# DERIVBASE: Inducing and Evaluating a Derivational Morphology Resource for German

**Britta Zeller*** **Jan Šnajder†** **Sebastian Padó***
*Heidelberg University, Institut für Computerlinguistik
69120 Heidelberg, Germany
†University of Zagreb, Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia
{zeller, pado}@cl.uni-heidelberg.de jan.snajder@fer.hr

## Abstract

Derivational models are still an underresearched area in computational morphology. Even for German, a rather resource-rich language, there is a lack of large-coverage derivational knowledge. This paper describes a rule-based framework for inducing *derivational families* (i.e., clusters of lemmas in derivational relationships) and its application to create a high-coverage German resource, DERIVBASE, mapping over 280k lemmas into more than 17k non-singleton clusters. We focus on the rule component and a qualitative and quantitative evaluation. Our approach achieves up to 93% precision and 71% recall. We attribute the high precision to the fact that our rules are based on information from grammar books.

## 1 Introduction

Morphological processing is generally recognized as an important step for many NLP tasks. Morphological analyzers such as lemmatizers and part of speech (POS) taggers are commonly the first NLP tools developed for any language (Koskenniemi, 1983; Brill, 1992). They are also applied in NLP applications where little other linguistic analysis is performed, such as linguistic annotation of corpora or terminology acquisition; see Daille et al. (2002) for an informative summary.

Most work on computational morphology has focused on inflectional morphology, that is, the handling of grammatically determined variation of form (Bickel and Nichols, 2001), which can be understood, overimplifying somewhat, as a normalization step. Derivational morphology, which is concerned with the formation of new words from existing ones, has received less attention. Examples are nominalization (*to understand* → *the understanding*), verbalization (*the shelf* → *to shelve*), and adjectivization (*the size* → *sizable*). Part of the reason for the relative lack of attention lies in the morphological properties of English, such as the presence of many zero derivations (*the fish* → *to fish*), the dominance of suffixation, and the relative absence of stem changes in derivation. For these reasons, simple *stemming* algorithms (Porter, 1980) provide a cheap and accurate approximation to English derivation.

Two major NLP resources deal with derivation. WordNet lists so-called "morphosemantic" relations (Fellbaum et al., 2009) for English, and a number of proposals exist for extending WordNets in other languages with derivational relations (Bilgin et al., 2004; Pala and Hlaváčková, 2007). CatVar, the "Categorial Variation Database of English" (Habash and Dorr, 2003), is a lexicon aimed specifically at derivation. It groups English nouns, verbs, adjectives, and adverbs into derivational equivalence classes or *derivational families* such as

$$ask_V \; asker_N \; asking_N \; asking_A$$

Derivational families are commonly understood as groups of derivationally related lemmas (Daille et al., 2002; Milin et al., 2009). The lemmas in CatVar come from various open word classes, and multiple words may be listed for the same POS. The above family lists two nouns: an event noun (*asking*) and an agentive noun (*asker*). However, CatVar does not consider prefixation, which is why, e.g., the adjective *unasked* is missing.

CatVar has found application in different areas of English NLP. Examples are the acquisition of paraphrases that cut across POS lines, applied, for example, in textual entailment (Szpektor and Dagan, 2008; Berant et al., 2012). Then there is the induction and extension of semantic roles resources for predicates of various parts of speech (Meyers et al., 2004; Green et al., 2004). Finally, CatVar has



*Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1201–1211,
Sofia, Bulgaria, August 4-9 2013. 

been used as a lexical resource to generate sentence intersections (Thadani and McKeown, 2011).

In this paper, we describe the project of obtaining derivational knowledge for German to enable similar applications. Even though there are two derivational resources for this language, IMSLEX (Fitschen, 2004) and CELEX (Baayen et al., 1996), both have shortcomings. The former does not appear to be publicly available, and the latter has a limited coverage (50k lemmas) and does not explicitly represent derivational relationships within families, which are necessary for fine-grained optimization of families. For this reason, we look into building a novel derivational resource for German. Unfortuantely, the approach used to build CatVar cannot be adopted: it builds on a collection of high-quality lexical-semantic resources such as NOMLEX (Macleod et al., 1998), which are not available for German.

Instead, we employ a rule-based framework to define derivation rules that cover both suffixation and prefixation and describes stem changes. Following the work of Šnajder and Dalbelo Bašić (2010), we define the derivational processes using derivational rules and higher-order string transformation functions. The derivational rules induce a partition of the language's lemmas into derivational families. Our method is applicable to many languages if the following are available: (1) a comprehensive set of lemmas (optionally including gender information); (2) knowledge about admissible derivational patterns, which can be gathered, for example, from linguistics textbooks.

The result is a freely available high-precision high-coverage resource for German derivational morphology that has a structure parallel to CatVar, but was obtained without using manually constructed lexical-semantic resources. We conduct a thorough evaluation of the induced derivational families both regarding precision and recall.

**Plan of the paper.** Section 2 discusses prior work. Section 3 defines our derivation model that is applied to German in Section 4. Sections 5 and 6 present our evaluation setup and results. Section 7 concludes the paper and outlines future work.

## 2 Related Work

Computational models of morphology have a long tradition. Koskenniemi (1983) was the first who analyzed and generated morphological phenomena computationally. His two-level theory has been applied in finite state transducers (FST) for several languages (Karttunen and Beesley, 2005).

