Parsing without lexicon: the MorP system

Gunnel Källgren
University of Stockholm
Department of
Computational Linguistics
S-106 91 Stockholm
Sweden
gunnel@com.qz.se
gunnel@ling.su.se

Abstract

MorP is a system for automatic word class assignment on the basis of surface features. It has a very small lexicon of form words (‰ entries), and for the rest works entirely on morphological and configurational patterns. This makes it robust and fast, and in spite of the (deliberate) restrictedness of the system, its performance reaches an average accuracy level above 91% when run on unrestricted Swedish text.

Keywords: parsing, morphology.

The development of the parser to be presented has been supported by the Swedish Research Council for the Humanities. The parser is called MorP, for morphology based parser, and the hypotheses behind it can be formulated thus:

a) It is to a large extent possible to decide the word class of words in running text from pure surface criteria, such as the morphology of the words together with the configurations that they appear in.

b) These surface criteria can be described so clearly that an automatic identification of word class will be possible.

c) Surface criteria give signals that will suffice to give a word class identification with a level of around or above 90% correctness, at least for a language with as

much inflectional morphology as Swedish.

A parser was constructed along these lines, which are first presented in Brodda (1982), and the predictions of the hypotheses were found to hold fairly well. The project is reported in publications in Swedish (Källgren 1984a) and English (Källgren 1984b, 1985, 1991a) and the parser has been tested in a practical application in connection with information retrieval (Källgren 1984c, 1991a). We also plan to use the parser in a project aimed at building a large tagged corpus of Swedish (the SUC corpus, Källgren 1990, 1991b). The MorP parser is implemented in a high-level string manipulating language developed at Stockholm University by Benny Brodda. The language is called Beta and fuller descriptions of it can be found in Brodda (1990). The version of Beta that is used here is a PC/DOS implementation written in Pascal (Malkior-Carlvik 1990), but Macintosh and DEC versions also exist.

The rules of the parser are partitioned between different subprograms that perform recognition of different surface patterns of written language. The first programs work on single words and segments of words and add their analy-

sis directly into the string. Later programs look at the markings in the string and their configurations. The programs can add markings on previously unmarked words, but can also change markings inserted by earlier programs. The units identified by the programs are word classes and two kinds of larger constituents: noun phrases and prepositional phrases. The latter constituents are established mainly as a step in the process of identifying word class from contextual criteria. After the processing, the original string is restored and the final result of the analysis is given in the form of tags, either after or below the words or constituents.

An interesting feature of the MorP parser is its way of handling non-deterministic situations by simply postponing the decision until enough information is available. The postponing of decisions is partly done with the use of ambiguous word class markers that are inserted wherever the morphological information signals two possible word classes. Hereby, all other word classes are excluded, which reduces the number of possible choices considerably, and later programs can use the information in the ambiguous markers both to perform analysis that does not require full disambiguation and to ultimately resolve the ambiguity.

AN EVALUATION OF THE PARSER

In an evaluation of the MorP parser, two texts of which there exists a manual tagging were chosen and cut at the first sentence boundary after 1,000 words. The texts were run through the MorP parser and the output was compared to the manual tagging of the texts.

MorP was run by a batch file that calls the programs sequentially and builds up a series of intermediate outputs from each program. Neither the programs themselves nor this mode of running them has in any way been optimized for time, e.g., unproportionally much time is spent on opening and closing both rule files and text files. To run a full parse on an AT/386 took 1 minute 5 seconds for one text (1,006 words), giving an average of 0.065 sec/word, and for the other text (1,004 words) it took 1 minute 1 second, average 0.061 sec/word. With 10,000 words, the average is 0.055 sec/word. The larger amounts of text that can be run in batch, the shorter the relative processing time will be, and if file handling were carried out differently, time would decrease considerably. The figures for runtime could thus be much improved in several ways in applications where speed was a desirable factor.

In evaluating the accuracy of the output, single tagged words have been directly compared to the corresponding words in the manually tagged texts. When complex phrases are built up, their internal analysis is successively removed when it has played its role and is of no more use in the process. The tags of words in phrases are thus evaluated in the following way: If a word has had an unambiguous tag at an earlier stage of the process that has been removed when building up the phrase, that tag is counted. (Earlier tags can be seen in the intermediate outputs.) If a word has had no tag at all or an ambiguous one and then been incorporated into a phrase, it is regarded as having the word class that the incorporation presupposes it to have. That tag is then compared to that of the manually tagged text.

The errors can be of three kinds: erroneous word class assignment,

unsolved ambiguity, and no assignment at all, which is rather a special case of unsolved ambiguity, cf. below. The figures for the three kinds are given below.

