja sama hlassið ... og klárurnar slást saman. (Huld. II, 151); s19 Þeir [ɔ: líkmenn] mættu opt ... berja klaka hálfan dag. (Ísaf. 1879, viðauki, 26); f20 Áburðurinn var barinn með kvíslum og klárum. (FJÞjóðh., 364); m20 svo var mörinn, sem þá var innan í bjórnum, barinn með sleggju á fiskasteininum eða börðusteininum, sem var á hverjum bæ. (Breiðdæla., 84); m20 Þarna var mörinn barinn eins og harður fiskur þangað til hann var orðinn svo mulinn, að líkast var mjöli. (Breiðdæla., 84); m20 Hann stendur með kláruna sína á vellinum og ber taðið í heimsku. (HKLSjfólk., 345); s20 Nú hófst vökuvinnan. Þá voru rakaðar gærur, tálgaður mör eða barinn. (ÁÓlaAld., 52); **berja fisk** m16 ath beria fisk suo sem vid þyrfti á þuj bui. (DI. XIII, 293 (1558)); m17 *Feginn heldur fiskinn vil ég berja. (HPSkv. II, 407); n18 Kvöð var engin nema berja fisk á staðnum. (Jarðab. V, 136); m18

bJ

málv.

bJ málv.

skýring

málv.

bJ

Bj

þjóðh.

þjóðh.

þjóðh.

þjóðh.

þjóðh.

þjóðh.

málfr.

adsciscit Dativum instrumenti et Accusativum objecti ... ut: at beria fiskinn sleggiunni. (JÓGrvOb.); f19 Lýist fiskr ef leingi er barinn. (GJ., 209); m19 var hann látinn berja fisk. (JÁÞj. II, 441); f20 Loks þurfti að berja í nestið góðan harðfisk, baka kökur og pottbrauð, gera ost og sjóða fornkjöt, sem þá var venjulega reykt, hangið kjöt svo feitt sem kostur var á. (Skírn. 1931, 66). ■ **4.** *róa (kröftuglega) á móti e-u* fm19 *eldharðan börðum austan vind, / upp eftir náðum. (SBrLj. I, 103); m19 Þeir eiga ljótan landsynninginn að berja. (Ísl. 2, 69); ms19 *Þungt er útnyrðinga lífs að berja. (GThLj. I, 6); f20 að lemja og berja harðan mótvind allan daginn með árum. (Ægir. 1907, 49). ■ **5.** *slá [gras] (við óhagstæð skilyrði)* s19 Þegar svo líður að slættinum mega bændr leigja verkamenn til að berja af þúfunum in fáu strá. (Skuld. 1880 nr. 125, 255); s20 Ekki munu menn ávallt hafa farið fúsir frá slættinum heima til að berja þúfurnar prestsins. (Múlaþ. 1969, 110).

berja e-ð áfram: berja e-ð blákalt áfram *knýja e-ð fram* m19 Vér erum komnir hér á þíng til þess, að berja hlutina blákalt áfram. (Alþ. 1855, 662).

berja e-ð fram: ■ **1.** *halda e-u (ákaft) fram, staðhæfa e-ð* m19 En ef prestastéttin yfir höfuð að tala er svo fátæk í samanburði við alþýðu, sem sumir eru að berja fram, þá... (Alþ. 1849, 401); f20 En þetta er barið fram án nokkurs samanburðar við önnur lönd og aðrar þjóðir. (Arnf., 129); f20 ég er ekki vanur að berja fram neina bölvaða vitleysu. (MagnStefBréf., 88); **berja e-ð blákalt fram** *a. halda e-u óhikað fram* m19 Eg álít nú að vísu óþarft að svara því marga, sem barið hefur verið fram blákalt á móti nefndarálitinu. (Alþ. 1855, 424); m19 lækningamennirnir berja það fram blákalt, að kláðinn sje innlendur. (Norðri. 1858, 65); m19 Allt fyrir það barði hann lygina blákalt fram. (Þús. I, 341); m19 þegar skynsamir menn berja fram meiningar sínar blákaldar ástæðulítið. (LKrVestl. II 1, 59 (1846)); m19 að berja aðra eins vitleysu blákalt fram. (Ísl. 2, 16); s19 Herra Jón Ólafsson barði það blákalt fram, að spítalagjaldið af síld þeirri, sem síðastliðið ár (1880) aflaðist hjer við land, muni nema 25.000 kr. (Skuld. 1882, 2); s19 sem nú berja það blákalt fram, að kláðinn sé upprættr. (Skuld. 1878, 91); s19 Þau [ɔ: vísindin] berja það fram blákalt að tvisvar tveir séu fjórir hvernig sem við látum. (Sunnf. II, 5); s19 Það dugar ekkert að berja það blákalt fram ... að uppástungur í þessa eða aðrar áttir, sjeu rangar. (Dagskrá. 1897, 226); *b. knýja e-ð fram* m19 hvað órýmilegt það væri sem þeir vildu berja fram blákallt. (Snp. I, 74); m19 þeir vilja berja það fram blákalt með lögum. (Norðurf. II, 71); m19 það getur opt verið, að menn vilji hleypa breytíngaratkvæðum til umræðu, og berja þau fram bláköld. (TíðÞjf., 73). ■ **2.** *ná e-u fram með hörku* m19 Eg ætla samt ekki að vera að berja fram þessa uppástúngu okkar, ef þíngið sér eitthvert annað ráð betra. (Alþ. 1855, 907); s19 Hefði jeg ... viljað berja fram, að ummæli mín í fundarboðinu skyldu gilda. (Ísaf. 1888, 210).

berja e-ð inn í e-n: *beita hörku til að láta e-n læra e-ð* f20 þessari borg er lýst í landafræði minni (sem er sú bezta landafræði á íslenzku, eins og eg hefi sagt, því það dugir ekki annað en berja það inn í þá). (Arnf., 134); f20 það sem hann kunni, og það var mikið, barði hann inn í okkur. (Skírn. 1923, 72).

berja e-ð í e-ð: berja augun í e-ð *verða starsýnt á e-ð* m19 Bretar börðu helzt augun í liðsending Napóleons. (Ísl. 1, 101).

berja e-ð í gegn: *knýja e-ð fram* m20 sem ráðherrann hafði á sínum tíma lofað þessu plássi og síðan barið í gegn á alþíngi. (HKLSj-fólk., 454).

berja e-ð niður: *kveða e-ð niður með hörku* fm19 að ef slíkar meiningar ekki eru barðar niður hjá almúga. (BThLj. II, 193); m19 þóktust sumir finna reykjarlykt, en aðrir börðu það niður. (JÁÞj. II, 275); **berja e-ð blákalt niður** m20 en allt tal hennar um það barði karl blákalt niður. (ÞTEyfs. I, 35).

berja e-ð saman: ■ **1.** *slá á e-ð svo að það þjappast saman* f20 Sé steypan vel barin saman, getur steinninn staðið án þess að skaddast. (Bún. 1903, 297); fm20 Þegar fer að hækka í tunnunni, er gott að berja kjötið saman með sleggju. (JSigMatr., 64). ■ **2.** *smíða e-ð (af vanefnum)* m19 Þegar einhvur smíðisgripur

er barinn saman ad handa hófi. (BúnSuð. I, 142); ms20 að kunnáttulitlir menn væru án eftirlits látnir berja saman fljótandi líkkistur. (GGSkút. II, 93). ■ **3.** ***semja e-ð (með erfiðismunum)*** s18 ad samanberia fa-einar Sorgarvisur i minningu þeirra ... Hiona. (OOlDraum. A, 2r); ms19 *jeg hef / drukkið í dag, svo alt snýst ótt í hring / á meðan þetta ber jeg saman stef. (GThLj. I, 203); m20 sem ég var búinn að berja saman hugðnæman stúf [ɔ: ræðustúf] rennur lestin inn á brautarstöðina. (ThVilhjGerv., 30). I‖

berja e-ð upp: ■ **1.** ***losa um [klakahrönn e. snjóskafl]*** f20 Fjara er borin í hús, 'barðir upp móðarnir' og fluttir í hús. (SafnF. III, 118); m20 Í hæstu flóðhrönn var hann að berja upp freðinn þarakamp fyrir sauði sína. (HéraðsBgf. II, 160). ■ **2.** ***safna [fé] með harðneskju*** m20 Hansen þessi ... hafði verið á snöpum úti um lönd að berja upp fé til slíkrar útgerðar. (ÁJakKast., 159); m20 Það er alltaf hægt að berja upp peninga. (IGÞorstLand., 179). A‖ G‖

berja e-ð úr e-m: ***kveða [tilt. skoðun e. hátterni e-s] niður*** m19 Bóndi var einarður maður, og ber þetta úr fólki. (JÁÞj. I, 391); s19 Það er svo eðlilegt og sjálfsagt, að ótrúlegt er, að það verði nokkurn tíma úr þjóðinni barið. (Ísaf. 1887, 143). G‖

berja e-ð út: ***fletja út [deig]*** s19 síðan skal berja það [ɔ: deigið] út með kefli, og brjóta það oft saman og berja það út í hvert skifti. (EJónssKvenn., 185). A‖

BERJA E-N: ***slá e-n, ráðast að e-m með höggum/barsmíð*** m16 bordu hann med hnefum / enn adrer gafu pustra i hans andlit. (Matt. 26, 67 (OG)); m16 gripinn / bundinn oc bardur / spyttur oc spieadur / eirnin pustradur. (CorvPass. A, IIIv); m16 erum klædfaer og verdum hnefum bardir. (1Kor. 4, 11 (OG)); m16 framselldi Iesum suipum bardan. (Mark. 15, 15 (OG)); s16 Þa tooku nockrer ad spijta a hann, i hanns Asioonu, og børdu hann med Knefum. (Eintal., 173); m17 Beria skal Barn til Asta. (JRúgm. 46); ms17 *innlendir barnir / þó öngvar fá varnir, / þeir ofríki líða. (HPSkv. II, 354); s17 Eingenn er sá barenn, sem Húsbóndans Skipan giörer. A‖ barður m16 barðan m16 málsh. barnir ms17 málsh.

(GÓlThes., 760); ms18 berja skal barn til aastar. (JÓGrvOb.); ms18 adsciscit Dativum instrumenti et Accusativum objecti ... ut: ... at beria mann griooti. (JÓGrvOb.); ms18 at beria mann med lurkum, keyrum og Svipum. (JÓGrvOb.); f19 Sá er enginn barinn, sem húsbóndans skipan gjörir. (GJ., 280); f19 Þann er ei vandt að verja, sem enginn vill berja. (GJ., 390); f19 Því veldr þrjózka þræls að hann er barinn. (GJ., 417); s19 hún vissi sem var, að faðir sinn mundi ekki trúa sér og berja sig eins og harðan fisk í þokkabót. (ÓDavÞj. III, 376); fm20 Hún vissi dæmi til að hrútar höfðu barið fjármenn til örkumla og bestar slíkt hið sama. (GFrRit. III, 290); m20 þetta er meinleysismaður, ég hef aldrei vitað hann berja gamalmenni. (HKLSjfólk., 294); m20 ekki skildust honum fyrirskipanirnar að heldur þótt hann væri barður. (HKLÍsl., 185); m20 barðir af meistaranum en skútyrtir af sveinum. (HKLHljm., 20); m20 Reyndi hann [ɔ: hrúturinn] ekkert að slíta sig lausan, bara að berja mig, svo var skapið mikið. (SkaftfÞjs., 172); s20 Það var lengi hefð á skútum, ef hann var tregur, að berja kokkinn. Þá fór alltaf að fiskast betur. (JÁrnVeturnóttak., 45); **berja e-n augum** ***sjá e-n, virða e-n fyrir sér*** m20 og hver er hann barði augum þá fraus honum blóð í æðum. (GrÞjóðs., 58); **berja e-n brigslum** ***dlasa e-m harkalega*** f20 og mótflokkarnir börðu hann látlaust brigslum. (Réttur. 1917 I, 16); m20 *Þó engan vilji ég brigzlum berja / beinum dugar ekki að verja / gamlan þaðan kominn kvitt. (JGÓlSög. 1938², 71); **berja e-n sundur og saman** s19 Þýzkir börðu þá sundr og saman, tóku yfir 10 þúsundir fanga. (Þjóð. 23, 34); **berja hrúta** „Tekið var báðum höndum um bita á milli sperra í baðstofu, þannig að fingurgómarnir sneru fram að andlitinu. Síðan átti að setja sig í hnút, þannig að hnén snertu bitann. Þetta átti svo að endurtaka án þess að láta fæturna koma í gólf. Fæstir gátu þetta nema 2–5 sinnum og afbragð þótti að geta barið 10 hrúta (eða 10 sinnum hrút) eða fleiri." (Tms. (S.-Þing.)); m20 Og fest gat kann höndum sínum upp um bita og 'barið hrúta', þótt hvorttveggja þetta þætti með ólíkindum með svo örkumlaðan mann. (Grímaný. II, 84). málsh. málfr. málsh. málsh. málsh. barður m20 þjóðh. hl‖ þjóðh. skýring

berja e–n frá sér: „Ekki máttum við sveifla bandspotta eða spýtu í kringum okkur því þá 'börðum við frá okkur englana'." (Tms. (Árn.)). A∫ þjóðtrú

berja e–n niður: *slá e–n svo að hann fellur* fm20 Ég áleit þá, og trúi því enn, að mennirnir, sem hjálpuðu mér upp úr sjónum, hafi verið sömu mennirnir, sem börðu mig niður á götunni. (IndEinSéð., 169). A∫

BERJA SIG: ■ **1.** *[fugl:] blaka vængjunum* fm20 Rjúpan flaug af í styggara lagi og barði sig aumkunarlega, þaut eftir kvíabólsstígnum eins og vængbrotin væri. (GFrRit. II, 226). ■ **2. berja sig utan** *slá óvitandi á líkama sinn* s19 fólkið barði sig utan fyrir vitleysuna og innvortis nagandi kvöl og skömm fyrir þessa athöfn. (EirÓl., 211).

BERJA E–U: *slá e. beina e–u harkalega [í e–a stefnu]* m19 sat þar þegjandi á kistunni og barði hælunum í hliðina. (JThSk. I, 56); s19 slóst hún upp á hana með skapraunarorðum, þreif af henni bogann og barði honum glottandi um eyru henni. (StollGoð., 40); f20 Þeir [ɔ: ungarnir] börðu vænglúrunum í vatnsskorpuna. (JTrRit. I, 275); **berja bægslunum** *brjótast e. þjösnast áfram* m19 Þar er lýst þrjótinum, sem einlægt ber bæxlunum, og hirðir ekkert um náttúruna. (Norðri. 1856, 53); s19 jeg, sem nú er að berja bækslunum fyrir lífi mínu og minna. (Þjóð. 27, 117); **berja e–u í vænginn** *afsaka sig með e–u* m19 En sé hann svo ófyrirleitinn, að ljóstra upp ást minni, þá ber ég því í vænginn, að ég hafi talað svona við hann. (Þús. I, 49); m19 Nú skal ég sjá, hverju þú ber í vænginn, mælti hún. (Þús. I, 119); f20 Þú hefir alt af barið einhverju í vænginn. (EHKvRit. V, 189); **berja höfðinu við steininn** *neita að beygja sig fyrir staðreyndum* s17 Illt er að berja Höfdenu vid Steinenn. (GÓlThes., 1825); s17 Hardt er ad berja Höfdenu wid Steinenn. (GÓlThes., 1446); f19 Bágt er að berja höfðinu við steininn. (GJ., 42); f19 Ad mótmæla honum mundi því verda ad berja høfdinu vid steininn. (Klp. VII, 69); s19f20 að stjórnin ... berji ekki höfðinu við steininn gegn því, sem sanngjarnar kröfur heimta. (JsJsRit. III, 270). A∫ málsh. málsh. málsh.

berja e–u fram: *halda e–u (ákaft) fram, staðhæfa e–ð* s19 Sjer sýndist ... hæpið að berja einhverju fram með ofurkappi, í staðinn fyrir með skynsemi og sönnunum. (Ísaf. 1879 viðauki, 36); f20 Hann [ɔ: Teitur] er ekki söguleikur, og það er ónákvæmni að berja því fram þvert ofan í yfirlýsingu mína í athugasemdunum. (JTrRit. VIII, 467). H∫

berja e–u í e–ð: berja augum í e–ð *verða starsýnt á e–ð* m19 lýtur samningurinn að ýmsu, er þau munu berja augum í. (Skírn. 1863, 59); s19 hafa þeir menn komizt á þingið er lengi hafa barið augum í svo mörg lagalýti. (Skírn. 1868, 159); s19 en börðu hins vegar augum í kostnaðinn og þær áþyngdir, er af því hlytu að rísa. (Skírn. 1871, 174). H∫

berja e–u niður: *breiða yfir e–ð* m17 eingenn skyllde draga Fiødur yfer Synder sijnar / beria þeim nidr eda laata sier lijtid til þeirra finnast. (FörstSkrift. G, VIIIv).

berja e–u við: *færa e–ð fram sem afsökun (fyrir e–u)* m16 kappsemin tijl at kenna Gudz ord heimtizt at predikorunum / swo at einginn meige þui vid beria. (CorvPost. II, 62r); s16 Nu so ad eingen þurfe þessu hier epter vid ad beria. (DietrPass. A, IIr); s17 Sumer beria því vid / ad þeir hafe laangann Kyrkiuveg. (DilherrPost. A, IIIr); m18 at berja einhverju vid, causari aliqvid, vel culpam in rem qvandam rejicere, causa qvadam prætensa et excogitata semet excusare, vel à crimine qvodam purgare. ... at berja við faatækt edur heilsu veiki. (JÓGrvOb.); s18 því hvørugr þeirra þurfti skulldum vid at beria. (LFR. IV, 44); f19 oc var því vidbarid at bændr hefdi fyrir þá sök skorast undann skattgialdi. (EspÁrb. III, 31); m19 Þó hann berði því við, að hjá sér væri mikill gullskortur. (Felsenb., 350); m19 Sumir berja því við, að þeir séu hræddir. (ÁrsrÞór. II, 39); m19 Því er opt barið við að menn þurfi að hafa brennivín til að hressa sig á. (NF. III, 130); m19 en hann barði við heilsulasleik sínum. (Þús. I, 4); s19 og þó sendi jeg þinginu 10 exemplör, svo því verður ekki barið við, að gengið hafi verið fram hjá þinginu. (Ísaf. 1891, 261); s19 allt af er því við barið, að ekkert verði gert fyrir peningaleysi. (Þjóð. 34, 76); s19 Sumir kunna að berja því við, að sökum harðæris og óárunar sé tíminn nú illa valinn til slíks fyrirtækis. skýring

(Suðri. 1885, 83); s19 Menn munu, ef til vill, berja því við, að ekki sé það hugsandi, að sjómenn vorir geti synt neitt í skinnklæðum. (Suðri. 1886, 134); s19 Því er einatt við barið, að nautn áfengisdrykkja sje því nær almenn í öllum löndum. (ÍslGT. VI, 134); s19 Þó að fjeleysi megi ef til vill við berja, ... þá er ekki hægt að kenna því um. (TímUpp. 1891, 65); s19 Karlinn barði bæði við kunnáttu- og hugleysi sínu. (FrEggFylg. I, 94); f20 Því er barið við, að vér séum svo fátækir, vér getum ekki farið að dæmi stórþjóðanna í þessu efni. (Fjallk. 1901, 23-1); f20 Hörmangarar hættu að láta skip sigla hingað 1745, börðu því við, að höfnin væri að fyllast af sandi. (Ægir. 1929, 92); f20 Kuldanum er mest barið við og er það nokkur ástæða. (Arnf., 109); f20 Barið væri við peningaleysi og verkfæraleysi. (Bún. 1902, 251); m20 barði [hann] því við, að heilsa sín myndi ekki þola loftslagsviðbrigðin. (SagaÍsl6., 310); m20 hinir flokkarnir börðu þá við tímaþröng vegna kosningaundirbúnings. (GBenSaga., 158); (Tms. (Skagaf.)).

berja e–m um e–ð: ***bregða e–m um skort*** *á e–u* m19 [hann] taldi á því öll vankvæði, og barði félaginu og stofnuninni um alla hæfilegleika til slíks. (Þjóð. 1853, 47).

BERJA SÉR: ■ **1.** ***slá sig utan (til að fá í sig hita)*** ms18 at beria sier til hita, semet ipsum calefaciendi causa verberare. (JÓGrvOb.); s19 menn þar inni voru að hrista og stappa af sér snjóinn og berja sér. (EirÓl., 47); s19f20 *Svo brauzt jeg upp í bræði í norðankóf / að berja mér og stappa niður fótum. (ÞErlRit. II, 186); m20 gengu [þeir] um gólf á sandinum og börðu sér til þess að halda á sér hita. (Blanda. VIII, 299); m20 Húskarlar þessir báru sig heldur krokulega og börðu sér mjög, milli þess sem þeir bundu baggana. (ThFrVer., 462); m20 Hann ber sér og ekur af kulda, klípur til skiptis stofugriðkuna og eldabuskuna. (TsjekovMað., 33); „Nokkur húsráð til þess að hita sér á höndum: að vinda loppungana, hræra flautir, fara í lúsaham, slá svensk, berja sér duglega." (Tms. (Rang.)). ■ **2.** „Staðan bein, fótum lyft sitt á hvað í lárétta stellingu frá mjöðm til hnés, en beygja um hnélið með sljóu horni. Meðan staðið er í vinstri fót er reynt að slá lófum saman undir hnésbót hægri fótar, skipt um fót og farið eins að. Ekki var færst úr stað, en hreyfing þess fótar sem staðið var í hverju sinni þurfti að vera létt og fjaðrandi, en ekki þunglamaleg." (Tms. (Eyjaf.)). ■ **3.** „Standa uppréttur, sveifla höndum eins langt aftur fyrir bak og unnt er, lófum skellt saman. Sömuleiðis fram fyrir sig, lófum skellt saman; skarplegar hreyfingar, öndun með nefi." (Tms. (Eyjaf.)). ■ **4.** „Standa uppréttur, berja höndum saman framan á brjósti í kross, svo langt að lófar nemi við axlir. Þetta skal gera skarplega og standa á öndinni meðan lotan varir." (Tms. (Eyjaf.)). ■ **5.** ***kvarta (um bágan hag e. bágt ástand), kveinka sér*** m17 *Ei mun bót að berja sér, / bernskan hefur sinn máta. (HPFlór. XVI, 6); m18 Nú hefir hann nokkra stund gripið til þess ráðs að berja sér aumkunarlega. (JÞorkÆf. II, 152 (1758)); m18 Ad beria sier. Penuriam conqveri, paupertatem exaggerare. (JÁLbs224 4to., 93); s18f19 *Margir kvarta, margir raupa, / Margir berja sér. (JHjaltTíð., 128); f19 Það eru ekki búmenn, sem ekki kunna að berja sér. (GJ., 376); f19 Þú veizt, að ekki er aldeilis óhætt að reiða sig á það sem búmenn segja, því sagt er þeir berji sér öðrum fremur. (BiskGörð., 9 (1811)).

berja sér í e–ð: berja sér í nestið ***vera í andarslitrunum, vera að dauða kominn*** m20 að loksins væri Ormur gamli á Grjótlæk farinn að berja sér í nestið. (GDanBolafl. II, 56); Að berja sér í nestið. Það mun lítið vera notað nú til dags. Eitt sinn heyrði ég ömmu mína tala um unga menn, sem komu frá sjóróðrum heim til sín upp í Hreppa, og voru þá með hósta. Sagði þá fólk, að þeir væru farnir að berja sér í nestið, og skildist mér þá, að átt hafi verið við hóstann, sem talinn var banvænn enda dóu þeir úr berklum litlu síðar. (Tms. (Árn.)); (Tms. (Árn., Mýr., Skagaf., S.-Þing.)).

berja sér niður: ***kvarta, bera sig illa*** ms18 at beria sier nidr, lamentari, seu præ moestitia semet ipsum terræ allidere. (JÓGrvOb.); s18 Eg ber mer nidr, — lamento. (HFLbs99fol., 192).

berja sér um e–ð: ■ **1.** ***kvarta yfir e–u*** f17 A medan þeir Ogudlegu eru ad telia sier þessar

A‖ skýring

A‖ þjóðh. skýring

A‖ þjóðh. skýring

A‖ þjóðh. skýring

J‖

skýring

málsh.

b‖

málv.

málv.

J‖ skýring

skýring

J‖

Tølur og beria sier um þeirra Idranarleyse. (NicSpeg., 667); f18 Kann sa ad beria sier umm Armood. (VídPost. I, 329); ms18 at beria sier umm nockut, alicujus causa qviritari. (JÓGrvOb.); f19 ad vér skulum berja ockur og barma um peningaleysi. (Árm. III, 115); m19 þykir landið vera svo gæðalítið, og eru því allt af að berja sjer um fátækt. (Þjóð. 1848, 6). ■ **2.** ***kvarta yfir skorti á e–u*** s18 Um þá síðari [ɔ: þekkingu] þurfum vér ecki svo sérlega ad berja oss. (MartEðl., 92v); s18f19 um allt Sudurland ... þarf qvennfólk ecki ad berja sér um hreyfingu. (JPétLækn., 73); f19 þá Sigurður ... vildi kaupa sjer kot og barði sjer um peninga til þess. (Ldsyrd. III, 264 (1828)).

HANN BER:

hann ber á e–u: hann ber á blikunni „getur líka hvesst aðeins, og er þá sagt að hann berji á blikunni eða blási þetta af sér." (Tms. (Dal.)). málv.

HANN BER E–Ð:

hann ber e–ð í sig: hann er að berja í sig hláku/blota „Talað var um að hann væri að berja í sig hláku eða blota, er veðurbreyting í þá átt var í aðsigi." (Tms. (Borgarf. v.)); **hann er að berja í sig lin** ***það er að draga úr frosti*** m20 Nú fer hann bráðum að berja í sig lin, sagði konan. (HKLSjfólk., 249); m20 Hann er að berja í sig lin. (HMatthVeð., 87); „Ef verið hafði ótíð, frost og hagleysa, en fór að draga úr frosti var sagt: Hann er að berja í sig lin." (Tms. (V.-Skaft. (Árn., A.-Barð.))). j↓ málv. k↓ málv.

HANN BER SIG: **hann ber sig í lin** ***„það dregur úr frosti"*** (Tms. (V.-Skaft.)); **hann er að berja sig upp í hláku** ***þíðviðri er í aðsigi eftir kuldakast*** (Tms. (Norður., Hnapp.)). k↑ j↑ málv.

