Decomposing Swedish Compounds Using Memory-Based Learning

Karin Friberg
Department of Swedish Language
Göteborg University
Sweden
karin.friberg@svenska.gu.se

Abstract

Swedish morphology differs significantly from English in several ways. This is something which makes natural language processing based on the English language not always applicable for Swedish material. One area where there is a difference is compounding. The word-forming process of compounding is very productive in Swedish. The compounds are mostly written as one word, without the segmentation point marked in any way. Thus segmentation has to be done in order to interpret the compounds.

In this study I have implemented a decomposer which finds the segmentation point in Swedish compounds, making it easier to handle compounds in natural language processing. Brodda's algorithm for heuristic compound segmentation guided the work. The decomposer is implemented in TiMBL, a memory-based learner.

1 Introduction

Compounding is, in Swedish, a very productive process. The number of possible compounds is huge, and it is not possible to list them. The use of compounds and the coining of new compounds is also significant. 10% of the words in Swedish newspaper text have been found to be compounds after removal of stopwords (Hedlund, 2002). The frequency of and the structure of Swedish compounds poses problems in natural language processing tasks which have to be dealt with.

Areas where this is important are for example information retrieval, machine translation and speech synthesis. In information retrieval a search query may contain a term which in a certain document only occurs hidden in a compound. Or it can be the other way around. The query may contain a compound but the concepts of any or both of the constituents may in a certain document only occur as simplex words.

In order to make use of the constituents of Swedish compounds, one must first determine which they are. One problem for languages where compounds are written with no white space between the parts, as in Swedish, is to find the parts, that is to determine the compound segmentation point.

In this paper I describe an experiment using memory-based learning for compound segmentation. The hypothesis is that this should work because n-grams around a segmentation point tend to contain grapheme clusters that do not occur in simplex words (Brodda, 1979). The learner is thus trained to distinguish clusters appearing around segmentation points from other clusters.

2 Background

There are different approaches for constructing systems that deal with segmentation of Swedish compounds, including the use of word lists, manually written rules or machine learning. The simplest approach would be to list the compounds in a word list with the segmentation point marked. However, since the process of compounding is so productive, it would be difficult to get sufficient coverage with

such a system.

2.1 Previous work

Sjöbergh and Kann (2006) present a successful segmentation system which uses word lists, although not with whole compounds. The system consists of a *first part list*, a *last part list*, and an *individual word list*. Sjöbergh and Kann report 99% success for both precision and recall.

A well-known system for analyzing Swedish words, including compounds, is SWETWOL (Karlsson, 1992). SWETWOL is based on manually written rules, and uses the Two Level formalism. SWETWOL gives several alternative interpretations if found, and the analyses include part of speech of all segments. Lexicalized compounds and derivations have been listed in SWETWOL and the system regards these as wholes in order to avoid incorrect decomposition.

Witschel and Biemann (2005) present a segmentation system for German compounds using Compact Patricia Tries, a type of classifier which they themselves present as something between a rule-based and a memory-based learner. They report results around 96% for the F-value.

In his thesis Kokkinakis (2001) mentions a rule-based segmentation tool based on Brodda's segmentation algorithm describing clusters at compound segmentation points. (See section 2.2.) Kokkinakis reports 96% precision.

2.2 Brodda's Segmentation Algorithm

The present study presents a compound segmentation system that, as Kokkinaki's system, is inspired by Brodda's segmentation algorithm, but instead of manually written rules uses memory-based learning.

Brodda's algorithm concerns Swedish grapheme combinations. Like other languages, the Swedish language has rules stating which combinations of consonants and vowels can appear at the beginning of a word, at the end or internally. In his algorithm, Brodda (1979) states that when two Swedish words are combined together in a compound, the result is often a consonant cluster in the segmentation point that is not allowed internally in simplex words, for example the cluster '**rkskr**' in '*ko***rkskr***uv*' ('*corkscrew*'). Brodda describes a six level hierarchy of clusters, from those that say nothing about the probability of a segmentation point, like '**ll**', to those that always signal a segmentation point, like '**ntst**'. Brodda's algorithm identifies 50% of tested compounds uniquely with the correct segmentation and another 40% with multiple segmentation points. 10% of the compounds are missed (Brodda, 1979).

2.3 TiMBL

TiMBL (Daelemans et al., 2004) is a memory-based learner used here to classify positions in presumed compounds as being segmentation points or not.

