On German verb sense disambiguation: A three-part approach based on linking a sense inventory (GermaNet) to a corpus through annotation (TGVCorp) and using the corpus to train a VSD classifier (TTvSense)

Dominik Mattern[1], *Wahed Hemati*[2], *Andy Lücking*[1], *and Alexander Mehler*[1]
[1] Goethe University Frankfurt, Text Technology Lab
[2] Shikenso GmbH

ABSTRACT

We develop a three-part approach to Verb Sense Disambiguation (VSD) in German. After considering a set of lexical resources and corpora, we arrive at a statistically motivated selection of a subset of verbs and their senses from `GermaNet`. This sub-inventory is then used to disambiguate the occurrences of the corresponding verbs in a corpus resulting from the union of `TüBa-D/Z`, `Salsa`, and `E-VALBU`. The corpus annotated in this way is called `TGVCorp`. It is used in the third part of the paper for training a classifier for VSD and for its comparative evaluation with a state-of-the-art approach in this research area, namely `EWISER`. Our simple classifier outperforms the transformer-based approach on the same data in both accuracy and speed in German but not in English and we discuss possible reasons.

Keywords:
verb sense disambiguation (VSD), word sense disambiguation (WSD)

Journal of Language Modelling Vol 12, No 1 (2024), pp. 155–212

1 INTRODUCTION

Ambiguity arises when a word or a multi-word constituent is associated with more than one meaning (Chierchia and McConnell-Ginet 2000, p. 38; see Kennedy 2011 for an overview). The multiple meanings of a word are referred to as *senses*. Choosing just one from the many senses of an ambiguous word in context is a process known as Word Sense Disambiguation (WSD) (Navigli 2009). Here we focus on *Verb Sense Disambiguation* (VSD), i.e., selecting a sense from the sense enumerations associated with a given verb. We present an approach to the disambiguation of German verbs. We briefly set the theoretical stage in Section 1.1 and review related NLP work in Section 1.2.

1.1 *Ambiguity and context variability*

VSD is a lexical issue: determining which of the verb's senses is appropriate in a given context.[1] Lexical ambiguity is expressed in terms of word sense enumerations: each meaning of an ambiguous word corresponds to one sense. Traditionally, lexical ambiguity is attributed to either polysemy (a single word form is associated with various senses) or homonymy (different senses happen to share the same orthographic (homograph) or phonological (homophone) representation) (Lyons 1977, p. 550). The two varieties of lexical ambiguity can be difficult to distinguish (though there are some guidelines, see Kroeger 2019, Section 5.3.3). Verb ambiguity is illustrated in (1), taken from Cruse (2000, p. 108):

(1) a. John expired last Thursday.
  b. John's driving licence expired last Thursday.
  c. ? John and his driving licence expired last Thursday.

[1] Thus, verbs exhibit *lexical ambiguity*. Other types of ambiguity known from nouns and adjectives and the phrases constructed out of those parts of speech are syntactic or structural ambiguity (*competent men and women*; Chierchia and McConnell-Ginet 2000, p. 38), as well as scope ambiguity (*Every schoolgirl crossed a road*; Dwivedi 2013).

The proper name *John* in (1a) calls for an interpretation of the verb *expire* in terms of "dying", while in (1b) an "end of period" reading is selected. Linguistic evidence for the polysemy of *expiring* is exemplified in (1c) (the question mark indicates semantic oddity): In the *antagonism test* (Kroeger 2019, Section 5.3.2), only different senses lead to the zeugma effect (the effect that the verb senses of conjoined verbs are antagonistic; for ambiguity tests see Zwicky and Sadock 1975; see Gillon 1990 for some critical discussion).

Disambiguation relies heavily on context information. For instance, keeping the two senses of *expiring* apart in (1) is based on world knowledge about proper names of persons and bureaucratic administrations. Accordingly, it is important to distinguish ambiguity from the general context variability of meanings (Cruse 2000, Chapter 6).[2] Let us illustrate the subtle differences between polysemy and context-variability by means of a positive and a negative example each. Consider the following sentences from German (since we are concerned with German VSD):

(2) a. Das Gerät **läuft** einwandfrei. (*The device works correctly.*)

b. Der Schaffner **läuft** zum Bahnhof. (*The ticket collector walks to the station.*)

c. ? Das Gerät **läuft** und der Schaffner auch. (? *The device is running and so is the ticket collector.*)

