Automatic Collection and Analysis of German Compounds

John Goldsmith
The University of Chicago

Summary: In this paper we report on an exploration of noun-noun compounds in a large German corpus. The morphological parsing providing the analysis of words into stems and suffixes was entirely data-driven, in that no knowledge of German was used to determine what the correct set of stems and suffixes was, nor how to break any given word into its component morphemes. To discover compounds, however, we used our prior knowledge of the structure of German nominal compounds, in a way that we will describe in greater length below.

The interest of this case derives from the fact that German compounds (unlike English compounds, but like those in many other languages, especially in the Indo-European family) include a linking element (Fugenelement in German) placed between the two stems. Traditional grammars report nine possible linker elements: e, es, en, er, n, ens, ns, s, and zero (see Duden 1995), and report as well that the Left Element determines which choice of linking element is appropriate for a given nominal compound.

1. Introduction
This project began with both a general and a very specific goal. One of the authors is currently developing a morphological analyzer that takes a large corpus as its input and returns a morphological analysis based on that corpus (see Goldsmith (in prep.)). Most of the morphological activity in European languages involves suffix-attachment to stems, but languages such as German and Dutch require that serious attention be paid to the prefix system, and an even wider range of languages (including both German and Dutch, but also such varied languages as English and Finnish) require an analysis of compounds.

The general goal, then, was to implement a compound-analyzer in the context of the unsupervised acquisition of morphology. The specific goal was to use this analysis to determine the linking element (see below) used by each member of the German lexicon that engages in compound formation as a Left Element.

2. The challenge of compounds
In general, the analyst cannot know whether a given language forms its compounds with fully inflected words or with stems (that is, inflected words minus the inflectional suffix), but the latter is by far the most common pattern. The challenge, then, is to determine whether an analysis of the non-compound words in a corpus will give rise to a sufficient inventory of stems (in the correct surface form, so to speak) so that actual compounds found in the corpus can be identified as concatenation of two such stems, possibly separated by a linker element chosen from a very small inventory. At the same time, it is critical that the analysis not over-recognize compounds, that is, that it not “recognize” compounds that are not there – an error that will typically arise if there exist true stems that are homographs of suffixes, or of subparts of suffixes. We have labelled this problem the Schweinerei problem (from Schweinerei “mess” [lit., pig + erei nominal derivational suffix]) because the word can be misanalyzed as a compound incorporating the linker er and the Right Element ei “egg”.

In addition, the challenge of identifying compounds raises the question as to whether there is a clear distinction to be drawn (in
German, and in other languages as well) between a (prefix + stem) structure and a compound (stem + stem) structure. Duden 1995, for example, characterizes one use of Haupt “head” as a prefix (e.g., in Hauptstadt “capital”), based, presumably, on the semantic bleaching that often accompanies long-time use of a word in various compounds. English has similar uses of the stem head, with cases ranging from head teacher, written with a space and in which the element head contributes a very clear semantics even though it has almost nothing to do with the original sense of head, all the way to headline, where the meaning of the word is barely, if at all, decomposable into two parts. In our work we have employed the definition of affix that is integrated into our automatic morphological analyzer, which is the following: after establishing a tentative set of candidate affixes, a set of affixes is identified which occurs with each given stem (a distinct set of prefixes and suffixes). If exactly the same set (of two or more suffixes) is used by two or more stems, then that set of affixes is “approved”, and the affixes are definitively identified as affixes (rather than as compounds, for example).

3. The challenge of German compounds

Compounding in German is common, ranging from the very frequent formation of compound nouns to the less common but also productive formation of compound verbs and adjectives. Multisegmented compounds, such as Anwendungsprogrammschnittstelle “applications program interface”, can be viewed as recursively applied binary compounds ([Anwendung “application” + Programm “program”] + Schnittstelle “interface”). We will refer to the element on the left of such a binary structure as the Left Element, the element on the right as the Right Element, and the sequence of linking characters used to join the Left Element and Right Element as the Linker.

In our example, the Linker s joins Anwendung and Programm, whilst the null Linker joins Anwendungsprogramm and Schnittstelle.

In German, the Linkers are e, es, en, er, n, ens, ns, s, and the zero morpheme null. In general, the Left Element, Linker, and Right Element are simply concatenated (Bewegung “movement” + s + Achse “axis” = Bewegungschse “axis of rotation”), although the Left Element is occasionally umlauted. (Huhn “hen” + er + ei “egg” = Hühnerei “hen’s egg”).

