Wordform- and class-based prediction of the components of German nominal compounds in an AAC system

Marco Baroni **Johannes Matiasek**
Austrian Research Institute for
Artificial Intelligence
Schottengasse 3,
A-1010 Vienna, Austria
{marco,john}@oefai.at

Harald Trost
Department of Medical Cybernetics and
Artificial Intelligence, University of Vienna
Freyung 6/2
A-1010 Vienna, Austria
harald@ai.univie.ac.at

Abstract

In word prediction systems for augmentative and alternative communication (AAC), productive word-formation processes such as compounding pose a serious problem. We present a model that predicts German nominal compounds by splitting them into their modifier and head components, instead of trying to predict them as a whole. The model is improved further by the use of class-based modifier-head bigrams constructed using semantic classes automatically extracted from a corpus. The evaluation shows that the split compound model with class bigrams leads to an improvement in keystroke savings of more than 15% over a no split compound baseline model. We also present preliminary results obtained with a word prediction model integrating compound and simple word prediction.

1 Introduction

N-gram language modeling techniques have been successfully embedded in a number of natural language processing applications, including word predictors for augmentative and alternative communication (AAC). *N*-gram based techniques rely crucially on the assumption that the large majority of words to be predicted have also occurred in the corpus used to train the models.

Productive word-formation by compounding in languages such as German, Dutch, the Scandinavian languages and Greek, where compounds are commonly written as single orthographic words, is problematic for this assumption.

Productive compounding implies that a sizeable number of new words will constantly be added to the language. Such words cannot, in principle, be contained in any already existing training corpus, no matter how large. Moreover, the training corpus itself is likely to contain a sizeable number of newly formed compounds that, as such, will have an extremely low frequency, causing data sparseness problems.

New compounds, however, differ from other types of new/rare words in that, while they are rare, they can typically be decomposed into more common smaller units (the words that were put together to form them). For example, in the corpus we analyzed, *Abend* 'evening' and *Sitzung* 'session', the two components of the German compound *Abendsitzung* 'evening session', are much more frequent words than the latter. Thus, a natural way to handle productively formed compounds is to treat them not as primitive units, but as the concatenation of their components.

A model of this sort will be able to predict newly formed compounds that never occurred in the training corpus, as long as they can be analyzed as the concatenation of constituents that did occur in the training corpus. Moreover, a model of this sort avoids the specific type of data sparseness problems caused by newly formed compounds in the training corpus, since it collects statistics based on their (typically more frequent) components.

Building upon previous work (Spies, 1995; Carter et al., 1996; Fetter, 1998; Larson et al., 2000), Baroni et al. (2002) reported encouraging results obtained with a model in which two-element nominal German compounds are predicted by treating them as the concatenation of a *modifier* (left element) and a *head* (right element).

Here, we report of further improvements to this model that we obtained by adding a class-based bigram term to head prediction. As far as we know, this it the first time that semantic classes automatically extracted from the training corpus have been used to enhance compound prediction, independently of the domain of application of the prediction model.

Moreover, we present the results of preliminary experiments we conducted in the integration of

compound predictions and simple word predictions within the AAC word prediction task.

The remainder of this paper is organized as follows. In section 2, we describe the AAC word prediction task. In section 3, we describe the basic properties of German compounds. In section 4, we present our split compound prediction model, focusing on the new class-based head prediction component. In section 5, we report the results of simulations run with the enhanced compound prediction model. In section 6, we report about our preliminary experiments with the integration of compound and simple word prediction. Finally, in section 7, we summarize the main results we obtained and indicate directions for further work.

2 Word prediction for AAC

Word prediction systems based on *n*-gram statistics are an important component of AAC devices, i.e., software and possibly hardware typing aids for disabled users (Copestake, 1997; Carlberger, 1998).

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 character. 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 typing letters, until the target word appears in the prediction window (or until she finishes typing the word).

The (percentage) *keystroke savings rate* (*ksr*) is a standard measure used in AAC research to evaluate word predictors. The *ksr* can be thought of as the percentage of keystrokes that a "perfect" user would save by employing the relevant word predictor to type the test set, over the total number of keystrokes that are needed to type the test set without using the word predictor.

Usually, the *ksr* is defined by

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

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. Typically, the user will need one keystroke to select among the predictions , and thus we assume that k_s equals 1.[1]

[1]In the split compound model, the user needs one keystroke to select the modifier and one keystroke to select the head.

The *ksr* is influenced not only by the quality of the prediction model but also by the size of the prediction window. In our simulations, we use a 7 word prediction window.

