Eva Ejerhed and Hank Bromley

A SELF-EXTENDING LEXICON: DESCRIPTION OF A WORD LEARNING PROGRAM

1. Introduction.

This paper describes the current implementation of the lexical analyzer (MORPH), of a finite state parser for Swedish (SWEDISH-SYNTAX). The MORPH program was developed by the authors in August 1985, and it is implemented in ZLISP on an LMI CADR Lisp machine. MORPH is a computational model of the recognition and analysis of written words, with the following brief characteristics:

MORPH is a word analyzer only, it does not synthesize or generate word forms.

- MORPH works left to right through a word, without pre-processing it in the form of suffix or prefix stripping.
- MORPH models both the process of word recognition/analysis and the process of word acquisition.
- MORPH works by combining the advantages of the two general methods for word recognition that are available: matching input words with words, and parsing input morphemes into words. By combining these methods, the disadvantages of using either method alone are avoided.

2. Considerations in designing MORPH.

As stated above, MORPH is a component of a larger system that is a parser for Swedish. At the Fourth Meeting of Scandinavian Computational Linguistics 1983 in Uppsala, I gave an overview of SWEDISH-SYNTAX, its organization and mode of operation (se also Church 1982, Ejerhed & Church 1983, and Ejerhed 1985). Since then, the work in Umeå on this system has been devoted to three things: (i) increasing the range of the syntactic constructions parsable by SWEDISH-SYNTAX by extending the grammar; (ii) adding a semantics component that assigns function-argument structures to a surface sentence on a given syntactic analysis of it; and (iii) increasing the number of words parsable by SWEDISH-SYNTAX.

A self-extending lexicon: description of a word learning program
Eva Ejerhed, Hank Bromley
Proceedings of NODALIDA 1985, pages 59-72

The version of SWEDISH-SYNTAX that I described in 1983 had a full form lexicon which listed each inflected form of a word along with its part of speech and morpho-syntactic features, e.g.

```
(defslex flicka noun +utr -pl -def)
(defslex flickan noun +utr -pl +def)
(defslex flickor noun +utr +pl -def)
(defslex flickorna noun +utr +pl +def)
```

The parser uses the part of speech information of a word in deciding constitutent structure for an input string, and it uses feature information when checking that agreement constraints are satisfied. Agreement in gender, number and definiteness between adjacent words in a noun phrase was and still is handled in SWEDISH-SYNTAX by a mechanism that filters out any combinations of non-agreeing words,like

```
* ett      vacker     flickan
  -utr      +utr       +utr
  -pl       -pl        -pl
  -def      -def       +def
```

This ensures that all noun phrases that are well formed with respect to constituent structure constraints (e.g. NP -- DET ADJECTIVE* NOUN), are also well formed with respect to agreement constraints. The way this works in SWEDISH-SYNTAX is not technically by unification, but by imposing the requirement that for immediate constituents of a noun phrase in Swedish no conflicts of values are allowed for the features of gender, number and definiteness. In the example above, there is both a gender conflict and a definiteness conflict For a more detailed technical description of the general properties of the agreement mechanism, see Church 1983, and for its application to Swedish syntax, see Ejerhed 1983.

We wanted to increase by a large amount the number of words that could be successfully dealt with by SWEDISH-SYNTAX, because a very large, and preferably self-extending, lexicon is necessary for most imaginable applications of natural language processing systems, such as natural language interfaces to data bases, or automatic information gathering (knowledge acquisition).

In designing MORPH, there were considerations of a theoretical as well as a practical nature. They were the following.

First, we wanted to test the hypothesis, that the proper role, and the only role, of morphology in word perception is in the processing and learning of unknown words. Unknown word processing proceeds by "decomposition first, retrieval second", to use the theoretical terminology of Job & Sartori 1984 in characterizing a possible model of word processing. The processing of known words, by contrast and on our hypothesis, proceeds by retrieval of the entire word and its associated properties (part of speech, features, and morphological decomposition) from a list ("retrieval first, decomposition second"). The source of this hypothesis, and the hybrid model of word processing, was research on speech synthesis (Byrd & Chodorow 1985. Church 1985). Associating pronunciations with written words is generally considered to be mediated by two processes, one that retrieves the stored pronunciation of a word, another that derives it from general letter to sound rules. Church 1985 gives a good argument why the task of associating pronunciations with words cannot be reduced entirely to either listing them or deriving them:

"Both approaches have their advantages and disadvantages; the dictionary approach fails for unknown words (e.g. proper names) and the letter to sound approach fails when the word does not follow the rules, which happens all too often in English. Most speech synthesizers adopt a hybrid strategy, using the dictionary when appropriate and letter to sound for the rest."