A hyphen can be used to emphasize the point of linkage between the Left Element+Linker and the Right Element. This effectively doubles the number of Linkers we consider, i.e. we add (e-es-en-er-n-ens-ns-s-and-) to our list. Duden 1995 reports that the hyphen is prescribed if the Left Element is an abbreviation and generally present if the Left Element is a proper name, and otherwise, it is generally employed to improve readability or to emphasize the individual components of the compound. Our actual results confirm some of these guidelines but also yield data that seem not to be covered by the guidelines. The leading hyphenated Left Elements in our data, for example, are (in order): US-, Tang-, and Ballett-Ballett is neither an abbreviation nor a proper name, nor does it seem that it leads to especially unreadable compounds; nevertheless, it is near the top of the list.

If the Left Element ends in the suffix -e or -en, this suffix is sometimes dropped (Schule “school” + Kind “child” = Schulkind “school-age child”). But there is another view of compounding in which no subtraction occurs. Rather, the form without the -e or -en (e.g. Linker, and Right Element correspond to the German terms Bestimmungswort, Fugenelement, and Grundwort, or to the English terms determinant, connecting morpheme, and head.

4 Umlauting of the Left Element (e.g. Land-Spiel=Länderspiel) can occur in conjunction with the null linker, the Linker e, and the Linker er. In these cases, the resulting form coincides orthographically with the plural form, but is not necessarily semantically motivated as a plural; see e.g. Duden 1995.

5 Žepi 1970, borrowing from Charles Hockett, refers to these as subtractive morphs.
schul) is the stem. Our corpus processing returns such suffixless stems. Furthermore, the stems returned by corpus processing can contain umlauts. In our task at hand of automatically assigning a linker distribution to lexicalized nouns, we simply have to add the -e or -en suffix and/or deumlaut the suffix to find the lexicalized noun for which we wish to determine a distribution of Linkers (schul -> schule; länd -> länd).

In general, the choice of a Linker (as well as umlauting and desuffixing) is determined by the Left Element.

Part-of-speech combinations of the Left Element and Right Element include noun-noun, noun-verb, verb-noun, adjective-noun, noun-adjective, etc. In this paper we are only concerned with noun-noun compounds, i.e. ones whose Left Element and Right Element are both lexicalized nouns. Non-nominal Left Elements exhibit fairly trivial Linker distributions.

Previous studies of automatic treatment of German compounds have not dealt with the treatment of the Linker element. Geutner 1995 describes the effect on a speech recognition system of the recognition of compounding in German as a productive and significant process. He notes that treatment of compounds decreases a substantial part of the nagging out-of-vocabulary problem, a major part of the cause for OOV being more significant in German than in English. Berton et al. 1996 also describe work aimed at improving OOV responses of a speech recognition system by allowing the language-model to include compounds. Results of that experiment showed that in the context of speech recognition, the addition of compounding (along with the removal of the compounds from the lexicon) could decrease the performance of the system, especially in the case where the compound was of high frequency, and the case where one of the compounds was phonologically short.

Our goals were formulated in the context of a system which must be equally robust in the context of analysis and generation; furthermore, we set out to obtain information that could be placed in our lexicon, but the analysis of compounds that we used did not need to be performed in real-time together with a user’s speech or keyboard input. On the other hand, we set quite stringent targets for the correctness of the materials that we obtain.

4. Linker distributions

To overcome the out-of-vocabulary problem, German natural language processing systems must accommodate compounds. Encoding in the lexicon for each noun a statistical distribution of Linkers governed by that noun when it is used as a Left Element provides the requisite lexical support. This information is critical for the generation of compound words and can increase the precision of compound analysis. We believe that this lexical approach is preferable to a rule-driven one both for computational efficiency and because the rules governing the selection of a Linker are tempered by such wide-ranging factors as gender, word-length, phonology, diachrony, and dialectal variation and are fraught with exceptions. Our broad-coverage German natural language processing system includes a lexicon with over 140,000 entries, including approximately 100,000 nouns, none of which contained Linker distribution information prior to our

9 For example, if in an examined corpus, the noun Staat were used 96 times with the Linker s, and 12 times with the Linker en, we would calculate the distribution (p(s)=0.89; p(en)=0.11).

