Predicting the Components of German Nominal Compounds

Marco Baroni[1] and **Johannes Matiasek**[1] and **Harald Trost**[2]

Abstract.
Word prediction systems (such as those embedded in most current augmentative and alternative communication systems) aim to predict what a user wants to type next on the basis of corpus-extracted *n*-gram counts. Good performance of such a system depends crucially on the size and quality of the underlying lexicon. Compounding is a common cross-linguistic way to form complex words. In German as in some other languages, compounds are commonly written as single orthographic strings. Because compounding is a very productive process, this leads to a considerable amount of orthographic words that cannot, even in principle, be listed in a lexicon. We present a solution to this problem based on the idea that compounds should not be predicted as units, but as the concatenation of their components. In particular, we designed a word prediction system in which the prediction of German two-element nominal compounds (by far the most common compound type in German) is split into the prediction of the modifier (left element) and the prediction of the head (right element). Both components are predicted on the basis of uni- and bigram statistics collected treating modifiers and heads as independent units, and on the basis of the type frequency of nouns in head and modifier context in the training corpus. We show that our system brings a dramatic improvement in keystroke saving rate over a word prediction scheme in which compounds are treated as units. In particular, our results indicate that the type frequency of nouns in head/modifier context in the training corpus is a very good predictor of which nouns will occur in head/modifier context in new text.

1 Introduction

Word prediction systems are writing programs that try to predict the next word that the user intends to type, or the completion of the word that the user is currently typing.

Besides having many other possible applications, word prediction systems are an important component of devices for augmentative and alternative communication (AAC) , i.e., software and possibly hardware typing aids for disabled users (see for example [1], [2], [3]). Besides functioning as typing aids, such devices can be connected to speech synthesizers to allow oral communication to people who cannot speak.

Obviously, successful word predictors have to deal with the specific characteristics of the target natural language(s). In this paper, we present a preliminary method to handle German words derived via productive *compounding* in AAC word prediction.

Compounding is a common and often very productive cross-linguistic way to form complex words. In many languages, including German, Dutch, the Scandinavian languages and Greek, compounds are commonly written as single orthographic words. For example, the equivalent of the English two word compound *evening session* is written in German as *Abendsitzung* (from *Abend* 'evening' and *Sitzung* 'session').

Productively formed compounds written as single words pose a challenge to most current AAC word predictors, which base their next word guesses on corpus-extracted *n*-gram counts (and, more in general, to any NLP system relying upon a *n*-gram language model, e.g., speech recognition systems).

Productive single-orthographic-word compound formation is problematic because it implies that speakers will keep creating new words that, as such, cannot in principle be in an already existing training corpus, no matter how large.

Moreover, the training corpora themselves will contain a large number of very rare words (new compound formations), causing sparseness of counts problems.

New/rare compounds are different from other types of new/rare words in that, while they are very low (or zero) frequency forms if taken as wholes, they can typically be decomposed into more common smaller units. For example, the compound *Abendsitzung* occurs only one time in the APA corpus (see below). However, both *Abend* and *Sitzung* occur thousands of times in the same corpus.

Thus, a natural approach to handling compounds is to try to predict them by treating them not as primitive units, but as the concatenation of their components.

In this paper, we present and evaluate different compound-splitting-based measures that can be used to predict the most common type of German compounds, i.e. compounds formed by a sequence of two nouns (N+N compounds).

While compound-splitting-based models have been proposed before in the domain of language modeling for speech recognition (see [4], [5], [6] and [7] among others), as far as we know this is the first time that an approach of this kind is evaluated in the context of the AAC word prediction task.

Moreover, as far as we know, this is the first time (in any domain in which language modeling techniques have been applied) that the training corpus type frequencies of nouns as compound modifiers/heads are used in split compound prediction. As we will show below, these measures turn out to be very good predictors of the components of compounds in the test set.

The remainder of this paper is organized as follows. In 2, we describe the AAC word prediction task and discuss related issues. In 3, we describe the basic properties of German compounds. In 4, we present our split compound prediction model. In 5, we describe how

[1] Austrian Research Institute for Artificial Intelligence, Vienna
[2] Department of Medical Cybernetics and Artificial Intelligence, University of Vienna

we tested our model. In 6, we discuss the results of our testing. In 7, we assess the general significance of the research reported here and we sketch directions for further work.