Translating this to the domain of syntactic and semantic analysis, we observed that the approach of listing full words will also fail for unknown words (e.g. the very large numbers of Swedish compounds, which are written as single graph words), and the approach of parsing all words will fail because of overrecognition (cf. the contribution of Doherty, Rankin, & Wirén to this conference). A hybrid model therefore seemed worth investigating, as an alternative to the currently popular full listing models in psycholinguistics and

computational linguistics (e.g. Wehrli 1985). For an overview of psycholinguistic work on morphology, see Henderson 1985, and for a set of recent studies of morphological constraints on word recognition, see Jarvella, Job, Sandström & Schreuder (forthcoming).

Second, we wanted a psycholinguistically and computationally plausible model of how words are entered into the lexicon of known words of a language user, as a function of language use. Most existing computational models of natural language deal with the processing of sentences relative to a fixed body of language knowledge, rather than with the acquisition of that knowledge. People acquire new knowledge about a language as part of the process of using it, and we have to try to simulate at least some aspects of language learning behavior on computers, it we want to shed further light on learning by people, and if we want to make computers more useful, languagewise.

Third, we wanted a unidirectional model of word recognition only, and not a bidirectional model of both word recognition and generation. There was a performance theoretic reason for this, as well as a practical one. In studies of language processing (e.g. Deutsch & Jarvella 1984), tasks involving perception and tasks involving production of the same linguistic structures have shown interesting differences, which undermines the idea that it is exactly the same knowledge of language (linguistic competence) that is at work in both performance processes. The practical consideration in modeling only word analysis was that it minimized the problem.

Fourth, and another practical consideration, was the limited time available for designing, implementing and testing the new lexical component, a total of four weeks.

Fifth, and last, the new lexical component had to provide information about words in a way that other components of SWEDISH-SYNTAX could use. In order for the lexical lookup routines to be of any use to the parser when presented with a written word, they have to serve the parser with the word plus its part of speech and features. In the case of ambiguous words, they have to hand the parser all of the analyses of the ambiguous word plus the information associated with each analysis of it. This meant that the names for parts of speech

and morpho syntactic features were already established, and prevented us from using directly any pre-exisiting morphological analyzer, such as Blaberg's two-level morphology for Swedish (Bläberg 1984), or Hellberg's Swedish morphology in the implementation of Doherty, Rankin & Wirén (this volume).

3. Description of the morphological analyzer.

3.1. Overview.

The two lexicon model of word processing that we arrived at has the following technical characteristics. There are two lexica, one of full words, called **word-lexicon**, and one of morphemes, called **morpheme-lexicon**. We will describe the morpheme-lexicon first.

3.2. The morpheme-lexicon.

The morpheme-lexicon is a lexicon of known morphemes, corresponding to the morphological decompositions of words given in the word-lexicon. The morpheme lexicon is a **hash table** (association list) with morphemes as **keys** and features as **values**. The advantage of this representation over other representations (discrimination networks, or treating words as Lisp atoms) is that the hash table enables a group of values to be associated with a single key and retrieved all at once. Since morpheme and word ambiguity is common, that is useful. Further, the morpheme-lexicon is sorted by values, meaning that morphemes with the same features are grouped together. The following is an example of the structure of entries in the morpheme-lexicon (note that "flick" here is a morpheme, not what Hellberg 1978 and others (Källgren 1986) have called a technical stem):

KEY	VALUE			
"flick"	(*	N	-WF	+UTR)
	1	2	3	4
	COMBINATORIAL FEATURES		BINARY FEATURES	

We will now describe the combinatorial features and the binary features of a morpheme. The former encode the combinatorial possibilities of the morpheme, and the latter are strictly binary features.

Position 1. The first combinatorial feature encodes what part of speech the morpheme combines with on its left, what it takes. The options are:

*	morpheme takes any part of speech on its left
NIL	morpheme takes nothing on its left
NOUN	morpheme takes specified part of speech on its left
VERB	
ADJ	
PREP	
ADV	
...	