10 See, for example, Žepić 1970
undertaking. Our goal was to identify stems and suffixes in a large German corpus, then post-process the results to yield Linker distributions for a large number of nouns in our lexicon. This goal was largely met. Both the stem/suffix identification and the subsequent post-processing were implemented to run fully automatically, so that the process can be applied to an arbitrarily large corpus, yielding distributions for a maximal number of lexicalized nouns.

5. Procedures

We now summarize the steps involved in first morphologically processing a corpus to detect stems and suffix, then using the stem/suffix information to find compounds, and finally post-processing the compound list to calculate Linker distributions for the nouns used as Left Elements.

Since the object of our inquiry has been noun-noun compounds, and since German nouns are capitalized, we restricted our processing to words in the corpus beginning with a capital letter. We therefore first applied our automatic morphological analyzer to the first 300,000 capitalized words in Microsoft's Encarta, an encyclopedia, to establish a list of 8,426 noun stems. These are identified by first automatically extracting the productive suffixes in the corpus; 74 were identified, in frequency dominated by the top six suffixes (en, e, er, s, ung, n); see Table 1.

When the algorithm identifies two distinct words as composed of the same stem followed by different suffixes, it accepts that stem as legitimate. For example, the string beobacht- (stem for “watch”) is identified as a stem because it appears in the corpus with the following five suffixes: -ere' -er/-ers/-ung/-tmgen. In addition, if a potential stem occurs as a free-standing word, we consider that to count as an appearance of the stem with a null suffix. For example, the stem Alaska “Alaska” appears with three “suffixes”: -s, -n, and Null. Thus any freestanding word which also appears with at least one (independently determined) suffix counts as a stem for our purposes. See Table 2. Table 2 illustrates the fact that this procedure includes in our list of stems noun compounds that are found in the corpus with more than one suffix. This is not a problem, and in fact is a good thing, because, as we noted above, compounds are frequently recursively composed out of pieces which are themselves compounds. With this list of stems in hand, we revisit the original corpus, checking each entry now for the possibility of one or more parses as compounds. Given the set of linkers (established in advance, as we have noted), we can very simply review each word to see if it can be parsed as the concatenation of an item from the list of stems + one of the linkers + another item from the list of stems + one of the 74 recognized suffixes. All forms that can be so parsed are added to a list of compounds found; in our corpus, we found 5522 compounds, based on 3866 distinct First Element stems. For each distinct FirstElement stem, we produce a record of the form:

( Left Stem, Linker { Exemplar1, Exemplar2,..., ExemplarN } )

where each Exemplar is the Right Element of a compound, and is itself of the form (Stem + Suffix).

Next, the compounds are filtered so that they only include unambiguous noun-noun compounds. This filtering process is described in the following section. Finally, the filtered set of data is used to calculate a distribution of Linker governance for each surviving Left Stem.

6. Filtering

In a compound such as Anwendungsprogramme (anwendung + s + programm + e), we call a (Left Stem + Suffix) pair such as (anwendung + s) a candidate, while a (Right Stem + Suffix) pair like (programm + e) is called an exemplar. Thus, our set of compounds is logically of the form:

( Candidate, { Exemplar1, Exemplar2,..., ExemplarN } )

For example, if the corpus contains Anwendungsprogramm “applications program”
and Anwendungsprogramme, "applications programs", then we would have the item
\( (\text{anwendung} + s), (\text{programm} + \text{null}), (\text{programm} + e) \ldots ) \)

Since our specific goal is to produce Linker distribution information for nouns used as the
Left Element in noun-noun compounds, we must now filter this raw data so that we end up with
candidates and associated exemplars that are unambiguously involved in noun-noun
compounding. This filtering process is now described.

In order to calculate meaningful linker
distributions, the raw data must first be passed
through a series of simple filters.

**Step 1** Left stems which are not the stems of
lexicalized nouns are excluded. The stem and
the lexicalized words may differ with regard to
umlauting, and in addition the lexicalized word
may contain the \(-\text{e/-en}\) suffix. For example, the
left stems \(\text{schul} \) and \(\text{land} \) correspond to the
lexical entries \(\text{Schule} \) and \(\text{Land} \), and are thus not
excluded. But this step does properly exclude
e.g. the candidate \(\text{ab} + \text{null}\) since \(\text{ab}\) is not a noun,
obviating compounds like \(\text{Abzug} \) and \(\text{Abbildung} \).