2 Word prediction for AAC

Word predictors provide the user with a *prediction window*, i.e. a menu that, at any time, lists the most likely next word candidates, given the input that the user has typed until the current point. If the word that the user intends to type next is in the prediction window, the user can select it from there. Otherwise, the user will keep entering letters, until the target word appears in the prediction window (or, of course, until she finishes typing the word).

Word prediction systems typically base their predictions on various forms of *n*-gram statistics extracted from one or more training corpora.

The (percentage) *keystroke savings rate* (*ksr*) is a standard measure used in AAC research to evaluate word prediction systems (see, for example, [1] and [2]). The *ksr* can be thought of as the number of keystrokes, in percent, that a "perfect" user could save by employing the relevant word predictor to type a certain corpus, over the total number of keystrokes that are needed to type the same corpus without using the word predictor.

Usually, the *ksr* is defined by

$$\mathit{ksr} = (1 - \frac{k_i + k_s}{k_n}) * 100 \quad (1)$$

where: k_i is the number of input characters actually typed, k_s is the number of keystrokes needed to select among the predictions presented by the model and k_n is the number of keystrokes that would be needed if the whole text was typed without any prediction aid. Here, we assume that the user will need one keystroke to select among the predictions in the prediction window, i.e. that k_s equals 1. In particular, in the simulations based on the split compound model reported below, we assumed that the user would need one keystroke to select the left element prediction and then one more keystroke to select the right element prediction.

The *ksr* is influenced not only by the quality of the prediction model but also by certain parameters of the prediction process, most importantly by the number of predictions to select from the user is presented with, i.e. by the size of the prediction window. In the simulations we report about below, we assumed a prediction window of 7 words, but comparable results were obtained with a prediction window of 5 words.

Using *ksr* as an evaluation measure has the drawback that an exact computation of the *ksr* is possible only by running a simulation of the prediction process. However, it is the measure that reflects best the benefits a disabled typist has when using a word prediction system.

In this paper, we focus entirely on the efficacy of various corpus-based-measures in predicting compounds as sequences of their parts. This means that we will not discuss the important problem of how such complex predictions should be integrated with the regular, non-compound predictions in a fully developed word prediction system. There are several possibilities: compound completions could be presented together and in competition with non-compound completions (so that, from the user point of view, compound prediction is indistinguishable from simple word prediction); alternatively, the user could explicitly ask for compound completions – for example, by typing a special diacritic before starting to type the compound. Intermediate solutions are also possible: for example, left elements of compounds could be presented together with simple words, but right element completions could be proposed only after the user explicitly indicates that she is planning to form a compound by typing a specific character (or, alternatively, potential left elements of compounds could be followed by a special symbol in the prediction list).

All these solutions involve some cost, either in terms of extra-keystrokes (for systems in which compound prediction is explicitly driven by the user) or in terms of a likely degradation in the overall quality of the predictions (for systems in which compound and simple word predictions are merged). In the simulations below, we penalize the split compound model by counting an extra-keystroke for the selection of the right element, but the true impact of compound and non-compound prediction integration must be assessed in the context of a fully developed word prediction system.

3 Compounding in German

Compounding is an extremely common and productive way to form words in German.

In order to understand the properties of German compounds, we conducted a study of the compounds in the APA corpus, a corpus of German newswire containing over 28 million words.

In order to identify and parse the compounds in the APA corpus, for purposes of analysis, training and testing, we ran each wordform in the corpus through the XEROX morphological analyzer ([8]).[3]

In our analysis of the APA corpus, we found that almost half (47%) of the word types were compounds. At the same time, the compounds accounted for a small portion of the overall token count (7%), which suggests that, as expected, many of them are productively formed *hapax legomena* or very rare words (83% of the compounds had a corpus frequency of 5 or lower).

By far the most common type of German compound is formed by a sequence of two nouns (62% of the compounds in our corpus have this shape). Thus, we decided to limit ourselves to the analysis of compounds of this shape, i.e. to N+N compounds. In future research, we plan to verify to what extent the model presented here can be extended to compounds made of more than two constituents and/or of constituents coming from other syntactic categories. In principle, extension of the same model to other types of (right-headed) compounds should be relatively straightforward.

Following standard linguistic terminology, we refer to the left element of a N+N compound as the *modifier* of the compound, and to the right element as the *head* of the compound.