BERJAST: ■ **1.** ***heyja bardaga e. baráttu*** m16 þu gudz madr ... berst godre barattu truarinnar / hondla suo eilift lif. (1Tím. 6, 12 (OG)); s17 Eg hefe barest eina goda Baraattu. (GÞorlPost. I Rr, IVr); f19 Betri er sá sem berst, enn hinn sem óþrífst. (GJ., 52); m19 Sá fær litlu afkastað, sem einsamall er að bjástra og berjast. (ÁrsrÞór. I, 4); m19 Ég er að berjast þetta einn og hef of lítið að styðjast við. (JHall. II, 27); m20 Nú á dögum er bara barist útí loftið af eintómum bjánaskap og þrjósku. (HKLSjfólk., 370); **berjandisk og bölvandisk** m20 að hafa hrútinn Séra Guðmund og bróður hans í krónni hinumegin við flórinn, berjandisk og bölvandisk alla nóttina. (HKLSjfólk., 179); K↓ skýring l↓

berjast eins og ljón fm20 Konungshugurinn hefur það til að leggja sig í lífsháska, þegar engin von er um undankomu, snúast þá öndverður gegn ofurefli og berjast þá eins og ljón. (GFrRit. VI, 401); **berjast í bökkum** ***a. eiga í erfiðleikum fjárhagslega*** m19 Þó berst bóndi ... í bökkum í öllum meðalárum. (Þjóð. 9, 10); s19 Þegar maður verður að berjast í bökkum til að hafa af fyrir ómegð sinni. (Þjóð. 35, 115); m20 Alla ævi barðist hann í bökkum, enda fádæma ráðlaus í fjármálum. (BergJMann., 74); m20 Leingi hefur Þjóðviljinn barizt í bökkum og berst enn. (HKLSjhl., 252); ***b. [viðureign:] standa því sem næst jafnt*** m18 stryded bardest i bòckum. Vario, (ancipiti) Marte pugnatum est. (JÁLbs224 4to., 999); s19 er svo að sjá ... sem þar á sléttunum hafi barizt í bökkum ófriðurinn af beggja hálfu. (Þjóð. 29, 113); **berjast í vök** ***eiga í erfiðleikum*** f20 *Hýstu aldrei þinn harm. Það er bezt. / Að heiman, út, ef þú berst í vök. (EBenLj. II, 249). n↓ skýring

■ **2.** ***[tveir e. fleiri innbyrðis:] heyja bardaga e. baráttu*** m19 við ... börðustum fyrir utan laufskálann. (Felsenb., 34); f20 Það veit líka á úrkomu, þegar hrafnar fljúgast á og berjast, en á þurt veður, ef þeim kemur vel saman og þeir kvaka á fluginn. (SÞórVeð., 65); fm20 Þær voru að berjast margar og voru reiðar. ... Það veit á hvassviðri, þegar sauðskepnurnar láta svona í kyrru veðri. (GFrRit. II, 271); s20 Tarfarnir berjast miskunnarlaust allan fengitímann. (JÁrnVeturnóttak., 139). K↓ þjóðtrú þjóðtrú

■ **3.** ***brjótast áfram*** m20 hlíðarklæddir menn börðust út í myrkrið með smíðatól, ljósker og kaðla. (Andv. 1960, 234). ■ **4.** ***hreyfast ákaft e. slást til*** m17 *hiartad bardest i brioste heitt, / bæde var lijf og salinn þreytt. (HPPass. II, 12); m19 hjartað barðist ekki af ekka. (JThSk. I, 11); s19 *Hið blakka hár hans berst um stafn / sem berji vængjum úfinn hrafn. (MJÞLeik., 53); s19f20 Klettarnir næst fossinum eru með dökkri gljáandi skán, þar sem vatnið hefir barist um þá. (ÞThFerð. I, 376); **byltandisk og berjandisk** s19f20 Já, nema skáldið sé eins og örkin gamla, byltandisk og berjandisk í brimróti veraldarflóðsins, umfaðm- D↑ l↑

andisk og innigeymandisk allar skepnur illar og góðar. (MJBrHH., 16). ■ **5.** *vera laminn* A↓ s19 Lauf barðist burt af skógum [ɔ: í haglhríð]. (Þjóð. 27, 97). m20 missti hann hrúta tvo af steinkasti, eða þeir börðust til dauða. (Vísagn. III, 308).

berjast af: *komast af* s19 hefi jeg ekki þurft mikið að brúka hana [ɔ: reikningslist] um dagana, jeg hefi einhvernveginn barist af án þess. (Nf. XVIII, 9).

berjast á: *takast á* m20 Þar er auðsénn mikill vindhraði, og tvær áttir berjast á. (Náttúrufr. 1953, 191). K↓

berjast á e–ð: *ráðast á e–ð, heyja bardaga til að ná e–u* s16 Sem Ioab bardist nu a Borgena. (2Sam. 11, 16 (GÞ)); s16 hann ... settist vm Samariam / og bardest a Borgena. (1Kong. 20, 1 (GÞ)); f19 svo að þeir gátu farið að berjast á kastalann. (Skírn. 1833, 9). K↓

berjast á móti e–u: *beita sér mjög gegn e–u* **berjast með hnúum og hnefum á móti e–u** *beita sér af alefli gegn e–u* m20 þeir, sem berjast með hnúum og hnefum á móti hlutfallskosningum. (BjBenLand. I, 195). L↓ o↓

berjast fram: *komast af, ná að fleyta sér* s18 og fleztir á 10 hr. leigumálum beriaz fram einsamlir. (LFR. IV, 171). M↓

berjast fram úr e–u: *takast á við e–ð og leysa það* s19 neitaði þó alltaf að taka við hjálp frá öðrum, og vildi sjálfur berjast fram úr þeim vandræðum. (Fróði. 1885, 201); s19f20 Eg barðist fram úr að læra dönsku. (JsJsRit. II, 126). M↑

berjast fyrir e–u: *beita sér mjög fyrir e–u* f20 að innlendir menn skuli hafa barist fyrir því með öllum meðölum, að leggja niður landsins peninga stofnun. (Alm. 1903, 66); **berjast eins og ljón fyrir e–u** m20 Alþýðuflokksforystan berst eins og ljón fyrir þeirri helstefnu, sem leitt hefur til atvinnuleysis. (Réttur. 1951, 11); L↓

berjast fyrir e–u með hnúum og hnjám *beita sér af alefli fyrir e–u* s19 ákvörðun þá, sem hann ... hafði barizt fyrir með hnúum og hnjám. (Þjóð. 36, 54). o↓

berjast fyrir e–m: *heyja baráttu e. leggja mikið á sig í þágu e–s* ms17 þeim sem berjast fyrer barnafiolda sinum med eimd og armædu. (SafnF. XII, 263). L↓

berjast gegn e–u: *beita sér mjög gegn e–u* **berjast gegn e–u hnúum og hnefum** *beita sér af alefli gegn e–u* f20 Þeir hafa barizt gegn þessu hnúum og hnefum og ótal mörgu öðru. (EHKvEitt., 59 (1905)). L↓ o↑

berjast í e–u: *basla við e–ð* ms17 ad berjast j þessum barattusama bwskap. (SafnF. XII, 6). N↓

berjast í móti e–m: *stofna til andstöðu við e–n, beita sér gegn e–m* m16 suo at eigi synuzt þier beriazt gudi i moti. (Post. 5, 39 (OG)); f20 Sárast tekur hann það, að Arnljótur skuli hafa barizt í móti sér í þessu máli. (PEÓlJSig. IV, 103). L↑

berjast til e–rs: *heyja bardaga til að ná e–u* m19 að þá gengu Norðmenn opt hraustlega fram, er þeir börðust eigi til krossanna. (Skírn. 1849, 133); s19 Austurríkismenn ... urðu ... að láta af hendi við þá það er þeir höfðu til barizt. (Alm. 1883, 27); s19 Því sannleikrinn er ekki dauðr bókstafr; hann er þvert á móti herfang, sem hver einstakr verðr að berjast til. (Nanna. II, 47); s19 að hann hafði í huga að berjast til valda nær sem færi gæfist. (Skuld. 1880 nr. 119, 186). K↓

berjast um: *hreyfast ákaft e. brjótast um* m19 Óvinurinn hamaðist, þandi út klærnar, barðist um með vængjunum. (JHall. I, 301); m19 Margir eru farnir að slétta tún sem áður börðust um í þýfinu. (MelBr., 3); m19 hjarta hennar fór að berjast um í brjóstinu af hræðslu. (JThSk. I, 12); **berjast um á hæl og hnakka** *brjótast um af mikilli ákefð e. ofsa* ms18 at beriast um á hæl og hnacka, turbulenter se gerere, calcibus et occipite subnixus (JÓGrvOb.); m19 drengur barðist um á hæl og hnakka og orgaði. (JÁÞj2. III, 78); O↓ a↓ skýring

berjast um hnakka og hæl *brjótast um af mikilli ákefð e. ofsa* s17f18 berst eg nu um hnacka og hæl ad svara i skyndi. (ÁMTorf., 262). a↑

berjast um e–ð: *heyja innbyrðis bardaga e. baráttu um e–ð* s19 menn nærri börðust um að fá þá náð að bera lík hans til grafar. (Nf. XXIII, 51); fm20 á sléttunni uppi við Sátujökul berjast þær um vatnið, Vestari-Jökulsá og Strangakvísl. (PHannÓb., 62). K↓

berjast undan e–u: *brjótast undan e–u* f20 Og samheldni Bandamanna var sömuleiðis O↑

sterkasti kastalinn, er þeir börðust undan áþján Englendinga og náðu frelsi sínu. (Fjallk. 1902, 15-2).

berjast við e–ð: ■ **1.** ***basla við e–ð, fást með erfiðismunum við e–ð*** ms18 at berjast vit eitthvad, satagere. (JÓGrvOb.); s18 ad beriast vid baráttu, adversis premi. (HFLbs99fol., 213); m19 Í ... 94. bl. ... Þjóðólfs hefir einhver jeg farið að berjast við, að halda uppi svörum fyrir sýslumanninn í Rángárvallasýslu. (Þjóð. 1853, 138); s19 Guðbrandur var gáfnadaufur og barðist við lagalærdóm í 12 ár utanlands. (FrEggFylg. I, 17); fm20 að hún ætti að selja sér eyna og vera ekki að berjast við þennan búskap. (Gráskinnahm. II, 76). ■ **2.** ***eiga í baráttu við e–ð*** ms18 at beriast vid faatæktina. (JÓGrvOb.); **berjast við skuggann sinn** ***berjast vonlausri baráttu*** s19 Hinn heiðraði höf. berst alveg við skuggann sinn, þar sem hann er að bera af sjer athugasemdina. (Ísaf. 1891, 397); **berjast við öndina** ***vera í andarslitrunum*** f17 ma sia a þeim sem vid øndena beriast / huørsu þeir þunglega kueliast. (MollMan. I, Ir); m18 Ad Beriast vid öndena. In mortis agone versari, constitutum esse, extrema agere, animam agere. (JÁLbs224 4to., 93); s18 vari óróleiki sá lengi, er fylgir andarslitrum, kallaz at hinn siúki beriez vid øndina. (LFR. IX, 187); f19 bardist sídan vid øndina í fullar 15 kluckustundir, og gaf hana loksins upp. (MStSmás., 264); m19 Ef maður lætur ljós smádrepast, og kvelur það, þá berst maður leingi við öndina. (JÁÞj. II, 550); m19 var eins og hann berðist við öndina og gæti ekki dáið. (JÁÞj. I, 618); s19 vita menn ekki, hversu sárt það er foreldrunum að sjá barn sitt berjast við öndina tímum saman, opt ár eptir ár. (Þjóð. 39, 10); **eiga við e–ð að berjast** ***eiga við e–ð að etja*** f18 Átti hann að berjast við þverúð og þrályndi flestra af hinu eldra fólki. (JHBisk. I, 104); m19 Við þessa flaungun átti hún leingi að berjast. (JÁÞj. II, 113); s19 Pasteur ... átti þó við mikla erfiðleika að berjast. (Alm. 1888, 30).

berjast við e–n: ***heyja bardaga e. baráttu við e–n*** ms18 at beriast vid mann, certare cum aliqvo, ut in adagio: þar er ei vid blaamenn at beriast, sem brwdir eiga land ad veria. (JÓGrvOb.); f19 Þar er ei við börn að berjast, sem hann er. (GJ., 392); m19 að nefndin í þessu tilliti átti ekki, ef svo má segja, við barn að berjast, þar sem frumvarpið var. (Alþ. 1853, 747). m19 berjast sem haukr við klár. (Sch.); m20 Austan- og vestanvindar eru sjaldan eins hvassir. Þeir berjast oft hvor við annan þannig, að skýin dregur upp frá vestri og koma svo strax aftur frá austri. (Landnlng. I, 15).

Það berst: ■ **1.** **það berst í barnástum með e–um** ***sagt í tilefni af því að ungmenni kljást e. glettast*** m20 Látið þið þau eiga sig, það berst bara í barnástum með þeim. (Þvígl. III, 226); s20 Nei, látið þið þau vera. Það berst bara í barnástum með þeim. (StÞórðNú., 116). ■ **2.** **það berst í bökkum** ***a. afkoman er tæp, það tekst naumlega að framfleyta sér*** s19 Í góðu árunum má ekki láta sjer lynda að berjist í bökkum. (Ísaf. 1890, 281); ***b. [e–að] stendur því sem næst jafnt, [e–að] stendur enn óútkljáð*** m18 þad Berst i böckum. Dubio ancipiti, vario Marte pugnatur. (JÁLbs224 4to., 93); m19 Barst lengi í bökkum með hvorumtveggju, unz Svartfellingar rjeðu til meginbardaga hjá þorpi er Duga heitir. (Skírn. 1863, 102); s19 Á þinginu í Washington berst nokkurn veginn í bökkum með þeim, þjóðvaldsmenn eru í meiri hluta í öldungaráðinu, en hinir á fulltrúaþinginu. (Skírn. 1876, 160); s19 er sagt, að ... hjer hafi barizt í bökkum, en Rússar hafi þó haldið Bjela. (Ísaf. 1877, 106); s19 Berst í bökkum enn í Tonkin. (Nf. XXII, 114); *c.* ms18 Þat berst í Bòckum, adv. hoc ripas allidit, loqvendi modus de re, qvæ ultro citro, huc et illuc ventilatur et reciprocatur nec effectum sortitur. Metaphora similitudinis, ducta à re qvadam intra ripas rivis fluitante. (JÓGrvOb.); **það berst í bökkum fyrir e–m** ***afkoma e–s er tæp, e–m tekst naumlega að framfleyta sér*** s18 hvørsu mikid, sem ... lausamenn sýnast ad mega ávinna med bralli sínu ... er þó, sem med íllann leik berjist í bøckum fyrir flestum jafnvel í gódum árum. (MStGAlv. I, 67;) s18 Og fyrir þessum berst í bøckum. (LFR. V, 72); **það berst í bökkum með e–um** ***viðureign e–ra stendur jafnt, það gengur hvorki né rekur í viðureign e–ra***

N‖ skýring K‖ málv. málsh. þjóðtrú skýring málv. málv. n‖ skýring n‖ n‖

ARNE JÖNSSON

Application-Dependent Discourse Management for Natural Language Interfaces: An Empirical Investigation

Abstract

This paper presents results from a refined analysis of a series of Wizard of Oz-experiments with focus on discourse management. The study is part of a larger project aimed at designing a general natural language interface. It is argued that a natural language interface needs different referent resolution mechanisms for different combinations of background systems and scenarios. I present three mechanisms that can be used: proximity, object hierarchy and goal-directed control. Finally I suggest the use of a Natural Language Interface Management System (NLIMS) for customization of natural language interfaces to different applications.

1 Introduction

The language used by a user when communicating with a computer application (database, expert system etc.) in natural language will differ from the language used in a spoken face-to-face communication between humans as well as the language used in written communication.[1]

The development of natural language interfaces more sophisticated than simple question-answering requires knowledge of the characteristics of such an interaction, not only of the language but also of how the discourse progresses. In a series of experiments (Dahlbäck & Jönsson 1989; Jönsson & Dahlbäck 1988) we have studied these phenomena and presented results on the characteristics of the interaction between a human and a natural language interface.

We have conducted experiments using five different background systems (see section 2) and 21 subjects, all computer novices. After every simulation we have interviewed the subject in order to find out how they liked the 'system'. No

[1] In Jönsson & Dahlbäck (1988) we elaborate more on this.

subject understood that it was a simulation and many (too many?) were *not* surprised with the system's capabilities. We have focused on computational aspects of the interpretation of the Subjects/Users utterances. This means that we have not considered the utterances typed by the system (simulator).

There are 1047 utterances which we have divided into four different categories; Initiative, Response, Resp/Init and Clarification, (cf. Linell, Gustavsson, & Juvonen (1988)). What poses problems from a computational point of view are mainly the user Initiatives and especially the indexical initiatives. There are 193 indexical subject utterances, which is 49% of the initiatives. Dahlbäck & Jönsson (op. cit.) further analyze these indexical utterances and also provide some other results. But the results have to be refined in order to see what mechanisms different kinds of NLI's should adopt when interpreting an indexical utterance.

2 Scenarios and Background Systems

In our experiments we have varied the type of background system and the scenario. The analysis of these experiments exposes considerable differences between different combinations of background systems and scenarios. These differences must be considered, as this study is concerned with exploring computational mechanisms for handling indexicals. Consequent our background systems and scenarios require a presentation.

The *scenario* is the characteristics of the task the subject has to solve. We used two different types of scenario, namely information retrieval and configuration. They differ in that a *configuration*, besides data retrieval, must deal with a dynamically updated object, such as an order or specification, during the discourse, while *information retrieval* is less complex and involves only retrieving facts from an information base.

Further, we use two different kinds of *background systems*, simple database systems and advisory systems. A *database system* simply returns facts from the database. The subject must state an information seeking question, he or she cannot ask for an opinion or evaluation from the system. An *advisory system*, on the other hand, can also be used for consultation. The subject can ask for advice i.e. the system's evaluation of certain facts. In both systems we have most answers as pre-stored texts, or text fragments which are completed, and the simulator (wizard) selects the appropriate text.

The different experimental settings as a function of the combination of background system and scenario are shown in Table 1. The name refers to the background system and the number of experiments of each combination is stated inside the parenthesis. Section 4 contains some examples from our simulations.

There are two systems of the most simple combination, namely database background system and information retrieval; one where the subjects ask questions concerning publications at our department (PUB), and one where they can ask questions about one of the education programmes (C-line). These systems allow the subjects to ask questions and also to refer to previous pieces

Scenario	Background system: Data base	Background system: Advisory system
Information retrieval	PUB (4) C-line (5)	Wines (4)
Config-uration	Travel (3)	HiFi (5)

Table 1: Scenarios and background systems.

of discourse, but they do not allow questions like *Which books have I ordered*? (configuration) or *Is that a good course*? (advice).

A bit more sophisticated is the information retrieval advisory system for selecting wines where the subjects can get advice on whether a wine is suitable for some specific meal, dialogue 1. However, it is not possible to select one wine for the first course and then order that wine, i.e. configuration is not possible. This does not prevent the user from referring to previously mentioned wines.

```
.
U:9> What do you recommend with crab-filled avocado?
S:10> Searching ...
       With crab-filled avocado a white wine is suitable
       Do you want more information about any specific wine?
U:11> Yes  in the price range from 30 to 50 SEK
S:12> Please wait
       Do you want rhine-, mosel- or burgundy wines. Or all.
U:13> Rhine wine
S:14> Searching ...
       The following wines exist ...  listing ....
.
```

Dialogue 1. Wines:1 (The corpus is in Swedish and translated for verbatim correctness.)

Another combination is configuration and database. This is found in the travel system. Here the subjects can order a trip to a holiday resort by retrieving information from a database and using these facts to configure a trip. However, they can not get advice, only access the database. This is discussed in section 4.2.

Finally, we have the HiFi dialogues, which are both advisory system and configuration. Here the subjects can get advice on HiFi equipment and also select their own outfit. The fact that a system is an advisory system does not prevent the use of canned texts. In fact we try to use pre-stored texts and sentence fragments as much as possible in order to get a uniform setting and amplify the belief that the subject interacts with a computer.

3 Discourse Management

The research reported here is part of a larger project on designing a general Swedish natural language interface. We have a pilot version of the system, called FALIN (Ahrenberg, 1989) which is a constraint-based, object-oriented model for a natural language dialogue system. FALIN provides a content structure which is an exhaustive description of the utterance. The content structure can be seen as an instantiated case frame and can be used for accessing the database or knowledge base. As an example, the content structure for utterance U:104 *ordered turntable* in dialogue 2, HiFi:3, would look like:

$$\begin{bmatrix} \text{CLASS:} & \text{\$Question;} \\ \text{SPEAKER:} & \text{Subject\#3;} \\ \text{ADDR:} & \text{System;} \\ \text{BASIS:} & \text{Subject\#3.Order;} \\ \text{ASPECT:} & \text{\$Turntable} \end{bmatrix}$$

When FALIN builds a content structure it can access information from what is called the `Discourse.State` for finding the referent to an anaphoric expression. The `Discourse.State` contains information about which of the many objects introduced in the discourse is currently active and also pending objects. This information is used to determine the referent, i.e. to select which of the many possible objects in the discourse that a certain anaphoric expression refers to. Thus, one needs some kind of dynamic discourse representation. This representation can be constructed either from the discourse using certain coherence criteria (see below) or like Grosz (1977), and Reichman (1985), from the task structure. In the present case the task structure is reflected by the conceptual structure of the knowledge base, see figure 1. The conceptual model of the knowledge base is a static structure and not a dynamically created description of the discourse.

The problem of managing discourse is normally divided in two parts: segmentation and finding the focused object. **Segmentation** is the process of finding out which parts in the discourse belong together and where one changes topic. The problem of segmentation is connected to the problem of coherence in discourse, where *cue words* (Reichman 1985), *coherence relations* (Hobbs 1985), *recency and temporal progression* (Webber 1988; Merkel 1989) are regarded as major coherence criteria.

The **focusing** problem is to find, given a segment, which of the many different objects that are actually talked about, is in focus. This is often done by

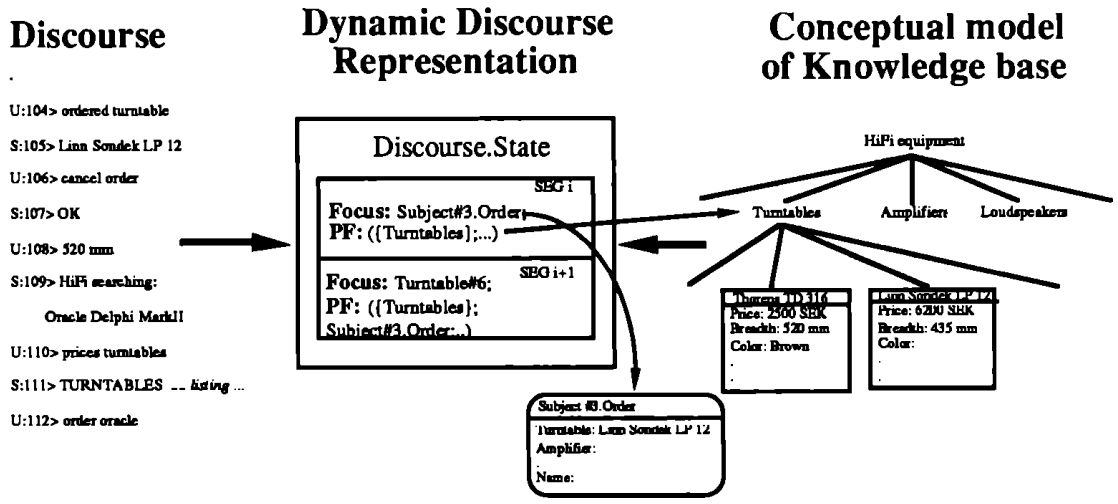


Figure 1: The Dynamic Discourse Representation

maintaining both a reference to the focused object (or objects), if any, and also a, possibly empty, list of objects that might be focused later, potential foci (PF) (Sidner 1983).

4 Empirical Findings

In this section I will illustrate the various strategies found by examples from our corpus. However, as can be seen from Table 1 there are few experiments of each combination of scenario and background system. Therefore I will not give any quantitative data on the occurrence of the different phenomena reported here.

4.1 Some General Findings

First of all there are **individual differences**, as can be seen from the examples in dialogues 2 and 3, both taken from HiFi, the most complex combination of scenario and background system. Dialogue 2 is a good example of what has been described as 'telegraphic' (Guindon, Shuldberg & Connor 1987), or 'computerese' (Reilly 1987).

This user started with a less fragmented language, but when he found that it was possible to interact in this 'cryptic' fashion he started to explore the database—not the language component as it may seem. Compare this dialogue fragment with dialogue 3.

The subject in dialogue 3 is a secretary and she is the subject which uses the most sophisticated language constructions. I have not selected a special part of the dialogue, although utterance 15 is our longest. This subject writes more

```
.
U:104> ordered turntable
S:105> Linn Sondek LP 12
U:106> cancel order
S:107> OK
U:108> 520 mm
S:109> HiFi searching:
         Oracle Delphi MarkII
U:110> prices turntables
S:111> TURNTABLES  .... listing...
U:112> order oracle
.
```

Dialogue 2. HiFi:3

```
.
U:11> What is the price if I instead take a cheaper pair of
        loudspeakers
S:12> What loudspeakers?
U:13> The cheapest there are
S:14> 70 600 SEK
U:15> Now I want to change to a better pair of loudspeakers which
        are at the most 29 400 SEK more than the ones I have now.
        The new items I'm getting should be as appropriate as possible
        for a small room.
```

Dialogue 3. HiFi:1

than one sentence in many of her turn-takings and illustrates the fact that certain non-linguistic factors, like type-writing skills, affect the language.

As noted above, **cue words**, **coherence relations**, and **temporal progression** are regarded as major coherence criteria. In our corpus I found no cue words used by the subject to signal the beginning of a new segment or termination of a segment and temporal progression is not significant for identifying segment shifts.

Although these findings seem pessimistic concerning the future of NLIs there are other results which are more optimistic. Especially the use of what I call Resp/Init, i.e. responses from the system which also include a **system initiative**. If the system does not only respond to the user's input, but also puts forward a follow up question, the dialogue becomes simpler. This is seen in dialogues 1 and 4. In dialogue 1 the system takes an initiative by hinting on a probable new piece of information to the user. In dialogue 4 the system follows a plan to help the user fulfil his goal.

Note that I do not consider the interpretation of the indexical utterances U:8 or U:10 in dialogue 4 problematic as they are responses to questions posed by the

```
.
U:6> We would like to go to Corfu
S:7> What hotel class do you want?
U:8> The best
S:9> How long will you be gone?
U:10> For two weeks
S:11> Vingfritid searching
        Hotel Hilton +++++ costs 5575 SEK/person
U:12> How much does the cheapest class cost?
S:13> Vingfritid searching
        Hotel Royal +++ costs 2245 SEK/person
.
```

Dialogue 4. Travel:2

system and, as such computationally easier to handle than the initiative taken in for instance U:12. This is also seen in dialogue 1 where for instance utterance U:11 is a response that could be interpreted using case-frame expectations like those described in Carbonell & Brown (1988).

4.2 Techniques for Discourse Management

The motivation for my study was to explore which computational mechanisms to use for a certain combination of background system and scenario. I have identified three different mechanisms for finding the referent to an indexical, that are applicable to different dialogue categories. I call them *Proximity*, *Object hierarchy* and *Goal directed control*.

Object **proximity** is found as the major criterion in the database information retrieval dialogues. In these dialogues the subject discusses one task and when that task is finished the object being discussed is dropped and a new object is introduced. This means that finding the referent is no problem; it is the closest object at hand, i.e. the object in focus. If the proximity heuristic fails to find the referent, some simple **recency** criteria can be used, like searching the segment stack for the referent. Carbonell & Brown (1988) present a Multi-Strategy approach to anaphora resolution using case frame information. They restrict themselves to a single previous sentence but suggest the use of recency if context information is necessary. Dialogue 5 from the PUB system gives an example.

This dialogue is very representative for all the PUB dialogues and many of the C-line dialogues. In this dialogue the signals to manipulate the segment stack and creating new segments is found from the syntactic/semantic structure, for example the segment shift in U:15 in dialogue 5. Further, the next utterance, utterance PUB:2:21, is: *What books are there on artificial intelligence*? This utterance cannot be seen as a continuation of any object on the segment stack, and thus the segment stack is popped and a new segment is created.

Proximity can also be viewed as a simple variety of goal-directed control (see

```
.
U:13> Who has written a book on linguistics?
S:14> PUB Searching. There are three authors of books about
        linguistics. Do you want to see them all?
U:15> No only one.
S:16> PUB Searching. Brown and Yule has written a book on
        linguistics.
U:17> What is the title
S:18> PUB Searching. Discourse analysis
U:19> Where is it.
S:20> PUB Searching. Owner: Mats Wir'en
.
```

Dialogue 5. PUB:2

below). However, I regard this mechanism separately since my goal is to find the simplest possible mechanism for each combination of background system and scenario. The reason is that one should not use a more sophisticated mechanism than necessary for a certain application.