The verb form *läuft* has two different meanings in sentences (2a) and (2b), which can be paraphrased with "it works" and "it walks", respectively. It is noteworthy, but by no means a rule, that the same German word form receives a different English translation for each sense. For this reason, we will have a particular focus on multilingual

[2] Context-sensitive effects of contents include indexicality (the first person pronoun *I*, for example, is not ambiguous despite referring to a potentially different person on each occasion of use; Kaplan 1989), coercion (e.g., type-shifting the noun *novel* to an eventive argument in *He began the novel*; Moens and Steedman 1988; Pustejovsky 1995; de Swart 2011), co-composition or co-predication (as observed, for instance, with "interactive verb-argument compositions" such as *Pat swallowed the lemonade* vs. *Pat swallowed her worries*; Pustejovsky 1991, 1995; Asher *et al.* 2017; Cooper 2011).

WSD resources. (2c) shows that polysemy is indicated by the antagonism test, which leads to a zeugma effect. The two senses are correctly kept apart in our approach.

However, *laufen* 'to run' can also be used to denote directed or undirected movement (Jackendoff 1983):

(3) a. Er **läuft** so schnell es geht zum Zug. (*He runs to the train as fast as possible*; $run_1 = go\text{-}to(x,y)$)

b. Sie **läuft** durch den Park. (*She runs through the park*; $run_2 = move(x)$)

c. Sie **laufen** zum Zug und durch den Park. (*They run to the train and through the park.*)

In contrast to (2), *laufen* 'to run' in (3) passes the antagonism test without giving rise to a zeugma effect, which provides evidence for a shared verb sense in both conjuncts. Furthermore, both verb occurrences are translated to the same English word form. With regard to semantics, both directed and undirected movements follow from interactive meaning composition (Pustejovsky 1991), so no sense enumeration is needed. Thus the pattern in (3) is due to a single sense of the verb. Since (3a) and (3b) are attributed to different senses in our account, we observe some overgeneralization of lexical ambiguity.

What about figurative language use such as metaphor or metonymy? Cruse (2000, p. 112) puts them among polysemy, namely as non-linear types of polysemy.[3] However, this classification lacks empirical support: metonymic uses of a noun phrase, for instance, do not seem to rest on ambiguity, but rather on a "transfer of meaning" (predicate transfer, in this case) (Nunberg 1995).[4] Consequently, we take figurative speech to be a matter of inference, not of WSD.

A note on terminology: We use the terms "valence" or "subcategorization" for the syntactic arguments of a verb. For example, a tran-

[3] They are non-linear because they lack a linear specialization relationship towards their "siblings".

[4] To briefly rehash one of Nunberg's arguments: the noun phrase *ham sandwich*, even when used metonymically in a restaurant in order to refer to its orderer, still preserves its basic meaning since it can be picked out by discourse anaphora: *The ham sandwich seems to be enjoying it* (*it* = the ham sandwich).

sitive verb such as *eat* takes a subject and a complement – hence, there are two noun phrases on its valence or subactegorization list. These elements are mapped onto the verb's argument structure and linked to content representations (linking) (Wechsler *et al.* 2021). There are different approaches to representing contents; we will refer to semantic arguments of content representations as *semantic roles*.[5]

VSD for German 1.2

Word Sense Disambiguation (WSD) in general is essential for many (if not all) Natural Language Processing (NLP) applications that require semantic information. The disambiguation of verbs, VSD, is of particular importance when it comes to Semantic Role Labeling (SRL) (Palmer *et al.* 2010). This is due to the fact that the argument structure or subcategorization frame of verbs can differ with their senses. Consider again *laufen* 'to run' from (2). While (2a) and (2b) select for a nominal nominative subject, the subject is linked differently to the semantic arguments provided by the verb sense-specific predication. Such argument structure linking can be achieved in various ways including selectional restrictions (e.g. ±ANIMATE) (Soehn 2005) or lexical frames (respectively parameterized states of affairs; e.g. *operating-frame* vs. *movement-frame*) (Wechsler *et al.* 2021).[6] Thus, if the representation of meaning fails already on the level of verb occurrences in sentences, because it is not able to distinguish between different senses connected with the same form, then a precondition for determining the corresponding sentence meaning is missing (Levin 1993). This leads us to the assessment that any reasonable approach to sentence or text meaning representation (which goes beyond black box

[5] WSD approaches usually refrain from using argument structures in the grammar-theoretic sense and employ a direct mapping from syntactic arguments to semantic representations, as is done in Semantic Role Labeling (SRL). Hence, the term "argument structure" when used in these contexts is to be understood either in terms of syntactic subcategorization or semantic roles.