Many recent approaches automatically induce morphological information from corpora. They are either based solely on corpus statistics (Déjean, 1998), measure semantic similarity between input and output lemma (Schone and Jurafsky, 2000), or bootstrap derivation rules starting from seed examples (Piasecki et al., 2012). Hammarström and Borin (2011) give an extensive overview of state-of-the-art unsupervised learning of morphology. Unsupervised approaches operate at the level of word-forms and have complementary strengths and weaknesses to rule-based approaches. On the upside, they do not require linguistic knowledge; on the downside, they have a harder time distinguishing between derivation and inflection, which may result in lower precision, and are not guaranteed to yield analyses that correspond to linguistic intuition. An exception is the work by Gaussier (1999), who applies an unsupervised model to construct derivational families for French.

For German, several morphological tools exist. Morphix is a classification-based analyzer and generator of German words on the inflectional level (Finkler and Neumann, 1988). SMOR (Schmid et al., 2004) employs a finite-state transducer to analyze German words at the inflectional, derivational, and compositional level, and has been used in other morphological analyzers, e.g., Morphisto (Zielinski and Simon, 2008). The site canoonet[1] offers broad-coverage information about the German language including derivational word formation.

## 3 Framework

In this section, we describe our rule-based model of derivation, its operation to define derivational families, and the application of the model to German. We note that the model is purely surface-based, i.e., it does not model any semantic regularities beyond those implicit in string transformations. We begin by outlining the characteristics of German derivational morphology.

### 3.1 German Derivational Morphology

As German is a morphologically complex language, we analyzed its derivation processes before implementing our rule-based model. We relied on traditional grammar books and lexicons, e.g., Hoeppner (1980) and Augst (1975), in order to linguistically

[1] http://canoo.net

justify our assumptions as well as to achieve the best possible precision and coverage.

We concentrate on German derivational processes that involve nouns, verbs, and adjectives.[2] Nouns are simple to recognize due to capitalization: *stauen*$_V$ – *Stau*$_N$ (*to jam – jam*), *essen*$_V$ – *Essen*$_N$ (*to eat – food*). Verbs bear three typical suffixes (*-en*, *-eln*, *-ern*). An example of a derived verb is *fest*$_A$ – *festigen*$_V$ (*tight – to tighten*), where *-ig* is the derivational suffix. Adjectivization works similarlty: *Tag*$_N$ – *täglich*$_A$ (*day – daily*).

This example shows that derivation can also involve stem changes in the form of umlaut (e.g., *a* → *ä*) and ablaut shift, e.g., *sieden*$_V$ – *Sud*$_N$ (*to boil – infusion*). Other frequent processes in German derivation are circumfixation (*Haft*$_N$ – *inhaftieren*$_V$ (*arrest – to arrest*)) and prefixation (*heben*$_V$ – *beheben*$_V$ (*to raise – to remedy*)). Prefixation often indicates a semantic shift, either in terms of the general meaning (as above) or in terms of the polarity ( *klar*$_A$ – *unklar*$_A$ (*clear – unclear*)). Also note that affixes can be either Germanic, e.g., *ölen – Ölung* (*to oil – oiling*), or Latin/Greek, e.g., *generieren – Generator* (*to generate – generator*).

As this analysis shows, derivation in German involves transformation as well as affixation processes, which has to be taken into account when modeling a derivational resource.

### 3.2 A Rule-based Derivation Model

The purpose of a derivational model is to define a set of transformations that correspond to valid derivational word formation rules. Rule-based frameworks offer convenient representations for derivational morphology because they can take advantage of linguistic knowledge about derivation, have interpretable representations, and can be fine-tuned for high precision. The choice of the framework is in principle arbitrary, as long as it can conveniently express the derivational phenomena of a language. Typically used for this purpose are two-level formalism rules (Karttunen and Beesley, 1992) or XFST replace rules (Beesley and Karttunen, 2003).

In this paper, we adopt the modeling framework proposed by Šnajder and Dalbelo Bašić (2010). The framework corresponds closely to simple, human-readable descriptions in traditional grammar books. The expressiveness of the formalism is equivalent to the replacement rules commonly used in finite state frameworks, thus the rules can be compiled into FSTs for efficient processing.

The framework makes a clear distinction between inflectional and derivational morphology and provides separate modeling components for these two; we only make use of the derivation modeling component. We use an implementation of the modeling framework in Haskell. For details, see the studies by Šnajder and Dalbelo Bašić (2008) and Šnajder and Dalbelo Bašić (2010).

The building blocks of the derivational component are *derivational rules (patterns)* and *transformation functions*. A derivational rule describes the derivation of a *derived* word from a *basis* word. A derivational rule $d$ is defined as a triple:

$$d = (t, \mathcal{P}_1, \mathcal{P}_2) \tag{1}$$

where $t$ is the transformation function that maps the word's stem (or lemma) into the derived word's stem (or lemma), while $\mathcal{P}_1$ and $\mathcal{P}_2$ are the sets of inflectional paradigms of the basis word and the derived word, respectively, which specify the morphological properties of the rule's input and output. For German, our study assumes that inflectional paradigms are combinations of part-of-speech and gender information (for nouns).

A transformation function $t : S \rightarrow \wp(S)$ maps strings to a set of strings, representing possible transformations. At the lowest level, $t$ is defined in terms of atomic string replacement operations (replacement of prefixes, suffixes, and infixes). The framework then uses the notion of higher-order functions – functions that take other transformations as arguments and return new transformations as results – to succinctly define common derivational processes such as prefixation, suffixation, and stem change. More complex word-formation rules, such as those combining prefixation and suffixation, can be obtained straightforwardly by functional composition.

Table 1 summarizes the syntax we use for transformation functions and shows two example derivational rules. Rule 1 defines an English adjectivization rule. It uses the conditional $try$ operator to apply to nouns with and without the *-ion* suffix (*action – active*, *instinct – instinctive*). Infix replacement is used to model stem alternation, as shown in rule 2 for German nominalization, e.g., *vermacht*$_A$ – *Vermächtnis*$_N$ (*bequethed – bequest*).