Sometimes nouns in modifier position have a special inflectional shape. In some cases, the form that a noun takes in modifier position can be analyzed as a plural or genitive form, but in other cases the modifier form only occurs in compounds. For example, the modifier *Leitungs* in compounds such as *Leitungsteil* 'portion of piping' does not correspond to an independent inflected form of *Leitung* 'piping', since the "linking" suffix *-s* is only attached to this noun in modifier context (see, among others, [9] and [10] for a discussion of linking suffixes).

Moreover, the same noun can take two or more different shapes in modifier context. For example, we found that in our corpus the noun *Doktor* 'doctor' occurred as modifier both in its singular form and in the plural form *Doktoren*.

The second element of a N+N compound is labeled the head of the

[3] Notice that, while the XEROX analyzer was used to identify the target compounds, the analyzer does not constitute part of the system used in the simulations reported below (alternative methods to identify compounds could also be explored – for example, using a data-driven algorithm along the lines of [7]).

compound since it is the element that determines the basic semantic and morphosyntactic properties of the compound. In particular, the gender, number and case of the compound are determined by the head – if the head is a feminine plural, for example, then the whole compound will behave, in terms of agreement, like a feminine plural.

An interesting property of the distribution of modifiers and heads that emerged from our corpus analysis is that not all nouns are equally likely to occur in modifier/head position: certain nouns are very frequent in head and/or modifier position, whereas other nouns never occur in compounds (less than one fourth of the nouns in the corpus ever occur as compound modifiers, and less than one fourth of the nouns in the corpus ever occur as compound heads). As we will see, this observation led us to the adoption of the type frequency of nouns as modifiers or heads in the training corpus as a potentially good predictor of modifiers and heads in the test set.

While German is the focus of our current research, we believe that our compound prediction model could also be extended to other compounding languages, in particular to those languages, like Dutch ([11], [10]) and Swedish ([12]), whose compounding patterns strongly resemble those of German (in terms of characteristics such as productivity, right-headedness and the presence of linking suffixes).

4 The split compound prediction model

Based on our analysis of the frequency, productivity and structural properties of German compounds, we constructed a model in which we try to predict N+N compounds by treating them as the sequence of a modifier and a head and by relying on the distributional properties of modifiers and heads as independent units in the training corpus.

4.1 Modifier prediction

In our model, modifiers are predicted on the basis of weighted probabilities deriving from the following three terms: the unigram and bigram training corpus frequency of nouns as modifiers or independent words, and the training corpus type frequency of nouns as modifiers:[4]

$$P_{mod}(w) = \lambda_1 P(w) + \lambda_2 P(w|c) + \lambda_3 P_{modtype}(w) \qquad (2)$$

We computed the unigram and bigram frequency of modifiers counting both their occurrences in modifier context and their occurrences as independent words. For example, all occurrences of *Abend* in the training corpus, both as an independent word and in compounds such as *Abendsitzung*, are used to compute the *n*-gram counts for this noun.

As we remarked above, some of the wordforms that occur in modifier position never occur as independent words, since they are special forms of the relevant nominal paradigms that only occur in compounds. For these forms, the *n*-counts were obviously entirely determined by their frequency as modifiers.

Similarly, if a noun occurred in more than one form in modifier context (such as in the *Doktor/Doktoren* example above), the two (or more) modifier forms of the noun were treated as different entities for the purposes of our counts.

As we observed above, the probability of nouns to occur in modifier context is not uniform, i.e. there are nouns that occur as modifiers of a large number of compounds, and other nouns that never occur in this context. Thus, all else being equal, if we are trying to guess the modifier of a compound, we should *a priori* be more inclined to choose a noun that often occurs as a modifier in the training corpus than a noun that never forms a modifier in the training corpus. This is encoded in our system by the third term we are using to predict modifiers, i.e. the number of times (in terms of type frequency) that a noun occurred in modifier context in the training corpus (this is somewhat related to the *head and tail probability* idea of [4]).

4.2 Head prediction

Heads are predicted on the basis of weighted probabilities deriving from three terms analogous to the ones used for modifiers: the unigram and bigram frequency of nouns as heads or independent words, and the type frequency of nouns as heads:

$$P_{head}(w) = \lambda_1 P(w) + \lambda_2 P(w|c) + \lambda_3 P_{headtype}(w) \qquad (3)$$

Like for modifiers, the unigram frequency of nouns as heads is computed considering both their occurrences as independent words and their occurrences in head context.