Position 2. The second combinatorial feature encodes what part of speech a morpheme makes. The options are:

NOUN
VERB
ADJ
PREP
ADV
...

Position 3. The third combinatorial feature encodes whether or not the morpheme can occur in word final position. There are just two options:

-WF	morpheme combines with something on the right, i.e. it is non word final
+WF	morpheme combines with nothing on the right, i.e. it is word final

The word-finality feature enabled us to do morphological analysis without zero morphemes, which seemed desirable. An example of how it is used is the following:
sko / __ #(w b) has features (* NOUN +WF +UTR -PL -DEF)
sko / __ +(m b) has features (* NOUN -WF +UTR) and no more.
It will get the features for number and definiteness from what it combines with, e.g. **sko+n, sko+r, sko+r+na.**

Position 4. Binary features. For the noun, adjective and verb system, they are:

+UTR	+AUX
-UTR	-AUX
+PL	+REAL
-PL	-REAL
+DEF	+FIN
-DEF	-FIN
	+PRS
	-PRS

The binary features are strictly binary, i.e. the complement of +UTR is -UTR, etc. Binary features are always fully specified, thus there are no redundancy rules that mediate between partially and fully specified feature values (the absence of a specification for a feature means that the item can pass the requirement of both the value + and the value - for that feature). There were two reasons for this, one being that it is doubtful that redundancy rules are of any practical use in a processing model, the other being that the absence of redundancy rules facilitates debugging and extending the lexicon. At any point of working with it, the feature specifications you see is what the processor gets, too. Below are samples of groups of entries of lexical morphemes and partial entries of some grammatical morphemes.

```
"flick"   ( *  N   -WF   +UTR)
"gryt"    ( *  N   -WF   +UTR)
"klock"   ( *  N   -WF   +UTR)
"pojk"    ( *  N   -WF   +UTR)
"drull"   ( *  N   -WF   +UTR)
"mån"     ( *  N   -WF   +UTR)
"himl"    ( *  N   -WF   +UTR)

"a" ( (A  A  +WF  +UTR  +PL  +DEF)   a / de söt+a tå+r+na
      (A  A  +WF  +UTR  +PL  -DEF)   a / söt+a tå+r
      (A  A  +WF  +UTR  -PL  +DEF)   a / den söt+a tå+n
      (A  A  +WF  -UTR  +PL  +DEF)   a / de söt+a bi+n+a
      (A  A  +WF  -UTR  +PL  -DEF)   a / söt+a bi+n
      (A  A  +WF  -UTR  -PL  +DEF)   a / det söt+a bi+et
      (N  N  -WF  +UTR  -PL)         a / flick+a+n
```

```
        (N  N  +WF  +UTR  -PL  -DEF)    a / flick+a
        (N  N  +WF  -UTR  +PL  +DEF)    a / bi+n+a
        (V  V  +WF  -AUX +REAL -FIN +PRS)  a / verk+a (inf)
        (V  V  +WF  -AUX -REAL) )       a / verk+a (impv)

"n" ( (N  N  +WF  +UTR  -PL  +DEF)    n / flick+a+n
      (N  N  +WF  -UTR  +PL  -DEF)    n / bi+n

"or" ( (N  N  -WF  +UTR  +PL)         or / flick+or+na
       (N  N  +WF  +UTR  +PL -DEF) ) or / flick+or

"na" ((V  A  +WF  +UTR  +PL  +DEF)    na / de druck+na ko+r+na
      (V  A  +WF  +UTR  +PL  -DEF)    na / druck+na ko+r
      (V  A  +WF  +UTR  -PL  +DEF)    na / den druck+na ko+n
      (V  A  +WF  -UTR  +PL  +DEF)    na / de druck+na bi+n+a
      (V  A  +WF  -UTR  +PL  -DEF)    na / druck+na bi+n
      (V  A  +WF  -UTR  -PL  +DEF)    na / det druck+na bi+et
      (N  N  +WF  +UTR  +PL  +DEF))   na / flick+or+na
```