Ksr is not a function of perplexity, but it is generally true that there is an inverse correlation between *ksr* and perplexity (Carlberger, 1998).

3 Compounding in German

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

In an analysis of the APA newswire corpus (a corpus of over 28 million words), we found that almost half (47%) of the word *types* were compounds. However, the compounds accounted for a small portion of the overall *token* count (7%). This 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 the N+N type, i.e., a sequence of two nouns (62% of the compounds in our corpus have this shape). Thus, we decided to limit ourselves, for now, to handling compounds of this shape.

In German, nominal compounds, including the N+N type, are *right-headed*, i.e., the rightmost element of the compound determines its basic semantic and morphosyntactic properties.

Thus, the context of a compound is often more informative about its right element (the *head*) than about its left element (the *modifier*).

In modifier context, nouns are sometimes followed by a *linking suffix* (Krott, 2001; Dressler et al., 2001), or they take other special inflectional shapes.

As a consequence of the presence of linking suffixes and related patterns, the forms that nouns take in modifier position are sometimes specific to this position only, i.e., they are bound forms that do not occur as independent words.

We did not parse special modifier forms in order to reconstruct their independent nominal forms. Thus, we treat all inflected modifier forms, including bound forms, as unanalyzed primitive nominal wordforms.

4 The split compound prediction model

In Baroni et al. (2002), we present and evaluate a split compound model in which N+N compounds

are predicted by treating them as the sequence of a modifier and a head.

Modifiers are predicted on the basis of weighed probabilities deriving from the following three terms: the unigram and bigram training corpus frequency of nominal wordforms as modifiers or independent words, and the training corpus type frequency of nominal wordforms as modifiers:[2]

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

The type frequency of nouns as modifiers is determined by the number of distinct compounds in which a noun form occurs as modifier.

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:

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

The type frequency of nouns as heads is determined by the number of distinct compounds in which a noun form occurs as head.

Given that compound heads determine the syntactic properties of compounds, bigrams for head prediction are collected by considering not the immediate left context of heads (i.e., their modifiers), but the word preceding the compound (e.g., *die Abendsitzung* is counted as an instance of the bigram *die Sitzung*).

For reasons of size and efficiency, single uni- and bigram count lists are used for predicting modifiers and heads.[3] For the same reasons, and to minimize the chances of over-fitting to the training corpus, all *n*-gram/frequency tables are trimmed by removing elements that occur only once in the training corpus.

We currently use a simple interpolation model, in which all terms are assigned equal weight.

4.1 Improving head prediction

While we obtained encouraging results with it (Baroni et al., 2002), we feel that a particularly unsatisfactory aspect of the model described in the previous section is that information on the modifier is not exploited when trying to predict the head of a compound. Intuitively, knowing what the modifier is should help us in guessing the head of a compound. However, constructing a plausible head-prediction term based on modifier-head dependencies is not straightforward.

The word-form-based compound-bigram frequency of a head, i.e., the number of times a specific head occurs after a specific modifier, is not a very useful measure: Counting how often a modifier-head pair occurs in the training corpus is equivalent to collecting statistics on unanalyzed compounds, and it will not help us to generalize beyond the compounds encountered in the training corpus. Moreover, if a specific modifier-head bigram is frequent, i.e., the corresponding compound is a frequent word, it is probably better to treat the whole compound as an unanalyzed lexical unit anyway.

POS-based head-modifier bigrams are not going to be of any help either, since we are considering only N+N compounds, and thus we would collect a single POS bigram (N N) with probability 1.[4]

We decided instead to try to exploit a semantically-driven route. It seems plausible that modifiers that are semantically related will tend to co-occur with heads that are, in turn, semantically related. Consider for example the relationship between the class of fruits and the class of sweets in English compounds. It is easy to think of compounds in which a member of the class of fruits (bananas, cherries, apricots...) modifies a member of the class of sweets (pies, cakes, muffins...). Thus, if you have to predict the head of a compound given a fruit modifier, it would be reasonable, all else being equal, to guess some kind of sweet.

4.1.1 Class-based modifier-head bigrams

While semantically-driven prediction makes sense in principle, clustering nouns into semantic classes is certainly not a trivial job, and, if a large input lexicon must be partitioned, it is not a task that could be accomplished by a human expert. Drawing inspiration from Brown et al. (1990), we constructed instead semantic classes using a clustering algorithm extracting them from a corpus, on the basis of the *average mutual information* (MI) between pairs of words (Rosenfeld, 1996).[5]

[2] Here and below, 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.

[3] 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, preliminary experiments indicated that the empirical effect of this distortion is minimal.