**Step 2.** Left stems with multiple parts of
speech are excluded. For example, \(\text{gut}\) can be an
adjective ("good") or a noun ("property"). Since
German compounds can be built with e.g. a verb
or adjective as the Left Element, we cannot
automatically determine whether a compound
starting with the Left Element \(\text{gut}\) is combining
the adjective or the noun. We therefore
eliminate the candidate \(\text{gut} + \text{null}\).\(^{12}\)

A special instance of excluding multiple parts of
speech is the case of verb stems. When a verb is
used as the Left Element of a compound, the
verb stem, i.e. the infinitive without the final
\((e)n\), is used. This leads to a number of
ambiguous Left Elements such as \(\text{blut}\) (noun
\(\text{Blut} \) "blood"; verb \(\text{bluten} \) "bleed") and \(\text{block}\)
(noun \(\text{Block} \) "block"; verb \(\text{blocken} \) "block"),
which are excluded, since it cannot be
automatically determined whether the
compounding is based upon the verb stem or the
homographic noun.

**Step 3.** Cases in which the division between
the Left Stem and the Linker is ambiguous are
excluded. For example, the candidate \(\text{mark} \)
"mark" + \(\text{en}\), with exemplars such as
\(\text{Welmeister} + \text{schaf} \) "world championship" and
\(\text{nam} + \text{e} \) "name", is excluded, since there is an
alternate division: \(\text{marke} \) "brand" + \(\text{n}\).\(^{13}\)

**Step 4.** Combinations of Left Stem and
Linker in which the final character of the Left
Element and the initial character of the Linker are
identical are excluded.

This is for phonological reasons, and applies
both to vowels and consonants. Thus, the
candidate \(\text{boden} \) with the exemplar \(\text{es} + \text{ter}\) is
properly rejected, as is \(\text{industrie} \) "industry" + \(\text{er}\),
with exemplars like (\(\text{zeugnisse, null}\)).\(^{14}\)

These first four filters remove invalid and/or
ambiguous candidates; next, a few more filters
are applied to remove invalid and/or ambiguous
exemplars. If this filtering of exemplars results
in a candidate being left with no valid
exemplars, then the candidate is of course
removed from the list.

**Step 5.** Exemplars whose stem is not a
lexicalized noun are excluded. This is a
reasonable filtering step, since we are interested
in noun-noun compounds. The exemplar \(\text{bella} + \text{null}\)
(associated with the candidate \(\text{Ara} \) "parrot"
+ \(\text{null}\)), derived from the compound \(\text{Arabella}\),
for example, is excluded in this step.

**Step 6.** Exemplars in which the division
between the Stem and the Suffix is ambiguous
are excluded. For example, the exemplar \(\text{kamm} \)
"comb" + \(\text{er}\) (associated e.g. with the candidate
\(\text{architekt} \) "architect" + \(\text{en}\)) is ambiguous with the
exemplar \(\text{kammer} \) "chamber" + \(\text{null}\), and is
therefore excluded.

**Step 7.** Cases in which the division between
the Linker and the Suffix is ambiguous are
excluded. Consider the candidate \(\text{Abfall} \) "trash"
+ \(\text{er}\), associated with the exemplar \(\text{fassung} \)

\(^{12}\) These, and other ambiguous cases, are logged to a
file for possible later manual review.

\(^{13}\) In this example, the alternate division is the
linguistically motivated one.

\(^{14}\) The proper parse of the compound
\(\text{Industriezeugnisse} \) is \(\text{Industrie} + \text{null} + \text{zeugnis} + \text{se} \)
"industry products", not \(\text{Industrie} + \text{er} + \text{zeugnis} + \text{se} \)
"industry certificates". Similarly, \(\text{Bodennester} \) is
parsed \(\text{Boden} + \text{null} + \text{nest} + \text{er} \) "ground nests", not
\(\text{Boden} + \text{n} + \text{ester} + \text{null} \) "ground ester". Note that
excluding the candidates \(\text{industrie} + \text{er} \) and \(\text{boden} + \text{n} \)
does not affect the candidates \(\text{industrie} + \text{null} \) and
\(\text{boden} + \text{null} \).
"fixture" + null. The exemplar is excluded, since there is an alternate division of linker and stem: abfall "trash" + null, with the exemplar erfassung "acquisition" + null. Another example of this kind of ambiguity is Blut-s-tau vs. Blut-stau, -- that is, Blut "blood" + s associated with Tau "dew" + null over against Blut "blood" + null associated with Stau "congestion" + null.