[6] Resources used for SRL differ in the granularity and nomenclature of their argument vocabularies. A recent resource addresses this inter-operability issue by providing yet another synset-based vocabulary but with links to FrameNet (Fillmore and Baker 2010), VerbNet (Schuler 2006), PropBank (Bonial *et al.* 2015) and `WordNet` (Fellbaum and Miller 1998) roles (Di Fabio *et al.* 2019).

models based e.g. on current neural networks) must perform VSD as a preprocessing step. Hence, there is already a history of lexical representations and WSD, including lexical resources (Miller 1995; Schuler 2006; Baker *et al.* 1998) and sense annotated corpora (Edmonds and Cotton 2001; Snyder and Palmer 2004; Pradhan *et al.* 2007; Navigli *et al.* 2013).

However, existing resources focus on English; there is little research on WSD in high resource languages such as German, especially for verbs. German WSD was featured on SemEval as a task or partial task only twice (Lefever and Hoste 2010, 2013), in both cases as part of a multilingual disambiguation task only involving a small number of nouns (see Figure 1).

To promote NLP for or based on SRL and related tasks in German, a correspondingly large dataset with high verb lemma coverage and a standardized sense inventory is needed. The present work aims to fill this gap by means of a three-layer architecture of VSD which integrates (1) the modeling and post-processing of verb sense representations with (2) the generation of training data annotation and (3) the machine learning based thereon. This approach, first elaborated in Hemati (2020) and considerably extended and further validated here, is compared in detail with related resources below. Such resources have been provided in few previous works on German verbs (for an evaluation of WSD algorithms for German *nouns* see Henrich and Hinrichs 2012):

1. The “Elektronische Valenzwörterbuch” (*electronic valence dictionary*) of German verbs, E-VALBU (Kubczak 2009), contains the 638 verbs from the printed VALBU (Schumacher *et al.* 2004), plus 30 new verb lemmas from the domain of a general science vocabulary. Grammatical descriptions and disambiguation of the E-VALBU verbs are based on their usage context in DEREKO (Dipper *et al.* 2002) and are obtained using corpus-assisted lexicographical methods (Schumacher 1986). For that reason, E-VALBU, though being a reference corpus, is of limited coverage.
2. Scheible *et al.* (2013) developed a rule-based *SubCat-Extractor*, which obtains subcategorization information from parsed corpora annotated with STTS (Schiller *et al.* 1999) such as the TIGER corpus (Brants *et al.* 2004). The SubCat-Extractor was applied