[2] We ignore adverb derivation; the German language distinguishes between adverbial adjectives and adverbs, the latter being a rather unproductive class and thus of no interest for derivation (Schiller et al., 1999).

| Function | Description |
|---|---|
| $sfx(s)$ | concatenate the suffix $s$ |
| $dsfx(s)$ | delete the suffix $s$ |
| $aifx(s1, s2)$ | alternate the infix $s1$ to $s2$ |
| $try(t)$ | perform transformation $t$, if possible |
| $opt(t)$ | optionally perform transformation $t$ |
| $uml$ | alternate infixes for an umlaut shift: $uml = aifx(\{(a, \ddot{a}),\ (o, \ddot{o}), (u, \ddot{u})\})$ |
| Examples | |
| 1 (EN) | $\big(sfx(ive) \circ try(dsfx(ion)), \mathcal{N}, \mathcal{A}\big)$ "derive *-ive* adjectives from nouns potentially ending in *-ion*" |
| 2 (DE) | $\big(sfx(nis) \circ try(uml), \mathcal{A}, \mathcal{N}\big)$ "derive *-nis* nouns from adjectives with optional umlaut creation" |

Table 1: Transformation functions and exemplary derivational rules in the framework by Šnajder and Dalbelo Bašić (2010)

$\mathcal{N}$ and $\mathcal{A}$ denote the paradigms for nouns (without gender restriction) and adjectives, respectively.

### 3.3 Induction of Derivational Families

Recall that our goal is to induce derivational families, that is, classes of derivationally related words. We define derivational families on the basis of derivational rules as follows.

Given a lemma-paradigm pair $(l, p)$ as input, a single derivational rule $d = (t, \mathcal{P}_1, \mathcal{P}_2)$ generates a set of possible derivations $L_d(l, p) = \{(l_1, p_1), \ldots, (l_n, p_n)\}$, where $p \in \mathcal{P}_1$ and $p_i \in \mathcal{P}_2$ for all $i$. Given a set of derivational rules $\mathcal{D}$, we define a binary derivation relation $\rightarrow_{\mathcal{D}}$ between two lemma-paradigm pairs that holds if the second pair can be derived from the first one as:

$$(l_1, p_1) \rightarrow_{\mathcal{D}} (l_2, p_2) \quad (2)$$
$$\text{iff} \quad \exists d \in \mathcal{D}.\, (l_2, p_2) \in L_d(l_1, p_1)$$

Let $\mathcal{L}$ denote the set of lemma-paradigm pairs. The set of *derivational families defined by $\mathcal{D}$ on $\mathcal{L}$* is given by the equivalence classes of the transitive, symmetric, and reflexive closure of $\rightarrow_{\mathcal{D}}$ over $\mathcal{L}$.

Note that in addition to the quality of the rules, the properties of $\mathcal{L}$ plays a central role in the quality of the induced families. High coverage of $\mathcal{L}$ is important because the transitivity of $\rightarrow_{\mathcal{D}}$ ranges only over lemmas in $\mathcal{L}$, so low coverage of $\mathcal{L}$ may result in fragmented derivational families. However, $\mathcal{L}$ should also not contain erroneous lemma-paradigm pairs. The reason is that the derivational rules only define *admissible* derivations, which need not be morphologically valid, and therefore routinely overgenerate; $\mathcal{L}$ plays an important role in filtering out derivations that are not attested in the data.

## 4 Building the Resource

### 4.1 Derivational Rules

We implemented the derivational rules from Hoeppner (1980) for verbs, nouns, and adjectives, covering all processes described in Section 3.1 (zero derivation, prefixation, suffixation, circumfixation, and stem changes). We found many derivational patterns in German to be conceptually simple (e.g., verb-noun zero derivation) so that substantial coverage can already be achieved with very simple transformation functions. However, there are many more complex patterns (e.g., suffixation combined with optional stem changes) that in sum also affect a considerable number of lemmas, which required us to either implement low-coverage rules or generalize existing rules. In order to preserve precision as much as possible, we restricted rule application by using *try* instead of *opt*, and by using gender information from the noun paradigms (for example, some rules only apply to masculine nouns and produce female nouns). As a result, we end up with high-coverage rules, such as derivations of person-denoting nouns (*Schule*$_N$ – *Schüler*$_N$ (*school – pupil*)) as well as high-accuracy rules such as negation prefixes (*Pol*$_N$ – *Gegenpol*$_N$ (*pole – antipole*)).

Even though we did not focus on the explanatory relevance of rules, we found that the underlying modeling formalism, and the methodology used to develop the model, offer substantial linguistic plausibility in practice. We had to resort to heuristics mostly for words with derivational transformations that are motivated by Latin or Greek morphology and do not occur regularly in German, e.g., *selegieren*$_V$ – *Selektion*$_N$ (*select – selection*).

In the initial development phase, we implemented 154 rules, which took about 22 person-hours. We then revised the rules with the aim of increasing both precision and recall. To this end, we constructed a development set comprised of a sample of 1,000 derivational families induced using our rules. On this set, we inspected the derivational families for false positives, identified the problematic rules, and identified unused and redundant rules. In order to identify the false negatives, we additionally sampled a list of 1,000 lemmas and used string distance measures (cf. Section 5.1) to retrieve the 10 most similar words for each lemma not

| Process | N-N | N-A | N-V | A-A | A-V | V-V |
|---|---|---|---|---|---|---|
| Zero derivation | – | 1 | 5 | – | – | – |
| Prefixation | 10 | – | 5 | 5 | 2 | 9 |
| + Stem change | – | – | 3 | – | 1 | – |
| Suffixation | 15 | 35 | 20 | 1 | 14 | – |
| + Stem change | 2 | 8 | 7 | – | 3 | 1 |
| Circumfixation | – | – | 1 | – | – | – |
| + Stem change | – | – | 1 | – | – | – |
| Stem change | – | – | 7 | – | – | 2 |
| *Total* | 27 | 44 | 49 | 6 | 20 | 12 |

Table 2: Breakdown of derivation rules by category of the basis and the derived word

already covered by the derivational families. The refinement process took another 8 person-hours. It revealed three redundant rules and seven missing rules, leading us to a total of 158 rules.