The general approach to specifying word structure in MORPH is exceedingly simple. Word = morpheme* minus those sequences of morphemes ruled out by the morphotactic constraints encoded in the combinatorial features. That those features go a long way towards specifying admissible sequences in real cases is illustrated by the following example of three morphemes and their specifications:

```
"av"  ( (NIL PREP -WF)          av / av+led+a
        (NIL PREP +WF) )        av / av flick+a+n
"led" ( (*   V   -WF)           led / led+er
                                      av+led+er
                                      bransch+led+ande
        (*   V   +WF -REAL) )   led / led!
"ning"( (V   N   -WF)           ning / led+ning+en
        (V   N   +WF +UTR -PL -DEF) ) ning / led+ning
```

Given this set of three morphemes **av, led, ning,** the maximal number of theoretically possible words (up to trimorphemic words) made out of them are: $3 + 3^2 + 3^3 = 39$.

3 monomorphemic words:

OK av, OK led, *ning

9 bimorphemic words:

*avav	*ledav	*ningav
OK avled	*ledled	*ningled
*avning	OK ledning	*ningning

27 trimorphemic words:

*avavav	*avledav	*avningav
*avavled	*avledled	*avningled
*avavning	OK avledning	*avningning
*ledavav	*ledledav	*ledningav
*ledavled	*ledledled	*ledningled
*ledavning	*ledledning	*ledningning
*ningavav	*ningledav	*ningningav
*ningavled	*ningledled	*ningningled
*ningavning	*ningledning	*ningningning

The specifications given for the three morphemes in each instance makes the correct prediction whether the word is admissible or inadmissible, which suggests that the framework is adequate for capturing such classical notions of morphology as prefix, root, and suffix, and the distinction between free and bound morphemes. However, we do not regard the framework for specifying admissible sequences as a complete and definitive proposal. It will have to undergo revisions in the amount and nature of morphological structure that it attributes to words.

3.3 The word-lexicon.

The word-lexicon is a lexicon of known words. It is also a hash table, with words as keys and their parts of speech, feature properties and morphological decomposition as values. It is sorted alphabetically. In the current version there is no semantics attached to either morphemes or words, but we plan to introduce this in the following way. Each morpheme will always have semantic information associated with it. Each word that has an idiosyncratic and non- compositional semantics will have that semantics associated with the word as a whole. However, most composite words have a straightforward compositional semantics, and for those there will be no listing in the word-lexicon of the semantics for the word as a whole. For such words, the semantics will be retrieved from each constituent morpheme via the morphological decomposition

(in accordance with a hypothesis of word perception of Caramazza et al 1985).
Below are examples of the structure of entries in the word-lexicon.

```
"flicka" (((NOUN +WF -PL -DEF +UTR)) ("flick" "a"))
"flickan" (((NOUN +WF +DEF -PL +UTR)) ("flick" "a" "n"))
"flickor" (((NOUN +WF +PL -DEF +UTR)) ("flick" "or"))
"flickorna" (((NOUN +WF +DEF +PL +UTR)) ("flick" "or" "na"))
```

These are copied with one minor change from the current entries in the word-lexicon, and they reveal an unnecessary feature of each entry, +WF. Since every word, by definition is word final, this feature should be removed. The entries above are very simple and do not illustrate what the entries of words with ambiguous analyses look like. Below are two examples.

```
"betalade" (((VERB +WF -AUX -PRS +FIN +REAL)
             ("be" "tal" "ade"))
            ((ADJECTIVE +WF +UTR +PL +DEF)
             (ADJECTIVE +WF +UTR +PL -DEF)
             (ADJECTIVE +WF +UTR -PL +DEF)
             (ADJECTIVE +WF -UTR +PL +DEF)
             (ADJECTIVE +WF -UTR +PL -DEF)
             (ADJECTIVE +WF -UTR -PL +DEF)
             ("be" "tal" "a" "d" "e")))

"bildrulle" (((NOUN +WF -PL -DEF +UTR) ("bil" "drull" "e"))
             ((NOUN +WF -PL -DEF +UTR) ("bild" "rull" "e")))
```

4. Using MORPH.

The central function in the MORPH program is called lookup-word, and it has the following easy to understand definition:

```
(defun lookup-word  (word)
 (cond ((lookup-word-in-lexicon word))
       ((analyze-word-and-add-to-lexicon word))
       ((ask-about-word-and-add-to-lexicon word)) ))
```