In section three I discussed whether the referent is to be found using information from the discourse or from a conceptual model of the database. Object proximity is one example of using the discourse as the source for finding the referent. If a conceptual model of the database is used we get what I call **object hierarchy**. Grosz (1977) used partitioned semantic networks—focus spaces—which reflected the task structure for interpreting discourse. Bobrow & Webber (1980) use a general object-centred knowledge representation for syntactic and semantic processing. Figure 1 gives an example of a conceptual model of the database for the HiFi dialogues. Dialogue 6 taken from the travel system dialogues is a good example of when it is more or less impossible to find the referent using only information about the discourse.

Here the subject discusses excursions in utterances S:33 to U:40. Thereafter the topic is dropped and the subject orders his travel. Twenty eight utterances later he once more raises the question of excursions in U:68. However, this question does not refer to excursions in general (as before), but to excursions on Corfu. Information about where the user wants to go is found in the subject's order and using the hierarchically organized database the information on excursions is found by searching this hierarchy.

Finally we have **Goal directed control**. I use the phrase goal directed control to describe the mechanism where the system uses knowledge about what goals the user has to fulfil. Grosz & Sidner (1986) identify a discourse segment purpose which provides information on what purpose a certain segment has in the discourse. Carberry (1989) presents a number of discourse goals and rules for utterances in information-seeking dialogues. She also identifies a number of plans for nonprimitive tasks. Litman (1985) provides not only plans but also

```
.
S:33> Do you want VingFritid to book any excursions?
U:34> Which ones exist?
S:35> VingFritid searching. Corfu excursions: Town sight-seeing,
        Island tour, Buss to Glyfada, Boat tour to Kassiopi,
        Afra by night.
U:36> What is Island tour and Afra by night
S:37> Island tour: .. listing...
        Afra by night: .. listing...
U:38> Can't  I book these excursions when I arrive?
S:39> Yes you can.
U:40> Then I will not book any excursion now
.
.
U:68> Can you send us a brochure on the excursions?
.
```

Dialogue 6. Travel:2

meta plans. These studies all report the importance to model and use the users expectations and communication goals when interpreting an utterance.

Dialogue 2 is an example where knowledge about the user's goal can help. Utterance U:108 signals a segment shift from one turntable model to another. The user wants to find a turntable in the database with breadth 520 mm.[2] However, 520 mm could also be interpreted as a question whether the canceled turntable was 520 mm or if there is any item in the database that is 520 mm.

To handle this we assume that the discourse state consists of both focused object and potential foci (PF) objects. After utterance S:107, Turntables is in PF as turntables are currently discussed, so 520 mm could be interpreted as either the breadth of the canceled turntable or of any turntable. Now, the user's goal is to complete an order and after canceling one item it is probable that he will order another of the same type. So a search for another turntable occurs.

To summarize:

- Proximity is the major criteria in the database information retrieval dialogues
- Object hierarchy is important for database configuration dialogues
- Goal directed control is necessary for advisory system configuration dialogues.

[2]In this particular dialogue, the user has discussed the breadth of the items before and therefore breadth is the aspect being discussed. I will avoid too many different details and therefore omit the mechanisms to handle this.

5 Discussion

This paper is somewhat pessimistic concerning the building of a general NLI, as the mechanisms for handling indexicals differ depending on both task structure and type of background system. However, developing the various mechanisms is one problem and knowing which mechanism to use another. This second problem, I believe, can be solved by the use of Natural Language Interface Management System (NLIMS). Kelly (1983) created a natural language interface using Wizard of Oz experiments for customizing the lexicon and grammar for a calender system and Good, Whiteside, Wixon & Jones (1984) report similar ideas for developing a command language. My idea is to use a NLIMS for deciding the priority of the different discourse strategies. Such a system should have a component for morphological and syntactic analysis, a collection of different semantic components and a collection of mechanisms for discourse management. The language engineer first builds a prototypical interface using knowledge about the type of background system and also maybe some information concerning the future use of the system. Then he runs a series of simulations. These simulations are used for updating the grammar and lexicon automatically (Jönsson & Ahrenberg, 1989) and also for selecting the appropriate mechanisms for discourse management. The discourse handler can consist of more than one mechanism with different priorities, i.e. if the referent cannot be found using one mechanism another is tried. Of course the system must be updated after a number of runs which also could imply that new simulations need to be performed.

6 Acknowledgements

This research is very much related to the experiments I conducted with Nils Dahlbäck and as always I am indebted to him. Lars Ahrenberg read previous versions of the paper and forced me to sharpen my terminology. I am also obliged to Magnus Merkel and Mats Wirén for valuable discussions. But of course all remaining mistakes are my own.

References

Ahrenberg, L. 1989. A Constraint-Based Model for Natural Language Understanding and a Pilot Implementation, Research Report, LiTH-IDA-R-89-22, Linköping University.

Bobrow, R. J. & Webber, B. L. 1980. Knowledge Representation for Syntactic/Semantic Processing. *Proceedings of AAAI-80.*

Carberry, S. 1989. A Pragmatics-Based Approach To Ellipsis Resolution. *Computational Linguistics*, 15(2):75–96.

Carbonell, J. G. & Brown, R. D. 1988. Anaphora Resolution: A Multi-Strategy Approach. *Proceedings of the 12th COLING Conference.* Budapest.

Dahlbäck, N. & Jönsson, A. 1989. Empirical Studies of Discourse Representations for Natural Language Interfaces. *Proceedings of the Fourth Conference of the European Chapter of the ACL.* Manchester.

Good, M. D., Whiteside, J. A., Wixon, D. R. & Jones, S.J. 1984. Building a User-Derived Interface. *Communications of the ACM*, 27(10):1032–1043.

Grosz, B. J. 1977. The Representation and Use of Focus in Dialogue Understanding. Unpublished Ph.D. Thesis. University of California, Berkely.

Grosz, B. J. & Sidner, C. L. 1986. Attentions, Intentions, and the Structure of Discourse. *Computational Linguistics*, 12(3):175–204.

Guindon, R., Shuldberg, K. & Connor, J. 1987. Grammatical and Ungrammatical structures in User-Adviser Dialogues: Evidence for Sufficiency of Restricted Languages in Natural Language Interfaces to Advisory Systems. *Proceedings, 25th ACL*. Stanford, CA.

Hobbs, J. R. 1985. On the Coherence and Structure of Discourse. Report No. CSLI-85-37.

Jönsson, A. & Ahrenberg, L. 1989. Extensions of a Descriptor-Based Tagging System into a Tool for the Generation of Unification-Based Grammars. I. Lancashire [Ed.] *Research in Humanities Computing*. Oxford University Press, Oxford.

Jönsson, A. & Dahlbäck, N. 1988. Talking to a Computer Is not Like Talking to Your Best Friend. *Proceedings of the First Scandinavian Conference on Artificial Intelligence*. Tromsø, Norway. IOS, Amsterdam.

Kelly, J. F. 1983. An empirical methodology for writing User-Friendly Natural Language computer applications. *Proceedings of the CHI '83*.

Linell, P., Gustavsson, L. & Juvonen, P. 1988. Interactional Dominance in Dyadic Communication. A Presentation of the Initiative-Response Analysis. *Linguistics*, 26(3).

Litman, D.J. 1985. Plan Recognition and Discourse Analysis: An Integrated approach for Understanding Dialogues. Ph.D. thesis. TR 170, The University of Rochester.

Merkel, M. 1989. Temporal Structure in Discourse. *Proceedings of the Seventh International Conference of Nordic and General Linguistics*. Faroe Islands.

Reichman, R. 1985. *Getting Computers to Talk Like You and Me*. Cambridge, Mass., MIT Press

Reilly, R. 1987. Ill-formedness and miscommunication in person-machine dialogue. *Information and software technology*, 29(2):69–74.

Sidner, C. 1983. Focusing in the Comprehension of Definite Anaphora. M. Brady and R. C. Berwick [Eds.] *Computational Aspects of Discourse*:267–330. Cambridge, Mass., MIT Press.

Webber, B. L. 1988. Tense as Discourse Anaphor. *Computational Linguistics*, 14(2):61–73.

Natural Language Processing Laboratory
Department of Computer and Information Science
Linköping University,
S-581 83 LINKÖPING, SWEDEN
Internet: ARJ@LIUIDA.SE

JÖRGEN PIND

Computers, Typesetting, and Lexicography

Abstract

As part of the general strategy of computerizing the lexicographic work process at the Institute of Lexicography, we have adopted Donald E. Knuths typesetting program TEX as our typesetting engine. The main characteristics of the program will be briefly described, followed by a discussion of its advantages for lexicographic work.

TEX has already been used for the typesetting of a 1300 page etymological dictionary of Icelandic. A number of other projects are under way.

Special notice will be paid to the problem of coding as it relates to the making of dictionaries. The advantages of a generic, or logical, coding over typographic coding will be emphasized. However, doubts will be raised about the possibility of providing a set of tags which are completely neutral with respect to typographic considerations.

1 Introduction

In this paper I want to discuss one particular aspect of computational lexicography, namely the typesetting of dictionaries. This is perhaps not an issue which is central to computational lexicography, yet it is a subject which deserves study, especially now when the arts of typesetting have been moving onto the desktop. I will show you the approach we have adopted at the Institute of Lexicography, and remark on how it fits into our overall strategy for computational lexicography.

Let me begin, in all modesty, by quoting myself. In 1986 I was invited to give a talk at the NordData Conference in Stockholm. At that time we were just embarking on widespread use of computers at the Institute, and I attempted to draw up a schematic diagram of a 'Lexicographers' workbench' (see figure 1), commenting that a number of features had not been implemented. "This holds especially for the 'manuscript writer'. Our work has not yet reached the stage where this is in great demand, but we envisage the possibility of using the database to turn out manuscripts for a typesetting program like TEX." (Pind 1986:87).

Well, this was written before we even had a version of TEX running at the Institute! As a matter of fact, though we expected that typesetting would be

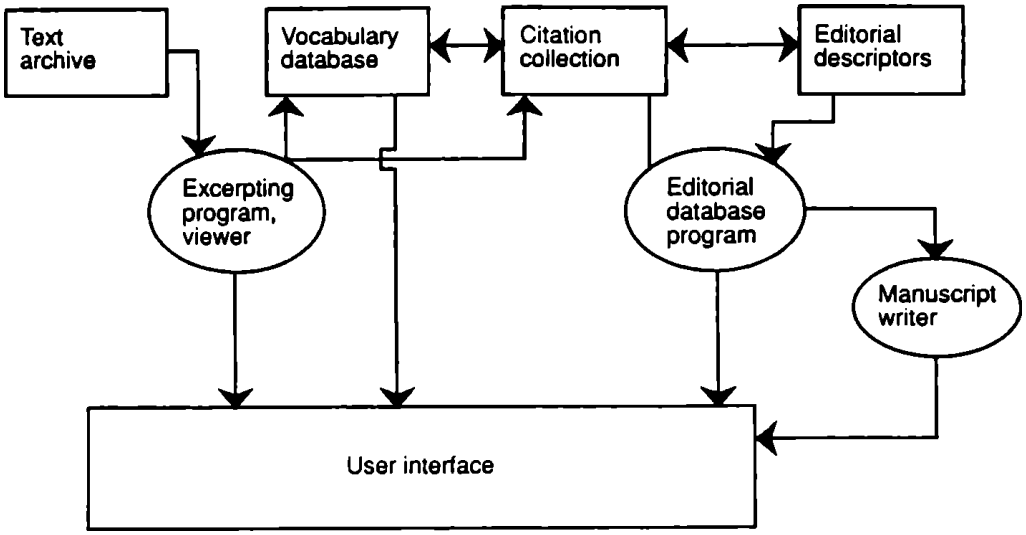


Figure 1: The lexicographer's workbench, 1986 vintage.

something that we would deal with much later, a lot of work over the past couple of years has been devoted to the typesetting side of lexicography.

There are two major reasons for this. The first is that the editor wants to be able to print proofs which are as closely related to the final form of the dictionary as possible. Thus a 'manuscript writer' has in fact been implemented as a feature of the 'workbench' we are currently working on. The relationship of this 'manuscript writer' to the work on the verbal dictionary has already been touched on in the paper by Björn Þór Svavarsson and Jörgen Pind in this volume.

The second reason is the fact that we have been engaged in producing Icelandic dictionaries from manuscripts, rather than from a database. Foremost among these is an Icelandic etymological dictionary by the late Ásgeir Blöndal Magnússon, former editor at the Institute, which will appear later this year.[1] We have also embarked upon a series of reprints of older Icelandic lexicographic works. These works have been coded in the TEX typesetting language.

2 Named Categories and Visual Formatting

In recent years a revolution has been taking place in the typesetting industry where numerous 'desktop publishing' programs have gradually been replacing the traditional tools of the printer. How far has this revolution affected the

[1] As a matter of fact, it was published on the 2nd of November 1989 as planned.

dictionary publisher and maker? I want to argue that such systems are not suited for the making of dictionaries.

Traditionally, dictionaries have been produced from collections of slips which have been used to ease the task of keeping the dictionary entries in alphabetical order and to allow them to expand as needed, without unduly affecting entries which follow alphabetically. The slips have then often been used, with minimal markup, as the manuscript for the printer. In the past few years attempts have, however, been made to use database systems to ease the arduous task of handling the collections of slips, with some success (cf. the paper by Björn Þór Svavarsson and Jörgen Pind in this volume). If we consider for a moment the nature of the database system, it is obvious that one of its major strengths is the fact that it allows the user to assign names or tags to the individual fields in the database. Thus we can easily imagine a database system for lexicographic work which knows about categories such as *headword*, *pronunciation*, *grammatical code*, *semantic field*, *usage notes*, and so on.

One of the typographical requirements for a dictionary is that *some* of these categories should be reflected in the typesetting itself. This shows for example in the use of different fonts in dictionaries, typically used to distinguish some of the categories. Note that only some of the categories will be thus reflected, since a typical dictionary contains many more categories than would be distinguished by typographic means. Some distinctions will thus be lost in the printed dictionary which are kept in the database systems.

Ideally, the lexicographer would like to use the database to automatically generate 'scripts' for typesetting, simply by instructing the database to print relevant typographic codes around some of the fields and not others. An even better approach would be to tag all the categories in the typesetting script and then instruct the typesetting system as to which ones should affect the typesetting process and which ones should not be reflected typographically. This latter approach is easy enough to accomplish if the typesetting system allows 'generic' or abstract coding of the input.

The desktop publishing systems mentioned at the beginning of this section do not allow such abstract coding (indeed very few of them are able to deal with traditional typesetting codes), since they are almost universally based on the idea of 'direct manipulation' or 'visual formatting'. The user manipulates a pointing device, such as a mouse, to mark parts of the text for, say, a font change. The notion of abstract coding plays no part at all in the formatting, and thus it is impossible in such a system to form a link between the categories of the database system and the typesetting. However, this is, of course, of the utmost importance for the lexicographer. A priori, I would have thought that this limitation of the desktop publishing systems would rule them out as being suitable for lexicographic work, and I was thus rather surprised when I came across the following description of the approach taken at the dictionary of Old English in Toronto.

> The typographical complexity of the dictionary entries—with a number of special characters, several languages, and many subsec-

> tions and cross-references which are distinguished by type—emphasizes the importance of interactive formatting. Because the working copy of the entry on the screen depicts the final appearance of a page, we hope to improve consistency...
>
> ...For example to put a keyword in bold in a citation, an editor can activate the area to be formatted by ranging over it with the mouse, and then use the mouse to select and apply the property bold from the Character Looks Menu (Healy 1985:248).

The system being described is a Xerox workstation, running publishing software similar to programs running on the Macintosh computer.

As mentioned earlier, this approach is severely handicapped by the fact that there is no easy way in a visual formatting system to form links to the categories of the database system being used. I would therefore like to argue that the requirements which need to be made of a typesetting system for lexicographic work are twofold.

- The typography should be of the highest order.
- The system must be able to work with generic or logical markup.

These requirements are met by a number of systems. We have chosen to work with TEX. In the following pages I will describe the way we have used TEX. While some of you are undoubtedly familiar with TEX, I will presume that not everyone is, and ask those knowledgeable to bear with me while I give a short tutorial introduction to TEX.

3 What is TEX? A Tutorial Introduction

TEX is a typesetting system 'intented for the creation of beautiful books' to quote TEX's author, Professor Donald E. Knuth of Stanford University. Those who have read his *TEXbook* will also know that the previous quote continues with 'and especially books that contain a lot of mathematics'.

TEX is indeed the premier system for typesetting mathematics available in the world today, so it is perhaps somewhat surprising to find it used for the making of dictionaries, indeed dictionaries which contain *no* mathematics at all! I will attempt to describe why we have found TEX to be eminently suitable for the typesetting of our dictionaries.

3.1 The Beginnings of TEX

It is perhaps rather surprising that we should be able to use TEX at all considering that it was created for one express purpose, viz. to allow Don Knuth to typeset his own magisterial treatise on the *Art of Programming* in what he felt would be an acceptable manner. These books started out being typeset in lead in the time-honoured manner of many generations of printers. When subsequently revisions of the original volumes were being prepared, the computer had made inroads into the field of typesetting and, to quote Knuth,

> ... when I received galley proofs they looked awful—because printing technology had changed drastically since the first edition had been published. The books where now done with phototypesetting instead of hot lead Monotype machines; and (alas!) they were being done with the help of computers instead of by hand (Knuth 1986f:96).

This was in 1977. This lead Knuth to temporarily abandon the project of writing the *Art of Computer Programming* while he would make up his own system for the typesetting, a task which he estimated would take about one year. In fact it took nine years of concentrated work to finish TEX and it's companion program METAFONT, which is a system for generations of letterforms.

The source code for the TEX system has graciously been put in the public domain by Knuth. The programs are written in WEB which is a special system for 'literate programming' (Knuth 1984b). A WEB program is processed by two programs. TANGLE makes a Pascal program from the WEB source which can then be compiled by a Pascal compiler, while WEAVE makes a TEX script from the same source, containing the source code with comments and detailed indices. Running this script through TEX produces a typeset version of the program. Knuth has thoroughly documented the TEX and METAFONT programs in his five volume work *Computers and Typesetting* (Knuth 1986a-e).

3.2 The Nature of TEX

TEX can be described as a document compiler or a typesetting language. Both terms require some clarification.

In the history of computer science, many computer languages have evolved. Some of these have been general purpose languages like Pascal or C, others have been specifically crafted for some particular task. TEX is an example of a special purpose language, and so is METAFONT. TEX as a language has primitive constructs which relate to the traditional art of printing.

The objects which TEX handles are 'boxes' and 'glue', to use Knuth's terminology (see figure 2). The smallest boxes which TEX manipulates are those surrounding the individual letters. Larger boxes can be built out of the undecomposable boxes surrounding the letters. Thus a line of type is also considered a box from TEX's point of view. Glue is the stuff which gets put between words and other boxes (though not between the boxes making up individual words). Leading, the distance between consecutive lines of type, is implemented in TEX through interline glue. This 'boxes and glue' model turns out to be surprisingly powerful and enables TEX to perform extraordinary feats of typesetting for example in the typesetting of mathematics.

Some of TEX's algorithms are quite well known. This is especially true for the paragraph setting algorithm (Plass and Knuth 1982), as well as the hyphenation algorithm devised by Frank Liang (Liang 1983).

The algorithm for setting paragraphs minimizes the 'demerits' associated with the setting of a particular paragraph. These demerits reflect, among other things, the 'badness' of individual lines of the paragraph which are calculated

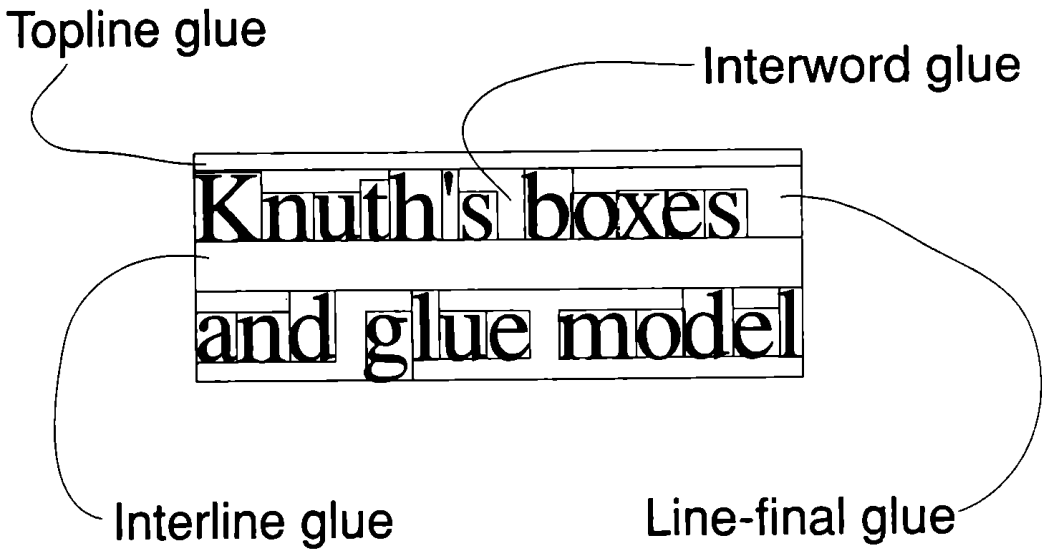


Figure 2: TEX's boxes-and-glue model

by noting the extent to which the inter-word glue has to stretch or shrink. TEX sets the paragraph by minimizing these demerits. The interesting thing to note is that this means that the paragraph as a whole is typeset in one go and a word coming late in a paragraph can influence the setting of lines coming earlier in the paragraph.

Liang's algorithm for word hyphenation is pattern-based, but departs from older versions by using both variable length patterns and patterns which both allow and inhibit hyphenation points. I will not discuss this any further here, but simply note that his method gives excellent results in a number of languages besides English. In particular the Icelandic hyphenation table does a very creditable job of hyphenating.

TEX has numerous primitives (around 300) for dealing with typesetting and also a very powerful macro programming language. It is this latter which gives TEX its status as a programming language.

Here are a few examples of the primitive operations which TEX operates with. Note that primitives and TEX macros are expressed with 'control sequences'. These usually start with a special 'escape character' which is typically \, the backslash.

- **\kern**. This command is followed by a dimension specification (e.g. in printers' points) and moves the placement of two boxes relatively to each other. Note that boxes come in horizontal and vertical versions and **\kern** can be used to position boxes both vertically and horizontally, depending on the 'mode' TEX is in. Usually kern is used to bring boxes closer together

(e.g., letter pairs like 'V' and 'A' which, because of kerning, are printed as 'VA' rather than 'VA'.

- `\looseness`. Changes to looseness mean that TEX will attempt to set a particular paragraph in more or fewer lines than the optimal setting calls for. By setting `\looseness=1` an attempt is made to open the paragraph and set it one line longer than would be the case if no `\looseness` is specified.

- `\fontdimen`. This command enables one to query the 'current font' for font parameters like the *x-height*, normal spacing, etc.

- `\penalty`. Controls the desirability of breaking at a particular point. Penalties can be both positive (making a break less likely) and negative (indicating desirable break points). Infinite penalties (having a value greater than 10000) either force (`\penalty=-10000`) or prohibit (`\penalty=10000`) a break at a particular point.

- `\sfcode`. The 'space factor code' is used to control the stretching of spaces after individual characters. Using the `\sfcode` makes it possible to, say, stretch spaces after periods more than after ordinary characters.

- `\hyphenchar`. Very few things are hard-wired into TEX. Even the hyphen# ation character can be changed. By setting `\hyphenchar\tenrm='\#`, TEX will use the hash-mark as the hyphenation character for the 10 pt Roman font (witness the first line of this paragraph).

3.3 TEX as a Programming Language

TEX has a very powerful macro language which can be used to write macros at almost any level of abstraction. The execution of these macros takes place through a process of macro expansion, where the macros are gradually reduced to primitives of the TEX language. Since macros can call other macros, it is possible to structure the code in a systematic way by gradually moving from primitive constructs to more abstract ones.

TEX observes a block structure, like most other programming languages. The block structure is achieved by using the symbols for 'open' and 'close group' which are usually the curly braces { and }. Using grouping, it is a simple matter to structure code, such that the likelihood of naming conflicts are lessened.

The TEX macro language, like most other macro languages, uses registers for the different 'data types' which are available. These registers come in five varieties:

Count registers are used for keeping integer values (32 bit). TEX has primitive operations for integer arithmetic only, but this is usually not a problem. The following piece of TEX code declares a count register named `\figno` which is initialized to 0:

```
\newcount\figno
\figno=0
```

The code for the figure macro would then take care of placing the figure and assigning a number to it which would be incremented for each figure. This last operation is achieved by:

```
\advance\figno by 1
```

Dimension registers are used for printers' dimensions, points, picas, millimeters, etc. The following piece of code declares a dimension register and then initializes it.

```
\newdimen\pagewidth
\pagewidth=170mm
```

Next come the glue registers or 'skip' registers. These contain glue specifications. The following example illustrates the definition of the `\smallskip` macro which makes use of the `smallskipamount` glue register:

```
\newskip\smallskipamount
\smallskipamount=3pt plus 1pt minus 1pt
\def\smallskip{\vskip\smallskipamount}
```

The `\smallskipamount` register is set to 3pt plus 1pt minus 1pt. The macro `\smallskip` is defined as a vertical skip (`\vskip`) of `\smallskipamount`.

Finally, we come to the box registers which are used for holding the boxes gradually acccumulated for each page. Boxes have three dimensions, as mentioned before. These can be queried or set, using the primitives `\wd`, `\ht`, and `\dp` for the width, height, and depth, respectively.

3.4 Defining Macros

We have already seen one example of how macros are defined. This is done with the `\def` primitive. Macros can take arguments, it is even possible to have macros which check for optional arguments, a higly useful feature. A typical macro with arguments is the following simple macro for setting headwords in bold face. (The percent sign % is usually a comment character in TEX. Anything coming after the % on a line is ignored by TEX.)

```
\def\hword#1{% macro for the headword
    {\bf#1\mark{#1}}}
```

This sets the headword in boldface (`\bf`) and defines a 'mark'. This mark can, for instance, be used to establish the range of entries on a particular page of a dictionary. The parameters are denoted by # and they are numbered consecutively, starting with #1.

Like any good programming language, TEX offers the user a conditional testing mechanism. One application of this is to print different types of proofs. For instance, it is possible to redefine the `\hword` macro in such a manner that TEX will write the headwords to a special file when the dictionary is being proofed. It

is then a straightforward matter to check whether the list of headwords thus generated is in correct alphabetical order. This can be accomplished in the following manner:

```
\newwrite\hwordfile % first a file is defined
\newif\ifproofmode  % A conditional is declared
\proofmodetrue      % Are we printing proofs? Yes we are.
\ifproofmode \message{**** Printing proofs *****}
\immediate\openout\outfile=\jobname.hwrd

\def\hword#1{% macro for the headword
    {\bf#1\mark{#1}}
    \immediate\write\outfile{#1}}
    ...
\else \message{**** Final run *****}
\def\hword#1{% macro for the headword
    {\bf#1\mark{#1}}}
\fi
   ...
```

The conditional construction

```
\if
   ...
\else
   ...
\fi
```

thus makes it easy to print different versions of the same manuscript according to need.

This has only been the briefest of introductions to TEX as a programming language, but it should, I hope, reveal to the reader something of the flavour of the TEX language.