Table 2 shows the distribution of rules with respect to the derivational processes they implement and the part of speech combinations for the basis and the derived words. All affixations occur both with and without stem changes, mostly umlaut shifts. Suffixation is by far the most frequently used derivation process, and noun-verb derivation is most diverse in terms of derivational processes.

We also estimated the reliability of derivational rules by analyzing the accuracy of each rule on the development set. We assigned each rule a confidence rating on a three-level scale: L3 – very reliable (high-accuracy rules), L2 – generally reliable, and L1 – less reliable (low-accuracy rules). We manually analyzed the correctness of rule applications for 100 derivational families of different size (counting 2 up to 114 lemmas), and assigned 55, 79, and 24 rules to L3, L2 and L1, respectively.

### 4.2 Data and Preprocessing

For an accurate application of nominal derivation rules, we need a lemma list with POS and gender information. We POS-tag and lemmatize SDEWAC, a large German-language web corpus from which boilerplate paragraphs, ungrammatical sentences, and duplicate pages were removed (Faaß et al., 2010). For POS tagging and lemmatization, we use TreeTagger (Schmid, 1994) and determine grammatical gender with the morphological layer of the MATE Tools (Bohnet, 2010). We treat proper nouns like common nouns.

We apply three language-specific filtering steps based on observations in Section 3.1. First, we discard non-capitalized nominal lemmas. Second, we deleted verbal lemmas not ending in verb suffixes. Third, we removed frequently occurring erroneous comparative forms of adjectives (usually formed by adding *-er*, like *neuer / newer*) by checking for the presence of lemmas without *-er (neu / new)*.

An additional complication in German concerns prefix verbs, because prefix is separated in tensed instances. For example, the 3rd person male singular of *aufhören (to stop)* is *er hört auf (he stops)*. Since most prefixes double as prepositions, the correct lemmas can only be reconstructed by parsing. We parse the corpus using the MST parser (McDonald et al., 2006) and recover prefix verbs by searching for instances of the dependency relation labeled PTKVZ.

Since SDEWAC, as a web corpus, still contains errors, we only take into account lemmas that occur three times or more in the corpus. Considering the size of SDEWAC, we consider this as a conservative filtering step that preserves high recall and provides a comprehensive basis for evaluation. After preprocessing and filtering, we run the induction of the derivational families as explained in Section 3 to obtain the DERIVBASE resource.

### 4.3 Statistics on DERIVBASE

The preparation of the SDEWAC corpus as explained in Section 4.2 yields 280,336 lemmas, which we cover with our resource. We induced a total of 239,680 derivational families from this data, with 17,799 non-singletons and 221,881 singletons (most of them due to compound nouns). 11,039 of the families consist of two lemmas, while the biggest contains 116 lemmas (an overgenerated family). The biggest family with perfect precision (i.e., it contains only morphologically related lemmas) contains 40 lemmas, e.g., *halten*$_V$, *erhalten*$_V$, *Verhältnis*$_N$ (*to hold*, *to uphold*, *relation*), etc. For comparison, CatVar v2.1 contains only 82,676 lemmas in 13,368 non-singleton clusters and 38,604 singletons.

The following sample family has seven members across all three POSes and includes prefixation, suffixation, and infix umlaut shifts:

*taub*$_A$ (*numb*$_A$), *Taubheit*$_{Nf}$ (*numbness*$_N$), *betäuben*$_V$ (*to anesthetize*$_V$), *Betäubung*$_{Nf}$ (*anesthesia*$_N$), *betäubt*$_A$ (*anesthetized*$_A$), *betäubend*$_A$ (*anesthetic*$_A$), *Betäuben*$_{Nn}$ (*act of anesthetizing*$_N$)

## 5 Evaluation

### 5.1 Baselines

We use two baselines against which we compare the induced derivational families: (1) clusters obtained with the German version of Porter's stemmer (Porter, 1980)[3] and (2) clusters obtained using string distance-based clustering. We have considered a number of string distance measures and tested them on the development set (cf. Section 4.1). The measure proposed by Majumder et al. (2007) turned out to be the most effective in capturing suffixal variation. For words $X$ and $Y$, it is defined as

$$D_4(X,Y) = \frac{n-m+1}{n+1} \sum_{i=m}^{n} \frac{1}{2^{i-m}} \quad (3)$$

where $m$ is the position of left-most character mismatch, and $n+1$ is the length of the longer of the two strings. To capture prefixal variation and stem changes, we use the n-gram based measure proposed by Adamson and Boreham (1974):

$$Dice_n(X,Y) = 1 - \frac{2c}{x+y} \quad (4)$$

where $x$ and $y$ are the total number of distinct $n$-grams in $X$ and $Y$, respectively, and $c$ is the number of distinct $n$-grams shared by both words. In our experiments, the best performance was achieved with $n = 3$.

We used hierarchical agglomerative clustering with average linkage. To reduce the computational complexity, we performed a preclustering step by recursively partitioning the set of lemmas sharing the same prefix into partitions of manageable size (1000 lemmas). Initially, we set the number of clusters to be roughly equal to the number of induced derivational families. For the final evaluation, we optimized the number of clusters based on $F_1$ score on calibration and validation sets (cf. Section 5.3).