However, following [6], we decided to compute the bigram counts of nouns as compound heads by considering not their immediate left context, i.e. the modifier, but the word preceding the compound. For example, a sequence such as *die Abendsitzung* ‘the evening session’ is counted, for purposes of head prediction, as an instance of the bigram *die Sitzung* ‘the session’.

We prefer this approach to trying to predict heads on the basis of their modifiers for two reasons. First, the latter strategy would not serve the purpose of generalizing beyond the compounds found in the training corpus (e.g., storing *Abend Sitzung* as a bigram is not more helpful than storing *Abendsitzung* as an unanalyzed unigram, in terms of predicting other compounds).[5]

Second, as we stated above, heads determine the basic semantic and morphosyntactic properties of compounds, and thus they are likely to be linked, semantically and syntactically, to their left phrasal context. As a matter of fact, the left context of a compound is likely to be a better predictor of the compound head than of the modifier.

For reasons of size and efficiency, we decided to use single *n*-gram count lists for modifiers and heads. This has a distorting effect on the bigram counts (words occurring before compounds are counted twice, once as the left context of the modifier and once as the left context of the head). However, we ran preliminary experiments that indicated that the empirical effect of this distortion is minimal, at least for purposes of compound prediction: the results obtained using separate modifier and head *n*-gram lists were almost identical to those obtained with the merged lists.

Finally, as with modifiers, not all nouns are equally likely to occur in head context (less than one fourth of the nouns in the APA corpus also occur as compound heads). Thus, we added the type frequency of nouns in head context as a third term to the head prediction model.

5 Evaluation procedure

In order to evaluate our split compound prediction model, we ran a series of simulations, comparing the *ksr* achieved using our model

[4] Here and in the next equation, c stands for the last word in the left context of w; w is the suffix of the word to be predicted minus the (possibly empty) prefix typed by the user up to the current point.

[5] If a *modifier-head* “bigram” is frequent, i.e. the corresponding compound is a frequent word, it is probably better to treat is as an unanalyzed lexical unit anyway.

to the *ksr* allowed by a model in which compounds are treated as unanalyzed words.

We split the APA corpus described above into a training set containing 25,466,500 words (corresponding to the newswire articles from January to September 1999) and a test set containing 2,754,052 words (corresponding to the newswire articles of October 1999). In order to train and test the split compound model, all words in both sets were run though the XEROX morphological analyzer, and all N+N compounds were split into their modifier and head surface forms.

For the main batch of simulations reported in the next section, the *n*-gram and type frequency tables extracted from the training set were trimmed, removing entries with a count of 5 or lower (but see figure 1 and related discussion for results of experimentation with different cutoff points). After this trimming, the unigram table used for whole word prediction contained 129,591 entries, the bigram table used for whole word prediction contained 320,108 entries, the unigram table used for split word prediction contained 105,402 entries, the bigram table used for split word prediction contained 323,061 entries, the modifier type frequency table contained 4,422 entries and the head type frequency table contained 6,639 entries.

The test set contained 28,104 compound types and 123,025 compound tokens (no trimming was applied to the test set). Notice that the percentage *ksr*'s reported below were computed by treating only these compound word tokens as targets to be predicted.

6 Results

We first ran two simulations with combined terms, using unigram and bigram probabilities (with equal weights) to predict compounds in the compound-as-a-simple-word model (in which compounds are treated as units in training and prediction), and unigram, bigram and modifier/head-frequency-based probabilities (with equal weights) to predict compounds in the split compound model (in which modifiers and heads are predicted separately).

The compound-as-a-simple-word model achieved a *ksr* of 51.5%. The split compound model achieved a combined (modifier + head) *ksr* of 57.9% (the *ksr* for modifier-only prediction was 58.1% and the *ksr* for head-only prediction was 57.6%). Thus, the split compound model led to an improvement in (compound target) *ksr* of more than 6%.

This result clearly indicates that the split compound approach is well worth pursuing. However, we must also remark that, despite the dramatic difference in *ksr* between the two models, the compound-as-a-simple-word model did, by itself, achieve a respectable *ksr* (given the differences in terms of target language and training and testing corpora among word prediction systems, it is difficult to make claims about what a "good" *ksr* is, but, intuitively, a *ksr* over 50% could already be of great help to a user).