3.5 TEX in Iceland

The Institute has been responsible for introducing TEX into Iceland. I have earlier described the steps undertaken to make TEX work with Icelandic (Jörgen Pind 1988). In particular:

- It was necessary to make a set of patterns for TEX to achieve correct (or nearly correct) hyphenation. The patterns were generated by Frank Liang's program PATGEN, using as input a 210.000 word dictionary made by the Institute for IBM in Iceland to use in IBM spelling checkers.[2]

[2] I am very grateful to Mr. Gunnar M. Hansson, general manager of IBM Iceland, for allowing us to use this material for this purpose.

- The Computer Modern Fonts had to be adapted to Icelandic by adding a few characters (e.g., 'ð' (eth) and 'þ' (thorn)).

- Changes had to be made to the standard macro collections to allow for new fonts and some differences in character definitions.

With these changes, TEX has been found to work admirably for Icelandic and has already been used to typeset a number of books. I guess Iceland must be unique in having brought out a number of TEXed books and yet no mathematics book has been typeset with the Icelandic version of TEX as yet!

4 Typography and Dictionaries

4.1 Some General Observations

The typesetting of dictionaries usually presents few problems. Dictionaries are usually set in two or three columns which are rather narrow. This can often lead to difficulties with line-breaking, since the narrow columns leave relatively little latitude for the paragraph-breaking algorithm. For this reason, it is advantageous to choose a font with a narrow set width, and, secondly, it is necessary to allow the typesetting program more flexibility in stretching and compressing interword spaces than is normal in books which are set to the full width of the page. In TEX this flexibility is controlled with the primitive `\tolerance`.

When the columns are set in register, as is usually the case, widow lines are bound to occur because the leading (interline glue in TEX) is not allowed to vary. These can be got rid of by stretching or shrinking the paragraph (or paragraphs on the previous page or pages). In TEX this is controlled by the `\looseness` primitive. If one is prepared to accept *full* widow lines (as we occasionally did in the etymological dictionary), it is possible to achieve this in TEX by setting the glue register `\parfillskip` equal to 0 pt, thus drawing the last line of a paragraph out to the full width of the column.

If the columns are not set in register (as is, for example, the case in the Oxford English Dictionary where the quotations are set in smaller type, thus forcing variable leading), it is much easier to control for widow lines since the space between paragraphs can easily be varied (this is done in TEX with the `\parskip` primitive).

It is custumary in dictionaries to print words at the top of the page, showing the range of the entries on that page. This process can very easily be automated in TEX, using the `\mark`. By `\marking` all headword entries and defining suitable macros for the outputting of the headlines, this process becomes completely automatic. Note that though I mention here the necessity of `\marking` the headwords, it is in fact *not* necessary to mark them individually. By a suitable definiton of the `\hword` macro this can be programmed (see the previous definitions of the `\hword` macros).

5 Work Finished and in Progress

The major performance test of TEX for lexicographic work was the typesetting of the etymological dictionary by Ásgeir Blöndal Magnússon. This book runs to 1231 two-column pages with forty pages of introductory material. TEX took care of the typesetting of all the pages except for two pages which contain illustrations demonstrating the use of the dictionary. These two pages were designed with a drawing program.

Originally, it was never intended that the etymological dictionary would be typeset with TEX. When keyboarding of the manuscript began in 1985, we did not have TEX, and the coding of the manuscript was such that it would be easy to transfer it to a printer for typesetting with traditional printers' typesetting codes. However, in January 1989, when we were ready to turn the manuscript over to the printer, it turned out that they did not have all the characters needed for the typesetting, and would also have difficulties with all the diverse floating accents which the book contains. At that point I decided to make some trial runs with TEX, using PostScript fonts (Adobe Times Roman). It turned out that no problems where encountered which could not rather easily be solved. Even the fact that PostScript has a fairly limited character repertoire could be remedied by drawing the missing characters with Fontographer, a font generating program running on the Macintosh (Altsys Corporation 1989).

Figure 3 shows a sample page from the dictionary.

Our major project in the future will, of course, be the dictionary of verbs outlined in the paper by Jón Hilmar Jónsson in this volume. The editing will take place in a database system, and the output of that system, a TEX script, will be generically coded.

Additionally, we have just embarked on a project to reprint some older Icelandic lexicographic works. Work is now in progress on four older dictionaries. These are all coded in the TEX language, and the intention is to bring these out in new editions. These are dealt with as textual objects, though the generic coding would, of course, considerably ease the task of putting them online, if that should be decided at a later stage (cf. Alshawi et al. 1989).

6 Issues of Coding

In recent years, more and more attempts have been made to use database systems for the creation of dictionaries. When a database is used for a dictionary, it becomes possible to *name* the fields which are being entered. The database programmer has quite a lot of freedom in the choice of these names and therefore in the choice of categories which are dealt with in the dictionary. I shall assume here that the final aim of the project is to produce a printed dictionary, though, of course, if it is made up using a database system it becomes possible to 'publish' it in computerized form, say, on a CD-ROM disk.

hreyfingu í leðju eða for, sbr. *lóna* af *lón*. Sjá so. *öðla*.

ádess. ⊙**ádeis** h. (18. öld) 'óhreinindi; óhapp; ádrepa'; af fs. *á* og *dess* af so. *dessa* (< **det(t)sa* < **dantisōn*), sbr. *að dessa niður á e-m* 'þagga niður í e-m' og *dessast* 'saurgast, versna'. Eiginl. 'það sem dettur á e-n eða skellur á e-m'. Sjá *dess*.

aðili k. 'hlutaðeigandi'; **aðild** kv. 'hlutdeild', sbr. *sakaraðild*, *réttaraðild* o.s.frv. Orð þessi lutu í öndverðu að skyldu og rétti ættingja (eða tengdamanna) í málaferlum, sk. *aðal* (1) og *aðall*.

Aðill k. fnorr. karlmannsnafn, sbr. *aðall* og *aðili*.

Aðils k. karlmannsnafn; sbr. sæ. *Adils*, sæ. rúnar. *Aþisl* < **Aðgísl*, fe. *Eadgils*. Forliðurinn *að-* á skylt við *aðal-* (2) og *óðal*, sbr. fsæ. pn. *Adi*; um viðliðinn sjá *gísl* (1).

adíu, **adjö** uh. (18. öld) 'kveðjuorð'. To. úr d. *adjø* < fr. *adieu* < *a Dieu*, eiginl. 'guð veri með þér'.

admíráll, **aðmíráll** k. (nísl.) 'sjóliðsforingi'. To. úr d. *admiral* < ffr. *a(d)miral* (s.m.) < arab. *amir* 'höfðingi'. Sjá *emír*.

Adólf k. karlmannsnafn; tökunafn, líkl. ættað úr þ., sbr. nhþ. *Adolf*, fhþ. *Athalwolf*, *Athulf*, gotn. *Athaulfs*; líkl. < **aþa-wulfaz*. Sjá *aðall* og *úlfur*.

adressa kv. (19. öld) 'heimilisfang'; **adressera** s. 'skrifa heimilisfang, ...'. To. úr d. *adresse*, *adressere* ættuð úr fr. *adresser*, sbr. lat. *ad* 'til' og *directum* (l.h.) 'beint'.

aðsjáll l. 'níðskur, naumur í útlátum' < **at-séall*; e.t.v. leitt af gamalli forskeyttri so., sbr. gotn. *atsaihwan* 'gaumgæfa' og ísl. *sjá að sér*.

-aður, †**-aðr** k. viðsk. no. eins og *munaður*, *unaður*. Skiptist á við *-uður* (s.þ.) og er komið af germ. **-ō-þu-*. Þetta viðsk. er runnið af verknaðarviðsk. **-þu-* < ie. **-tu-* sem skeytt var við stofn ō-sagna. Víxl *-að-* og *-uð-* eru upphaflega háð sérhljóði eftirfarandi endingar, t.d. nf. et. **-aþuʀ* > *-uðr*, en ef. et. **-aþaʀ* > *-aðar*, og gegndu þessar tvær myndir viðsk. í upphafi sama hlutverki, en síðar hefur *-að-* verið að mestu sérhæft í verknaðarmerkingu, en *-uð-* að mestu í gerandmerkingu. Sjá *-uður*, *-naður* og *-nuður*.

aðventa kv. 'jólafasta'. To., komið úr lat. *adventus* 'koma', ɔ: koma eða fæðing Krists í heiminn.

aðventistar k.ft. kristinn trúflokkur; nafngiftin lýtur að trú þeirra á endurkomu Krists.

aðvífandi lh.nt.: *koma a.* 'koma að eins og af tilviljun'. Sjá **vífa* (2).

1 af fs. (ao.) 'frá, burt'; sbr. fær., nno. og sæ. *av*, d. *af*, gotn. *af*, fe. *æf*, *of*, fhþ. *ab(a)*, lat. *ab* (< **ap*), gr. *ápo/apó*; sk. *afar*, *afr* (2), *aftur*, *at* (4), *efja*, *eftir*, *efsa*, *öfund*, *öfugur* og e.t.v. *aftann*. Sjá *af-* (2).

2 af- forskeyti; sbr. fær., nno. og sæ. *av*, d. *af-*, gotn. *af-*, fe. *of-*, fhþ. *ab-*, *aba-*, *abo-*, lat. *ab-*, gr. *apo-*, fi. *apa-*. Sjá fs. *af*. Ýmist gamalt forskeyti eins og t.d. í *afbragð*, *aflát*, *afráð*, *afrek* o.s.frv. eða síðar forskeytt fs. eða ao., sbr. t.d. *afdráttur*, *afhýða*, *afrækja* o.fl. Forskeytið heldur oft eiginlegri (staðarlegri) merkingu sinni, sbr. t.d. *afbjarga*, *affjalla*, *afhús*, *afhvarf*, en stundum verður tákngildi þess niðrandi eða herðandi, t.d. *afgelja*, *afgera*, *afát* 'ofát', *afgamall*, *afkostir*, *afstopi* 'ofstopi', eða meira eða minna óeiginlegt, t.d. í *afráð*, *afrek*.

áfa kv., merking ekki fullljós, en líkl. 'fjandskapur, mein', sbr. físl. *iǫll ok ófu / færik ása ſonum* (Lokas.). Sumir telja að *áfa* sé í ætt við lo. *æfur* og *ófa* kv., en stofnsérhljóðið, germ. **ē*, er annars óþekkt í þeirri orðsifi. Aðrir ætla að *ófu* (í Lokas.) sé eiginl. s.o. og *áfá* og þá < **ófo* < **áfǫ́*. Enn aðrir tengja orðið við *vofa* kv.; lítt sennilegt; *áfa* er stakorð og ritháttur ekki öruggur, e.t.v. stendur *ófu* fyrir *ófu* og orðið þá s.o. og *ófa* og tengt lo. *æfur*. Allt óvíst.

áfá kv. (18. öld) 'áhrif, t.d. af vínanda', sk. *áfengur* l. 'sem hrífur á'; **áfengi** h. 'vínandi' og *áfang* h. E.t.v. < **anfa(n)hō* dregið af forskeyttri so. **anfa(n)han*, sbr. fhþ. *anafāhan* 'byrja' (eiginl. 'grípa á'), eða myndað af so. *fá* (1) eða öllu heldur samb. *fá á*.

áfang h. † 'átak, hnjask, ofbeldi'; e.t.v. leitt af forskeyttri so. **anfa(n)han* 'grípa í, byrja', sbr. fhþ. *anafang* 'átak, hrifs, byrjun'; sk. *áfá* og *áfengur*. Sjá *fá* (1).

áfangi, †**áfangr** k. Sjá *áivangr*.

afar ao. 'mjög', einnig forskeyti **afar-**, sbr. **afarkostir**; líklega sama orð og gotn. *afar* 'á eftir, síðar', fhþ. *avar*, *abur* 'aftur', sbr. nísl. **afur-** (< **afr-*) sem notað er sem forskeyti í líkri merk. og *afar-* (*afurverði*, *afurnagandi*) og *af-* (2) sem stundum er haft í herðandi merkingu, t.d. *afkostir* s.s. afarkostir, *afgamall* 'mjög gamall'; *afar* sýnist vera einsk. miðstig af fs. eða ao. *af*, sbr. fi. *ápara-* 'aftari, síðari'. Aðrir telja að *afar* sé sk. gotn. *abrs* 'sterkur'. Sjá *afr* (2).

af-baka s. (16. öld) 'aflaga, skekkja'; sbr. nno. *avbakleg* 'öfugsnúinn, óhægur, erfiður, afskekktur', *avbekt* 'þver, öfugur', sæ. máll. *dbäklig* 'luralegur, ólögulegur', fær. *avbekladur* 'illa troðinn, aflagaður (um skó)'. Myndun orðsins er óljós, þótt það sé sýnilega tengt no. *bak*. F.J. (1914) ætlar að það merki í öndverðu 'að bakfletta trjávið, höggva ávala af trjám' og styðst þar m.a. við umsögn B.H., en það samræmist lítt merkingu og formi nno. og sæ. orðmyndanna. Sjá *bak* og *bekill*; ath. *bækill*. **-baldi** k. (nísl.) 'ofsafenginn maður', sk. *baldinn* l. og *ofbeldi* h. **-bragð** h. 'e-ð frábært'; sbr. nno. *avbragd* og fær. *avbragd-* í *avbragdsstyrki* 'mikið afl'. Leitt af so. **ab-bregðan* eða *bregða af*, sbr. *afbrugðinn* 'frábrugðinn, ólíkur' og *afbrúðig(u)r*. **-brúðig(u)r** l., **af-brýði** (†**af-brygði**) kv. Sjá *ábrúðig(u)r*. **-danka** s. (nísl.) 'svipta metorðum eða stöðu'; **-dankaður** l.

Figure 3: A sample page from the etymological dictionary

The traditional way of making a dictionary has been to proceed in a somewhat different manner, writing the dictionary entries on slips of paper.[3]

While the comparison between slips of paper, a file cabinet, and a database system is often made, this comparison is somewhat misleading since categories on the written sheets or slips are usually *not* named. In the case of dictionaries this is most clearly the case. An example will show this. Figure 4 shows a slip from the collection which was used in the making of the first standard dictionary of

letur (-urs, pl. ds.) [le:dɔǫ, lɛ:tɔǫ] n. **1. a.** Skrift, Typer: *gotneskt, latneskt l.; færa e-ð í letur,* optegne n-t, føre i Pennen; *sett l.,* en Slags Halvfraktur, nærmende sig til Schwabacherlypen. — ***b.** *leturs land,* Papir (BóluHj. 255); *letra rolla* (egl. Typefaar) (BóluHj. 217) = *prentsmiðja.* — **2.** Indskrift: *l. á steini.* **-band** [-r-ban·t] n. Forkortelse, Abbreviatur. **-breyting** [-brei:diŋk, -brɛi:t-] f. Udhævelse. **-gerð, -gjörð** [-ɣer·ð, -ɣör·ð] f. **1.** Bogstavskrift, Typernes Karakter: *leturgerðin er alt önnur,* Typerne er af en helt anden Karakter. — **2.** Skrivning: *hvorugur þeirra hafði numið svo mikið í leturgjörð, að þeir mættu rita nöfn sín* (JThMk. 382). — **3. a.** *(samning rits)* Optegnelse, Affattelse af et Skrift.

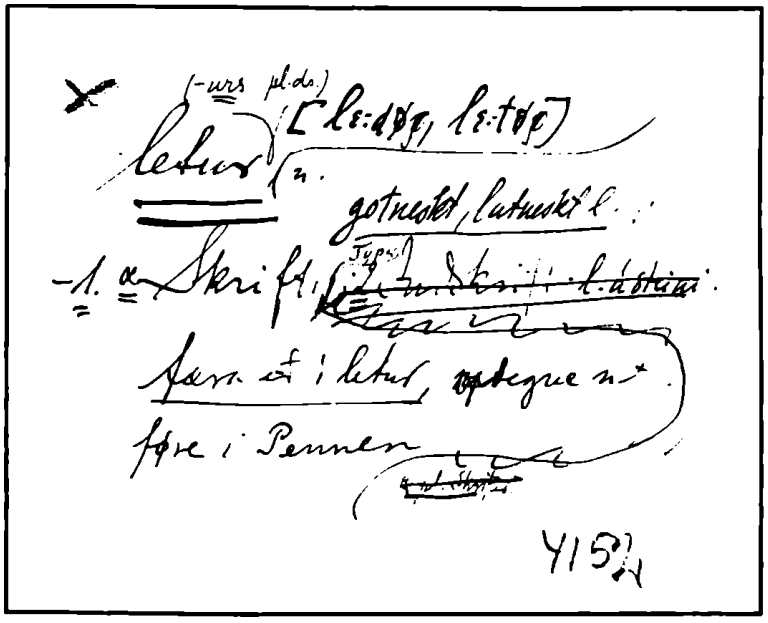


Figure 4: A sample entry from the dictionary by Sigfús Blöndal *and one of the dictionary slips on which it is based*

modern Icelandic, Sigfús Blöndal's Icelandic-Danish dictionary (Blöndal 1923). It is quite obvious that no categories as such are marked on the slip. They can, however, be *inferred* from the slip by the use of markings which indicate different fonts. The slip implicitly marks categories by the use of underlining and other typographical marks.

[3] Actually, using dictionary slips was quite a breakthrough in the making of dictionaries. This can be seen if one has a look at the 18th century monumental Icelandic dictionary by Jón Ólafsson from Grunnavík (which was never finished). This was written out as a single manuscript, at first having reasonable space between the entries, but gradually deteriorating into complete chaos as entries were added to the manuscript.

This approach is quite natural, considering that dictionary editors, such as Sigfús Blöndal, were working with the sole aim of producing a printed dictionary. They thought of their work as that of producing a *text*, and their approach was quite plainly a 'typographical' one where the only things they needed to keep distinct in the manuscripts were changes which would show up on the printed page, like font changes.

This approach has no doubt been almost universally followed, at least until quite recently. Some published dictionaries have been made available to researchers. These are generally typographically coded and bringing them online has often proved to be a formidable task (Alshawi et al. 1989).

This discrepancy between the database representation of a dictionary and the printed, typographical, representation is quite unfortunate and various steps have been taken to close the gap. This is currently not too difficult a task and I want to discuss here briefly how one could achieve this aim with TEX.

A programming language such as TEX makes it possible to code the manuscript at any level of abstraction which one finds most convenient. The primitives which TEX deals with are for the most part typographical ones, as already discussed. However, it is by no means necessary to use these primitives directly. Let me illustrate this by taking the entry from the Icelandic-Danish dictionary shown in figure 4 as an example. This can be typographically coded as follows in TEX (the phonetic transcription has been left out):

```
\bold{letur (-urs,} pl. ds.) [...] n. 1. a. Skrift, Typer:
\ital{gotneskt, latneskt l.; føra e-ð í letur}, optegne n-t,
føre i Pennen; \ital{sett l.}, en slags Halvfraktur,
nærmende sig til Schwabachertypen. --- \bold{*b.}
\ital{leturs land}, Papir (BóluHj. 255);
\ital{letra rolla} (egl. Typefaar) (BóluHj. 217)
= \ital{prentsmiðja}.
```

This example should be mostly self-explanatory. The instructions `\bold` and `\ital` change respectively to the bold and italic fonts. This representation is fairly close to the one given on the slips themselves, as depicted in figure 4. Note incidentally the somewhat strange use of fonts in the first line where parentheses do not balance correctly with respects to fonts. This use is probably quite natural for the printer (who has, after all, been taught that a delimiter character, for example, should belong to the same font as the preceding text). To someone accustomed to the notions of 'blocking' and 'environments' from computer science this manner of font change does seem illogical.

If we care to analyze the example from a functional perspective, we can easily see that it contains a number of different categories. There is the headword, which is printed in bold type, and so is the grammatical ending signifying the genitive. Here we have an example, ever so common in dictionaries, of one font being used for disparate categories. Additionally, there are examples of use and phrases shown in italic type, of sectioning (using numbers and letters of the alphabet), and of source references ('BóluHj.' being the Icelandic 19th century poet Hjálmar Jónsson).

A different way of coding would be to code the categories directly without any reference whatsoever to their typographical implementation. This approach, which has quite a short history, has been variously named 'logical' or 'generic' coding, and can thus be distinguished from the *visual* coding shown above. Generic coding has recently received increased attention through the standardization of the SGML (Standard Generalized Markup Language) (ISO 1986, Barron 1989, Bryan 1989). Similar concepts have been expressed in other languages and formatters, though SGML carries it to its logical conclusion: SGML is simply a manner of coding a manuscript, and has really nothing do do with typesetting, or database manipulation. It does, however, embody a manner of representing the structures which are to be found in a particular document.

In particular, as regards TEX, Leslie Lamport's macro package LATEX is very much geared towards logical coding (Lamport 1986; see also Lamport 1988). LATEX is a macro package used for general document processing. It uses the concept of separate 'style files' to capture the different formatting needs of reports, articles, books, etc. Furthermore, it defines categories such as 'titles', 'sections', 'chapters', 'footnotes', and so forth to express the different logical categories of documents.

The TEX macro language is such that one can easily implement macros to any degree af abstraction required. Using such an approach, it would be easy enough to code the above example from Sigfús Blöndal's dictionary in the following manner (I have formatted it here for easier readability):

```
\hword{letur} (\decl{-urs}, \xx{pl. ds.}) \phon{[...]}
\pos{n.}
\sense{1.}
 \subsense{a.} \trans{Skrift, Typer}:\V
  \exampl{\ic{gotneskt, latneskt l.; færa e-ð   letur},
  \da{optegne n-t, føre i Pennen}};
  \exampl{\ic{sett l.},
  \da{en slags Halvfraktur, nærmende sig til
             Schwabachertypen}}. ---
 \subsense{*b.}
  \exampl{\ic{leturs land}, \da{Papir} \source{BóluHj. 255};
  \exampl{\ic{letra rolla} \da{(egl. Typefaar)}
         \source{BóluHj. 217)}
  = \xrf{prentsmiðja}.
\sense{2.}
```

This, I hasten to add, is just a demonstration of the manner by which it would be possible to proceed. In particular, in no way is this coding based upon a study of the entries in this dictionary, a study which it would be necessary to undertake if it were desired to code the dictionary in this manner.

The categories mentioned above should be easy enough to understand since they have been given names which are fairly self-explanatory (the categories `\ic` and `\da` stand respectively for 'Icelandic' and 'Danish') and it will thus not be

necessary to give detailed explanations for each of them. It is, of course, immediately apparent that the manuscript gets considerably more complicated when such a system of coding is employed. After all, a lot of categories are delimited which will not find any particular realization in the printed text. By working from such a manuscript it is much easier to set up a one-to-one relationship with a database representation which of course is considerably more difficult when dealing only with a visually coded manuscript.

The astute reader will probably object to the choice of terms for the entries labelled `\sense` and `\subsense` in the above extract, since these only refer to numbers and letters and cannot strictly be said to denote the sense. This is, of course, true. In this case it would have been better to label the whole passage belonging to the particular sense, leaving out the numbers and letters and letting TEX assign these automatically. The point here is simply that it is possible to approach the task of coding in different ways, and it is difficult to specify once and for all a finite set of categories that will take care of all the entities one could conceivably want to code.[4]

One example will illustrate this. The etymogical dictionary, like all of its kind, contains *n* different accents which have to be coded for. In TEX, accents are expressed with special macros which make use of an `\accent` primitive. Thus one would write `\=a` to get 'ā', where the `\=` signifies a floating bar accent, or `\'a` to get á etc. But this command will not always give the correct result. Thus if one attempts to put an acute accent on top of a 'k' by writing `\'{k}` the result is ḱ. The correct version should look like 'ǩ'. This reflects a limitation of the `\accent` primitive in TEX which can be circumvented by writing special purpose macros for letters like 'ǩ'.

To obtain this effect it is necessary to write a special purpose macro in TEX. However, in that case, it is of course necessary to know about the fonts being used for typesetting. One of the major premises of generic markup is that such knowledge is not necessary, indeed it is not necessary to know how the text will eventually be used, say, whether it will be printed or put into a database.

6.1 Visual Coding and Direct Manipulation

The approach to coding which has been described here, is language-based and thus contrasts very much with the 'direct manipulation' approach which has in recent years been popularized especially on the Macintosh computer. As regards typography, the direct manipulation approach entails that the user points to or 'clicks' on words or letters on the screen and then typically chooses the relevant font from a menu. This was the pattern of usage which was embodied in MacWrite, the archetypical Macintosh word-processing program. The effects of the font changes could be immediately seen on the screen, in a WYSIWYG 'What you see is what you get' representation. The user interface was immediately hailed as a breakthrough, which of course it was, and yet, as time has shown, it has its problems. This can be seen in the evolution of word-processing programs

[4] I guess The DANLEX Group (1987) would want to argue differently, since they have attempted to provide a taxonomy of all the different categories which can occur in a dictionary.

for the Macintosh which tend to move them closer to a language-based representation. Thus the notion of 'style sheets', an idea borrowed from Brian Reid's program *Scribe*, has now been carried over into almost every word-processing program for the Macintosh (Reid and Walker 1980). Using style sheets, it becomes possible to mark sections in a semi-generic or logical manner. Unfortunately the notion of style sheets only applies to paragraphs, and is thus useless for the making of dictionaries where one is mainly interested in categories at a much finer granularity (i.e. sub-paragraph categories).

As demonstrated in this paper, a language-based formatter like TEX can easily be accommodated to a manuscript generated from a database and thus it can deal with categories at any level. I can state without hesitation that our experience using TEX has shown that it is eminently suited for lexicographic work.

References

Alshawi, Hiyan, Bran Boguraev, and David Carter. 1989. Placing the Dictionary On-Line. Bran Boguraev and Ted Briscoe [Eds.]. *Computational Lexicography for Natural Language Processing*:41–63. Longman, London.

Altsys Corporation. 1989. Fontographer, Users's Guide. Plano, Texas.

Barron, David. 1989. Why use SGML? *Electronic Publishing*, 2(1):3–24.

Blöndal, Sigfús. 1923. *Íslensk-dönsk orðabók*. Reykjavík.

Bryan, Martin. 1988. *SGML: An Author's Guide to the Standard Generalized Markup Language*. Wokingham, Addison-Wesley.

The DANLEX Group. 1987. *Descriptive Tools for the Electronic Processing of Dictionary Data*. Lexicographica, Series Major, 20. Max Niemeyer Verlag, Tübingen.

Healy, A. diPaolo. 1985. The Dictionary of Old English and the Final Design of its Computer System. *Computers and the Humanities*, 19:245–249.

ISO. 1986. International Standard 8879: Standard Generalized Markup Language (SGML). s.l.

Knuth, Donald E. 1984a. Literate Programming. *Computer Journal*, 27(2):97–111.

Knuth, Donald E. 1984b. *The TEXbook*. Addison-Wesley, Reading, Massachusetts.

Knuth, Donald E. 1986a. *TEX: The Program*. Computers and Typesetting, vol B. Addison-Wesley, Reading, Massachusetts.

Knuth, Donald E. 1986b. *The* METAFONT*book*. Computers and Typesetting, vol C. Addison-Wesley, Reading, Massachusetts.

Knuth, Donald E. 1986c. METAFONT: *The Program*. Computers and Typesetting, vol D. Addison-Wesley, Reading, Massachusetts.

Knuth, Donald E. 1986d. *Computer Modern Typefaces*. Computers and Typesetting, vol E. Addison-Wesley, Reading, Massachusetts.

Knuth, Donald E. 1986e. Remarks to Celebrate the Publication of Computers and Typesetting, *TUGboat* 7:95–98.

Lamport, Leslie. 1986. LATEX. *A Document Preparation System*. Addison-Wesley, Reading, Massachusetts.

Lamport, Leslie. 1988. Document Production: Visual or Logical. *TUGboat* 9:8–10.