Table of results of word class assignment

	Number of words	Correct word class N	%	Number of errors Wrong	Zero	Ambig.
Text 303	1,006	920	91.5	54	29	3
Text 402	1,004	917	91.3	43	40	4
Total	2,010	1,837	91.4	97	69	7

These results are remarkably good, in spite of the fact that many other systems are reported to reach an accuracy of 96-97%. (Garside 1987, Marshall 1987, DeRose 1988, Church 1988, Ejerhed 1987, O'Shaughnessy 1989.) Those systems, however, all use 'heavier artillery' than MorP, that has been deliberately restricted in accordance with the hypotheses presented above. This restrictiveness concerns both the size of the lexicon and the ways of carrying out disambiguation. It is always difficult to define criteria for the correctness of parses, and the MorP parser must be judged in relation to the restrictions and the limited claims set up for it.

All, or most, errors can of cause be avoided if all disturbing words are put in a lexicon, but now the trick was to get as far as possible with as little lexicon as possible. Rather than trimming the parser by increasing the lexicon, it should first be evaluated as it is, and in accordance with its basic principles, before any amendments are added to it. It should also be noted that MorP has been tested and evaluated on texts that are quite different from those on which it was first developed.

If we look at the roles that different parts of the MorP parser play in the analysis, we see that the lexical rules (which are only 435 in number) cover 54% of the 2,010 running words of the texts. The two texts differ somewhat on this point. One of them (text 402) contains very many quantifiers which are found in the lexicon, and that text has 58% of its running words covered. Text 303 has 50% coverage after the lexical rules, a figure that is more "normal" in comparison with my earlier experiences with the parser. As can be seen from the table, the higher proportion of words covered by lexicon in text 402 does not have an overall positive effect on the final result. The fact that a word is covered by the lexical rules is by no means a guarantee that it is correctly identified, as the lexicon only assigns the most probable word class.

The first three subprograms of MorP work entirely on the level of single words. After they have been run, disambiguation proper starts. The MorP output in this intermediate situation is that 75% of the running words are marked as being unambiguous (though some of them later have their tags changed), 11% are marked as two-ways ambiguous, and 14% are unmarked. In practice, this means that the latter are four-ways ambiguous, as they can finally come out as nouns, verbs, or adjectives, or remain untagged.

The syntactic part of MorP, covered by four subprograms, performs both disambiguation and identification of previously unmarked words, which, as stated above, can be seen as a generalization of the disambiguation process. This part is entirely based on linguistic patterns rather than statistical ones. Of course, there is "statistics" in the disambiguation rules as well as in the lexical assignment of tags, in the sense that the entire system is an implementation of my own intuitions as a native speaker of Swedish, and such intuitions certainly comprise a feeling for what is more or less common in a language. Still, MorP would certainly gain a lot if it were based on actual statistics on, e.g., the structure of noun phrases or the placement of adverbials. The errors arising from the application of syntactic patterns in the parsing of the two texts however rarely seem to be due to occurrence of infrequent patterns, but more to erroneous disambiguation of the words that are fitted into the patterns.

Next, I will give a few examples from the texts of the kind of errors that will typically occur with a simplified system like MorP. Errors can arise from the lexicon, from the morphological analysis, from the syntactic disambiguation, and from combinations of these. In text 402, there is also a misspelling, the non- existent form **utterst** for the adverb **ytterst** 'ultimately'. This is correctly treated as a regularly formed adverb, which shows some of the robustness of MorP.

We have only a few instances in these texts where a word has been erroneously marked by the lexicon. Most notorious is the case with the word **om** that can either be a preposition, 'about', or a conjunction, 'if'. It is marked as a preposition in the lexicon and a later rule retags it as a conjunction if it has not been amalgamated with a following noun phrase to form a prepositional phrase by the end of the processing. Mostly, however, it is impossible to decide the interpretation of the word **om** from its close context, as if-clauses almost always start with a subject noun phrase. In the two texts, **om** occurs 17 times, 9 times as a preposition and 8 times as a conjunction. One of the conjunctions is correctly retagged by the just mentioned rule, while the others remain uncorrected. Regrettably, one of the prepositions has also been retagged as a conjunction, as it is followed by a that-clause and not by a noun phrase. Of the 7 erroneously marked conjunctions, 3 are sentence-initial, while no occurrence of the word as a preposition is sentence-initial. A possible heuristic would then be to have a retagging rule for this position before the rules that build prepositional phrases apply. A remarkable fact is that none of the conjunctions **om** is followed by a later **så** 'then'. A long- range context check looking for 'if - then' expressions would thus add nothing to the results here.

The case with **om** is a good and typical example of a situation where more

statistics would be of great advantage in improving and refining the rules, but where there will always be a rest class of insoluble cases and cases which are contrary to the rules.