A memory-based learner stores classified instances in memory and later, given test instances, compares these to the stored instances by calculating the distance or similarity between them. It then assigns the class of the majority of the nearest neighbors to the instance to be classified.

The properties of the instances are described in feature vectors in the form of a list of an arbitrary number of features, separated by commas, followed by a class label and ended by a period. (See Example 1 in section 3.1.)

After testing for optimal settings for the learner, I chose 'Overlap', which counts the number of features that coincide between two vectors, and 'MVDM' (Modified Value Difference Metric) which considers some values to be more alike than others, making the distance smaller. For example, an '*n*' can be considered to be more similar to an '*m*' than to a '*p*'. The parameter feature weighting concerns how much weight each feature is given. For this I used the setting 'Gain Ratio'which normalizes the weight considering the number of values the feature may take. Finally, there is class voting which is about how much influence each neighbor has. Here I chose 'Majority Voting' where the class with the largest part of the k nearest neighbors is chosen regardless of the distance.

3 Method

The aim of this study is to have TiMBL learn to classify every position in a string as a compound segmentation point or not.

To train a memory-based learner, data is needed to supply instances of all classes. In this case the classes are: segmentation point = Y, no segmenta-

tion point = N. To this end I used compounds from the on-line medical lexicon MedLex (Kokkinakis, 2004). I had at my disposal a list of 5 786 compounds with the segmentation point marked. Since MedLex is a medical lexicon constructed by adding medical terms to a learners' dictionary, a great part of these compounds, but far from all, are from the medical domain. I put aside one tenth of the compounds for testing and used nine tenths for training.

3.1 Feature vectors

The starting point of this study is, as mentioned, Brodda's segmentation algorithm (Brodda, 1979), which tells us that clusters not allowed in simplex words signal probable compound segmentation points. With this in mind I created feature vectors representing the graphemes around a supposed compound boundary.

The feature vector I used as baseline had the following components: the word in question (ignored in training and testing), four graphemes before and four graphemes after the supposed segmentation point, three graphemes before and three graphemes after the segmentation point, two graphemes before and two graphemes after the segmentation point, and finally the class label.

abstinens∼besvär,nens,besv,ens,bes,ns,be,Y.
abstinens∼besvär,inen,sbes,nen,sbe,en,sb,N.

Example 1. A positive (Y) and a negative (N) example of the baseline feature vector.

The graphemes closest to the segmentation point are repeated in several features in order to give graphemes closer to the compound boundary more weight and also to catch cases where the cluster around the segmentation point is smaller than four graphemes on either side.

I limited the negative cases of training instances to represent points one and two positions to the right and to the left of the predetermined compound boundary. For the batch testing below, the test cases were limited to testing five positions in each word, the supposed segmentation point and two positions to the left and two to the right of the boundary.

3.2 Experiments

I did some experimenting with the feature vector, ignoring a varying number of features. No combination of ignored features gave a positive result. I went on trying to group similar graphemes. Since the clusters Brodda mentioned were mostly consonant clusters, I replaced all vowels with one and the same symbol, hoping to get more matching of close neighbors. This had a devastating result. I tried the same approach with the so called heavy vowels '*å*', '*ä*', '*ö*', '*y*', '*u*' and '*i*' which are mostly found in stems (Brodda, 1979). This gave a somewhat better result, but again worse than with no vowel substitution.

I also tried a case of consonant substitution. Swedish phonotactics tells us that if a word starts with a cluster of three consonants the first one has to be '*s*', the second a voiceless plosive, '*p*', '*t*' or '*k*', and the last one '*r*', '*l*', '*j*' or '*v*' (Lundström-Holmberg and af Trampe, 1987). With this in mind I replaced all voiceless plosives with a common symbol. This did not have as bad result as the vowel substitutions, but was still no improvement to the baseline.

I finally tried extending the feature vector with features representing graphemes across the segmentation point. First, a feature consisting of one grapheme on each side (1+1). I then added another one with varying numbers of graphemes on either side of the segmentation point (2+2, 1+2, 3+1, etc.). This, finally, gave results improving on the original vector. (See Table 1 below.)

abstinens∼besvär,nens,besv,ens,bes,ns,be,sb,ensb,Y.
abstinens∼besvär,inen,sbes,nen,sbe,en,sb,ns,nens,N.

Example 2. The feature vector giving the best results, with boundary features 1+1 and 3+1.