The (moderate) success of the no-split model is probably due to the fact that the training and testing sets, being composed of newswire from the same agency and from close time periods, are extremely similar, and thus they are likely to contain largely overlapping sets of low frequency compounds.

Indeed, we observed a correlation between the minimum frequency threshold used to trim the bigram/unigram tables and the difference in performance between the two models: the higher the trimming threshold, the larger the difference. This is illustrated in figure 1, where we plotted the *ksr*'s obtained using the split compound and the compound-as-a-simple-word models in a series of simulations with frequency cutoff points increasing from 1 to 9 in steps of 2.

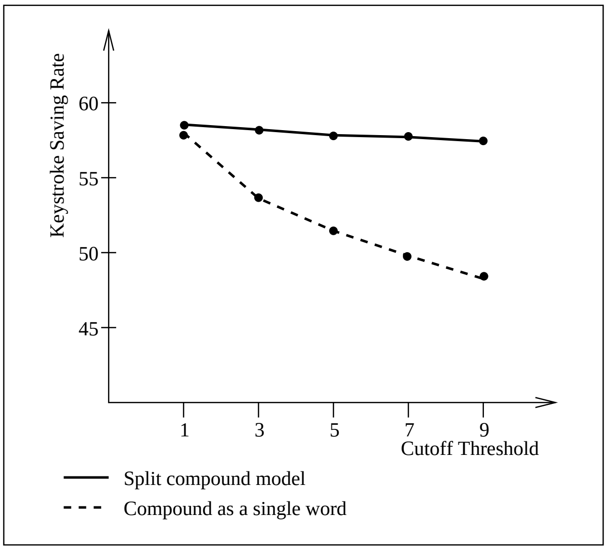

Figure 1. Changes in *ksr* by varying the minimum frequency threshold

The figure shows that, as the cutoff threshold increases, the difference in *ksr* between the two models also increases, with the performance of the simple word model being negatively affected in a more severe way than the performance of the split compound model. The drop in *ksr* from the simulation with the lowest trimming filter to the simulation with the highest trimming filter is of almost 10% (from 58.0% to 48.4%) with the simple word model, but of less than 1% with the split compound model (from 58.4% to 57.5%).

It is extremely likely that this is due to the fact that the test set contains a number of low frequency compounds that are characteristic of the APA newswire style and topics, which disappear from the no-split-based *n*-gram tables when higher minimum frequency filters are applied. This does not affect the no-split model as much, since this model can handle compounds that were not in the training set, as long as their components were in the training set. Thus, we expect that the compound-as-a-simple-word model, being more dependent on corpus-specific low frequency elements, would perform dramatically worse on test sets that are less close, in terms of topics and style, to the training corpus. This must of course be tested in future research.

We also ran a series of simulations in which only one predictor term at a time was used to guess the target words. The results are reported in table 1.

predictor	no-split unigram	no-split bigram	split unigram	split bigram	head/mod typefq
compound *ksr*	46.2	27.6	50.0	36.2	55.2
modifier *ksr*	N/A	N/A	50.7	32.5	58.0
head *ksr*	N/A	N/A	49.4	39.8	52.4

Table 1. Predicting compounds with one-term models

First of all, notice that the superiority of the split compound approach is not due to a single term (and/or to the choice of weights we used in the simulation with combined terms), as the split compound approach outperforms the whole word approach with respect to both unigram- and bigram-based prediction.

Moreover, the results indicate that the type frequency of nouns as heads and modifiers in the training corpus is a very good predictor of heads and modifiers in the test set. This is particularly true for modifier prediction, for reasons that are at the moment not entirely clear, but are perhaps related to the presence of special nominal forms that only occur as modifiers, and not as independent words (see discussion in 3 above).

Less surprisingly, the results in table 1 also show that bigram probabilities are better suited to predict compound heads than modifiers. The likely reason for this, as we remarked above, is that compounds inherit most of their semantic and all of their morphosyntactic properties from their heads. Thus, to mention just the most obvious consequence of this fact, adjectives and determiners preceding a compound must agree with its head but not with the modifier.

7 Conclusion

The results reported here confirm that the special treatment of compound words can lead to considerable improvements in the performance of a word prediction system.

In our simulations we could show that a word prediction system in which (N+N) compounds are treated as sequences of a modifier and a head performs considerably better than a system in which compounds are not analyzed, at least when the systems are tested on compound targets only.