[4] Even if the model handled other compound types, very few POS combinations are attested within compounds.

[5] We are aware of the fact that other measures of lexical association have been proposed (Evert and Krenn, 2001, and

MI values were computed using Adam Berger's trigger toolkit (Berger, 1997).[6] The same training corpus of about 25.5M words (and with N+N compounds split) that we describe below was used to collect MI values for noun pairs. All modifiers and heads of N+N compounds and all corpus words that were parsed as nouns by the Xerox morphological analyzer (Karttunen et al., 1997) were counted as nouns for this purpose.

MI was computed only for pairs that co-occurred at least three times in the corpus (thus, only a subset of the input nouns appears in the output list). Valid co-occurrences were bound by a maximal distance between elements of 500 words, and a minimal distance of 2 words (to avoid lexicalized phrases, such as proper names or phrasal loanwords).

Having obtained a list of pairs from the toolkit, the next step was to cluster them into classes, by grouping together nouns with a high MI. For space reasons, we do not discuss our clustering algorithm in detail here (we motivate and analyze the algorithm in a paper currently in preparation).

In short, the algorithm starts by building classes out of nouns that occur with very few other nouns in the MI pair list, and thus their assignment to classes is relatively unambiguous, and it then adds progressively more ambiguous nouns (ambiguous in the sense that they occur in a progressively larger number of MI pairs, and thus it becomes harder to determine with which other nouns they should be clustered). Each input word is assigned to a single class (thus, we do not try to capture polysemy). Moreover, not all words in the input are clustered (see *step 5* below).[7]

Schematically, the algorithm works as follows (the input vocabulary of *step 1* is simply a list of all the words that occur at least once in the MI pair list):

- *step 1:* Rank words in input vocabulary on the basis of how often they occur in the MI pair list (from least to most frequent);
- *step 2:* Shift top word from ranked list and determine with the members of which existing class it has the highest average mutual information;
- *step 3:* If highest value found in *step 2* is 0, assign current word to new class; else, assign it to class corresponding to highest value;
- *step 4:* If ranked list is not empty, go back to *step 2*;
- *step 5:* Discard all classes that have only one member.

This is a heuristic clustering procedure and there is no guarantee that it will construct classes that maximize MI. A cursory inspection of the output list indicates that most classes constructed by our algorithm are intuitively reasonable, while there are also, undoubtedly, classes that contain heterogeneous elements, and missed generalizations. Table 1 reports a list of ten randomly selected classes that were constructed using this procedure.

Alleinstehende, Singles, Alben, Platten, Platte, Sound, Hits, Hit, Live, Songs, Single, Album, Pop, Studio, Rock, Fans, Band
Atrophie, Hartung, Neurologe
Magische, Magie
Bilgen, Tivoli, Baur, Scharrer, Streiter, Winkel, Pfeffer, Schmid, M
Effizienz, Transparenz
Harm, Radar, Jets, Flugzeugen, Typs, Abwehr, Raketen, Maschinen, Angriffen, Flugzeuge, Kampf
Relegation, Birmingham, Stephen
Partnerschafts, Partnerschaft, Kooperation, Bereichen, Aktivitäten
Importeure, Zölle
Labyrinths, Labyrinth

Table 1: Randomly selected noun classes

The algorithm generated 3744 classes, containing a total of 14059 nouns (about one third of the nouns in the training corpus).

Class-based modifier-head bigrams were then collected by labeling all the modifiers and heads in the training corpus with their semantic classes, and counting how often each combination of modifier and head class occurred.

references quoted there) and are sometimes claimed to be more reliable than MI, and we are planning to run our clustering algorithm using alternative measures.

[6] The trigger toolkit returns directional MI values (i.e., separate MI values for the pairs N1 N2 and N2 N1). Since we were not interested in directional information, we merged pairs containing identical nouns by summing their MI. We realize that this is not mathematically equivalent to computing symmetric MI values, but it is a practical approximation that allowed us to use the trigger toolkit for our purposes.

[7] We also experimented with an iterative version of the algorithm that tried to cluster all words, through multiple passes. The classes generated by the non-iterative procedure described in the text, however, gave better results, when integrated in the head prediction task, than those generated with the iterative version.

Like the other tables, class-based bigrams were trimmed by removing elements with a frequency of 1.

4.1.2 The class-based head prediction model

We compute the class-based probability of a compound head given its modifier in the following way:

$$P_{class}(h|m) = P(Cl(h)|Cl(m))P(h|Cl(h)) \quad (4)$$

where

$$P(Cl(h)|Cl(m)) = \frac{\text{count}(Cl(m), Cl(h))}{\text{count}(Cl(m))} \quad (5)$$

and

$$P(h|Cl(h)) = \frac{1}{|Cl(h)|} \quad (6)$$

The latter term assigns equal probability to all members of a class, but lower probability to members of larger classes.