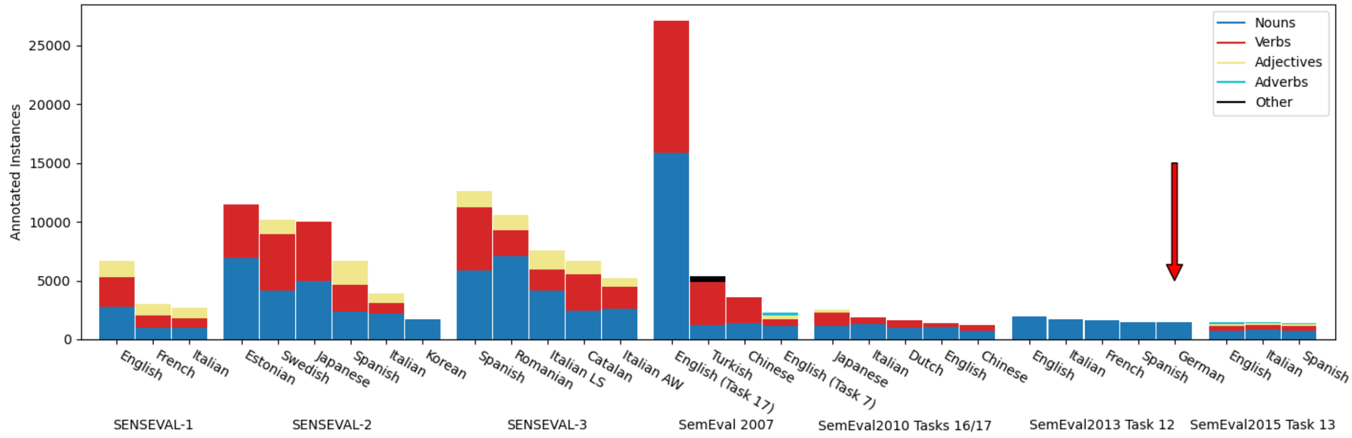

Figure 1: Number of annotated instances in the training and test corpora used for WSD during the SENSEVAL/SemEval Tasks. The SENSEVAL/SemEval tasks are a series of evaluations designed to assess the effectiveness of computational systems in understanding the meaning (or "sense") of words in different contexts, crucial for word sense disambiguation (WSD). Only supervised WSD tasks are included. Some of the languages are missing from `SENSEVAL-2` and `-3`, specifically Basque, Czech, Dutch and English from `SENSEVAL-2` and Basque, Chinese and English tasks from `SENSEVAL-3`. The papers for these languages did not include complete POS breakdowns of the corpora. In short, the distribution shows that verb-related WSD is a rare topic, especially for German

to SdeWac (Faaß and Eckart 2013). Although not explicitly connected to VSD, the resulting subcategorization lexicon of German verbs may contain different syntactic argument frames for a given verb, which often correlates with different semantic construals (as with the Levin 1993 classes). Since the verbs are retrieved from a large web-crawled database, the SubCat-Extractor resource has reasonable coverage. However, no explicit link to meaning labels is established.

3. VSD on a restricted class of verbs, namely perception verbs, was carried out by David *et al.* (2014). The focus of this paper was on distinguishing between perception verbs exhibiting literal and non-literal meanings. To this end, the authors selected one example of an optical, an acoustic, an olfactory, and a haptic verb each. The four verbs were assigned to 3 to 4 senses (1 literal and 2 to 3 non-literal), based on a corpus survey. Then a database was created by manually annotating 50 randomly chosen sentences for each selected perception verb in terms of the previously defined senses (i.e., 200 sentences in total). A decision tree was trained on the resulting dataset exploiting various features, partly drawing on the resource of Scheible *et al.* (2013). The classifier reached accuracies between 45.5 % and 69.4 %, however, due to the rather special focus of the approach it is difficult to generalize it to other VSD phenomena.

4. Henrich (2015) presents the most comprehensive work on VSD in German. She analyzed various corpora, including manually annotated and automatically created ones. In particular, she created a new German resource for WSD, namely WebCAGe (*Web-Harvested Corpus Annotated with GermaNet Senses*). WebCAGe rests on a semi-automatic alignment of Wiktionary glosses and GermaNet senses. Wiktionary was used to enlarge the set of sample sentences, most notably by exploiting links to Wikipedia articles. Following the "one sense per discourse" heuristics (Gale *et al.* 1992), occurrences of target words in external but linked sources are likely to be used in the same sense as that of the pivot word from a Wiktionary gloss. It should be noted that WebCAGe contains only words with more than one GermaNet sense, that is, words that are polysemous in GermaNet's sense – unambiguous

words are excluded on purpose (since WebCAGe is designed as a *disambiguation* dataset). The resource creation process was semi-automatic, as the large-scale annotation is done automatically, followed by a manual post-correction. The resulting dataset was evaluated by lexicographers. The focus of WebCAGe, however, was on WSD (i.e, nouns, verbs, and adjectives). As a result, Henrich (2015) does not achieve high coverage for German verbs: the disambiguation resource includes 3,190 tagged verb tokens which belong to 897 polysemous verbs in `GermaNet`, exhibiting 3.6 verb senses on average (Henrich 2015, p. 118).[7]

5. A cross-lingual, multimodal approach to VSD was taken by Gella *et al.* (2019). They provide the MultiSense image dataset, which comprises 9,504 images annotated with English verbs and their translations into German and Spanish. MultiSense covers 55 English verbs with 154 (German) and 136 (Spanish) unique translations. The dataset is divided into 75 % training, 10 % validation and 15 % test splits. The best performing model in a translation task was a mixed one which used visual and textual features. MultiSense departs from the sense enumeration paradigm (see Section 1.1) and delegates disambiguation to a translation process (namely translating the pivot verb into verbs of the remaining two target languages). Since the target language verbs are not disambiguated either, it is obvious that this approach only works for VSD if the target verbs are unambiguous – which is probably rarely the case (as a simple example reconsider (2)).[8]