In particular, the likelihood that a specific noun is a head or a modifier, computed on the basis of the type frequency of the noun in head/modifier context in the training corpus, appears to be a very helpful measure to use in predicting the parts of compounds.

While we believe that the results reported here are very encouraging, several key issues have to be dealt with in future research.

First of all, the results reported here concern the prediction of target compounds, but we have not yet fully implemented a system in which compound prediction is integrated into a general word prediction scheme (see [13] for some preliminary attempt). As we observed in 2 above, this integration is likely to have a negative impact on the performance of the system.

Moreover, we conducted our experiments with training and testing corpora that were similar to each other. However, since the true power of the split compound system relies on the fact that it can predict compounds that were not found in the training corpus, a more fair assessment of our model (and of the potential pitfalls of the whole word model) should be conducted on a test set that is further removed from the training corpus.

Another important point related to the data we used here is the following: Obviously, the best way to assess the performance of an AAC word prediction system would be by collecting training and test data from the population of intended users of such systems. Whether the results we report for German newswire text will also hold for text generated by users of AAC systems remains to be seen.

In terms of the model we used to predict compound parts, we believe that several improvements are possible. For example, we are currently exploring the possibility of adding lemma-based, POS-based and semantic-class-based terms to our model (see [13] for preliminary results using class-based head prediction). Moreover, we have assumed here that all compounds should be predicted via the split compound model, but we also intend to explore a mixed approach, in which frequent, lexicalized compounds are treated as unanalyzed words (this, in turn, will require a procedure to identify lexicalized compounds). Moreover, the split compound model should be extended to handle other compound types, beyond the N+N type.

Finally, we plan to test our model with data from other languages. At the very least, we hope that the model will produce similarly encouraging results when applied to languages, such as Dutch and Swedish, whose compounding patterns share strong similarities with those of German.

ACKNOWLEDGEMENTS

We would like to thank the anonymous reviewers for helpful comments, and the Austria Presse Agentur for kindly making the APA corpus available to us. This work was supported by the European Union in the framework of the IST programme, project FASTY (IST-2000-25420). Financial support for ÖFAI is provided by the Austrian Federal Ministry of Education, Science and Culture.

REFERENCES

[1] J. Carlberger, *Design and Implementation of a Probabilistic Word Prediction Program*, Royal Institute of Technology (KTH), 1998.

[2] A Copestake, ‘Augmented and alternative NLP techniques for augmentative and alternative communication’, Proceedings of the ACL workshop on Natural Language Processing for Communication Aids, 1997.

[3] K. McCoy, and P. Demasco, ‘Some Applications of Natural Language Processing to the Field of Augmentative and Alternative Communication’, Proceedings of the IJCAI-95 Workshop on Developing AI Applications for People with Disabilities, Montreal, 1995.

[4] P. Fetter, *Detection and Transcription of OOV Words*, Verbmobil Report 231, 1998.

[5] D. Carter, J. Kaja, L. Neumeyer, M. Rayner, F. Weng, and M. Wirèn, ‘Handling Compounds in a Swedish Speech-Understanding System’, *Proc. ICSLP-96*.

[6] M. Spies, ‘A Language Model for Compound Words’, *Proc. Eurospeech ’95*, pp.1767-1779, 1995.

[7] M. Larson, D. Willett, J. Kohler, and G. Rigoll, ‘Compound splitting and lexical unit recombination for improved performance of a speech recognition system for German parliamentary speeches’, *Proc. ICSLP-2000*.

[8] L. Karttunen, K. Gaál, A. Kempe, *Xerox Finite-State Tool*, Xerox Research Centre Europe, Grenoble, 1997.

[9] J. Goldsmith and T. Reutter, ‘ Automatic Collection and Analysis of German Compounds’, in F. Busa F. et al. (eds.), *The Computational Treatment of Nominals*, Universite de Montreal, Canada, pp.61-69, 1998.

[10] A. Krott, *Analogy in Morphology*, Max Planck Institute for Psycholinguistics, Nijmegen, 2001.

[11] G. Booij, ‘Compounding in Dutch’, *Rivista di Linguistica*, **4**(1), pp.37-59, 1992.

[12] S. Åberg, *Compounding in Swedish*, personal communication.

[13] M. Baroni, J. Matiasek and H. Trost, ’Wordform- and class-based prediction of the components of German nominal compounds in an AAC system.’, *OeFAI Technical Report TR-2002-12*, 2002.