Class-based probability is added to the wordform-based terms of equation 3 obtaining the following formula to compute head probability:

$$P_{head}(w) = \lambda_1 P(w) + \lambda_2 P(w|c) + \lambda_3 P_{ishead}(w) + \lambda_4 P_{class}(w|m) \quad (7)$$

5 Evaluation

The new split compound model and a baseline model with no compound processing were evaluated in a series of simulations, using the APA newswire articles from January to September 1999 (containing 25,466,500 words) as the training corpus, and all the 90,643 compounds found in the Frankfurter Rundschau newspaper articles from June 29 to July 12 of 1992 (in bigram context) as the testing targets.[8]

In order to train and test the split compound model, all words in both sets were run though the morphological analyzer, and all N+N compounds were split into their modifier and head surface forms.

We first ran simulations in which compound heads were predicted using each of the terms in equation 7 separately. The results are reported in table 2.

As an independent predictor, the class-based term performs slightly worse than wordform-based bigram prediction.

We then simulated head and compound prediction using the head prediction model of equation 7.

[8] In other experiments, including those reported in Baroni et al. (2002), we tested on another section of the APA corpus from the same year. Not surprisingly, *ksr*'s in the experiments with the APA corpus were overall higher, and the difference between the split compound and baseline models was less dramatic (because many compounds in the test set were already in the training corpus).

model	$P(w)$	$P(w\|c)$	P_{ishead}	$P_{class}(w\|m)$
head *ksr*	42.2	30.0	47.1	29.4

Table 2: Predicting heads with single term models

The results of this simulation are reported in table 3, together with the results of a simulation in which class-based prediction was not used, and the results obtained with the baseline no-split-compound model.

Model	split w/ classes	split no classes	no split
head *ksr*	51.2	48.8	N/A
compound *ksr*	50.1	48.8	34.9

Table 3: Predicting heads and compounds

When used in conjunction with the other terms, class bigrams lead to an improvement in head prediction of more than 2% over the split compound model without class-based prediction. This translates into an improvement of 1.3% in the prediction of whole compounds. Overall, the split compound model with class bigrams leads to an improvement of more than 15% over the baseline model.

The results of these experiments confirm the usefulness of the split compound model, and they also show that the addition of class-based prediction improves the performance of the model, even if this improvement is not dramatic. Clearly, future research should concentrate on whether alternative measures of association, clustering techniques and/or integration strategies can make class-based prediction more effective.

6 Preliminary experiments in integration

In a working word prediction system, compounds are obviously not the only type of words that the user needs to type. Thus, the predictions provided by the compound model must be integrated with predictions of simple words. In this section, we report preliminary results we obtained with a model limited to the integration of N+N compound prediction with simple noun prediction.

In our approach to compound/simple prediction integration, candidate modifiers are presented together and in competition with simple word solutions as soon as the user starts typing a new word. The user can distinguish modifiers from simple words in the prediction window because the former are suffixed with a special symbol (for example an underscore). If the user selects a modifier,

the head prediction model is activated, and the user can start typing the prefix of the desired compound head, while the system suggests completions based on the head prediction model.

For example, if the user has just typed *Abe*, the prediction window could contain, among other things, the candidates *Abend* and *Abend_*. If the user selects the latter, possible head completions for a compound having *Abend* as its modifier are presented.

Modifier candidates are proposed on the basis of $P_{mod}(w)$ computed as in equation 2 above. Simple noun candidates are proposed on the basis of their unigram and bigram probabilities (interpolated with equal weights).

We experimented with two versions of the integrated model.

In one, modifier and simple noun candidates are ranked directly on the basis of their probabilities. This risks to lead to over-prediction of modifier candidates (recall that, from the point of view of token frequency, compounds are much rarer than simple words; the prediction window should not be cluttered by too many modifier candidates when, most of the time, users will want to type simple words).

Thus, we constructed a second version of the integrated model in which $P_{mod}(w)$ is multiplied by a penalty term. This term discounts the probability of modifier candidates built from nominal wordforms that occur more frequently in the training corpus as independent nouns than as modifiers (forms that are equally or more frequent in modifier position are not affected by the penalty).

The same training corpus and procedures described in section 5 above were used to train the two versions of the integrated model, and the baseline model that does not use compound prediction.