In order to gain a better verb-related database for NLP in German beyond these resources, we created the TTLab German Verb Sense Corpus (`TGVCorp`). `TGVCorp` is a German corpus with a very high degree of coverage regarding the annotation of the senses of a high number of frequent verbs. Since the annotation of data is time-consuming and therefore cost-intensive, we developed a generic procedure to quickly

[7] In total WebCAGe contains 10,750 tagged word tokens which belong to 2,607 distinct polysemous words in `GermaNet` (Henrich 2015, p. 118).

[8] A further issue might reside in the *prima facie* appealing use of images as a *lingua franca*: While mundane, concrete actions can be depicted straightforwardly, it is difficult to see how more abstract contents such as those needed for attitude verbs are captured.

create high-quality training data for WSD. This procedure integrates three methods for the automatic generation of annotations employing translation models, language models and an inductive heuristics based on sense compression. `TGVCorp` contains manually annotated data for 1,560 ambiguous verb lemmas covering more than 78% of the verb tokens in `COW` (Schäfer and Bildhauer 2012), which is one of the largest openly accessible corpora for German. We use neural network-based tools for WSD and demonstrate their adaptation to VSD. We reproduce the experiments of Henrich (2015) and compare our approach with hers. In direct comparison to Henrich 2015, our most efficient model offers a performance increase of 8.4%, creating a new gold standard. We additionally present a simple method for generalizing senses that allows us to disambiguate verbs that are not present in the training set. With our approach, we achieve the highest verb token coverage for German VSD while maintaining state-of-the-art performance.

The paper is organized as follows: Section 2 describes `TGVCorp` and our procedure for creating it semi-automatically. Section 3 presents our supervised classifier for VSD based on `TGVCorp`. Finally, Section 4 concludes and discusses future work.

2 FROM RAW TEXTUAL DATA TO A SENSE-DISAMBIGUATED TEXT CORPUS: A THREE-LEVEL ARCHITECTURE

In this section we first describe the selection of the sense inventory underlying `TGVCorp`. We then turn to the generation of `TGVCorp` and evaluate its coverage using a larger set of different (genre- and topic-diverse) corpora. Finally, we describe the annotation of senses in this corpus, which are used in the remainder of the paper to train a supervised VSD classifier.

The significant expansion of annotation of verb senses in corpora is needed to train better classifiers for VSD. That is, instead of training new classifiers all the time, we rely on the idea of expanding the database and its quality to arrive at better NLP methods. To support the generation of such a resource on the example of VSD, each target verb requires a list of its senses with sufficient information per sense so that they can be adequately captured, identified, and distinguished from each other by annotators. Creating our own list from scratch would be too complex, so we used existing inventories to gain a working basis. Hence, the first step was to determine which inventory is most appropriate for German VSD (Section 2.1). Likewise, we had to choose a corpus to start with, so in addition we examined several corpora (Section 2.2). Since human annotation is costly, we combined several methods to map the selected corpus to the selected inventory while minimizing annotation effort and keeping data quality high (Section 2.3).

2.1 *Sense inventories*

A sense of a word *w* is a generally accepted meaning of *w* represented as a gloss, a paraphrase or as a synset in a WordNet (Fellbaum 1998). In a sense inventory these senses are enumerated per word. Independent of the question whether word senses can be enumerated as discretizable units, inventories map words to finite discrete sets of senses, each representing a certain meaning of the corresponding word. However, it is doubtful that there are periods of time in which the senses of a word can be completely discretized, so that one knows exactly where one sense begins and another ends (Rieger 1989, 2001). The discrete approach comes up against the fact that natural languages are permanently affected by change as a result of constantly changing contexts of language use (Keller 1990) – see Steels 2011–12 for a consideration of language dynamics from the point of view of evolutionary processes. This dynamic cannot be represented by sense lists, which are based on the implicit assumption of sufficiently stable senses, without actually measuring this stability: *Is the stability of the senses of words equally distributed?* (Most likely not.) *What does this stability depend on? Are the periods during which particular senses are observed sufficiently long so that a valid WSD can be performed? What does this mean for the selection of appropriate text corpora? Are these even sufficiently available for these periods?* Ideally, these and related questions should be clarified in order to make sense inventories a valid representation format.