Liang, Franklin M. 1983. *Word Hy-phe-na-tion by Computer.* Report STAN-CS-83-977. Stanford University, Department of Computer Science.

Pind, Jörgen. 1986. The Computer Meets the Historical Dictionary. *Nordisk DATAnytt* 16(10):41–43.

Pind, Jörgen. 1988. Umbrotsforritið TEX. Íslenskun þess og gildi við orðabókargerð. *Orð og tunga*, 1:175–219.

Plass, Michael, and Donald E. Knuth. 1982. Choosing Better Line Breaks. Jurg Nievergelt, Giovanni Coray, Jean-Daniel Nicoud, and Alan C. Shaw [Eds.]. *Document Preparation Systems: A Collection of Survey Articles*:221–242. North-Holland, Amsterdam.

Reid, Brian K., and Janet H. Walker. 1980. *Scribe: Introductory Users's Manual.* [3. ed.] Unilogic, Pittsburgh.

Institute of Lexicography
University of Iceland
101 Reykjavík
Iceland
jorgen@lexis.hi.is

BJÖRN Þ. SVAVARSSON & JÖRGEN PIND

Database Systems for Lexicographic Work

Abstract

At the Institute of Lexicography of the University of Iceland work revolves around very large collections of data in computers. There are mainly two types of databases which are kept in computer storage.

The first one is a relatively simple database containing the whole vocabulary of the main collection of the Institute, along with the age, number of citation, word class, word type, and oldest citation for each word. For maintaining this database there is no need for a very complex or powerful database system, but it must be fast since it contains over 600,000 words.

The second database is much more complex. It contains the lexicographic analysis and is used to construct the dictionary itself. A system like this must be more "intelligent" and more flexible than the first one, but speed is not as important a feature.

This paper describes some of the properties of the database systems we have been working on under MS-DOS and UNIX at the Institute.

1 The Background

The Institute of Lexicography at the University of Iceland was established in 1947 with the major purpose of making a historical dictionary of the Icelandic language, covering the period 1540 to the present. Over the past four decades, major effort has been put into the excerption, building up a collection of some 2.6 million citations. The state of excerption is now such that it is 'complete' for the 16th, 17th and 18th centuries, 'fairly complete' for the 19th century, and 'incomplete but substantial' for the 20th century. The collection encompasses a vocabulary in excess of 600,000 words.

When confronted with such a massive amount of material, the question arises as to what would be the natural way of proceeding with the editing. The first question one would presumably ask is: How *has* it been done? Well, the answer to that question is pretty well known. Traditionally, after the material has been collected, or at least a substantial part of it, the editor sits down with the first few hundred slips from the first box and tries not to think too much about the

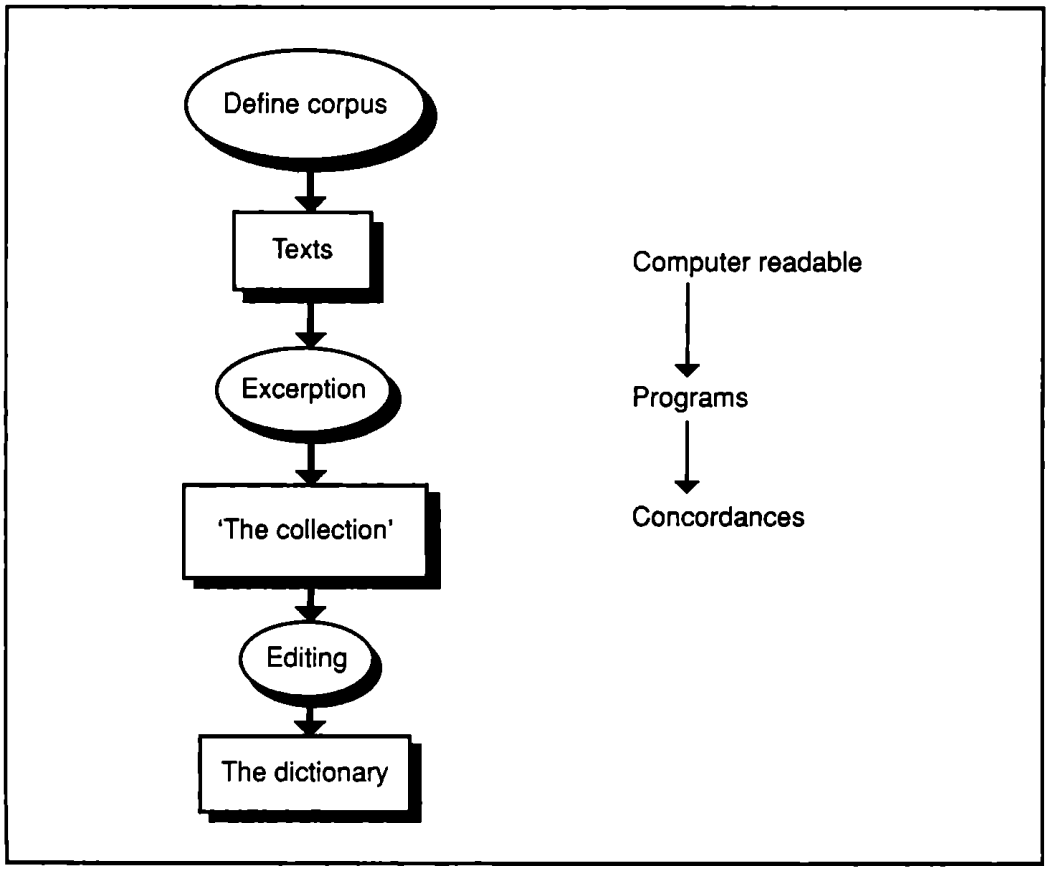


Figure 1: The procedure of making of historical dictionary

2.6 million slips awaiting! As material gathers, it is brought out in installments over a period of decades, sometimes even extending to a century or two. This procedure, almost universally adhered to in the making of historical dictionaries, is shown on figure 1.

We want to emphasize here that the historical dictionary, because of its enormous scope and sheer size, poses some problems which can conveniently be labelled 'order of magnitude effects.' They are not primarily problems of theoretical difficulty (although such problems are also abundant, as discussed in the paper by Jón Hilmar Jónsson in this volume), but simply problems which are related to the enormous bulk of these books. The major aim of this paper will be to discuss some ways in which we have attempted to deal with this problem. We would also like to emphasize that we are dealing with a project which is in some respects unique, since the whole editorial process will be computerized. This is in contrast to projects for computerizing existing dictionaries, e.g. the Oxford English Dictionary (Raymond and Tompa 1988) or the Svenska Akademiens Ordbok (Rydstedt 1988).

In 1983 a pilot editing project along traditional lines was undertaken. A reasonably full dictionary text was made up, containing words from a small subsection of the alphabet. By comparing this sample with a standard dictionary

of the Icelandic language (Sigfús Blöndal 1920–1924), it emerged that it would take close to two centuries to finish the editing using traditional methods and present levels of staffing. At the time this seemed to us an unacceptable way of proceeding. There were two reasons for this. On the one hand we felt the time scale to be absolutely unacceptable, on the other there are well-known problems with the traditional approach which we felt we could do something to solve by following another approach to the editing task.

2 A Different Strategy

Based on the experience gained in 1983, a new strategy for the editing of the historical dictionary has gradually been taking shape in the work carried out at the Institute. This strategy stands in rather sharp contrast to earlier methods and approaches. It is characterized by heavy reliance on computational tools. Indeed, we think it would be fair to say that without the existence of such tools this strategy would not be feasible.

The contrast between our new methodology and the traditional one can be captured with two terms borrowed from computer science. The traditional methodology can be characterized as **depth-first**, whereas ours comes much closer to being a **breadth-first** strategy.

A depth-first strategy means that every detail in the editorial process must be resolved to completion as the need arises. There is no chance of delayed commitment, no matter how much the lexicographer so desires! Since the editorial process turns out pages evenly and constantly, any revisions, which may be called for at a later date, may lead to the necessity of redoing the analysis, taking the manuscript apart, as it were, and putting it back together again.

How much more efficient it would be if it were possible to carry out the analysis independently of making up the manuscript. This is a goal not readily achieved, however, since the lexicographic analysis needs to be linked to the sources, i.e. the citations, in some manner. Traditionally, that link has been achieved by interspersing the analysis with quotations from the collection of citations.

It is precisely this reliance on alphabetical ordering, coupled with the enormous wealth of material which has to be accounted for, which gives rise to the aforementioned 'order of magnitude' effects. The following points are worth mentioning:

- There is a time lag of decades between the editing of different parts of the dictionary with many *generations* of lexicographers involved.

- The editor has a very *limited view* of the task at hand since he is labouring in a small corner of the dictionary.

- The alphabetical ordering of the text forces the editor to deal with material which is in no way related.

These problems are acute because of the size of the project. With a single volume dictionary, which takes perhaps 10 years to complete, it is relatively easy to bypass these problems. Not so with the historical dictionary. The research carried out at the University of Waterloo on the computerized Oxford English Dictionary has brought this out (Raymond and Tompa 1988). It turns out, for instance, if the cross-references of the OED are examined, that references to previous letters of the alphabet are much more frequent than those to subsequent letters. Most cross-references are, however, to words starting with the same letter. While the latter is to be expected (many cross-references are probably to closely related words), the former result can only be explained by the individual editor's limited view of the whole project.

A breadth-first strategy, of the sort now being developed at the Institute of Lexicography, proceeds with the editorial work from the top down, by making a number of passes through the citation collection, deepening the analysis at each stage.

There are numerous advantages to such an approach:

- Since the editing is computer-based it can be made available on the computer at an early stage.
- It is possible to deal with coherent parts of the vocabulary at any one time.
- It opens the way for defining significant phases in the work which can be finished in the relatively short time span of 5–10 years.

The first stage in the editorial process, which began in late 1983, involved a complete pass through the main collection, making up a database of the total vocabulary.

This database is of twofold use. In the first place it is of tremendous value for work in linguistics, especially those areas dealing with word-formation and morphology. The database has already been used to investigate the nature of compounding in Icelandic (Kristín Bjarnadóttir 1990). Such research is not only of theoretical importance, but is also relevant in the making of practical language tools on the computer, such as the ubiquitous spelling-checker. Spelling-checkers are usually dictionary-based. Dictionary-based checkers, however, have some limitations in most Germanic languages where the process of compounding is quite active. Some dictionary-based checkers now offer compound word parsing (and one Icelandic checker is completely based on the idea of word parsing). Due to the limited knowledge about compounding, however, it has generally not been possible to state the constraints which compounding obeys, and thus the mechanisms tend to overgeneralize wildly.

Secondly, the database is of great value for further lexicographic work as it, at once, opens up multiple search paths into the collection and frees the editor from the 'tyranny' of the alphabet (since a collection of written dictionary slips permits only one search key!). Now, however, the editor can access the collection on the basis of grammatical category, age of citations, oldest source, etc.

3 The Vocabulary Database

3.1 The Nature of the Database

As already mentioned, the first computational project involved a database covering the whole vocabulary of the main collection of the Institute. This database has 8 fields which can be briefly described as follows:

The word itself. The choice of words was primarily based on the orthographical form.

Word class. This, of course, has traditionally been indicated on the dictionary slips.

Age of oldest citation. The age is established to the nearest third of a century or to a greater time period if it is not possible to uniquely position the citation in this time frame (of 33 years). See also note below on the **source**.

Age of most recent citation. This is coded in the same manner as the **age of oldest citation**.

Number of citations. This is noted exactly for 1 to 5 citations. All words having more than five citations are marked as such with no finer distinctions being made.

Word type. This is the only type of information which cannot be read directly from the citation slips. We felt, however, that it would be advantageous to attempt a rough classification of the vocabulary according to word-type so words are marked as being compounded, affixed, or noncompounded.

Source. Finally, the source for the oldest citation is noted. This is often followed by a note detailing the exact age of the citation, e.g. a specific year. In fact, this field is built up of two subfields: abbreviation for the source and reference of page, exact age, etc.

Word in reverse. Part of the word is kept reversed in this field. This is done for the indexing of word endings.

Actually, while the Vocabulary Database makes up a relatively simple dictionary, it took quite a while to prepare. This was mainly due to the enormous vocabulary in the main collection. While the total number of citations is approximately 2.6 million, the total number of different words contained in the vocabulary database now stands at 608,205. This shows that the number of 'singletons' (words attested by only one citation) is relatively high, as noted by Jón Hilmar Jónsson (this volume).

Work on the database started in October 1983, we reached the 100,000 mark in May the following year, and keyboarding finished on the 7th of March 1986! The keyboarding was done on a Victor 9000 microcomputer using a specially made BASIC program.

After the keyboarding was completed, proofs were read, mostly in 1987. Due to the simple nature of the database, it was possible to perform numerous integrity and consistency checks with specially written programs. This considerably eased the onerous task of proofreading. The database reached its present form at the end of 1988, or five years after work on it was originally started.

Though we did not keep an exact account of the work involved, we would guess that it probably amounted to about 10–12 man-years. This gives some indication of the magnitude of the task of compiling a true dictionary of the material contained in the collections of the Institute.

3.2 The Computer Database

As already mentioned, the Vocabulary Database contains over 600,000 records at the present time. Putting the database online under the MS-DOS operating system (which has been our platform for most of this period) was never considered a viable option, since we felt that this operating system would have considerable difficulty in coping with a database of this size. This was one of the reason for a decision made late in 1987 to change to the Unix operating system.[1]

Our main computers are two IBM 6150 machines (perhaps better known as IBM RT/PC), running the AIX operating system. At present their total disk capacity is 840 Mb. We also have two IBM PS/2 machines running AIX PS/2 with 230 Mb of disk space. The RTs are connected by *Ethernet* and the PS/2s will also shortly be linked to the network. The *Network File System* (NFS) runs on top of the *Ethernet* connecting the machines.

The Vocabulary Database can conveniently be described by the relational database model. It consists of one main table, containing the records for all the words, along with a secondary table which contains information about the sources used to gather the citations for the main collection. These two tables are linked on the source fields as shown in figure 2.

There are a number of relational database managers available for AIX, among the better known are *Oracle* and *Informix*. Due mainly to considerations of cost, we decided to use the *Informix* database system. This is a fully relational system with the SQL query language. Unfortunately, it turned out that *Informix* has a nasty peculiarity[2] in that it does not allow fields with a variable number of characters (`VARCHAR`). Now this, obviously, makes life pretty difficult for the lexicographer!

For this reason the database grows to approximately 150 Mb when it has been indexed under *Informix*[3]. While the database is quite voluminous, it is also quite fast. It will instantly find any word and search on indexed keys is also fast.

[1] There were, of course, other reasons as well. Unix is well-known for its outstanding collection of tools, its preeminence in dealing with text files, the easy access to e-mail, etc.

[2] When compared with the description of relational database systems given, for example, by Date (1986).

[3] In textform it is about 30 Mb.

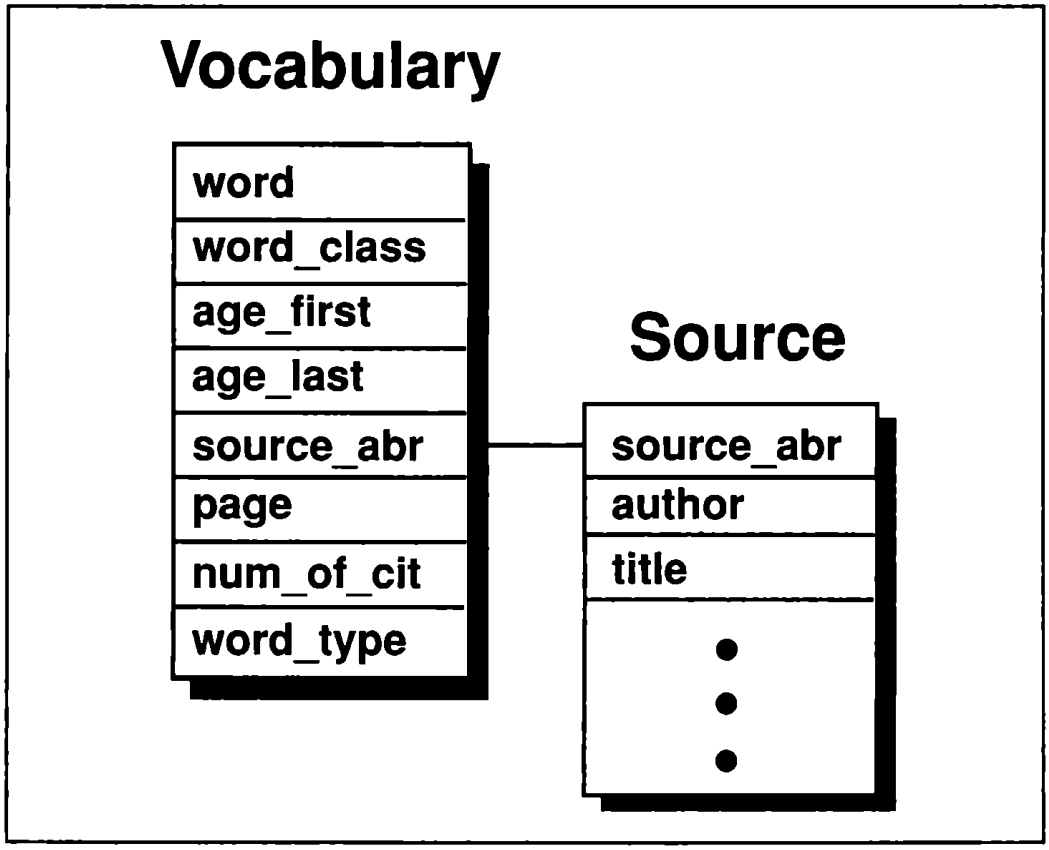


Figure 2: The tables of the Vocabulary Database as they are configured at present. The link on the source fields is shown.

3.3 Searching the Database

We will not particularly elaborate on the existing possibilities for searching the database. It is evident, from the description given so far, that we now have the possibility of searching the collection in ways not possible earlier. The following are examples of queries which have been put to the database by members of staff at the Institute, as well as by other researchers.

1. Find all words which occur for the first time in the works of the 19th century poet Jónas Hallgrímsson.
2. Find all adverbs ending in 'is'.
3. What is the proportion of verbs in the total vocabulary?
4. List all nouns for which there are more than five citations in the collection, with examples attested both in the 16th or 17th centuries and in the 20th century.
5. List all noncompound nouns for which first examples are attested in the first third of the 19th century.

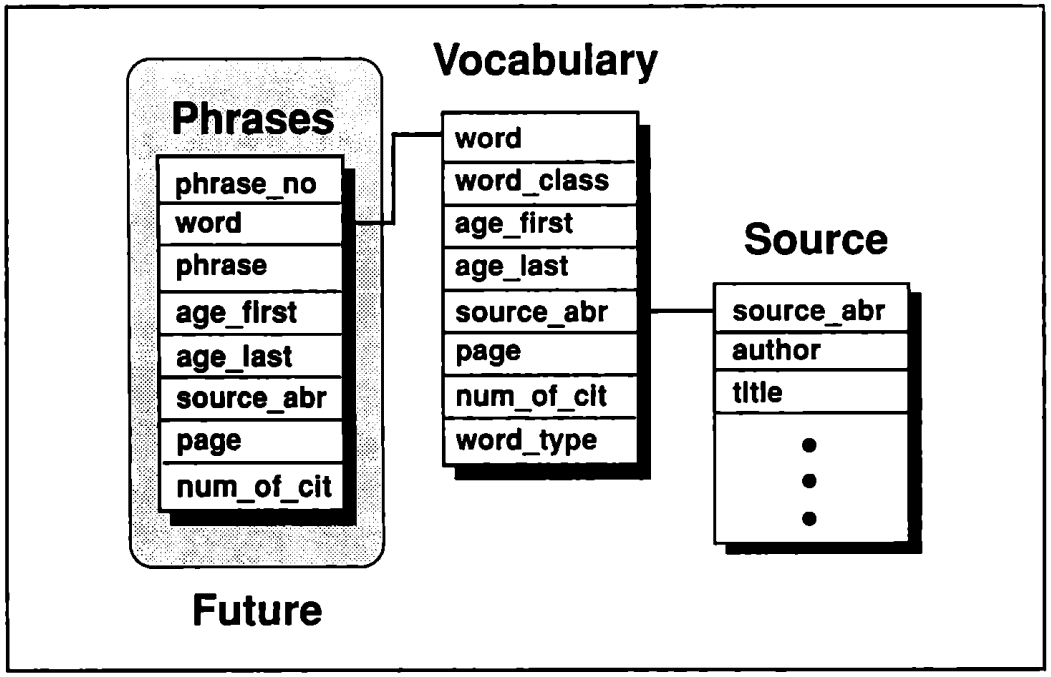


Figure 3: Relationship of tables in the Vocabulary Database if a table of phrases is attached to it.

3.4 Enhancing the Vocabulary Database

We have given some thought to the possibility of enhancing the Vocabulary Database with other kinds of information. As described by Jón Hilmar Jónsson in his paper (this volume), the main thrust of the editorial process concerns the description of verbs. Evidently, quite a lot of material which is of relevance to the verbs is filed under other word classes in the collection. This holds, for example, for phrases and idioms which are often filed under the noun rather than the verb. Some attempts were made by those collecting the citations to file them under all the relevant categories, but it goes without saying that in a task of this magnitude, stretching over decades, it is inevitable that various inconsistencies of practice will arise. While this extension has not been implemented, figure 3 gives an example of how it could be carried out.

4 Editing, Excerption: Computational Approaches

While material was being gathered for the Vocabulary Database, work was also being carried on in various other areas relating to the dictionary project as a whole. Four things in particular stand out:

1. The editorial process was begun in 1983 and gathered momentum during 1984 and 1985.

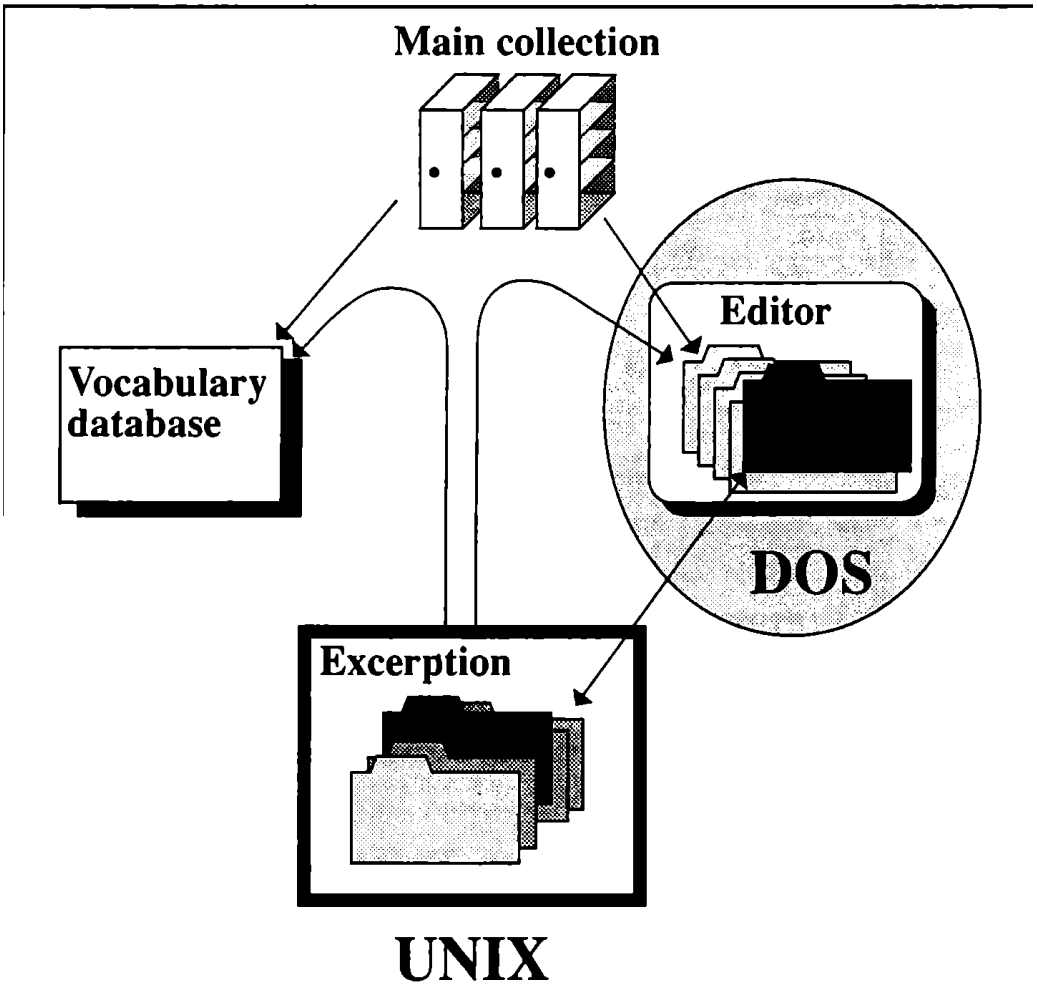


Figure 4: Computerized tasks at the Institute, as divided between MS-DOS and Unix at present.

2. The Institute started a collection af machine-readable texts. These texts can readily be used to augment the collections. It should also be noted that although most of the texts are modern texts from publishers and printers we have also endeavoured to acquire older texts. Some of these have been specially keyboarded at the Institute.

3. Excerpting for the main collection continues. All new citations are now entered directly to the computer.

4. We have adopted the TEX typesetting system for typesetting dictionaries. This is further discussed in the paper by Jörgen Pind (this volume).

Figure 4 shows in a schematic way all the major activities where computers have been brought to bear on the lexicographic work. This figure shows this activity as it is carried out at present under MS-DOS and Unix. We are presently

in the process of moving all these tasks to Unix. A number of points are worth discussing in some detail.

The editorial process was carried out using the *Revelation* database system for MS-DOS—which later became *Advanced Revelation* (Cosmos 1987). Revelation is a database system, based on the Pick Operating system which has enjoyed some success in the commercial environment (Rochkind 1985). The main reason for originally choosing Revelation was the absence of any (significant) constraints on the length of individual fields; using, as it does, completely variable length fields. The main characteristics of Revelation are summarized in the following:

The benefits of Advanced Revelation are:

- It is very flexible. It is easy to reorganize datafiles and reconstruct applications.
- It has a user-friendly interface.
- It has variable field lengths. Each field can range from 0 to 65 kilobytes, and predefinition of length is unnecessary.
- It is possible to define as many as 65k fields in each record, and the number of records in one file is only limited by disk space.
- It is possible to have many files open concurrently, and these can be related.
- It has its own procedural programming language (R/BASIC), similar to BASIC, but more structured.
- It has a powerful query language, similar to SQL.

The major disadvantages of Advanced Revelation are:

- It is too slow.
- It is only available on computers running MS-DOS which is a primitive operating system.[4]
- Since the system has only been available on computers running MS-DOS, disk storage is limited.

In spite of these disadvantages we have found Revelation to be a singularly useful product for lexicographic work, and it is with some sadness that we take leave of it now that we have moved over to Unix! The combination of variable length record fields with a very powerful programming language has turned out to be ideal.

[4] As of 1990 it is also available under OS/2.

4.1 The Nitty Gritty of the Editing System

We will now describe the editing system as it was implemented under *Advanced Revelation.*

As Jón Hilmar Jónsson has already described in his paper, the editing of verbs proceeds mainly with reference to the formal, syntactic and morphological behaviour. The editing is based on the citations and proceeds in a number of steps. In effect, it is possible to view the editing as being, to a large extent, a continuation of the excerption in that the citations are provided with grammatical markers. This is in complete contrast with the methodology traditionally employed (Kuhn 1982).