These models were tested by treating *all* the nouns in the test corpus as prediction targets. The integrated test set contained 90,643 N+N tokens and 395,731 more nouns. The results of the simulations are reported in table 4.

Model	integrated no penalty	integrated w/ penalty	simple pred only
compound *ksr*	47.6	45.9	34.9
simple n *ksr*	40.5	42.5	45.6
combined *ksr*	42.5	43.5	42.6

Table 4: Integrated prediction

Because of the simple noun predictions getting in the way, the integrated models perform compound prediction worse than the non-integrated split compound model of table 3. However the integrated models still perform compound prediction considerably better than the baseline model.

The integrated model with modifier penalties performs worse than the model without penalties when predicting compounds. This is expected, since the modifier penalties make this model more conservative in proposing modifier candidates.

However, the model with penalties outperforms the model without penalties in simple noun prediction. Given that in our test set (and, we expect, in most German texts) simple noun tokens greatly outnumber compound tokens, this results in an overall better performance of the model with penalties.

The integrated model with penalties achieves an overall *ksr* that is about 1% higher than that achieved by the baseline model.

Thus, these preliminary experiments indicate that an approach to integrating compound and simple word predictions along the lines sketched at the beginning of this section, and in particular the version of the model in which modifier predictions are penalized, is feasible. However, the model is clearly in need of further refinement, given that the improvement over the baseline model is currently minimal.

7 Conclusion

The main result concerning German compound prediction that was reported in this paper pertains to the introduction of class-based modifier-head bigrams to enhance head prediction.

We presented a procedure to cluster nominal wordforms into semantic classes and to extract class-based modifier-head bigrams, and then a model to calculate the class-based probability of candidate heads using these bigrams.

While we evaluated our system in the context of the AAC word prediction task, we believe that the class-based prediction model we proposed could be extended to any other domain in which *n*-gram-based compound prediction must be performed.

The addition of class-based head prediction to the split compound model of Baroni et al. (2002) leads to an improvement in head prediction (from a *ksr* of 48.8% to a *ksr* of 51.2%). This translates into an improvement of 1.3% in whole compound prediction (from 48.8% to 50.1%). Overall, the split compound model with class bigrams led to an improvement of more than 15% over a no split compound

baseline model.

This result was presented in the context of the AAC word prediction task, but we believe that the class-based prediction model we proposed could be extended to any other domain in which *n*-gram-based compound prediction must be performed.

While the results we report are encouraging, the improvement obtained with the addition of the class-based model is hardly dramatic. It is clear that further work in this area is required.

In particular, we plan to experiment with different measures of association to determine the degree of relatedness of words, and with alternative clustering techniques.

Moreover, we hope to improve the overall performance of the compound predictor by resorting to a better interpolation strategy than the uniform weight assignment model we are currently using.

We also reported results obtained with a preliminary model in which split compound prediction is integrated with simple noun prediction. This model outperforms the baseline model without compound prediction, but only of about 1% *ksr*. Clearly, further work in this area is also necessary. In particular, as suggested by a reviewer, we will try to exploit morpho-syntactic differences between simple nouns and modifiers to help distinguishing between the two types.

Acknowledgements

We would like to thank an anonymous reviewer 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

M. Baroni, J. Matiasek, and H. Trost, 'Predicting the Components of German Nominal Compounds', to appear in *Proc. ECAI 2002*.

A. Berger: Trigger Toolkit, publicly available software, 1997.
http://www-2.cs.cmu.edu/ aberger/software.html

P. Brown, V. Della Pietra, P. DeSouza, J. Lai, and R. Mercer, 'Class-based n-gram models of natural language', *Computational Linguistics* 18(4), pp.467-479, 1990.

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

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

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.

W. Dressler, G. Libben, J. Stark, C. Pons, and G. Jarema, 'The processing of interfixed German compounds', *Yearbook of Morphology 1999*, pp. 185-220, 2001.

S. Evert and B. Krenn, 'Methods for the Qualitative Evaluation of Lexical Association Measures', *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, Toulouse, France, 2001.

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

L. Karttunen, K. Gal, and A. Kempe, *Xerox Finite-State Tool*, Xerox Research Centre Europe, Grenoble, 1997.

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

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', *Proceedings of the 6th International Conference of Spoken Language Processing* (ICSLP-2000), October 16-20., Peking, China, 2000.

R. Rosenfeld, 'A Maximum Entropy Approach to Adaptive Statistical Language Modeling', *Computer Speech and Language* 10, 187–228, 1996.

M. Spies, 'A Language Model for Compound Words', *Proc. Eurospeech '95*, pp.1767-1779, 1995.