The function lookup-word, when applied to a word, first tries to find it in the word-lexicon, and if successful, returns its values as its value. If unsuccessful, it invokes the function analyze-word-and-add-to-lexicon. This function analyzes the word character by character from left to right and finds all

internall, consistent morphological decompositions of it, relative to the current morpheme-lexicon. These analyses are returned as the value of the function, and the result incorporated in the word-lexicon. If the function analyze-word-... is unsuccessful, the program asks the user to supply information about the word, as a last resort. Several utilities for the manual extension of the lexica are included in MORPH, in order to minimize the amount of typing done. In that sense it is a good example of a lexicon writer's workbench. For example, if a new morpheme is exactly like a previously known one, with respect to part of speech and features (and degree and nature of lexical ambiguity), then this information is automatically copied when typing the morpheme it is like, at the appropriate point in the interactive sequence.

As intended, MORPH has replaced the old lexicon of SWEDISH-SYNTAX, and the interaction works out well. For example, in the case of an ambigous word like "betalade" above, the new lexicon hands the parser all of the analyses, and the parser picks the appropriate one according to the context in which the word occurs. The following tree diagrams produced by SWEDISH-SYNTAX illustrate the interaction between the lexicon and the parser.

Fig. 1 Analys av
Hon betalade boken.

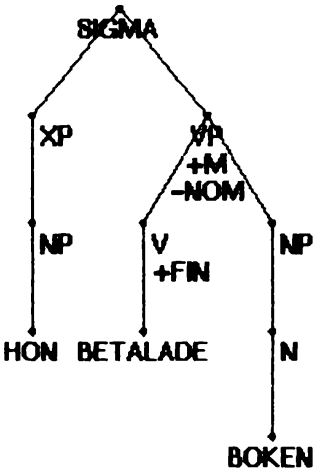
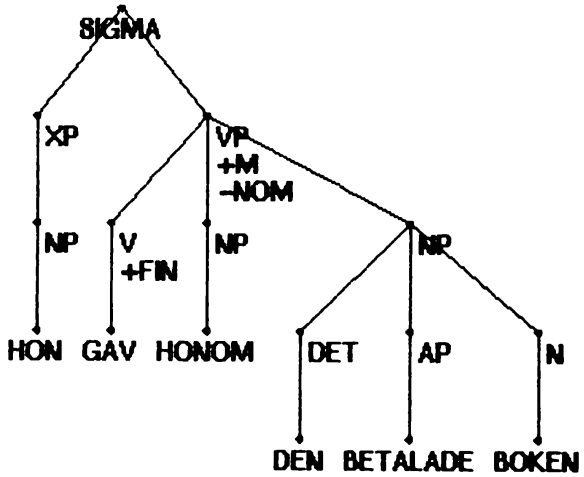


Fig. 2 Analys av
Hon gav honom den betalade boken.



Finally, we would like to present some preliminary figures that show how long it takes for the program to do lookup-word by lookup-word-in-lexicon (R = retrieval) on the one hand, and by analyze-word-and-add-to-lexicon (A = analysis) on the other hand. The words we tested are presented in descending order of morphological complexity, and the last word is an example of an ambiguously decomposable word.

#kyrk+klock+s+in+vig+ning#	A	0.219 sec
	R	0.004 sec
#bibliotek+s+in+vig+ning#	A	0.215 sec
	R	0.004 sec
#industri+av+veckl+ing#	A	0.313 sec
	R	0.004 sec
#ut+veckl+ing#	A	0.089 sec
	R	0.005 sec
#barn+cykel+nyckel#	A	0.223 sec
	R	0.004 sec
#till+tal+ande#	A	0.227 sec
#från+tag+ning#	A	0.065 sec
#kyrk+tag+ning#	A	0.079 sec
#bil+drull+e# #bild+rull+e#	A	0.068 sec
	R	0.003 sec

ACKNOWLEDGEMENTS

The work reported here was supported by a grant from the Tercentenary Foundation of the Bank of Sweden. We are grateful to the following persons for discussing issues of computational morphology with us, and for making contributions to the improvement of MORPH: Olli Bläberg, Robert Jarvella, Remo Job, Fred Karlsson, Lauri Karttunen, Kimmo Koskenniemi, Görel Sandström and Rob Schreuder.