4 Results

The outcome important in this case is 'positive recall', that is the procentage of found segmentation points and 'positive precision', the procentage of true segmentation points among the declared segmentation points. Positive recall is a measure that mirrors the procentage of missed positives and positive precision mirrors the procentage of false positives. In many cases having high recall is more essential than having high precision since a user getting superfluous results easily can discard the incor-

Feature properties	Missed positives	Recall (positive)	False positives	Precision (positive)
Baseline	41	92.9	40	93.1
Ignore feature 6	41	92.9	40	93.1
Ignore features 5 and 6	42	92.7	52	91.2
Ignore features 4 and 6	46	92.1	47	91.9
Ignore feature 2	57	90.2	57	90.2
Ignore feature 3	64	88.9	61	89.4
Vowel substitution	94	83.8	107	81.9
Heavy vowel substitution	64	88.9	71	87.9
Voiceless plosive substitution	47	91.9	51	91.3
Boundary feature 1+1	38	93.4	43	92.6
Boundary feature 1+1, 2+2	40	93.1	38	93.4
Boundary feature 1+1, 2+1	35	94.0	37	93.6
Boundary feature 1+1, 1+2	35	94.0	40	93.2
Boundary feature 1+1, 3+1	**32**	**94.5**	**36**	**93.8**
Boundary feature 1+1, 3+2	40	93.1	34	94.1
Boundary feature 1+1, 4+1	41	93.8	36	93.0

Table 1: The effect on precision and recall, when experimenting with the feature vector, testing on five positions in a total of 579 words.

rect ones. Furthermore, there are often multiple correct answers. Declared false positives are not seldom morpheme boundaries or alternative or even additional compound boundaries.

Looking at the false positives in this case, it is clear that the learner has not been able to detect all forbidden clusters. No Swedish words begin with '**ttk**', '**tst**', '**lk**', '**tg**', '**rs** or end with '**rnskl**' '**sn**' or '**mk**'. The fact that these clusters are missed follows from never explicitly having stated which the forbidden clusters are. Among the false positives is also the case of '*Bricker+-blåsa*' which has been declared to have its segmentation point before the hyphen, as well as after. One way to go from here might be to combine this learner with a filter removing false positives containing forbidden start or end clusters and segmentation points before hyphens.

The results of the false positives also reveals a well-known problem in decomposing Swedish compounds. It is the handling of the link morpheme. In in the process of compounding a link morpheme, most often an '**s**', is sometimes inserted between the parts of a compound. The problem is to determine whether an '**s**' in the vicinity of a segmentation point, belongs to the component to the right, or if it is in fact a link morpheme, which is closer bound to the component to the right.

The results from experimenting with the feature vector can be seen in Table 1 above. The best result, with the vector including boundary features 1+1 and 3+1, gave 94,5% recall, which means that almost 19 out of 20 segmentation points were found. This is somewhat better than the 90% reported by Brodda (1979). The precision in the tables above are not quite comparable to Brodda's results since tests have not been done in every position of the words, only the compound boundary and two positions preceding and following it.

To see how well the learner manages to decompose whole compounds, I tried the best feature vector (boundary feature 1+1 and 3+1) on 12 randomly selected words. (See Appendix A.) All 12 segmentation points were found, that is 0% failure. 3 words, 25%, were unambiguously decomposed correctly, and 9 words, 75%, had multiple segmentation. This can be compared to Brodda's 10% failure, 50% unambiguous segmentation and 40% multiple segmentation (Brodda, 1979). However, 8 out

of 15 false positives have been decomposed at morpheme boundaries. In other words, there are only 7 instances of completely false positives, that is segmentation within a morpheme.

5 Conclusion

A conclusion that can be drawn from the experiments above is that it is possible to state that some features are more alike than others, hence the good results of the MVDM setting. Considering the fact that no improvements could be reached by ignoring features, it appears correct to assume that substrings up to the length of four graphemes on either side of the segmentation point give information about permissible clusters. It is also important to consider information across the segmentation point. However, replacing groups of graphemes with a common substitute is not successful, which tells us that every grapheme has its own way of behaving.

The performance of the TiMBL decomposer must be considered satisfactory. Several of the failures are proposed boundaries at morpheme borders, which could be of use in some applications. Finally, the precision could be further improved by simple filters, removing forbidden start and end clusters and compound boundaries before hyphens.