Step 8. Cases in which the entire compound, i.e. candidate plus exemplar, is lexicalized are excluded. For example, there is a candidate Ara "parrot" + null associated with the exemplar Rat "council" + null. The exemplar is excluded, however, since the candidate plus the exemplar yields Ararat "Ararat", which is lexicalized.

A small amount of noise survives the filtering process. For example, the Linker ns is improperly included in the linker distribution of the noun Ar, based on the proper noun Arnsberg, which resembles a compound noun: Ar-ns-berg. This minimal amount of noise is further reduced by thresholding: Any candidate (Left Element + Linker) for which there is only one remaining exemplar does not contribute to the distribution. After this final filtering, the surviving (Left Element + Linker) candidates and their associated surviving exemplars are used to calculate linker distributions for each Left Element.

Of the 8,496 candidates entering the filtering and thresholding process, 1361 of them survive. Of these, 20 share a common Left Element with another candidate15; thus we are able to calculate a Linker distribution for 1341 lexicalized nouns.

7. Linker Distributions
The filtering described in the previous section yields a set of reliable candidates and exemplars for noun-noun compounding. For example, (anwendung + s), (programm + null), (programm + e) ... ) survives the filtering process.

Based on these vetted candidates and exemplars, we now calculate a Linker governance distribution for lexicalized nouns used as the Left Element of a noun-noun compound.

First, from each set of exemplars associated with a given candidate, we squeeze out the exemplars with a common stem. In our example, the exemplar (programm + e) is removed, since the exemplar (programm + null) is also associated with the candidate (anwendung + s).

Next, for each Left Stem, we simply tally the total number T of exemplars associated with that Left Stem. Then, for each Linker associated with Left Stem, we calculate its probability by tallying the number of exemplars associated with the candidate (Left Stem + Linker), then dividing by T.

We wish to incorporate this data into our lexicon as follows. For each noun entry N, derive the distribution D(N) of Linkers governed by N16. For example, for the entry Staat, the distribution (en = 0.11; s = 0.89) is calculated.

8. Conclusions
Our goal in this effort has been to evaluate and, ultimately, to use for practical ends the analysis of large-scale German corpora in order to determine a morphological property of individual German noun stems -- the choice of Linker element used in compounding.

Our results support the strategy of using large-scale natural language corpora as a source for automatic processing and as a means to gather specific lexical information. While linker information is sparsely distributed across the corpora we have studied, the largely automatic character of our search allows us to have increasingly certain information about this property.

15 For example, the candidates Stand+null and Stand+es share the Left Stem Stand.

16 The number of noun entries for which any distribution is calculated is, of course, dependent upon the corpus processed. Every step of processing described in this paper is fully automated, so that an arbitrarily large corpus can be processed, limited only by computational resources.
References

Berton, Andre, Pablo Fetter and Peter Regel-Brietzmann. 1996. Compound Words in Large-Vocabulary German Speech Recognition Systems. Proceedings of the 1996 International Conference on Spoken Language Processing, ICLSP. Part 2.

DUDEN Grammatik der deutschen Sprache, pp. 465 ff. 1995. Der Duden in 12 Bänden. Vol. 4. Dudenverlag. Mannheim.

Geutner, P. 1995. Using Morphology towards better large-vocabulary speech recognition systems. Proceedings of the 1995 International Conference on Acoustics, Speech, and Signal Processing. Vol. 1.

Goldsmith, John. In preparation. Unsupervised Learning of the Morphology of a Natural Language.

Hockett, Charles. 1957. Two Models of Linguistic Description. In Martin Joos (ed.) Readings in Linguistics. University of Chicago Press.

Horacek, Helmut. 1996. On Expressing Metonymic Relations in Multiple Languages. Machine Translation, vol. 11: 109-158. Kluwer Academic Publishers. The Netherlands.