Figure 2: Senses of the German verb *abtragen* 'to dismantle' in two sense inventories: Duden (download: February 14, 2024) (left) and `Wiktionary` (download: February 14, 2024) (right)

Source: Duden

abtragen / *to dismantle*

Senses:

[1.a] Wiktionary[1]
[1.b] Wiktionary[1]
[1.c] Wiktionary[4]
[2] Wiktionary[3]
[3] Wiktionary[2]
[4] Wiktionary[6]

Source: Wiktionary

abtragen / *to dismantle*

Senses:

[1] schichtweise entfernen
[2] Kleidung so lange benutzen, bis sie kaputt ist
[3] bezahlen
[4] *Haushalt, gehoben:* das Geschirr vom Tisch räumen
[5] *Medizin:* operativ entfernen
[6] *Geometrie:* Strecke auf Gerade festlegen

In any event, these time-related dynamics and delimitation-related uncertainties are probably two reasons why different dictionaries contain sense inventories of different composition and detail. This is illustrated by Figure 2, which shows the sense inventory of the verb *abtragen* 'to dismantle' as represented by Duden[9] and Wiktionary.[10] While there are three overlaps (Wiktionary[x], $x = 2, 3, 4$), there is one case where a Wiktionary sense (Wiktionary[1]) is divided into two Duden senses (1.a, 1.b) and one case of senses that the other resource does not know (Wiktionary[5]) – in 2019 (download: May 1, 2019), Duden[4] was unknown to Wiktionary. While the first deviation can be seen as a difference in semantic resolution, the second raises the more fundamental question of the "true set" of different senses assumed to exist independently of scientific observation, which in turn evokes the question which of the actual senses of the verb are not "listed". In other words, should we opt for Duden, Wiktionary, or the union of all such resources – and what does that leave open (assuming we have solved all the problems of sense matching or ontology matching as induced)?

[9] `https://www.duden.de/`
[10] `https://de.wiktionary.org/`

	Wiktionary: – verbs: 14 649 – senses: 29 894	**Duden:** – verbs: 19 278 – senses: 31 404
GermaNet: – verbs: 10 764 – senses: 18 336	– same verbs: 8 440 (78%,58%) – same sense num.: 2 798 (15%,9%) – same senses: 1 844 (10%,6%)	– same verbs: 10 319 (96%,54%) – same sense num.: 6 120 (33%,31%)

Figure 3: `GermaNet` in relation to `Wiktionary` and `Duden`; *same verbs:* word-form-based counting; *same number of senses:* based on the same number of distinguished senses (not necessarily the same); *same senses:* based on assignable senses

A more systematic summary of the differences is given in Figure 3. Using version 12 of `GermaNet` as a reference, it shows the overlap between this resource and Wiktionary and Duden in terms of verb forms, sense numbers, and in the case of Wiktionary, senses (using the mapping between the two resources). We see both remarkably low overlaps in terms of the verbs mapped (52% of the Duden verbs are mapped by this version of `GermaNet`) and, even more so, in terms of the sense inventory sizes. Again, this raises the question what alignments and potential unions would be necessary to arrive at a more complete ("truer") inventory – a task that is beyond the scope of this paper. Moreover, the first deviation in scale is related to the fact that different NLP applications require different granularities of word senses (Navigli 2009), which induces a third source of dynamics. Consequently, one might argue for an intrinsic approach that uses, e.g., transformers (Devlin *et al.* 2018) to represent senses indirectly as a result of postprocessing contextualized word representations rather than enumerating them in advance (see Pilehvar and Camacho-Collados 2021, p. 94 for an example).

While this approach has the advantage of adaptability (through fine-tuning) to ever-new corpora, it also has the disadvantage that senses appear as ephemeral entities that make identifications and comparisons across corpus boundaries difficult: ultimately, such an approach lacks a sufficient degree of explicitness necessary for delineating indisputably existing senses (see the introduction) as nameable objects of humanities research which ultimately make them a subject of separate studies. In light of these arguments, we pursue the path of using sense inventories to view word senses as *discrete, designatable* and *nameable* entities – and see this as a kind of working hypothesis.To survey all dictionaries and sense inventories available for German is beyond the scope of this paper. Therefore we focus on frequently used resources, that is, Duden (Duden *et al.* 1980), Wiktionary (Wiktionary 2019; Mehler *et al.* 2018) and `GermaNet` (Hamp and Feldweg 1997; Kunze and Lemnitzer 2002; Henrich *et al.* 2012) as a taxonomy:[11]