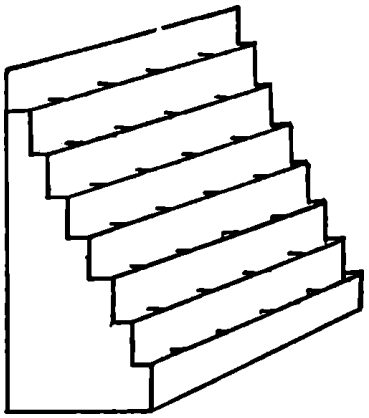


Figure 5: Kuhn's sorting board

Traditionally, the editing proceeds with the editor spreading all the slips attesting the occurences of a particular word out on a table, attempting to sort them manually into semantic categories. Kuhn has a nice illustration of an 'exceedingly useful' sorting board (shown in figure 5) which has been used to assist in the preliminary sorting done for the *Middle English Dictionary.*

It goes without saying that the strategy employed at our Institute is diametrically opposed to Kuhn's approach since the editing process starts out by augmenting the citations with various entries detailing such factors as conjugation, subject, object, meaning, etc.

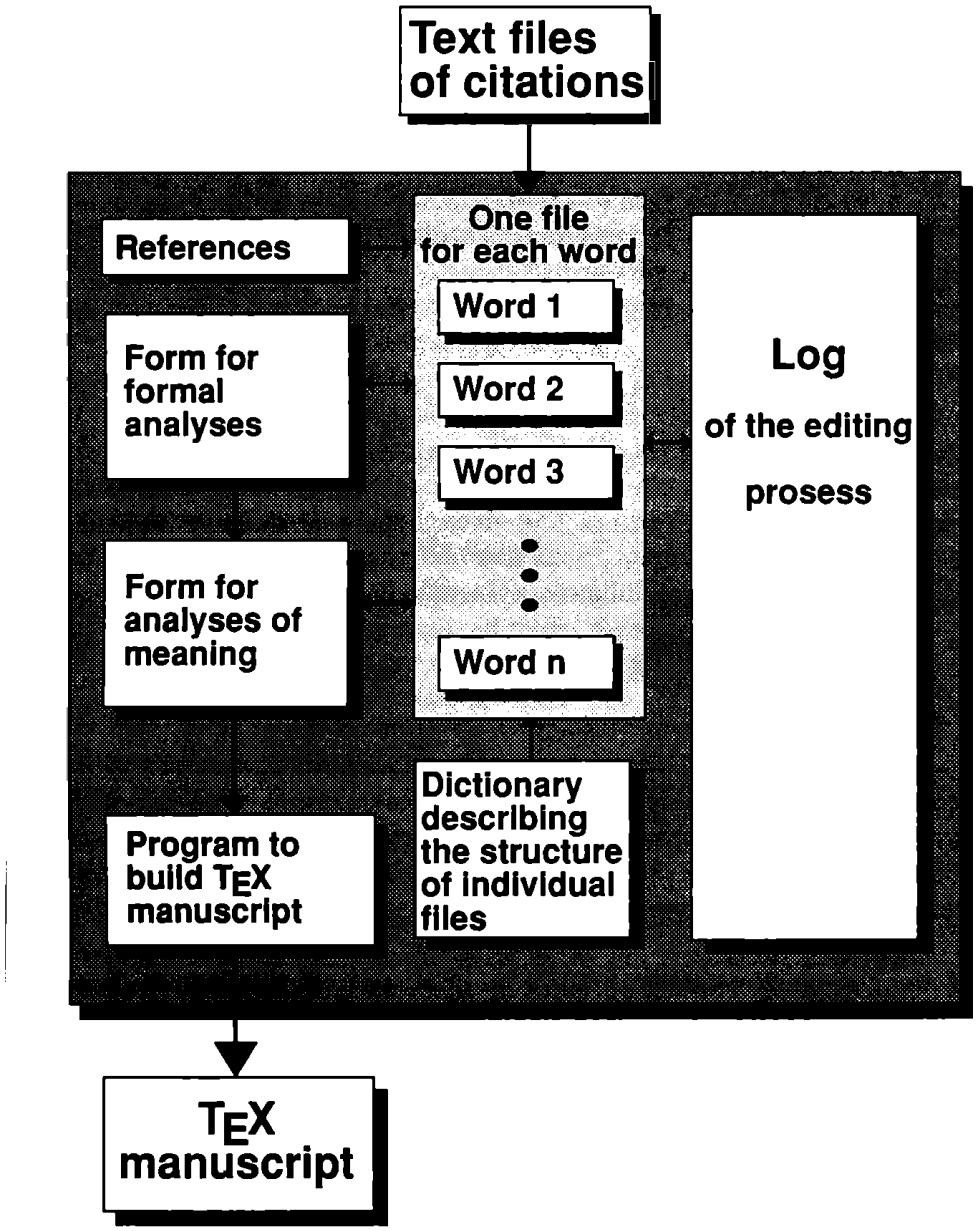


Figure 6: The structure of the Lexicographic Database as implemented under Advanced Revelation.

The method used in the editing is also reflected in the structure of the database program which is used for the editing. Figure 6 shows in a schematic way how the editing takes place. All citations are keyboarded by a secretary into files which are then imported into the database system. One file keeps a log of the editing process, noting which verbs have been analysed, the state of analysis, and other such information of a general nature. Furthermore, it contains

fields with detailed information associated with most of the citations. This file functions as a log of the editing process.

The citations for each word are kept in separate files which also contain the analysis. Jón Hilmar Jónsson, in his paper, describes the fields which are entered into these files and the editorial process as such. A special 'dictionary' contains information about the structure of the files. Each time a word is imported into the database the dictionary is consulted regarding the structure of the file which is to be made for the imported word.

Two screen input forms are used for the editorial analysis, one for the formal analysis, the other for the semantic analysis. When the editor starts on a particular phase of the analysis, he or she can use the query mechanism to arrange the citations in any desired order and browse through the collection while entering the analysis, thus dealing with similar citations at each point. After each pass through the citations, it is possible to rearrange the citations using the fields which have been entered. The command given is the `select` command. The following command is, for instance, used to order the verb 'koma' (come) on the fields *form*, followed by *voice*, followed by *conjugation*:

```
select koma by form by mynd by beyging
```

This possibility of ordering the citations prior to the analysis and at each subsequent step, is one of the main attractions of a database system such as *Advanced Revelation* as it ensures consistency of treatment and also speeds up the analysis considerably. Another advantage of Revelation is the fact that the entry forms have been defined in such a way that they carry over default values from the preceding citation. When the citations are sorted such carry-over defaults also serve to ease the data entry task and ensure consistency.

For the person responsible for the maintainance of the system it is very important that changes to the system, especially the entry forms, can be easily achieved. A special forms editor (which, incidentally, goes by the name 'painter') makes it very easy to change the layout of forms. It is also easy to change the structure of the files.

When the analysis of each word is finished the system will output a TEX-coded manuscript. This is done by a special program written in R/BASIC, Revelation's procedural language. This program uses a query command to select the citations which are marked as suitable for the printed dictionary.

To summarize briefly at this point:

- The editorial process augments the citations by attaching entries to the citations.
- These entries make up the sort key which is used to deliver automatically the major structural lines of each article.
- The database's "report writer" has been changed so that instead of producing the normal columnar reports it turns out manuscripts for TEX which can then be directly typeset.

The discerning reader is now probably wondering how we guarantee the integrity of our database, when on the one hand we have a collection of citations, and on the other hand citations contained within the editorial database. If errors are found in the editorial database, where will they be corrected, in the citation database, in the editing database, or in both places?

Obviously, this is a weakness of our approach as implemented in the Advanced Revelation database system of which we are fully aware. This, in fact, brings us right up to the present date as regards the development of our lexicographers "workbench". It is obvious that the computerized citation collection needs to be integrated with the editorial database.

5 Computational Lexicography under Unix

How should this database be implemented under Unix? It is quite clear that a database system like *Informix* is not at all up to this task since it cannot deal with fields having variable length records. Presumably the *Oracle* DBMS would be able to cope and this is already used for at least one large lexicographic database, CELEX in the Netherlands. However, Oracle is an expensive system, and, furthermore, other considerations have lead us to consider a different approach.

Our experience with TEX, which is a completely open system with all source code freely available, has brought home to us the importance of having programs which are available in source code form. If the program does not behave quite as desired one can always change the program code. This lesson has been reinforced with our experience of Unix, where an abundance of excellent software in source code form is also available. For this reason, we will be making a serious attempt to create a system where we can use available source code which is (preferably) available in the public domain (e.g. TEX and GNU) or commercially. The details of the actual implementation are currently being worked out, so here we will limit ourselves to an overall view of the toolset, as we have currently come to view it.

Perhaps the major concern for us is the following: Since the computerized citation collection is still being added to, we need some way of keeping access to the collection open, while at the same time using the citations, or at least a part of them, for the editorial work. Now, by referring to the description given earlier of the editing process as it was implemented in the Revelation database, it can be seen that quite a lot of the information which gets added to the citations in the editorial process is only dependent on the citation itself. As a matter of fact, this holds for 25 out of the 29 features analysed. It would, therefore, seem natural to detach these features from the editorial database and store them along with the citations. This would leave the citation collection intact and still allow us to progress with the editorial process. In this way, the formal analysis would be taken care of.

In this manner the citation collection, thus augmented, could be input directly to a sort routine similar to that illustrated earlier for Advanced Revelation.

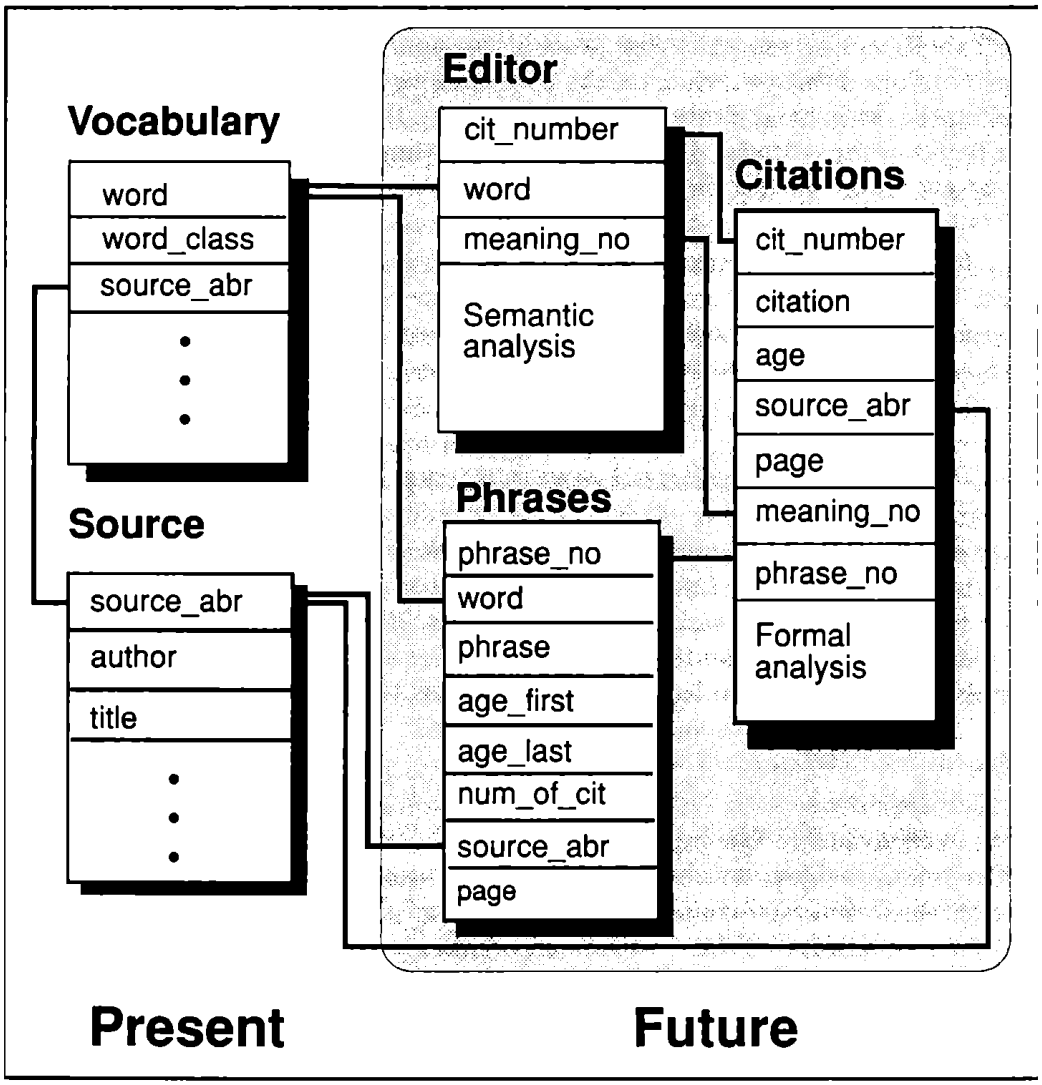


Figure 7: Our present ideas of how the lexicographic database can be implemented under Unix in the future

From this point on, we need to rely on a specifically made editorial program or a collection of programs for further analysis. Such a collection of program needs to be able to handle the following:

- It must keep a journal of the editorial process.
- As the citation collection grows, there must be some mechanism for notifying the editor that further citations have been added to the collection. These need to be incorporated into the editorial database.
- The generation of TEX manuscripts should be automated.

Work on the implemention of such a procedure has just been started under

Unix so it is not possible for us to elaborate in detail as to how the programs will be implemented, but figure 7 shows our present ideas.

References

Kristín Bjarnadóttir. 1990. *Um stofnhlutagreiningu samsettra orða.* BA thesis in preparation.

Sigfús Blöndal. 1920–1924. *Íslensk-dönsk orðabók.* Reykjavík.

Cosmos. 1987. *Advanced Revelation—Documentation.* 4 volumes. Cosmos, Seattle, Washington.

Date, C. J. 1986. *An Introduction to Database Systems.* Addison-Wesley, Reading, Mass.

Jón Hilmar Jónsson. 1989. A Standardized Dictionary of Icelandic Verbs. (This volume).

Jörgen Pind. 1989. Computers, Typesetting, and Lexicography. (This volume).

Kuhn, Sherman. 1982. On the Making of the Middle English Dictionary. *Dictionaries: Journal of the Dictionary Society of North America* 4:14–41.

Raymond, Darrel R., and Frank Wm. Tompa. 1986. Hypertext and the Oxford English Dictionary. *Communications of the ACM* 31:871–879.

Rochkind, Marc. J. 1986. Pick, Coherent and Theos. *Byte* 10(11):231–239.

Rydstedt, Rudolf. 1988. Creating a Lexical Database from a Dictionary. Martin Gellerstam [ed.]. *Studies in Computer-Aided Lexicology.* Data Linguistica, 18. Almqvist & Wiksell, Stockholm.

Institute of Lexicography
University of Iceland
Reykjavík 101
Iceland
bjorn@lexis.hi.is
jorgen@lexis.hi.is

ANNA SÅGVALL HEIN

Lemmatising the Definitions of Svensk Ordbok by Morphological and Syntactic Analysis. A Pilot Study

Lemmatizing Dictionary Definitions

Abstract

In this paper we present the results of a study of the definition vocabulary of *Svensk ordbok*. It is part of our on-going work on the generation of a machine-tractable dictionary from this dictionary, in specific, of making its definitions exploitable to a parser. Aiming, in particular, at the automatic lemmatisation of the definition vocabulary, the study includes an automatic morphological analysis of a subset of it, and an examination of its results. Two major issues were addressed, i.e. the coverage of the dictionary in relation to the definition vocabulary, and the feasibility of homograph resolution by syntactic analysis.

1 Introduction

The primary lexical source of the Swedish parser developed in the project *A Lexicon-Oriented Parser for Swedish, LPS*, (Sågvall Hein 1987a), is *Svensk Ordbok* 'A Dictionary of Swedish' (1986). It comprises 58,536 head words, lemmas, representing 65,568 lexemes. The *lemma* is defined as a group of word forms belonging to the same word class and the same inflectional pattern (Allén 1970: Introduction). The distinct senses that can be expressed by a lemma are referred to as *lexemes* (Allén 1981: 382).

Svensk Ordbok is drawn from a lexical data base, developed in the *Lexical Data Base, LDB*, project (Allén 1976). Thus, the lexical material is not only machine-readable, but also systematically organized in a flexible file handling system (Sjögreen 1988). The database was, however, primarily compiled for human use, thus presupposing linguistic background knowledge to be fully exploited. In other words, it is not readily *machine-tractable* in the sense of the word used by e.g. Wilks (1988).

The basic semantic information in *Svensk Ordbok* is provided by the *definitions of the word senses*, expressed in a subset of Swedish. In order to make this information useful to the parser, we have to make the implied linguistic knowledge explicit, and to formalize it. A step towards this goal is to parse the definitions, morphologically and syntactically. Here we will discuss the problems involved in lemmatizing the (graphic) words of the definitions automatically. The procedure involves two basic steps, i.e. an automatic morphological analysis, followed by a homograph resolution process, optionally carried out by automatic syntactic analysis. In the carrying through of such a scheme, aspects of the implied linguistic background knowledge will be brought up and concretized. In specific, the following questions are addressed:

- To what extent are the words of the definitions explicitly defined?
- To what extent are the homographies of the definitions solvable within the contexts in which they occur?

The results that we present are based on an automatic morphological analysis of a subset of the definition vocabulary.[1] First, however, we give a short description of the definitions and the underlying principles behind them.

2 The Definitions of *Svensk Ordbok*

In the semantic model developed in the *LDB* project and applied in *Svensk Ordbok*, every lexeme is supposed to have a well identifiable and relatively pregnant *kernel* meaning, around which a number of subordinate, derived meanings are grouped (Järborg 1988). The kernel senses of the lexemes are described by means of a *definition*, and, optionally, a *definition complement*, whereas for the derived meanings their relations to the kernel meaning are stated. The definition "is subject to the restriction that it should be capable of replacing the lexeme in question (the definiendum) in all syntactic or morphological contexts (sometimes with the help of some natural transformation). Given this restriction, the definition should describe the kernel sense in an analytic way, i.e. with each semantic factor being recognized during the analysis being assigned a separate word or phrase in the definition. The words of the definition should, as far as possible, be more semantically central in the lexicon than the definiendum, as a first step towards establishing an inherent defining vocabulary."(Järborg ibid.: 144).

The corpus of the definitions (disregarding definition complements) comprises 360,144 tokens and 43,184 inflectional types (distinct graphic words). This is the *primary corpus* of our study. In the pilot study reported on here, a subset of the primary corpus has been analysed comprising the 2,500 first (in alphabetical order) types, from *abborre* 'perch' to *behovet* 'the need'.

[1] By definition vocabulary, as opposed to *defining vocabulary* we simply understand the inventory of words actually used in the formulation of the definitions.

3 The Morphological Analysis

As a result of our previous work in the LPS project, we dispose of a morphological analyser of Swedish, comprising a stem dictionary covering the head words of *Svensk Ordbok*, along with a corresponding inflectional grammar (Sågvall Hein 1988; 1989). The parsing engine of the LPS parser is the *Uppsala Chart Processor, UCP* (Sågvall Hein 1987b).

```
AGAT :
        (* =   (WORD.CAT = VERB
              LEM = AGA.VB
              DIC.STEM = AGA
              INFL = PATTERN.ÄLSKA
              TENSE = SUP))
        (* =   (WORD.CAT = VERB
              LEM = AGA.VB
              DIC.STEM = AGA
              INFL = PATTERN.ÄLSKA
              PP = +
              GENDER = NEUTR
              ADJ.INF = STRONG
              NUMB = SING))
        (* =   (WORD.CAT = NOUN
              LEM = AGAT.NN
              DIC.STEM = AGAT
              INFL = PATTERN.FILM
              GENDER = UTR
              FORM = INDEF
              NUMB = SING
              CASE = BASIC))
```

Figure 1: An example of morphological analysis

The morphological descriptions generated by the LPS parser are represented as *Directed Acyclic Graphs, DAGs* (see e.g. Shieber 1986). Information on *word class*, *lemma*, *inflectional type*, and *dictionary stem*[2] is a compulsory part of each morphological description. For the rest, the information provided differs with the different word classes. For an illustration, we present an example of a morphological description generated by the morphological analyser (Fig. 1). The word *agat* (Fig. 1) has been recognized as a supine or past participle form of the verb *aga* 'flog', *internal homography*, or as the noun *agat* 'agate', *external homography*. However, with the automatic lemmatisation as the primary goal of the current study, internal homographies will, for the time being, be disregarded. In other words, *homography* in the following presentation should be understood as *external homography*. Likewise, when we present data on number of analyses below (Table 1), we disregard cases of internal homography.

As shown in Table 1, a single analysis was given in 73 percent of the cases, no analysis in 23 percent of the cases, and, finally, more than one analysis for a

[2]The information on dictionary stem is saved to be used as a tool in extracting a subdictionary for the syntactic analysis to follow; the information on inflectional type is included to provide a basis for subsequent statistical calculations.

Number of analyses	Absolute frequency	Relative frequency
0	572	23
1	1817	73
2	93	4
3	16	1
4	2	0
Total	**2,500**	**100%**

Table 1: Number of analyses resulting from the inflectional morphological analysis.

minority (5 percent) of the (graphic) words. Below we will examine the different cases, starting with analysis failure.

3.1 Lexical Gaps

The analysis failures are due to missing stems in the dictionary, excluding two cases of insufficient grammar coverage, and one case of an unforeseen use of parentheses (Table 2).

Type	Absolute frequency	Relative frequency
no stem: compound	452	79
no stem: derivative	80	14
no stem: proper noun	36	6
grammar coverage	2	< 1
orthographic convention	2	< 1
Total	**572**	**100%**

Table 2: Types of analysis failure.

Focusing for a moment on the missing stems, we have to state, that some 23% of the words in the definitions are not themselves head words in the dictionary, and, consequently, not explicitly defined. They are, basically, of two kinds, i.e. *proper nouns* and *derived words* (compounds and derivatives). The very fact that there are derived words missing in the dictionary should not surprise us (even if the high number of them did). A "complete" dictionary is theoretically impossible, due to, among other things, the rich potential for word formation, in specific, accidental compounding.

"Av samma skäl [utrymme] ges inte den triviala betydelsen hos en rad sammansättningar, avledningar och partikelverb. Denna betydelse täcks indirekt av ordboken genom beståndsdelarnas definitioner." 'For the same reason [space] the trivial meaning of a number of compounds, derivatives, and particle verbs is not given. This meaning is indirectly covered by the definitions of the constituent parts.' (*Svensk Ordbok* 1986: Preface). What can be objected to, however, is the

use of such words in the definitions. "Vidare har en strävan varit att hålla antalet ord som används i definitionerna relativt litet och att välja så enkla ord som omständigheterna medger." 'Further the aim has been to keep the number of words used in the definitions relatively small and to choose as simple words as the circumstances permit.' (ibid.). Here we won't discuss this issue further, but rather examine the derived words used in the definitions to see if they keep up with the *transparency claim*, i.e. if their meaning is derivable from the definitions of their constituent parts by means of a set of general word formation rules. The rules on which those derivations are based, are part of the background knowledge implied by the lexicographers, and as such of primary interest to us. Thus, if they fullfil the transparency claim, they should be formalized and included in the word formation component of the LPS machine dictionary as its first piece.

A subset of the implicitly defined words of the definitions are included in the dictionary as morphological examples. The very existence of the derived words is confirmed by the examples, but still, nothing is said about their use and meaning; they are found to be transparent. In all, there are 3,934 morphological examples. Once the word formation rules behind the derived words of the definitions have been formulated, the total of the morphological examples might be used as a test corpus for the resulting word formation component.

The missing *proper nouns* make up a much smaller number (36) than the derived words (532). Thirty-three of them name geographical units, and the remaining name persons. As long as the words that they define are included in the dictionary, we see no other solution than to enter them as head words and thus define them. A frequent type is made up of country names (e.g. *Algeriet* 'Algeria') used in the definitions of people from those countries (e.g. *algerier* 'Algerian': *person från Algeriet* 'person from Algeria').

Two words were not analysed due to limitations in the inflectional grammar. They represent a type that can be illustrated by the graphic word *ansikts-* 'face' in coordinated compounds such as *ansikts- och hårvård* 'face and hair care'. The phenomenon is regular and has to be taken care of partly by the morphological, partly by the syntactic grammar. In the first instance, the LPS inflectional grammar will be accordingly extended.

Short for *befordrats* '(has been) sent' or *befordras* 'is (being) sent' we find *befordra(t)s* among the definition words. This abbreviated form of expressing alternatives was not foreseen by the LPS inflectional grammar. Nor will it be, but the short forms of this kind will be spelled out.

3.1.1 Compounds and Derivatives

To keep up with the *transparency claim*, the word formation competence on which the understanding of (some parts of) the definitions is based, should be more stereotypical than what is generally the case. Further, we must require that the *constituent words are themselves defined*, and, that they are *not homographic* or *polysemous*. We begin our examination with the most dominant word

Type	Example	Absolute frequency	Relative frequency
N-N	*aktieägare* 'stockholder'	376	83
N-AP	*abborrliknande* 'resembling a perch'	22	5
N-PP	*arvsberättigad* 'entitled to an inheritance'	11	2
N-A	*alkoholfri* 'non-alkoholic'	18	4
Ab-N	*bakåtspark* 'kick backwards'	8	2
A-N	*allmängiltighet* 'universal applicability'	4	< 1
Ab-Ab	*akterifrån* 'from the stern'	4	< 1
Ab-AP	*bakomliggande* 'lying behind'	3	< 1
Ab-PP	*bakåtriktad* 'pointing backwards'	3	< 1
Ab-A	*antidemokratisk* 'antidemocratic'	2	< 1
A-A	*allmänkulturella* 'general cultural'	1	< 1
Total		**452**	**100%**

Table 3: Types of compounds.

formation type, i.e. the *compounds*. In Table 3 we present the distribution of the compounds by syntactic types.[3]

Among the 11 compound types that were found, the quite dominating one is the *N-N* type making up some 83 percent of the total number of compounds. The dominance of this type is in accordance with our intuition and with other studies of modern Swedish (see e.g. Blåberg 1988). Our figure is, however, considerably higher than that presented by Blåberg (68%). His figures are based on an investigation of a corpus of 3,971 compounds, identified in newspaper text from 1985. Our higher proportion of *N-N compounds* seems to support the transparency claim, indicating a restricted use of the variation offered by the word formation potential. Additional support is given by the parameter *number of different types*, whose value is much lower in our material than in that of Blåberg, i.e. nine[4] versus 22. Our types constitute a subset of the set identified by Blåberg, including, in addition: V-N, V-A, A-V, Ab-V, Numeral-V, Numeral-N, Numeral-A, Numeral-Numeral, N-Proprium, A-Proprium, Proprium-N, Proprium-V, Proprium-Proprium. It remains to be seen though, how stable our figures remain through out the material. We don't, however, expect to find such a rich variation of types involving proper nouns as does Blåberg, proper nouns in general, being outside the scope of *Svensk Ordbok*.

Among the 376 N-N compounds, there are in all 154 different first constituents (simplex or derived). The majority of them (146) are explicitly defined, seven of them implicitly, and one of them (*A4* in the compound *A4-format* not at all). Among those that are only implicitly defined, there is a dominating type, i.e. process nouns derived from verbs by means of the *-(n)ing* suffix, such as *avfyrnings-* 'firing' (cf. *avfyrningsmekanism* 'firing mechanism') of the verb

[3] AP is short for active present participle, and PP for passive past participle.

[4] This is the figure we arrive at if we adapt to Blåberg, who includes the active and the passive participles in the verb group.

avfyra 'fire'. It accounts for five of the seven cases. One of the remaining cases is in itself an N-N compound, i.e. *bastuba* 'bass tuba' (cf. *bastubeinstrument* 'bass tuba instrument'). The other one is an A-N compound, i.e. *andraklass* 'second class' as in *andraklassutrymme* 'second class area'. When one of the nouns of an N-N compound is in itself only implicitly defined, the derivation of the definition word has to proceed in two steps. We think that such a situation should be avoided, and the two constituents of a compound used in a definition both be *explicitly* defined.