NOTES

1. The old lexical analyzer was actually more intelligent than this description suggests, in that it enabled a grouping together of inflected forms of the same lexical item.

2. The position that morphology only plays a role in the processing of unknown words, and no role at all in the processing on known words is extreme, and it was deliberately chosen in order to investigate its consequences for a language processing system as a whole. In view of the fact that morphological structure appears to play a role in the processing of known (=high frequency) words, as well as unknown (=low frequency) words, we would now like to qualify our original hypothesis in one respect that concerns semantics. With respect to processing unknown words, the hypothesis stays the same. But with respect to the processing of unknown words, it is revised so that retrieval yields the entire word, its part of speech, its syntactic features and its morphological decomposition, but not the semantics of the word as a whole, in the case of normal, semantically compositional words. Only in the case of semantically non-compositional words (and phrases) does the word-lexicon carry the semantics of the word as a whole. This would imply that morphological structure is used as a control structure for semantic integration, in the case of compositional words.

REFERENCES

Bläberg, O., 1984, Svensk böjningsmorfologi - en tvånivåbeskrivning, Helsingfors universitet.

Byrd, R. & Chodorow, M., 1985, Using an on-line dictionary to find rhyming words and pronunciations for unknown words, Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics, pp 277-283.

Caramazza, A., Miceli, G., Silveri, M. & Laudanna, A., 1985, Reading mechanisms and the organisation of the lexicon: evidence from acquired dyslexia, Cognitive Neuropsychology 2(1), pp 81-114.

Church, K., 1982, A framework for processing finite state grammars of Swedish and English syntax, Report from the Department of General Linguistics, University of Umeå.

Church, K., 1983, Phrase-structure parsing: a method for taking advantage of allophonic constraints, Indiana University Linguistics Club.

Church, K., 1985, Stress assignment in letter to sound rules for speech synthesis, Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics, pp 246-253.

Deutsch, W. & Jarvella, R., 1984, Asymmetrien zwischen Sprachproduktion und Sprachverstehen, in C.F. Graumann & T. Herrmann (eds), Karl Bühlers Axiomatik - Fünfzig Jahre Axiomatik der Sprachwissenschaften, Frankfurt an Main, Klostermann, pp 173-199.

Doherty, P., Rankin, I. & Wirén, M., 1986, Erfarenheter av en implementering av Hellbergs system för svensk morfologi, i F. Karlsson (utg), Föredrag från V Nordiska Datalingvistikdagarna, 11-12 december, 1985, Helsingfors.

Ejerhed, E., 1983, Kongruens i en svensk finite state parser, föredrag vid 4:e svenska kollokviet i språklig databehandling - datorsimulering av verbalt beteende, 26 maj 1983, Lund.

Ejerhed, E., 1985, En ytstrukturgrammatik för svenska, i S. Allén et al (utg), Svenskans beskrivning 15, Institutionen för nordiska språk, Göteborgs universitet, s 175-192.

Ejerhed, E. & Church, K., 1983, Finite State Parsing, i F. Karlsson (ed), Papers from the Seventh Scnadinavian Conference of Linguistics, University of Helsinki, Department of General Linguistics, Publication No. 10 (II), pp 410-432.

Hellberg, S., 1978, The morphology of present day Swedish, Stockholm, Almqvist & Wiksell International.

Henderson, L., 1985, Toward a psychology of morphemes, in A.W. Ellis (ed), Progress in the psychology of language, Vol. 1, London, Lawrence Erlbaum Associates.

Jarvella, R., Job, R., Sandström, G. & Schreuder,R. (forthcoming), Morphological constraints on word recognition, Department of General Linguistics, University of Umeå.

Job, R. & Sartori, G., 1984, Morphological decomposition: evidence from crossed phonological dyslexia, The Quarterly Journal of Experimental Psychology 36A, pp 435-458.

Källgren, G., 1986, The role of the lexicon in heuristic parsing, paper presented at the 9th Scandinavian Conference of Linguistics, University of Stockholm, January 9-11, 1986.

Wehrli, E., 1985, Design and implementation of a lexical data base, Proceedings of the Second Conference of the European Chapter of the Association for Computational Linguistics, University of Geneva, pp 146-153.