### 5.2 Evaluation Methodology

The induction of derivational families could be evaluated globally as a clustering problem. Unfortunately, cluster evaluation is a non-trivial task for which there is no consensus on the best approach (Amigó et al., 2009). We decided to perform our evaluation at the level of pairs: we manually judge for a set of pairs whether they are derivationally related or not.

[3] http://snowball.tartarus.org

We obtain the gold standard for this evaluation by sampling lemmas from the lemma list. With random sampling, the evaluation would be unrealistic because a vast majority of pairs would be derivationally unrelated and count as true negatives in our analysis. Moreover, in order to reliably estimate the overall precision of the obtained derivational families, we need to evaluate on pairs sampled from these families. On the other hand, in order to assess recall, we need to sample from pairs that are not included in our derivational families.

To obtain reliable estimates of both precision and recall, we decided to draw two different samples: (1) a sample of lemma pairs sampled from the induced derivational families, on which we estimate precision (*P-sample*) and (2) a sample of lemma pairs sampled from the set of possibly derivationally related lemma pairs, on which we estimate recall (*R-sample*). In both cases, pairs $(l_1, l_2)$ are sampled in two steps: first a lemma $l_1$ is drawn from a non-singleton family, then the second lemma $l_2$ is drawn from the derivational family of $l_1$ (P-sample) or the set of lemmas possibly related to $l_1$ (R-sample). The set of possibly related lemmas is a union of the derivational family of $l_1$, the clusters of $l_1$ obtained with the baseline methods, and $k$ lemmas most similar to $l_1$ according to the two string distance measures. We use $k = 7$ in our experiments. This is based on preliminary experiments on the development set (cf. Section 4.1), which showed that $k = 7$ retrieves about $92\%$ of the related lemmas retrieved for $k = 20$ with a much smaller number of true negatives. Thus, the evaluation on the R-sample might overestimate the recall, but only slightly so, while the P-sample yields a reliable estimate of precision by reducing the number of true negatives in the sample.

Both samples contain 2400 lemma pairs each. Lemmas included in the development set (Section 4.1) were excluded from sampling.

### 5.3 Gold Standard Annotation

Two German native speakers annotated the pairs from the P-sample and R-samples. We defined five categories into which all lemma pairs are classified as shown in Table 3. We count R and M as positives and N, C, L as negatives (cf. Section 3).[4] Note that this binary distinction would be sufficient to compute recall and precision. However, the more

[4] Ambiguous lemmas are categorized as positive (R or M) if there is a matching sense.

| Label | Description | Example |
|---|---|---|
| **R** | $l_1$ and $l_2$ are morphologically and semantically related | *kratzig*$_A$ – *verkratzt*$_A$ (*scratchy – scuffed*) |
| **M** | $l_1$ and $l_2$ are morphologically but not semantically related | *bomben*$_V$ – *bombig*$_A$ (*to bomb – smashing*) |
| **N** | no morphological relation | *belebt*$_A$ – *loben*$_V$ (*lively – to praise*) |
| **C** | no derivational relation, but the pair is compositionally related | *Filmende*$_N$ – *filmen*$_V$ (*end of film – to film*) |
| **L** | not a valid lemma (mislemmatization, wrong gender, foreign words) | *Haufe*$_N$ – *Häufung*$_N$ (*N/A – accumulation*) |

Table 3: Categories for lemma pair classification

| | Agreement | Cohen's $\kappa$ |
|---|---|---|
| R-sample | 0.85 | 0.79 |
| P-sample | 0.86 | 0.70 |

Table 4: Inter-annotator agreement on validation sample

fine-grained five-class annotation scheme provides a more detailed picture. The separation between R and M gives a deeper insight into the semantics of the derivational families. Distinguishing between C and N, in turn, allows us to identify the pairs that are derivationally unrelated, but compositionally related, e.g., *Ehemann*$_N$ – *Ehefrau*$_N$ (*husband – wife*).

We first carried out a calibration phase in which the annotators double-annotated 200 pairs from each of the two samples and refined the annotation guidelines. In a subsequent validation phase, we computed inter-annotator agreements on the annotations of another 200 pairs each from the P- and the R-samples. Table 4 shows the proportion of identical annotations by both annotators as well as Cohen's $\kappa$ score (Cohen, 1968). We achieve substantial agreement for $\kappa$ (Carletta, 1996). On the P-sample, $\kappa$ is a little lower because the distribution of the categories is skewed towards R, which makes an agreement by chance more probable.

In our opinion, the IAA results were sufficiently high to switch to single annotation for the production phase. Here, each annotator annotated another 1000 pairs from the P-sample and R-sample so that the final test set consists of 2000 pairs from each sample. The P-sample contains 1663 positive (R+M) and 337 negative (N+C+L) pairs, respectively, the R-sample contains 575 positive and 1425 negative pairs. As expected, there are more positive pairs in the P-sample and more negative pairs in the R-sample.

| | Precision | Recall |
|---|---|---|
| Method | P-sample | R-sample |
| DERIVBASE (initial) | 0.83 | 0.58 |
| DERIVBASE-L123 | 0.83 | **0.71** |
| DERIVBASE-L23 | 0.88 | 0.61 |
| DERIVBASE-L3 | **0.93** | 0.35 |
| | R-sample | |
| Stemming | 0.66 | 0.07 |
| String distance $D_4$ | 0.36 | 0.20 |
| String distance $Dice_3$ | 0.23 | 0.23 |

Table 5: Precision and recall on test samples

## 6 Results

### 6.1 Quantitative Evaluation

Table 5 presents the overall results. We evaluate four variants of the induced derivational families: those obtained before rule refinement (DERIVBASE initial), and three variants after rule refinement: using all rules (DERIVBASE-L123), excluding the least reliable rules (DERIVBASE-L23), and using only highly reliable rules (DERIVBASE-L3).