Another problem with the derivation of meaning from the N-N compounds of the definitions concerns homography and polysemy. Among the first constituents of the N-N compounds we found four cases of homography, i.e. *akter, arm, babords*, and *back. Akter, babord*, and *back* are all nouns or adverbs, and *arm* is a noun or an adjective. For instance, *akter* is a noun 'stern' or an adverb 'aft', and *back*, a noun 'back, reverse' or an adverb 'back'. The word *akter* occurs in *akterdäck, aktermast*, and *aktervägg*, and in deciding whether the noun or the adverb interpretation of *akter* was the intended one, we were guided by the morphological example *akterdäck* presented under the (noun) head word *akter.* By analogy, we chose the noun interpretation in the other two cases, too. As regards *back* it occurs in *backverkan* 'reverse effect' and in *backåkning* 'downhill going'. No morphological examples are given, but intuitively *back* in *backverkan* should be understood as an adverb instead of as a noun. This intuition is confirmed by its context *vända med utnyttjande av backverkan* 'turn using reverse effect'. The context of *backåkning*, i.e. *(typ av) kälke för backåkning* 'kind of sledge for downhill going' confirms the inutition that *back* here is an occurrence of *backe* 'hill' rather than of *back.* In addition to the four cases of homography mentioned above, there are 36 cases (9%) of polysemy concerning the first constituents of our N-N material. Without scrutinizing the individual cases further, we conlude that the derivation of meaning from the constituent parts of a compound in the definition has to be based on the identification of their definitions. When homography is involved, a choice has to be made. Only when a morphological example is there to facilitate the choice can we maintain the transparency claim. The compound words with an ambiguous first or second component will all have to be explicitly defined in our machine dictionary.

In addition to the 376 N-N compounds there are 76 compounds of different syntactic structure (Table 3). They will all be examined with regard to explicit definition and homography of constituent parts.

The total number of derivatives (not explicitly defined) that were identified is 80, which means that their share of the total number of implicitly defined words is only roughly 15 percent as compared to 85 percent for the compounds. In Table 4 we present the distribution of the derivatives by different types.

The total number of derivatives given in Table 4 (90), is higher than that presented in Table 3 (80), for the following reason. The figures in Table 3, are based on a 'top-level' classification of the words. In other words, a word such as *bakningsredskap* 'baking tool' is classified as a compound only, disregarding the fact that its first constituent is a derivative not explicitly defined. However, the figures presented in Table 4, include also such indirect derivation, and, in some

Type	Example	Absolute frequency	Relative frequency
N {*-(n)ing*}	*annonsering* 'advertising'	44	49
N {*-het*}	*aktsamhet* 'carefulness'	9	10
N {*-nde*}	*anskaffandet* 'acquirement'	7	8
A {*-bar*}	*avgränsbar* 'delimitable'	3	3
A {*-orisk*}	*artikulatorisk* 'articulatory'	1	1
Pref-V	*avstämpla* 'stamp'	6	7
V Particle	*hugga av* 'cut off'	20	22
Total		**90**	**100%**

Table 4: Types of derivatives incl. particle verbs.

cases, a word contributes to more than one of the derivation types. For instance a word such as *avfjällning* which has to be derived in two steps, i.e. *fjälla* 'peel' (explicitly defined), *fjälla av* 'peel off', and the process *avfjällning* is counted as an instance of both the V Particle and the N {*-(n)ing*} type. Cases of two-step derivation, though, are rare.

The dominating type (49%) is that of the verbal nouns, formed by means of the *-(n)ing* suffix. Its 44 members (inflectional forms) are derived from 42 different verbs. 32 of these verbs are explicitly defined, whereas 10 only implicitly so. Further, six of the 32 explicitly defined verbs are polysemous, and, consequenlty, the nouns derived from them not uniquely defined. The ten implicitly defined verbs, are either particle verbs formed by means of the particle *av*, or prefixed verbs formed by means of the prefix *av*. Intuitively, we identify them as derived from the particle verbs e.g. *klippa av*. However, formally they might as well be derived from the corresponding prefixed verbs e.g. *avklippa*. The same kind of ambiguity is a problem in the identification of the verbs of prefixed past participles, e.g. *avhuggen*. It might be derived from *avhugga* as well as from *hugga av*. As regards the semantic distinction between the prefixed verb and the particle verb, the meaning of the prefixed verb seems to be more abstract than that of the particle verb (see further Hellberg 1976; Ejerhed 1979). If we allow the use of implicitly defined particle verbs or prefixed verbs as a basis of verbal nouns of the *(n)ing* type, a systematic ambiguity will be created. It should be avoided, as should also the use of prefixed past participles in the definitions, inherently ambiguous as they are.

Concerning the remaining derivative types, their representatives will be examined for homographies and implicitly defined types in the same manner as the verbal nouns. In discussing the derivatives and their aptness for being handled by the word formation component rather than being registered as lexical units, the productivity of the affixes is an additional parameter to be considered. An example of a suffix sequence with low productivity is *-or-isk* with an absolute (lexical) frequency of one (*artikulatorisk* 'articulatory') in our material and of two in the NFO material (Allén et al. 1980).

Finally, derived words, which can be segmented in more than one way should be avoided. An example of such an instance in our material is *alliansfri-het* 'non-alignment' or *allians-frihet*.

3.2 Homography

In the automatic lemmatization process, the morphological analysis has to be followed by a homograph resolution procedure. The need for such a procedure in our material is evident from the figures on alternative analyses presented in Table 1. Roughly five percent of the definition words are homographic. Here we will give a presentation of the different kinds of homographies that we found, and discuss the possibility of solving them by means of syntactic analysis.

Type	Absolute frequency	Relative frequency
A/V	29	30
N/V	25	26
A/N	13	14
N1/N2	10	10
V1/V2	5	5
A/Ab	3	3
Ab/Pp	3	3
N/Ab	2	2
N/Pn	2	2
N/I	1	< 1
V/I	1	< 1
V/Pp	1	< 1
C/Im	1	< 1
Total	**96**	**100%**

Table 5: Types of homographies.

In Table 5 we present a word class based overview of the different kinds of homographies resulting from the morphological analysis. First we state, that in roughly 85 percent of the cases, the homograph components belong to different word classes, whereas 15 percent concern homography *within* one word class (the nouns or the verbs). The preconditions for solving the homographies by syntactic means, should, of course, be better within the first category than within the second one, where we may come close to purely lexical disambiguation. However, for an evaluation of this general assumption, we need to know more about the actual forms that coincide, and present such data for the major types in Tables 6 to 10.

The overwhelming number of cases of A/V homography (90%) are due to coincidence between the adjective and a participial form of the verb. The differentiation between them is notoriously difficult, to a great extent due to their partly overlapping distribution. However, one of the principles that was adhered

A/V

Type	Example	Number of members
1. A/V(pp)	*ansedd* 'respected/considered'	9
2. A(weak)/V(pp/sup)	*ansedda* 'respected/considered'	3
3. A(neutr)/V(pp/sup)	*avancerat* 'advanced'	6
4. A(weak)/V(pp/pret))	*allierade* 'allied'	6
5. A/V(ap)	*anslående* 'impressive/impressing'	2
6. A(weak)/V(inf)	*anrika* 'high-born/concentrate'	3
Total		**29**

Table 6: Types of A/V homographies.

to in the formulation of the definitions, i.e. the *replacement restriction* (see 2), makes the task more realistic than would be the case with unrestricted text.

The nine members of the first type of Table 6 are in the basic form, and thus may occupy both an attributive and a predicative position. In all, they occur 73 times, and in 67 of these cases the adjectival interpretation should be preferred. Our choice was based on the following heuristics:

1. If the current word is in the attributive position, and not negated by an adverb, choose the adjective.
2. If the current word is the head of the definition of an adjective, and not negated by an adverb, choose the adjective.
3. If the current word is nominalized, choose the adjective.
4. If the current word occurs in the context *egenskapen att vara* ... the property of being '...', choose the adjective.
5. Else, choose the participle.

Examples:

1. [nn absess]: *begränsad ansamling var* 'limited amount of pus'
2. [av krystad]: *påfallande ansträngd och onaturlig* 'strikingly forced and unnatural'
3. [nn arv 1]: *övergång av egendom av (visst) värde från avliden till efterlevande* 'transition of property of (certain) value from deceased to surviving'
4. [nn begåvning/1]: *egenskapen att vara begåvd* 'the property of being talented'
5. [vb gälla 1/3]: *vara ansedd (som)* 'be considered (as)',
   [nn bricka/1]: *bärbar skiva begränsad av låg kant* 'portable plate surrounded by a low edge',

[av oavgjord/1]: *inte avgjord till någons fördel* 'not decided to anyone's advantage',
[nn ödesbygdsväg]: *väg genom ej bebodda trakter* 'road through not inhabited regions'.

The heuristic rule 2 is based on the replacement restriction; only if the definition belongs to the same syntactic category as the definiendum is it capable of substituting it. Further we require *same syntactic category* to mean *same word class*, (or same word class with respect to the head word of the definition). If, however, the homograph is preceded by a negating adverb, being a strong verb signal, the participle interpretation is chosen.

Concerning Type 2 (in Table 6) we have the same situation as with Type 1 with an additional supine form alternative. The supine form has a quite different distribution as compared to the adjective and the participle, and thus is easily distinguished in the syntactic analysis. (No supine forms, however, were found among the 31 occurrences of this type). In distinguishing between the adjectival and the participial forms, the heuristics listed above should be applied. It should also work out for Type 3, 4, and 5. The finite (past tense) form of Type 4 should be easily identified in the syntactic analysis. Finally, Type 6 should cause no problem.

A/N

Type	**Example**	**Number of members**
1. A/N	*alternativ* 'alternative'	8
2. A(neutr)/N	*basalt* 'basal/basalt'	1
3. A(weak)/N	*amerikanska* 'American (woman)'	4
Total		**13**

Table 7: Types of A/N homographies.

The dominating type of the adjective/noun homography is that between their basic forms (Type 1 in Table 7). Among the 70 occurrences of this type, 46 are adjectives, and, only 24 nouns. In the attributive position, there are hardly any problems in recognizing the adjectives on a purely syntactic basis. Making the homograph resolution though in the position after the finite verb requires additional knowledge. This is the case, for instance, for a word such as *bankrutt* 'bankrupcy' in the definition *göra bankrutt* 'become bankrupt' [vb bankruttera]. Further, Type 2 (Table 7) is an example of a causal coincidence between a neuter adjectival form and a noun. More systematic, however, is the homography between the weakly inflected adjective and the noun in Type 3. The rules for nominalizing the adjectives must be very restricted in the analysis grammar, if we are to solve these homographies syntactically, the reason being that the weakly inflected adjectives due to their inherent definitenesse constitute a productive basis for nominalization.

N/V

Type	Example	Number of members
1. N/V	*aga* 'flog(ging)'	6
2. N/V(imp)	*begär* 'desire/require'	3
3. N/V(ap)	*anseende* 'reputation/considering'	4
4. N/V(sup/Pp)	*agat* 'agate/flogged'	2
5. N/V(pret)	*bad* 'bath/asked'	3
6. N(+gen)/V(sup pass)	*ansats* 'approach(+gen)/(been) cultivated'	1
7. N(gen)/V(pres pass)	*begärs* 'desire(gen)/(is) required'	1
8. N(gen)/V(pret pass)	*bars* 'bar(gen)/(was) carried'	1
9. N(gen)/V(dep)	*andas* 'spirit(gen)/breathe'	1
10. N(pl)/V(pres)	*bakar* 'backs/bake(s)'	7
Total		**29**

Table 8: Types of N/V homographies.

In Table 8 we present the noun/verb homographies. All the different types seem to be solvable in their local contexts in combination with the syntactic prediction made by the word class marker of the definiendum, the substitution criterion.

N1/N2

Type	Example	Number of members
1. N1(utr)/N2(neutr)	*as* 'as/carcass'	5
2. N1(pl)/N2(pl)	*backar* 'backs/hills'	2
3. N1/N2	*bar* 'bar1/bar2'	1
4. N1(def sg)/N2(def sg)	*anden* 'the wild duck/the spirit'	1
5. N1/N2(pl)	*basar* 'bazaar/bass voices'	2
6. N1/N2(def)	*banan* 'banana/the path'	1
7. N1/N2(gen)	*askes* 'ascetisism/ash wood(gen)'	1
8. N1(indef gen)/N2(def gen)	*banans* 'banana(gen)/the path(gen)'	1
9. N1(gen)/N2(gen)	*bars* 'bar1(gen)/bar2(gen)'	1
Total		**15**

Table 9: Types of N1/N2 homographies.

In Table 9 and Table 10 we present the inherently most difficult homography cases, i.e. those in which the homograph components belong to the same word class, the noun class in Table 9 and the verb class in Table 10. Among the nouns there are four types (2, 3, 4, and 9) with no formal criteria to distinguish between them. Consequently, in these cases the homograph resolution amounts to purely lexical disambiguation, and the homography will be indifferent to the syntactic analysis. In Type 1 the homograph components differ with regard to gender only.

Its five members have, in all, 35 occurrences, but only in 9 of these cases is gender decisive for the choice. In Type 5, number is the distinguishing feature. It has two members with 10 occurrences. Three of these cases can be readily solved by syntactic means. Three of them can be solved, if number agreement is considered to be a precondition for coordinating nouns. For instance, *hög överbyggnad i för eller akter på medeltida fartyg* [nn kastell/2] 'high superstructure at the prow or at the stern of medieval vessels'. Two cases, finally, can be solved if phraseology is taken into account, e.g. *akter ut* 'astern'. In Type 6 and Type 8 definiteness is the distinguishing feature. It readily solves one of the three cases, i.e. *väg (till ngt) som ej följer den naturliga banan* 'road (to something) which doesn't follow the natural path' [nn bakväg]. As regards definiteness in the head word of prepositional phrases (the remaining two cases), the individual prepositions make their own demands, and the situation is more complicated. Finally, in Type 7, the homograph components differ with regard to case. It solves the homography, if the grammar doesn't admit elliptic NP heads.

V1/V2

Type	Example	Number of members
1. V1/V2	*avsluta* 'finish/conclude'	2
2. V1(+pres)/V2(+pres)	*avslutas* 'finish(pass)/conclude(pass)'	2
3. V1(pp/pret)/V2(pret)	*avlade* 'begetted/layed aside'	1
Total		**5**

Table 10: Types of V1/V2 homographies.

In Table 10 we present three types of verb homographies. The first two have no distinguishing formal feature, and thus cannot be solved by syntactic means. In the third type, however, there is a context in which they differ, i.e. in past participle constructions. In our material, this distinction doesn't emerge, and the homography cannot be solved syntactically.

Type	Example	Number of members
N1/N2/V	*backar* 'backs/hills/back(s)'	5
A/N/V(ap)	*avgörande* 'decisive/decision/deciding'	4
A/Ab/Pp	*bakom* 'stupid/at the back/behind'	1
A/Ab/C	*bara* 'bare/only/(as) long as'	1
N1/N2/Ab	*akter* 'stern/acts/aft'	1
N/Ab/Pn	*allt* '(the) universe/gradually/everything'	1
A/N/Ab	*akut* 'acute/casualty department/urgently'	1
V/Nl/Pn	*andra* 'state/second/other'	1
N1/N2/N3	*bas* 'bass voice/base/thrashing'	1
Total		**16**

Table 11: Three homograph components

Type	Example	Number of members
N1/N2/Ab/Pp	*bak* 'behind/baking/at the back/behind'	1
N1/N2/A/V	*bar* 'bar/bar/bare/carried'	1
Total		**2**

Table 12: Four homograph components.

In those cases where three or four analyses were generated (cf. Table 1) combinations of the homography types presented above (cf. Table 5) were involved. The actual combinations are presented in Table 11, and Table 12.

4 Conclusions

In this paper we have presented the results that we achieved in a pilot study of the definition vocabulary of *Svensk ordbok*. It is part of our on-going work on the generation of a machine-tractable dictionary from the lexical database from which the dictionary was drawn, in specific, of making its definitions exploitable to the LPS parser. Parsing them is a step towards this goal. The present study, aiming in particular at the automatic lemmatisation of the definition vocabulary, has included an automatic morphological (inflectional) analysis of a subset (2,500 graphic words) of it, and a, largely manual, analysis of the results. Two major issues were treated, i.e. the coverage of the dictionary in relation to the definition voaculary, and the feasibility of homograph resolution by syntactic analysis in the local (definition) contexts.

The morphological analysis was carried out by means of the LPS morphological analyser, disposing of a stem dictionary covering the head words of the dictionary. As a result of this analysis, we found that, roughly, 23 percent of the definition words were not explicitly defined. The lexical gaps were mainly due to the use of proper nouns (6%), and derived words (93%). As regards the missing proper nouns, they will be registered in our machine dictionary. The dominating type of the derived words are the compounds (79%), while the derivatives (incl. particle verbs and prefixed verbs) account for 14 percent of the gaps. The transparency claim with regard to the implicitly defined words cannot be maintained without certain modifications of the definition vocabulary. Two measures will be taken in the generation of the machine dictionary. First, derived words including homographic or polysemous constituents will be exchanged by explicitly defined words, or by derived words of unambiguous elements. Secondly, the use of deverbal nouns based on implicitly defined particle verbs or prefixed verbs, implying a systematic ambiguity, will be avoided, as will the use of prefixed past participles, inherently ambiguous as they are. In general, derivations of more than one step will be avoided. In other words, we require that the toplevel constituents of a compound or a derivative are themselves explicitly defined. These modifications, when applied to whole definition vocabulary, will bring it one step closer to a *defining* vocabulary.

Homography turned out to be a smaller problem than the lexical gaps. In all, external homographs constitute less than five percent of the lexical material that was examined. Data on the distribution of different types of homographies are presented. With a manual study of their contexts (definitions) as a basis, we conclude, that they can, to a large extent, be resolved by syntactic means, provided that the syntactic prediction made by the word class marker of the definiendum is taken into account. This information is of vital importance. A heuristics for distinguishing between adjectives and participles was proposed. It will be further evaluated in the course of the project.

References

Allén, S. 1970. *Nusvensk frekvensordbok baserad på tidningstext. 1. Graford. Homografkomponenter.* [Frequency dictionary of present-day Swedish based on newspaper material. 1. Graphic words. Homograph components.] Stockholm.

Allén, S. 1981. The Lemma-Lexeme Model of the Swedish Lexical Data Base. B. Rieger [Ed.]. *Empirical Semantics*:376–387. Bochum.

Allén, S., S. Berg, J. Järborg, J. Löfström, B. Ralph, & C. Sjögreen. 1980. *Nusvensk frekvensordbok baserad på tidningstext. 4. Ordled. Betydelser.* [Frequency dictionary of present-day Swedish based on newspaper material. 4. Morphemes. Meanings.] Stockholm.

Blåberg, O. 1988. A study of Swedish compounds. Report No. 29, Dept. of General Linguistics. University of Umeå.

Ejerhed, E. 1989. Verb-partikelkonstruktionen i svenska: syntaktiska och semantiska problem. [The verb-particle construction in Swedish: syntactic and semantic problems.] O. Josephson, H. Strand, & M. Westman [Eds.]. *Förhandlingar vid sammankomst för att dryfta frågor rörande svenskans beskrivning 11*:49–64. Dept. of Nordic Languages. University of Stockholm.

Hellberg, S. 1976. *Av* som partikel och preposition. ['Av' as particle and as preposition.] Dept. of Computational Linguistics. University of Göteborg.

Järborg, J. 1988. Towards a formalized lexicon of Swedish. *Studies in computer-aided lexicology*:140–158. Almqvist & Wiksell International. Stockholm. (Data linguistica 18).

Sågvall Hein, A. 1987a. Forskningsprogram för projektet En lexikonorienterad parser för svenska. [Research program for the project A Lexicon-Oriented Parser for Swedish.] Dept. of Computational Linguistics. University of Göteborg.

Sågvall Hein, A. 1987b. Parsing by means of Uppsala Chart Processor (UCP). L. Bolc [Ed.]. *Natural language parsing systems*:202–266. Springer Verlag. Berlin & Heidelberg.

Sågvall Hein, A. 1988. Towards a comprehensive Swedish parsing dictionary. *Studies in computer-aided lexicology*:268–294. Almqvist & Wiksell International. Stockholm. (Data linguistica 18).

Sågvall Hein, A. 1989. The LPS inflectional grammar. A listing of the rules. Dept. of Computational Linguistics. University of Göteborg.

Sjögreen, C. 1988. Creating a dictionary from a lexical database. *Studies in computer-aided lexicology*:299–338. Almqvist & Wiksell International. Stockholm. (Data linguistica 18).

Svensk ordbok. [A dictionary of Swedish.) 1986. Produced at Språkdata ([Dept. of Computational Linguistics. University of Göteborg.] Stockholm.

Wilks, Y., D. Fass, C. Guo, T. McDonald, & M. Slator. 1988. Machine tractable dictionaries as tools and resources for natural language processing. D. Vargha [Ed.]. *Proceedings of the 12th International Conference on Computational Linguistics (COLING)*:750–755. Budapest.

Center for Computational Linguistics
Uppsala University
Box 513
S-751 20 Uppsala
uduas@seudas21.bitnet

IVAR UTNE

What Should be Included in a Commercial Word Data Base, and Why?

The original title, which should be considered as synonym, was:

> *User considerations and market strategy as basis for profiling content in computational word bases, with special regard to the Norwegian Term Bank's data base NOT and a multilingual Norwegian word list*

Abstract

In the article I present definitions of the concepts quality and quality assurance, which are basis for proposing a general definition of quality of word bases. The quality of word bases is the specifications that customer and producer agree upon, and includes linguistic and other user relevant considerations. This is exemplified with the revision work of a term record format, and work with a word base of everyday language.

1 Introduction

In this paper I will propose guidelines that could be worth aiming at in order to compile word bases that the users wish, need, and will pay for. The presentation of the guidelines will be based on a presentation of the quality assurance concept, which is becoming very important in the industry and the service sector. This will be exemplified with some word bases, i.e. computer based word lists and terminological data bases. I will emphasize that the presented proposals are not a complete solution, but I hope they will present some ideas for further work.

The ideas will be based on experience with terminological data bases at the Norwegian Term Bank (NT, Norsk termbank), and with work on word lists based on everyday language at the Department of Scandinavian Languages and Literature (Nordisk institutt) in cooperation with the Norwegian Computing Centre for the Humanities (NAVFs edb-senter for humanistisk forskning). I have be working in or in close contact with these institutions for some years, and base

the descriptions on the current term record format on documents produced by personnel at NT. Much of these activities, and especially those of terminological work, have been based solely on support from the industry and users outside the University.

Note that the term *producer* in this presentation always will mean *producer of products and services*, and the term *product* will always mean *product and service.*

2 Word Base Quality — Quality Assurance

In order to design word bases and to get language services contracts including compilation of lists as part of them, we need to have a strategy to succeed in the market.

The main points for such a *market strategy* could be described as transfer of products and services to a market with regard to:

(a1) Users' wishes and needs

(a2) Identification of new paying user groups and their wishes and needs

(a3) Compare user wishes and needs with the needs of the researchers to gain cooperation and/or coordination in order to give more resources to research and to improve the product

(a4) Prices, which will not be further developed here

According to this I will concentrate on the quality of the products, which is the most important premise on an effective market strategy.

Quality of products (and services) is according to the International Standardization Organization defined as:

> The totality of features and characteristics of a product or service that bear on its ability to satisfy stated or implied needs. (ISO 8402-1986:2)

More shortly it could be stated as:

> Accordance with what specifications the customer and producer have agreed upon.

Another related concept is *quality assurance* (QA). According to the same standard QA is defined as:

> All those planned and systematic actions necessary to provide adequate confidence that a product or service will satisfy given requirements for quality. (ISO 8402-1986:4)

While ISO stresses the relation between producer and customer, Norwegian Petroleum Directorate (Oljedirektoratet) stresses saftey for personnel (and society) and Norwegian Society for Quality (NFK, Norsk Forening for Kvalitet) in some publications also expresses the importance of profit as result of effective production routines.

Norwegian Petroleum Directorate states aspects of QA in *Regulations concerning the licensee's internal control in petroleum activities on the Norwegian Continental Shelf with comments* (Oljedirektoratet 1985a:3):

> The regulations will be applicable to worker protection and the worker environment within the scope of the Act concerning worker protection and worker environment (the Working Environment Act).
>
> The regulations will also be applicable to protection against pollution in petroleum activities within the scope of the Act concerning pollution and refuse (the Pollution Act).

The QA requirements related to worker environment and public considerations is not yet precisely formulated in the legislation and regulations. An instructive presentation of QA and worker environment is Hellesøy 1988 (Norwegian text).

Norwegian Society for Quality states their views on profit for instance in:

> the title of a booklet called *Profit by quality* (The Norwegian original title is: *Lønnsomhet ved kvalitet*), NFK 1987a.
>
> the heading "Correct performed quality assurance increases the productivity" (The Norwegian original text is: "Riktig utført kvalitetssikring øker produktiviteten") in another booklet (The Norwegian original title is Kvalitet og Kvalitetsstyring which means Quality and Quality Control, NFK 1987b).

This means that QA implies existence of systematically controlled routines to ensure that:

(b1) For the *producer-customer* relation: A product will be completed according to what is agreed beforehand, which implies routines to secure that the order is unambiguous and that there exists routines during production to ensure that the production acts towards the agreed goal.

(b2) For the *employers*: That the producer has as effective routines as possible, to ensure low cost for both parts (employer).

(b3) For the *employees*: The personnel implied work takes place without any lack of sequrity and according to work environment requirements.

(b4) For the *society*: There shall be no considerable risks for workers' safety and pollution, and the national (here: Norwegian) language shall have preference for the sake of safety (A requirement stated by Norwegian Petroleum Directorate in Oljedirektoratet 1985b:4 may possibly be interpreted like this.)

In order to apply this on work with word bases we have to decide what is central aspects of word base quality and how customer and producer can agree on what alternative espressions or style and other non-linguistic specifications that are to be defined as goals.