A Results testing all positions in 12 randomly selected words

Below is the result of running 12 randomly selected words through the decomposer, testing in every position of the words. The strings in **boldface** are correctly decomposed. A plus sign (+) represents segmentation at a morpheme boundary. A minus sign (-) represents an incorrect decomposition within a morpheme.

anti+klimax

abort+rådgivning
abor-trådgivning
abortråd+givning

elektroretino+grafi
el-ektroretinografi
elektro+retinografi
elektrore-tinografi

folk+bokföring
folkbok+föring

immuno+logi
im-munologi

kommun+fullmäktige
kommunfull+mäktige

millimeter+rättvisa
milli+meterrättvisa
millimeterrätt+visa

overhead+projektor
o-verheadprojektor
over+headprojektor
overheadproj-ektor

senare+lägga
sen+arelägga

sommar+stuga
som-marstuga

tid+rymd

värme+bölja

B Missed positives using boundary features 1+1 and 3+1

an+fordran
bak+åt
bockhorns+klöver
cerebro+spinal
del+ta
dos+ekvivalent
döds+trött
hov+tång
in+jaga
kontant+insats
kull+kasta
land+sätta
Medel+svensson
mot+åtgärd
nefr+ektomi
nor+adrenalin

pappers+tiger
prust+rot
rep+övning
rid+sår
rök+dykare
silver+nitrat
skräm+skott
slät+struken
tinning+lob
tviste+mål
ur+skog
utom+kved
vak+natt
varnings+triangel
ved+bod
vid+öppen

C False positives using boundary features 1+1 and 3+1

The correct segmentation point is marked by two vertical bars (||). A plus sign (+) represents an incorrect segmentation by the learner at a morpheme boundary. A minus sign (-) represents an incorrect segmentation by the learner within a morpheme. In the string Bricker+-||blåsa the minus sign is a grapheme which belongs to the compound that is decomposed.

abor-t||rådgivning
akter||se-glad
ben||s-tomme
bockhorn+s||klöver
bockhorns||kl-över
Bricker+-||blåsa
elektro||en-cefalogram
hop-p||lös
hydro||ne-fros
hög+t||stående
konsument||om+budsman
kontan-t||insats
kort||s-lutning
kul-l||kasta
land+s||kamp
lek||s-kola
luft||om+byte
lång+t||gående
lång||s-ökt
löne||an+språk
mikrovåg+s||ugn
mot||åt+gärd
pastor+s||expedition
prus-t||rot
rid||s-år
rätte||sn-öre
sjuk||av+drag
skörbjugg+s||ört
spö-rs||mål
strepto||my-cin
stru-p||lock
tallko-tt||körtel
tillhand+a||hålla
tvis-te||mål
tätt||be+byggd
utom||k-ved

References

Benny Brodda. 1979. Något om de svenska ordens fonotax och morfotax: Iakttagelse med utgångspunkt från experiment med automatisk morfologisk analys. *PILUS nr 38*. Institutionen för lingvistik, Stockholm University. (Also in: ”I huvet på Benny Brodda” Festskrift till densammes 65-årsdag.)

Walter Daelemans, Jakub Zavrel, Ko van der Sloot, Antal van den Bosch. 2004. *TiMBL Memory-Based Learner, version 5.1. Reference Guide*. ILK Technical Report – ILK 04-02.

Hedlund, Turid. 2002. Compounds in dictionary-based cross-language information retrieval. *Information Research*, volume 7 No.2. 2002. Department of Information Studies. University of Tampere, Finland.

Fred Karlsson. 1992. SWETWOL: A Comprehensive Morphological analyser for Swedish. *Nordic Journal of Lingusitics 15*, pages 1-45.

Dimitrios Kokkinakis. 2001. *A Framework for the Acquisition of Lexical Knowledge: Description and Applications*. Department of Swedish Language, Språkdata, Göteborg University.

Dimitrios Kokkinakis. 2004. *MEDLEX: Technical Report*. Department of Swedish Language, Språkdata, Göteborg University. [www] <http://demo.spraakdata.gu.se/svedk/pbl/MEDLEX_work2004.pdf>. The reference created February 7, 2007.

Eva Lundström-Holmberg, Peter af Trampe. 1987. *Elementär fonetik*. Studentlitteratur, Lund.

Jonas Sjöbergh, Viggo Kann. 2006. Vad kan statistik avslöja om svenska sammansättningar. *Språk & stil 16*.

Hans Friedrich Witschel, Chris Biemann. 2005. *Rigorous dimensionality reduction through linguistically motivated feature selection for text categorization*. Proceedings of the 15th NODALIDA conference, Joensuu, 2005.