We measure the precision of our method on the P-sample and recall on the R-sample. For the baselines, precision was also computed on the R-sample (computing it on P-sample, which is obtained from the induced derivational families, would severely underestimate the number of false positives). We omit the $F_1$ score because its use for precision and recall estimates from different samples is unclear.

DERIVBASE reaches 83% precision when using all rules and 93% precision when using only highly reliable rules. DERIVBASE-L123 achieves the highest recall, outperforming other methods and variants by a large margin. Refinement of the initial model has produced a significant improvement in recall without losses in precision. The baselines perform worse than our method: the stemmer we use is rather conservative, which fragments the families and leads to a very low recall. The string distance-based approaches achieve more balanced precision and recall scores. Note that for these methods, precision and recall can be traded off against each other by varying the number of clusters; we chose the number of clusters by optimizing the $F_1$ score on the calibration and validaton sets.

All subsequent analyses refer to DERIVBASE-

| | Accuracy | | |
|---|---|---|---|
| Coverage | High | Low | *Total* |
| High | 18 | – | 18 |
| Low | 53 | 21 | 74 |
| *Total* | 71 | 21 | 92 |

Table 6: Proportions of accuracy and coverage for direct derivations (measured on P-sample)

| | P | R | | P | R |
|---|---|---|---|---|---|
| N-N | 0.78 | 0.68 | N-A | **0.89** | **0.83** |
| A-A | 0.87 | 0.70 | N-V | 0.79 | 0.68 |
| V-V | 0.55 | 0.24 | A-V | 0.88 | 0.73 |

Table 7: Precision and recall across different part of speech (first POS: basis; second POS: derived word)

L123, which is the model with the highest recall. If optimal precision is required, DERIVBASE-L3 should however be preferred.

**Analysis by frequency.** We cross-classified our rules according to high/low accuracy and high/low coverage based on the pairs in the P-sample. We only considered directly derivationally related ($\rightarrow_{\mathcal{D}}$) pairs and defined "high accuracy" and "high coverage" as all rules above the 25th percentile in terms of accuracy and coverage, respectively. The results are shown in Table 6: all high-coverage rules are also highly accurate. Most rules are accurate but infrequent. Only 21 rules have a low accuracy, but all of them apply infrequently.

**Analysis by parts of speech.** Table 7 shows precision and recall values for different part of speech combinations for the basis and derived words. High precision and recall are achieved for N-A derivations. The recall is lowest for V-V derivations, suggesting that the derivational phenomena for this POS combination are not yet covered satisfactorily.

### 6.2 Error analysis

Table 8 shows the frequencies of true positives and false positives on the P-sample and false negatives on the R-sample for each annotated category. True negatives are not reported, since their analysis gives no deeper insight.

**True positives.** In our analysis we treated both R and M pairs as related, but it is interesting to see how many of the true positives are in fact semantically unrelated. Out of 1,663 pairs, 90% are semantically as well as morphologically related (R), e.g., *alkoholisieren*$_V$ – *antialkoholisch*$_A$ (*to alcoholize – nonalcoholic*), *Beschuldigung*$_N$ – *unschuldig*$_A$ (*accusation – innocent*). Most R pairs result from high-accuracy rules, i.e., zero derivation, negation prefixation and simple suffixation. The remaining 10% are only morphologically related (M), e.g., *beschwingt*$_A$ – *schwingen*$_V$ (*cheerful – to swing*), *Stolzieren*$_N$ – *stolz*$_A$ (*strut – proud*). In both pairs, the two lemmas share a common semantic concept – i.e., being in motion or being proud – but nowaday's meanings have grown apart from each other. Among the M true positives, we observe prefixation derivations in 66% of the cases, often involving prefixation at both lemmas, e.g., *Erdenkliche*$_N$ – *bedenklich*$_A$ (*imaginable – questionable*).

| | TPs | FPs | FNs |
|---|---|---|---|
| Label | P-sample | P-sample | R-sample |
| R | 1,492 | – | 107 |
| M | 171 | – | 60 |
| N | – | 216 | – |
| C | – | 7 | – |
| L | – | 114 | – |
| *Total* | 1,663 | 337 | 167 |

Table 8: Predictions over annotated categories

**False positives.** We observe many errors in pairs involving short lemmas, e.g., *Gen*$_N$ – *genieren*$_V$ (*gene – to be embarrassed*), where orthographic context is unsufficient to reject the derivation. About 64% of the 337 incorrect pairs are of class N (unrelated lemmas). For example, the rule for deriving nouns denoting a male person incorrectly links *Morse*$_N$ – *Mörser*$_N$ (*Morse – mortar*). Transitively applied rules often produce incorrect pairs; e.g., *Speiche*$_N$ – *speicherbar*$_A$ (*spoke – storable*) results from the rule chain *Speiche*$_N$ $\rightarrow$ *Speicher*$_N$ $\rightarrow$ *speichern*$_V$ $\rightarrow$ *speicherbar*$_A$ *(spoke $\rightarrow$ storage $\rightarrow$ to store $\rightarrow$ storable)*. Chains that involve ablaut shifts (cf. Section 3.1) can lead to surprising results, e.g., *Erringung*$_N$ – *rangiert*$_A$ (*achievement – shunted*). Meanwhile, some pairs judged as unrelated by the annotators might conceivably be weakly related, such as *schlürfen*$_V$ and *schlurfen*$_V$ *(to sip – to shuffle)*, both of which refer to specific long drawn out sounds. About 20% out of these unrelated lemma pairs is due to derivations between proper nouns (PNs) and common nouns. This happens especially for short PNs (cf. the above example of *Morse*). However, since PNs also participate in valid derivations (e.g., *Chaplin – chaplinesque*),

one could investigate their impact on derivations rather than omitting them.