Leser, Martin. 1990. Das Problem der 'Zusammenbildungen': eine lexikalistische Studie, Wissenschaftlicher Verlag Trier. Trier.

Meyer, Ralf. 1993. Compound comprehension in isolation and in context: the contribution of conceptual and discourse knowledge to the comprehension of German novel noun-noun compounds. Max Niemeyer Verlag. Tübingen.

Rufener, John. 1971. Studies in the motivation of English and German compounds, Juris Druck Verlag. Zürich.

Shaw, J. Howard. 1979. Motivierte Komposita in der deutschen und englischen Gegenwartssprache, Gunter Narr Verlag. Tübingen.

Trost, Harald. 1991. Recognition and generation of word forms for natural language understanding systems: Integrating two-level morphology and feature unification. Applied Artificial Intelligence, Vol. 5.

Žepić, Stanko. 1970. Morphologie und Semantik der deutschen Nominalkomposita. Philosophische Fakultät der Universität Zagreb.
| Suffix | Words with this suffix | Tokens |
|--------|------------------------|--------|
| 1      | en                     | 2022   |
| 2      | e                      | 1377   |
| 3      | er                     | 843    |
| 4      | s                      | 1097   |
| 5      | ung                    | 774    |
| 6      | n                      | 535    |
| 7      | a                      | 441    |
| 8      | ie                     | 244    |
| 9      | m                      | 253    |
| 10     | es                     | 415    |
| 11     | ch                     | 149    |
| 12     | ten                    | 395    |
| 13     | te                     | 312    |
| 14     | on                     | 240    |
| 15     | el                     | 230    |
| 16     | i                      | 214    |
| 17     | in                     | 198    |
| 18     | ungen                  | 271    |
| 19     | o                      | 256    |
| 20     | se                     | 219    |
| 21     | y                      | 186    |
| 22     | ik                     | 133    |
| 23     | an                     | 132    |
| 24     | em                     | 213    |
| 25     | ter                    | 188    |
| 26     | um                     | 115    |
| 27     |                        | 5382   |

Table 1: German suffixes, determined automatically.

| Suffix | Words with this suffix | Tokens |
|--------|------------------------|--------|
| 28     | de                     | 155    |
| 29     | il                     | 59     |
| 30     | nt                     | 97     |
| 31     | land                   | 99     |
| 32     | al                     | 139    |
| 33     | us                     | 151    |
| 34     | tion                   | 87     |
| 35     | schaft                 | 76     |
| 36     | se                     | 50     |
| 37     | chen                   | 15     |
| 38     | tat                    | 56     |
| 39     | ische                  | 105    |
| 40     | as                     | 78     |
| 41     | tur                    | 38     |
| 42     | ur                     | 56     |
| 43     | ismus                   | 69     |
| 44     | ia                     | 67     |
| 45     | erung                  | 106    |
| 46     | ischen                  | 58     |
| 47     | ation                  | 77     |
| 48     | ers                    | 72     |
| 49     | end                    | 17     |
| 50     | reich                  | 24     |
| 51     | ien                    | 24     |
| 52     | ens                    | 59     |
| 53     | ium                    | 42     |
| 54     | mittel                  | 24     |
| 55     | sen                    | 21     |
| 56     | lich                   | 25     |
| 57     | os                     | 19     |

| Suffix | Words with this suffix | Tokens |
|--------|------------------------|--------|
| 58     | ner                    | 17     |
| 59     | ii                     | 29     |
| 60     | nen                    | 15     |
| 61     | szenstrum               | 11     |
| 62     | den                    | 13     |
| 63     | schen                  | 11     |
| 64     | sgebiet                | 13     |
| 65     | ons                    | 15     |
| 66     | iierung                | 12     |
| 67     | isten                  | 9      |
| 68     | 's                     | 9      |
| 69     | isch                   | 4      |
| 70     | der                    | 7      |
| 71     | shire                  | 7      |
| 72     | see                    | 5      |
| 73     | produkti on            | 6      |
| 74     | ill                    | 5      |
| 75     | nischen                | 1      |
| 76     | nische                 | 1      |

Table 2: Sample stems with suffixes found.
Table 3 Most common Left Elements in German corpus

Note: elements marked with ** were automatically filtered out since they did not meet the strict requirements for unambiguous noun-noun compounds.