Still, there are not many words in the sample texts where the tagging done by lexicon is wrong. This is remarkable, as the lexicon always assigns exactly one tag, not a set of tags, even if a word is ambiguous.

The morphological analysis carries out a very substantial task and, consequently, is a large source of errors. One example is the noun **bevis** 'proof', which occurs several times in one of the texts. It has a very prototypical verbal look, with the prefix **be-**, a monosyllabic stem seemingly ending in a vowel and followed by a passive **-s**, exactly like the verbs **beses, begås, betros, bebos**, etc. It is just a coincidence that the verb is **bevisa**, not **bevi**, and the noun is formed by a rare deletion rather that by adding a derivational ending. A similar error is when the noun **resultat** 'result' is treated as a supine verb, as **-at** is a very common, very productive supine ending.

Disambiguation of course also adds many errors, as the patterns for those rules are less clear than the patterns for word structure, and as all errors, ambiguities and doubtful cases from earlier programs accumulate as the processing proceeds. Often it is the ambiguous-marked words that are disambiguated wrongly or not at all. In one of the texts there is for instance the alleged finite verb **djungler** 'jungles'. A foregoing adverb has caused the ambiguous ending **-er** to be classified as signalling present tense verb rather than plural noun. The remaining ambiguities also often belong to this class of words, but on the whole, it is surprising how few of the ambiguous-marked words that remain in the output.

The set of words that are still unmarked by the end of the process is comparatively large. A possible heuristic might be to make them all nouns, as that is the largest open word class, and as most singular and many plural indefinite nouns have no clear morphological characteristics in Swedish. A closer look at the unmarked words reveals that this is not such a good idea: of 69 unmarked words, 25 are nouns, 18 adjectives, and 18 verbs. One is a numeral, one is a very rare preposition that is a homograph of a slightly more common noun, 2 are adverbs with homographs in other word classes, and 2 are the first part of conjoined compounds, comparable to expressions like 'pre- or postprocessing'. The hyphenated first part gets no mark in these cases. They could be done away with by manual preprocessing, as also the not infrequent cases occurring in headlines, where syntactic structure is often too reduced to be of any help. For the rest, a careful examination of their word structure and context seems promising, but more data is needed.

By this, I hope to have shown that parsing without lexicon is both possible and interesting, and can give insights about the structure of natural languages that can be of use also in less restricted systems.

REFERENCES

Brodda, B. 1982. An Experiment in Heuristic Parsing, in **Papers from the 7th Scandinavian Conference of Linguistics**, Dec. 1982. Department of General Linguistics, Publication no. 10, Helsinki 1983.

Brodda, B. 1990. Do Corpus Work with PC Beta, (and) be your own Computational Linguist to appear in Johansson, S. & Stenström, A.- B. (eds): **English Computer Corpora**, Mouton-de Gruyter, Berlin 1990/91 (under publication).

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

DeRose, S.J. 1988. Grammatical category disambiguation by statistical optimization. **Computational Linguistics** Vol. 14:1.

Ejerhed, E. 1987. Finding Noun Phrases and Clauses in Unrestricted Text: On the Use of Stochastic and Finitary Methods in Text Analysis. MS, AT&T Bell Labs.

Garside, R. 1987. The CLAWS word-tagging system, in Garside, R., G. Leech & G. Sampson (eds.), 1987.

Garside, R., G. Leech & G. Sampson (eds.). 1987. **The Computational Analysis of English.** Longman.

Källgren, G. 1984a. HP-systemet som genväg vid syntaktisk märkning av texter, in **Svenskans beskrivning 14**, p. 39-45. University of Lund.

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

Källgren, G. 1984c. Automatisk excerpering av substantiv ur löpande text. Ett möjligt hjälpmedel vid automatisk indexering? **IRI-rapport 1984:1.** The Swedish Law and Informatics Research Institute, Stockholm University.

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

Källgren, G. 1990. "The first million is hardest to get": Building a Large Tagged Corpus as Automatically as Possible. **Proceedings from Coling '90.** Helsinki.

Källgren, G. 1991a. Making Maximal use of Surface Criteria in Large Scale Parsing: the Morp Parser, Papers from the Institute of Linguistics, University of Stockholm (PILUS).

Källgren, G. 1991b. Storskaligt korpusarbete på dator. En presentation av SUC-korpusen. **Svenskans beskrivning 1990.** University of Uppsala.

Malkior, S. & Carlvik, M. 1990. **PC Beta Reference.** Institute of Linguistics, Stockholm University.

Marshall, I. 1987. Tag selection using probabilistic methods, in Garside, R., G. Leech & G. Sampson (eds.), 1987.

O'Shaughnessy, D. 1989. Parsing with a Small Dictionary for Applications such as Text to Speech. **Computational Linguistics** Vol. 15:2.