The specifications for word base quality should consider at least language, user relevant information and user interface. The consideration may include different values, e.g. that cultural considerations and rhytmic language may be of low or no importance; more in detail:

(c1) Good and communicative *language*

Good style/performance (aesthetic and cultural considerations with possible consequences for economics)

- Rhytmic or harmonius language
- Cultural considerations, tradition of "good" written language
- Subcultural considerations, e.g. firm or other local standards

Effective information transfer (economic and administrative considerations) means:

- Requirements for the selection and formation of terms according to ISO 704-1987(E):12–13 are that the terms should be:
  - linguistically correct

    "the term should conform to the norms of language in question" (op.cit.), e.g. letters (*ks* instead of *x*) and inflection paradigms
  - accurate, or motivated

    "the term should reflect, as far as possible, the characteristics of the concept which are given in the definition" (op.cit.), e.g. *magnetic tape* which is defined as "carrier of a magnetic recording, having the form of a tape" (based on op.cit.)
  - concise

    i.e. preciseness
  - permit, if possible, the formation of derivatives

    "alcohol — alcoholic, alcoholism, alcoholize" (op.cit.)
  - monosemous, if the terms are considered for standardization

    there shold be only one standardized term for a concept, and only one concept for one term
- Native (e.g. Norwegian) versus foreign (e.g. English) expressions
  - a native expression may be preferable because of accuracy and because it is linguistically correct (in the native language)
  - a foreign expression may be preferable in international communication, especially for abbreviations, formulas and symbols, but also for subjects with international traditions like chemistry

- "noise free" or neutral expressions
  - * not stigmatizing in the actual subculture
  - * related to relevant tradition

(c2) *User relevant* information

Reference to standard/authority documents/publications, i.e. controlling/ prescriptive documents (publications from language councils or academies, standards)

New use of symbols and other expressions, which usually are not included in dictionaries

Additional consensus with subject/user defined groups

(c3) *User interface* in a data base system

Transportable program and data, including copy protection

Stratified selection of information related to the purpose

Simple and logical user dialogue

Frequent update, especially when language services interact in multipurpose projects

For the implied interest groups in the collection of QA definitions listed above this means for:

(d1) *Customers*:

To decide what will be the goals we have to clarify and coordinate the needs and wishes of the customers with the word base knowledge/expertise. Central interfaces may be listed like this:

- What the users want and need
- What the users want, but don't need, e.g.:
  - * alternative synonyms in the target language
- What the users need, but don't ask for, e.g.:
  - * consistent terminology without use of synonyms in the same language
  - * consequent use of substandards (-norms), such as British English without US forms
  - * defined subsets of Norwegian
  - * accommodation to existing standards and regulations

In order to decide the quality requirements of the product the customer and producer must be in dialogue:

- To choose and design expressions according to the linguistic requirements above, cf. (c1).
  To choose information categories (types)

  - * In word lists: the selection of languages, references, definitions etc.
  - * In thesauri: any registration of deleted subject words/concepts after revision of an old thesaurus
- – To design the information according to needs and subject knowledge
  - * In terminological data bases: should the additional synonyms, references, or context excerpts be left out
  - * In word lists: e.g. the selection of variants (style) within Norwegian-Nynorsk

(d2) *Employers*:

Backup routines

Effective tools

Reference information

Housekeeping of information for other projects and for research

(d3) *Employees*:

Effective and user friendly tools for automatizing, to get rid of boring work and have overview and control

Proper procedure descriptions

Easy access to relevant information

Unambiguous references

(d4) *Society*:

Consideration of culture values, according to the prevalent values

Public considerations, e.g. proper language for effective communication which may imply a precondition for safety and good health

In the next section I will apply these definitions and principles to the word base work at my institution. For a discussion of quality assurance for language work in general it is referred to Utne 1987 (Norwegian text).

3 Record Format for a Terminological Data Base

The terminological data bases at NT include mainly bilingual dictionaries and also hierarchically structured thesauri. Both these data types have been presented at *Symposium for datamatstøttet terminologi og leksikografi* in 1985 and 1987 (Utne 1986, Utne 1988), and revised descriptions are distributed from NT (free of cost). An excerpt from the present description (NT 1989) is included in the appendix.

The terminological work has usually been part of multipurpose projects aiming at cost effective document production. Language services have in most occasions been looked upon as a totality in this context. It is very unusual that the supporters give grants for development of dictionaries. For the bilingual dictionaries the exceptions have been projects for oil companies, mostly in the period 1984–86. The update of the data is partly financed by subscription on continuously updated copies. NT is maintenance contractor for a thesaurus developed by NT in 1985–86.

A radical restructuring of the data base format for the terminological data base was performed in 1988. A presentation of this restructuring while in process is presented in Ebeling and Utne 1988.

In the following I will examplify application of the quality assurance concept on the restructuring of this data base. This application is based on my point of view which is partly from outside. The process was in practice not planned and performed according to the principles presented below, but has in fact followed most of them. My presentation will be an application connected to a possible and realistic strategy.

Lots of considerations about quality of linguistic expressions are not dependant of this revision of format and are therefore not listed below. The *general leading principles* for the revision have been to gain better quality of:

(e1) *Language*, by:

Error free data to a larger extent

(e2) *User relevance*, by:

Unambiguous classification of greater part of the data

More flexible introduction of new categories/classification

More stress on standardization

(e3) *User interface*, by:

More flexible presentation, excerption and introduction of different and more fine grained data types

Improved distribution

This implied some more applied leading principles (without explaining the links to the general principles in detail):

(f1) More strictly constructed hierarchical structure

Goal: To make more flexible excerpts of subsets of data possible, i.e. to extract different combinations of fields and parts of fields, and introduce more unambiguous links between a term and its abbreviation, reference or its context

Solution: A more strictly hierarchical structuring inside the term record, which means that:

- Abbreviations, contractions, symbols and formulas are unambiguously bound to their full forms and not only to their concept
- References are unambiguously bound to their term (or abbreviation etc.), context-excerpts, definitions etc
- Contexts are unambiguously bound to their terms etc
- Comments are unambiguously bound to all relevant kinds of fields

(f2) More formal notation language

Goal: Be more consequent in registration of all kinds of information, i.e. more formal and restricted formats for the information types, to make it easier to extract expressions from the same page or pages in a specified source or to extract terms from a specified subject area

Solution: Defined vocabulary (e.g. titles) and syntax for references
Program tools according to the goal

(f3) A productive notation system to include user relevant information

Goal: Be more flexible in introducing of new information classes, i.e. the possibility of introducing new information classes according to simple and defined routines

Solution: Supplementary information not traditionally included in dictionaries, is introduced because of the usefulness for the target groups. This means for instance that there is introduced a distinction between general abbreviations, project specific abbreviations, symbols, classification codes and formulas. Some of the categories are:

- International standardized symbol; shown, referred to an illustration, and/or described
- Chemical/mathematical formulas, e.g. NH_4HSO_3 for *ammonium hydrogensulfite*
- International classification codes, e.g. according to UN (United Nations), EC (European Community), CAS (Chemical Abstract Services) and widely used national standards
- Trade names for products of the concept
- References to official documents, e.g. standards, registers, laws and regulations
- Hazard classification, e.g. fire, poison
- References to figures which illustrate the concepts
- Area of application (not uniquely for this data base), for housekeeping, collecting subsets and for indicating meaning.

(f4) Existence of programs and other routines for error checks

Goal: Make update more free from errors, i.e. facilitate more throughout error checks

Solution: Program tools according to the goal

(f5) Standardization

Goal: As it has been a goal for years, one concept is one record

Solution: Synonyoms in the same language are collected in the same record.

Goal: As it has been a goal for years, there should be only one preferred full form term for each language.

Solution: This is called main term, and the others are called synonyms or deprecated terms.

Goal: The format should also include registration of alternative standards, like former main term, and main terms in other standards.

Solution: Introduction of fields for approval date and scope of a standard. This includes also out of date standards.

(f6) Transportable program and data

Goal: Program and data files produced for different machines and media

Goal: Routines for copy protection

Solution: Program tools for both these goals.

4 Word Lists

The development of word lists of everyday language was initiated by the researchers at the University about 20 years ago. The project is a cooperation between NT and Norwegian Computing Centre for the Humanities. Further development is financed by sale. Customers are mostly software houses, institutions and firms (graphic industry and newspapers) which are able to include the lists in existing software or to develop standalone programs.

The quality of these lists is partly based on general needs for spelling lists, special and general needs for lists with hyphenation marks (with special reference to different levels) and our estimation of user needs for other lists with grammatical information and lists of word parts. The development of spelling lists has been based on our own assumption that a frequency based and general correspondance vocabulary would suit the users' needs best. The development of lists with grammatical information and word composita is based on point (a3) in market strategy above, i.e. combination of researchers' and customers' needs. The development of lists with substandards also accommodates needs and wishes for accommodation to substandards and more personal ways of writing.

The existing word lists include spelling lists based on general need and a list with hyphenation-marks based on special needs:

- *Spelling word lists* for Norwegian-Bokmål and for Norwegian-Nynorsk, including a collection of high frequency word forms without any further coding.
- Word lists with *hyphenation marks*, based on special need, but also with general application. The marks express different levels, so that the different types of marks borders between:
  - *Compounds*, like data+base (which is written as one word in Norwegian)
  - *Pre- and suffixes* and the rest of the word like ex=plos=ion (Norwegian: eks=plo=sjon)
  - *Inflection morphemes*, like the definite plural in bil//ene (= the car//s)

The further development includes partly further refinement of the lists above and development of new word list types partly based on a combination of what is asked for in new products for text processing or data base tools with dialogues in natural language, and work with machine-aided translation with what is asked for, cf. combination of customer needs and wishes with the researchers' as market strategy. That means lists containing:

- *Word composita*, e.g. parts of latin and greek loan words.
  - The machine aided system may use lists like this to translate loan words which have identical parts, but related inflectional paradigms.
  - In text processing systems this may be used as part of hyphenation programs, and combined with supplementary rules partly also as part of spell checkers.
- Word lists including *grammatical information*, like part of speech and inflectional paradigms.
  - The most important part of a machine aided translation.
  - In text processing systems and especially in dialogue based data base tools this may be used in programs that are based on simpler syntax analysis or calculations of possible part of speech for words in a text string.

And as a combination with work to systematize *substandards* in the written Norwegian languages:

> Word lists with grammatical information that classifies the words and word forms (often inflectional paradigms) as moderate (as different as possible) and radical (approaching to each other) within the two official Norwegian languages. This is of importance to writers who need language checks to profile their writing closer to such

a substandard. This implies a great variety of possible written languages which computer systems should be able to control. The exact definitions of each such language are not objectively stated, but are to some extent a matter of personal or user groups decisions. Table 1 and 2 in Utne 1989 (paper at *Nordiske datalingvistikdage 1989*) present some examples of this diversity. Some of the examples are repeated in Table 1 below. A further explanation of the language situation is presented in Utne 1989.

Language		porridge	line	problems	boys
Mod.	Norw.-Bokmål:	grøt	linje	problemer	gutter
Rad.	Norw.-Bokmål:	graut	linje	problem	gutter
Rad.	Norw.-Nynorsk:	graut	linje	problem	gutar
Mod.	Norw.-Nynorsk:	graut	line	problem	gutar

Table 1. Spelling and inflection in Norwegian
(Mod. = Moderate, Rad. = Radical)

To some extent this substandard works as if there are slightly different written sublanguages inside each of the two official Norwegian languages. There is also some tradition for unofficial written standards, e.g. one more moderate than Norwegian-Nynorsk called Conservative Norwegian-Nynorsk, another more moderate than Norwegian-Bokmål called Conservative Norwegian-Bokmål (Norwegian: Riksmål) and a third one between the two official languages which is sometimes called Pan-Norwegian (Norwegian: Samnorsk). Of these three Conservative Norwegian-Bokmål has the widest use with its use in at least one of the most widespread newspapers, in lots of books and publications every year.

The list containing a diversity a form alternatives within each of the language and also to some extent other unofficial language standards can be considered as a multilingual dictionary. This total word base concept, which at the time beeing contains between 20 and 30 000 entries (which can be inflected according to different alternatives) is the base of such a multilingual Norwegian word base. This base will be developed both for research, included machine aided translation, and for commercial applications.

Other possibilities not worked out in detail yet are for instance:

Synonyms and words with related meaning

Deprecated words and expressions, and their substitutions

Lots of deprecated expressions in Norwegian-Bokmål have common or near related preferred words both in Norwegian-Bokmål and Norwegian-Nynorsk.

5 Conclusion

Through this discussion of commercial word bases I have formulated what are the concepts quality and quality assurance, and proposed a general definition of quality of word bases. While quality is the part of the product, quality assurance is the procedures to secure quality and also consider the interests of employer, employee and the society. The quality of word bases is the coordinated specification that customer and producer agree upon, and includes linguistic and other user relevant considerations.

The examplifications from the work at the University of Bergen concern the revision of a term record format and the work with a word base of every day language. The work with format revision emphasized language from the view of errors cheks, user relevance, and user interface. The work with word bases emphasized an ongoing work with a multilingual Norwegian word base which includes form variants and also unofficial languages. In the work with this word base there are combined interests between research and profit.

References

Ebeling, Jarle and Ivar Utne. 1988: New Record Format at The Norwegian Term Bank. *Nordisk tidsskrift for fagspråk og terminologi* (Nordic Journal of L.S.P. and Terminology), vol 6, no 1:7–14. ISSN 0108-77891

Hellesøy, Odd H. 1988: Kvalitetsstyring av arbeidsmiljø? (= Quality control of worker environment?). *Nordisk Ergonomi* Vol 6, no 1:11–16.

ISO (International Standardization Organization) 704-1987 (E): *Principles and methods of terminology.*

ISO (International Standardization Organization) 8402-1986 (E): *Quality — Vocabulary.*

NFK (Norsk Forening for Kvalitet) 1987a: *Lønnsomhet ved kvalitet.* Booklet.

NFK (Norsk Forening for Kvalitet) 1987b: *Kvalitet og kvalitetsstyring.* Booklet.

NT (Norsk termbank) 1989: *"NOT" User's Guide.* Ver. 1989-05-23. Booklet.

Oljedirektoratet (Norwegian Petroleum Directorate) 1985a: *Forskrift om rettighetshavers internkontroll i petroleumsvirksomheten på norsk kontinentalsokkel med kommentarer. Regulations concerning the licensee's internal control in petroleum activeties on the Norwegian Continental Shelf with comments.* ISBN 82-7257-183-8

Oljedirektoratet (Norwegian Petroleum Directorate) 1985b: *Forskrift om sikkerhet m.v. til lov om petroleumsvirksomhet. Regulations concerning safety in exploration, exploration drilling and recovery of petroleum deposits, etc.* ISBN 82-7257-186-2

Utne, Ivar. 1986: Terminologidata på mikromaskin ved Norsk termbank. *Symposium om datorstödd terminologi och lexikografi i Helsingfors den 13 og 14 december 1985*:52–58. Centralen för Teknisk Terminologi. Helsingfors 1986.

Utne, Ivar. 1987: Nye perspektiver på språklig kvalitet — introduksjon til kvalitetssikring for språklig arbeid. (New perspectives on language quality — Introduction to quality for language Work.) *Nordisk tidsskrift for fagspråk og terminologi* (Nordic Journal of L.S.P. and Terminology), vol. 5, no 1:14–21. (Norwegian text with English summary.) ISSN 0108-77891

Utne, Ivar. 1988: Terminologi, arbeidsinstrukser og lagerstyring — om kodeuttrykk i fagspråk. *Nordiske Datalingvistikdage og Symposium for datamatstøttet leksikografi og terminologi 1987.* Proceedings:273–285. Institut for Datalingvistik, Handelshøjskolen i København. Copenhagen.

Utne, Ivar. 1989: Machine aided translation between the two Norwegian languages Norwegian-Bokmål and Norwegian-Nynorsk. *Nordiske datalingvistikdage 1989.* Reykjavík. In press.

Strømgaten 53
N-5007 BERGEN
Norway

Appendix

From *"NOT" User's Guide*:2, ver. 1989-05-23

The Record Format

The terms in this data base are organized in concept records, which consist of one Norwegian section, one English section and one section that is common to both languages.

The Norwegian section consists of the following fields:

N	hvd	=	Norwegian main term (recommended for use)
N	syn	=	Norwegian synonym to the same concept (to be avoided)
N	fra	=	Norwegian synonym not recommended for use (must not be used)
N	krt	=	Norwegian contraction (used for reasons of space, as for instance in screen pictures, drawings, signs etc.)
N	frk	=	Norwegian abbreviation
N	def	=	Norwegian definition
	kon	=	context of term
	kom	=	comment to term
	ref	=	reference of term
	fig	=	reference to figure

The English section contains the following fields:

E	hvd	=	English main term
E	syn	=	English synonym to the same concept (to be avoided)
E	fra	=	English synonym not recommended for use (must not be used)
E	krt	=	English contraction (used for reasons of space, as for instance in screen pictures, drawings, signs etc.)
E	frk	=	English abbreviation
E	def	=	English definition
	kon	=	context of term
	kom	=	comment to term
	geo	=	geographical distribution of term
	ref	=	reference of term
	fig	=	reference to figure

The common section consists of information about the concept as a whole:

	brk	=	area of application
	sbl	=	international standardized symbol
	fml	=	chemical/mathematical formula
.	nr	=	national and international numbers
.	nvn	=	trade name
.	kom	=	comment to the concept as a whole
	kon	=	context of symbol, formula etc.
	kom	=	comment to symbol, formula etc.
	ref	=	reference of symbol, formula etc.
	fig	=	reference to figure of symbol, formula etc.

It is possible to introduce new fields at all levels.

Participants

Sture Allén
Språkdata
Göteborgs universitet
S-412 98 Göteborg
SVERIGE

Peter Ammundsen
EF-Kommissionen
Afd. for terminologi
Batiment Jean Monnet A2/137
L-2920 LUXEMBOURG

Poul Andersen
EUROTRA - DK
Københavns Universitet
Njalsgade 80
DK-2300 København S
DANMARK

Ásta Svavarsdóttir
Orðabók Háskóla Íslands
Árnagarði við Suðurgötu
IS-101 Reykjavík
ISLAND

Baldur Jónsson
Íslenskri málstöð
Aragötu 9
IS-101 Reykjavík
ISLAND

Annelise Bech
EUROTRA - DK
Københavns Universitet
Njalsgade 80
DK-2300 København S
DANMARK

Björn Þór Svavarsson
Orðabók Háskóla Íslands
Árnagarði við Suðurgötu
IS-101 Reykjavík
ISLAND

Lars Borin
Uppsala Universitet,
Centrum för Datorlingvistik
Box 513
S-751 20 Uppsala
SVERIGE

Anna Braasch
EUROTRA - DK
Københavns Universitet
Njalsgade 80
DK-2300 København S
DANMARK

Benny Brodda
Institutionen för lingvistik
Stockholms Universitet
S-106 91 Stockholm
SVERIGE

Boel Bøggild-Andersen
EUROTRA - DK
Københavns Universitet
Njalsgade 80
DK-2300 København S
DANMARK

Mats Dahllöf
Språkdata
Göteborgs Universitet
S-412 98 Göteborg
SVERIGE

Helle Degnbol
Ordbog over det norrøne prosasprog
Københavns Universitet, Amager
Njalsgade 76, DK-2300 København S
DANMARK

Einar Gunnar Pétursson
Stofnun Árna Magnússonar á Íslandi
Árnagarði við Suðurgötu
IS-101 Reykjavík
ISLAND

Eiríkur Rögnvaldsson
Málvísindastofnun Háskóla Íslands
Árnagarði við Suðurgötu
IS-101 Reykjavík
ISLAND

Eva Ejerhed
Institutionen för lingvistik
Umeå Universitet
S-901 87 Umeå
SVERIGE

Bengt Ek
Institutionen för lingvistik
Stockholms Universitet
S-106 91 Stockholm
SVERIGE

Gunnar Eriksson
Institutionen för lingvistik
Stockholms Universitet
S-106 91 Stockholm
SVERIGE

Jens Erlandsen
TextWARE A/S
Rådmandsgade 43, 1
DK-2200 København N
DANMARK

Hanne Fersøe
EUROTRA - DK
Københavns Universitet
Njalsgade 80
DK-2300 København S
DANMARK

Lisbet Frost
Volvodata AB
Avdeling 2630
S-405 08 Göteborg
SVERIGE

Barbara Gawron'ska-Werngren
Institutionen för lingvistik
Lunds Universitet
Helgonabacken 12
S-223 62 Lund
SVERIGE

Annelise Grinsted
Grinsted Products A/S
Edwin Rahrsvej 38
DK-8220 Brabrand
DANMARK

Bo Grönholm
Vienolavägen 10/5
SF-20210 bo
FINLAND

Maija Grönholm
Vienolavägen 10/5
SF-20210 bo
FINLAND

Guðrún Kvaran
Orðabók Háskóla Íslands
Árnagarði við Suðurgötu
IS-101 Reykjavík
ISLAND

Gunnlaugur Ingólfsson
Orðabók Háskóla Íslands
Árnagarði við Suðurgötu
IS-101 Reykjavík
ISLAND

Lars M. Gustafsson
Institutionen för lingvistik, Lund
Rådhustorget 7
S-261 31 Landskrona
SVERIGE

Arnbjørg Hageberg
Norsk leksikografisk Institutt
Universitetet i Oslo, Boks 1021
Blindern Oslo 3
NORGE

Halldóra Jónsdóttir
Ísafoldarprentsmiðja hf.
Þingholtsstræti 5
IS-101 Reykjavík
ISLAND

Steffen Leo Hansen
Institut for Datalingvistik
Handelshøjskolen i København
Dalgashave 15
DK-2000 Frederiksberg
DANMARK

Heimir Pálsson
Bókaútgáfan Iðunn
Bræðraborgarstíg 16
IS-101 Reykjavík
ISLAND

Helga Jónsdóttir
Orðabók Háskóla Íslands
Árnagarði við Suðurgötu
IS-101 Reykjavík
ISLAND

Håvard Hjulstad
Rådet for teknisk terminologi
Riddervolds gate 3
N-0258 Oslo 2
NORGE

Henrik Holmboe
Handelshøjskolen i rhus
Afdeling for Datalingvistik
Fuglesangs Allé 4
DK-8210 rhus V
DANMARK

Hrefna Arnalds
Ísafoldarprentsmiðja hf.
Þingholtsstræti 5
IS-101 Reykjavík
ISLAND

Hreinn Benediktsson
Málvísindastofnun Háskóla Íslands
Árnagarði við Suðurgötu
IS-101 Reykjavík
ISLAND

Höskuldur Þráinsson
Málvísindastofnun Háskóla Íslands
Árnagarði við Suðurgötu
IS-101 Reykjavík
ISLAND

Ingibjörg Johannessen
Ísafoldarprentsmiðja hf.
Þingholtsstræti 5
IS-101 Reykjavík
ISLAND

Bent Chr. Jacobsen
Ordbog over det norrøne prosasprog
Københavns Universitet, Amager
Njalsgade 76, DK-2300 København S
DANMARK

Janne Bondi Johannessen
Institutt for humanistisk informatikk
Universitetet i Oslo, Postboks 1102
Blindern, N-0317 Oslo 3
NORGE

Jón Aðalsteinn Jónsson
Orðabók Háskóla Íslands
Árnagarði við Suðurgötu
IS-101 Reykjavík
ISLAND

Jón G. Friðjónsson
Orðabók Háskóla Íslands
Árnagarði við Suðurgötu
IS-101 Reykjavík
ISLAND

Jón Hilmar Jónsson
Orðabók Háskóla Íslands
Árnagarði við Suðurgötu
IS-101 Reykjavík
ISLAND

Harri Jäppinen
SITRA Foundation
P.O. Box 329,
SF-00121 Helsinki
FINLAND

Niels Jæger
EUROTRA - DK
Københavns Universitet
Njalsgade 80
DK-2300 København S
DANMARK

Arne Jönsson
NLPLAB/IDA
Linköpings Universitet
S-581 83 Linköping
SVERIGE

Jörgen Pind
Orðabók Háskóla Íslands
Árnagarði við Suðurgötu
IS-101 Reykjavík
ISLAND

Keld Gall Jørgensen
Háskóla Íslands
IS-101 Reykjavík
ISLAND

Fred Karlsson
Institutionen för allmän språkvetenskap
Helsinki Universitet
Hallituskatu 11-13
SF-00100 Helsinki
FINLAND

Kirstín Flygenring
Íslenskri málstöð
Aragötu 9
IS-101 Reykjavík
ISLAND

Sabine Kirchmeier-Andersen
EUROTRA – DK
Københavns Universitet
Njalsgade 80
DK-2300 København S
DANMARK

Gregers Koch
Datalogisk institut ved Københavns Universitet
Universitetsparken 1
DK-2100 København Ø
DANMARK

Kristján Árnason
Málvísindastofnun Háskóla Íslands
Árnagarði við Suðurgötu
IS-101 Reykjavík
ISLAND

Gunnel Källgren
Institutionen för lingvistik
Stockholms Universitet
S-106 91 Stockholm
SVERIGE

Guðrún S. Magnúsdóttir
Språkdata
Göteborgs Universitet
S-412 98 Göteborg
SVERIGE

Margrét Jónsdóttir
Málvísindastofnun Háskóla Íslands
Árnagarði við Suðurgötu
IS-101 Reykjavík
ISLAND

Margrethe H. Møller
Institut for erhvervsforskning
Handelshøjskole syd
Gråbrødregade 2
DK-6000 Kolding
DANMARK

Magne Myhren
Norsk leksikografisk Institutt
Avd. for nynorsk
Universitetet i Oslo, Boks 1021
Blindern 0315 Oslo 3
NORGE

Søren Juul Nielsen
Institut for Datalingvistik
Dalgas have 15, 2Ø.083, 200
Frederiksberg
DANMARK

Sigurd Nordlie
Norsk leksikografisk Institutt
Universitetet i Oslo, Boks 1021
Blindern 0315 Oslo 3
NORGE

AnnCharlotte Nordstrøm
IBM Nordiska Laboratorier
Box 962
S-181 09 Lidingø
SVERIGE

Ole Norling-Christensen
DANLEX-Gruppen
& Gyldendals Ordbøger
Postboks 11 (Pilestræde 51)
DK-1001 København K
DANMARK

Stig Örjan Ohlsson
Københavns Universitet
Det humanistiske Edb-center
Njalsgade 80
DK-2300 København S
DANMARK

Susanne Nøhr Pedersen
EUROTRA – DK
Københavns Universitet
Njalsgade 80
DK-2300 København S
DANMARK

Jóhan Hendrik W. Poulsen
Føroyamálsdeildin
Fróðskaparsetur Føroya
FR-100 Tórshavn
FØROYAR

Pétur Rasmussen
Menntaskólinn við Sund
Gnoðarvogi
IS-104 Reykjavík
ISLAND

Eva Rode
Ordbog over det norrøne prosasprog
Københavns Universitet, Amager
Njalsgade 76, DK-2300 København S
DANMARK

Cristopher Sanders
Ordbog over det norrøne prosasprog
Københavns Universitet, Amager
Njalsgade 76, DK-2300 København S
DANMARK

Klaus Schubert
BSO/Research
Postbus 8348
NL-3503 RH Utrecht
HOLLAND

Bengt Sigurd
Institutionen för lingvistik
Lunds Universitet
Helgonabacken 12
S-223 62 Lund
SVERIGE

Sigurður Jónsson
Bókaútgáfan Iðunn
Bræðraborgarstíg 16
IS-101 Reykjavík
ISLAND

Stefán Briem
Orðabók Háskóla Íslands
Árnagarði við Suðurgötu
IS-101 Reykjavík
ISLAND

Bertha Sørensen
Institut for erhvervssprog,
Handelshøjskole Syd
Østervang 2
DK-6800 Varde
DANMARK

Anna Sågvall Hein
Språkdata
Göteborgs Universitet
S-412 98 Göteborg
SVERIGE

Torben Thrane
Københavns Universitet
Det humanistiske Edb-center
Njalsgade 80
DK-2300 København S
DANMARK

Ole Togeby
EUROTRA – DK
Københavns Universitet
Njalsgade 80
DK-2300 København S
DANMARK

Ivar Utne
Nordisk institutt,
Universitetet i Bergen
Strømgt 53
N-5007 Bergen
NORGE

Vilhjálmur Sigurjónsson
Bókaútgáfan Iðunn
Bræðraborgarstíg 16
IS-101 Reykjavík
ISLAND

Katri Vuorela
Uppsala Universitet
Finsk-ugriska Institutionen
Box 513
S-751 10 Uppsala
SVERIGE

Martin Zachariassen
Teldudeildin
Fróðskaparsetur Føroya
J.C. Svabosgøta 7
FR-100 Tórshavn
FØROYAR

Petur Zachariassen
Teldudeildin
Fróðskaparsetur Føroya
J.C. Svabosgøta 7
FR-100 Tórshavn
FØROYAR

Jordan Zlatev
Institutionen för lingvistik
Stockholms Universitet
S-106 91 Stockholm
SVERIGE

Annette Östling
Lundagatan 44
uppg. 4, 2 tr.
S-117 27 Stockholm
SVERIGE