Errors of the category L – 34% of the false positives – are caused during preprocessing by the lemmatizer. They cannot be blamed on our derivational model, but of course form part of the output.

**False negatives.** Errors of this type are due to missing derivation rules, erroneous rules that leave some lemmas undiscovered, or the absence of lemmas in the corpus required for transitive closure. About 64% of the 167 missed pairs are of category R. About half of these pairs result from a lack of prefixation rules – mainly affecting verbs – with a wide variety of prefixes (*zu-*, *um-*, etc.), including prepositional prefixes like *herum- (around)* or *über- (over)*. We intentionally ignored these derivations, since they frequently lead to semantically unrelated pairs. In fact, merely five of the remaining 36% false negative pairs (M) do not involve prefixation. However, this analysis as well as the rather low coverage for verb-involved rules (cf. Table 7) shows that DERIVBASE might benefit from more prefix rules. Apart from the lack of prefixation coverage and a few other, rather infrequent rules, we did not find any substantial deficits. Most of the remaining errors are due to German idiosyncrasies and exceptional derivations, e.g., $fahren_V - Fahrt_N$ (*drive – trip*), where the regular zero derivation would result in *Fahr*.

## 7 Conclusion and Future Work

In this paper, we present DERIVBASE, a derivational resource for German based on a rule-based framework. A few work days were enough to build the underlying rules with the aid of grammar textbooks. We collected derivational families for over 280,000 lemmas with high accuracy as well as solid coverage. The resource is freely available.[5]

Our approach for compiling a derivational resource is not restricted to German. In addition to the typologically most similar Germanic and Romance languages, it is also applicable to agglutinative languages like Finnish, or other fusional languages like Russian. Its main requirements are a large list of lemmas for the language (optionally with further morphological features) and linguistic literature on morphological patterns.

We have employed an evaluation method that uses two separate samples to assess precision and recall to deal with the high number of false negatives. Our analyses indicate two interesting directions for future work: (a) specific handling of proper nouns, which partake in specific derivations; and (b) the use of graph clustering instead of the transitive closure to avoid errors resulting from long transitive chains.

Finally, we plan to employ distributional semantics methods (Turney and Pantel, 2010) to help remove semantically unrelated pairs as well as distinguish automatically between only morphologically (M) or both morphologically and semantically (R) related pairs. Last, but not least, this allows us to group derivation rules according to their semantic properties. For example, nouns with *-er* suffixes often denote persons and are agentivizations of a basis word (Bilgin et al., 2004).

[5] `http://goo.gl/7KG2U`; license cc-by-sa 3.0

## Acknowledgments

The first and third authors were supported by the EC project EXCITEMENT (FP7 ICT-287923). The second author was supported by the Croatian Science Foundation (project 02.03/162: "Derivational Semantic Models for Information Retrieval"). We thank the reviewers for their constructive comments.

## References

George W. Adamson and Jillian Boreham. 1974. The use of an association measure based on character structure to identify semantically related pairs of words and document titles. *Information Processing and Management*, 10(7/8):253–260.

Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. 2009. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12(4):461–486.

Gerhard Augst. 1975. *Lexikon zur Wortbildung*. Forschungsberichte des Instituts für Deutsche Sprache. Narr, Tübingen.

Harald R. Baayen, Richard Piepenbrock, and Leon Gulikers. 1996. *The CELEX Lexical Database. Release 2. LDC96L14*. Linguistic Data Consortium, University of Pennsylvania, Philadelphia, PA.

Kenneth R Beesley and Lauri Karttunen. 2003. *Finite state morphology*, volume 18. CSLI publications Stanford.

Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2012. Learning entailment relations by global graph structure optimization. *Computational Linguistics*, 38(1):73–111.

Balthazar Bickel and Johanna Nichols. 2001. Inflectional morphology. In Timothy Shopen, editor, *Language Typology and Syntactic Description, Volume III: Grammatical categories and the lexicon*, pages 169–240. CUP, Cambridge.

Orhan Bilgin, Özlem Çetinoğlu, and Kemal Oflazer. 2004. Morphosemantic relations in and across Wordnets. In *Proceedings of the Global WordNet Conference*, pages 60–66, Brno, Czech Republic.

Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 89–97, Beijing, China.

Eric Brill. 1992. A simple rule-based part of speech tagger. In *Proceedings of the Workshop on Speech and Natural Language*, pages 112–116, Harriman, New York.

Jean C. Carletta. 1996. Assessing agreement on classification tasks: the kappa statistic. *Computational Linguistics*, 22(2):249–254.

Jacob Cohen. 1968. Weighted kappa: Nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70:213–220.

Béatrice Daille, Cécile Fabre, and Pascale Sébillot. 2002. Applications of computational morphology. In Paul Boucher, editor, *Many Morphologies*, pages 210–234. Cascadilla Press.

Hervé Déjean. 1998. Morphemes as necessary concept for structures discovery from untagged corpora. In *Proceedings of the Joint Conferences on New Methods in Language Processing and Computational Natural Language Learning*, pages 295–298, Sydney, Australia.

Gertrud Faaß, Ulrich Heid, and Helmut Schmid. 2010. Design and application of a gold standard for morphological analysis: SMOR in validation. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation*, pages 803–810.

Christiane Fellbaum, Anne Osherson, and Peter Clark. 2009. Putting semantics into WordNet's "morphosemantic" links. In *Proceedings of the Third Language and Technology Conference*, pages 350–358, Poznań, Poland.

Wolfgang Finkler and Günter Neumann. 1988. Morphix - a fast realization of a classification-based approach to morphology. In *Proceedings of 4th Austrian Conference of Artificial Intelligence*, pages 11–19, Vienna, Austria.

Arne Fitschen. 2004. *Ein computerlinguistisches Lexikon als komplexes System*. Ph.D. thesis, IMS, Universität Stuttgart.

Éric Gaussier. 1999. Unsupervised learning of derivational morphology from inflectional lexicons. In *ACL'99 Workshop Proceedings on Unsupervised Learning in Natural Language Processing*, pages 24–30, College Park, Maryland, USA.

Rebecca Green, Bonnie J. Dorr, and Philip Resnik. 2004. Inducing frame semantic verb classes from wordnet and ldoce. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 375–382, Barcelona, Spain.

Nizar Habash and Bonnie Dorr. 2003. A categorial variation database for English. In *Proceedings of the Anuual Meeting of the North American Association for Computational Linguistics*, pages 96–102, Edmonton, Canada.

Harald Hammarström and Lars Borin. 2011. Unsupervised learning of morphology. *Computational Linguistics*, 37(2):309–350.

Wolfgang Hoeppner. 1980. *Derivative Wortbildung der deutschen Gegenwartssprache und ihre algorithmische Analyse*. Narr, Tübingen.

Lauri Karttunen and Kenneth R Beesley. 1992. *Two-level rule compiler*. Xerox Corporation. Palo Alto Research Center.

Lauri Karttunen and Kenneth R. Beesley. 2005. Twenty-five years of finite-state morphology. In Antti Arppe, Lauri Carlson, Krister Lindén, Jussi Piitulainen, Mickael Suominen, Martti Vainio, Hanna Westerlund, and Anssi Yli-Jyr, editors, *Inquiries into Words, Constraints and Contexts. Festschrift for Kimmo Koskenniemi on his 60th Birthday*, pages 71–83. CSLI Publications, Stanford, California.

Kimmo Koskenniemi. 1983. *Two-level Morphology: A General Computational Model for Word-Form Recognition and Production*. Ph.D. thesis, University of Helsinki.

Catherine Macleod, Ralph Grishman, Adam Meyers, Leslie Barrett, and Ruth Reeves. 1998. NOMLEX: A lexicon of nominalizations. In *In Proceedings of Euralex98*, pages 187–193.

Prasenjit Majumder, Mandar Mitra, Swapan K. Parui, Gobinda Kole, Pabitra Mitra, and Kalyankumar Datta. 2007. YASS: Yet another suffix stripper. *ACM Transactions on Information Systems*, 25(4):18:1–18:20.

Ryan McDonald, Kevin Lerman, and Fernando Pereira. 2006. Multilingual dependency analysis with a two-stage discriminative parser. In *In Proceedings of the Conference on Computational Natural Language Learning*, pages 216–220, New York, NY.

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004. Annotating noun argument structure for NomBank. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, Lisbon, Portugal.

Petar Milin, Victor Kuperman, Aleksandar Kostic, and R Harald Baayen. 2009. Paradigms bit by bit: An information theoretic approach to the processing of paradigmatic structure in inflection and derivation. *Analogy in grammar: Form and acquisition*, pages 214–252.

Karel Pala and Dana Hlaváčková. 2007. Derivational relations in Czech WordNet. In *Proceedings of the ACL Workshop on Balto-Slavonic Natural Language Processing: Information Extraction and Enabling Technologies*, pages 75–81.

Maciej Piasecki, Radoslaw Ramocki, and Marek Maziarz. 2012. Recognition of Polish derivational relations based on supervised learning scheme. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation*, pages 916–922, Istanbul, Turkey.

Martin Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.

Anne Schiller, Simone Teufel, Christine Stöckert, and Christine Thielen. 1999. Guidelines für das Tagging deutscher Textcorpora mit STTS. Technical report, Institut fur maschinelle Sprachverarbeitung, Stuttgart.

Helmut Schmid, Arne Fitschen, and Ulrich Heid. 2004. Smor: A German computational morphology covering derivation, composition and inflection. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation*, Lisbon, Portugal.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*, pages 44–49, Manchester, UK.

Patrick Schone and Daniel Jurafsky. 2000. Knowledge-free induction of morphology using latent semantic analysis. In *Proceedings of the Conference on Natural Language Learning*, pages 67–72, Lisbon, Portugal.

Jan Šnajder and Bojana Dalbelo Bašić. 2008. Higher-order functional representation of Croatian inflectional morphology. In *Proceedings of the 6th International Conference on Formal Approaches to South Slavic and Balkan Languages*, pages 121–130, Dubrovnik, Croatia.

Jan Šnajder and Bojana Dalbelo Bašić. 2010. A computational model of Croatian derivational morphology. In *Proceedings of the 7th International Conference on Formal Approaches to South Slavic and Balkan Languages*, pages 109–118, Dubrovnik, Croatia.

Idan Szpektor and Ido Dagan. 2008. Learning entailment rules for unary templates. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 849–856, Manchester, UK.

Kapil Thadani and Kathleen McKeown. 2011. Towards strict sentence intersection: Decoding and evaluation strategies. In *Proceedings of the ACL Workshop on Monolingual Text-To-Text Generation*, pages 43–53, Portland, Oregon.

Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37(1):141–188.

Andrea Zielinski and Christian Simon. 2008. Morphisto - an open source morphological analyzer for German. In *Proceedings of the 7th International Workshop on Finite-State Methods and Natural Language Processing*, pages 224–231, Ispra, Italy.