1. **Duden** is a spelling dictionary of German, first published in 1880, which subdivides lemmata into senses. Duden senses are enumerated and further differentiated by enumerating more granular word senses. The feature descriptions and senses are combined with examples from German text corpora or with manually created examples. Verb entries may contain lists of synonyms, with each list roughly corresponding to one sense of the verb. However, Duden contains relations at the lemma level, not at the sense level, as the synonym lists are not connected to senses.
2. **Wiktionary** is a dictionary developed under the auspices of the Wikimedia Foundation according to the Wiki principle. Word senses are enumerated and distinguished by descriptions and examples. Wiktionary specifies relationships such as synonyms, antonyms, hypernyms and hyponyms at the sense level (but not necessarily: in some cases they are specified only at the lemma level – for the details of this model cf. Mehler *et al.* (2018)). These relations point at units at the level of superlemmas and not of senses.
3. `GermaNet` is a terminological ontology similar to WordNet (Miller 1995; Fellbaum and Miller 1998). Senses are grouped together into synsets which are networked by means of semantic relations. The `GermaNet` subgraph containing only verbs has a tree-like core structure based on hyponym/hypernym relations.

The choice of a sense inventory is essential to keep VSD manageable, and to be able to process corpora with existing tools or use them to extend existing corpora. `GermaNet`'s WordNet-like structure

[11] For a lexicographic overview of web-based German dictionaries, see Storrer 2010; see Sowa 2000 for the characterization of wordnets as terminological ontologies.

Table 1: Number of verb lemmas, synsets, and senses in Duden, GermaNet and Wiktionary. Duden and Wiktionary do not (fully) specify relations at the sense level. These resources do not group senses into synsets so the corresponding entries for the number of synsets for these resources are empty. GermaNet distinguishes between senses and synsets, where the former are exemplified by sense glosses. The last row shows the coverage of the resource's verbs by COW

	GermaNet	Duden	Wiktionary
#verb lemmas	10,764	19,278	14,649
#verb synsets	14,178	∅	∅
#senses (senses or sense glosses)	18,336	41,441	29,894
coverage	97.9 %	93.6 %	97.4 %

offers many advantages for ML because of the sense relations it represents. Moreover, GermaNet describes these relations completely at the level of senses. It is constantly maintained, with several text corpora already mapped on GermaNet and tools available for their processing (Henrich and Hinrichs 2013; Henrich *et al.* 2012, 2011). Table 1 shows the number of lemmas and senses maintained by these resources: Duden contains the largest number of verbs, but the gain in coverage of the verbs annotated in COW (Schäfer and Bildhauer 2012), one of the largest openly available corpora for German, is marginal. That is, the verbs in Duden that are not included in GermaNet are apparently rare: the 9,349 verbs contained in Duden, but not in GermaNet, have a COW coverage of only 1.36 %. Likewise, the 6,209 verbs contained in Wiktionary but not in GermaNet have a COW coverage of only 0.85 %. Given its many advantages and its sufficiently high COW coverage, we selected GermaNet, and specifically the then current version 14, as an inventory of word senses.

Corpus creation 2.2

Having decided on a verb sense inventory, the next step is to create the TTLab German Verb Sense Corpus (TGVCorp) in which a sufficiently large number of verbs from this inventory are disambiguated at the sense level. To this end, we consider three boundary conditions that an ideal corpus should fulfill: (C1) a relevant number of verb lemmas should be covered, whose occurrences (C2) cover a large part of verb tokens observable in a reference corpus and (C3), a sufficient number of example sentences per lemma should be annotated so that ML models can be trained with this data. We choose `COW` as the reference corpus for C2 and use it to determine which verbs to disambiguate, and `TüBa-D/Z Treebank` as the text repository for examples for C3, coincidentally following the approach of Henrich (2015). This section describes how we arrive at these choices, giving an overview of existing German corpora and `COW` in particular in the process.

We want to prioritize high verb-token coverage (C2) over high verb-lemma coverage (C1), as this naturally helps with finding sufficient examples per lemma (C3). To do this, we process verbs according to their rank frequency distribution. This follows the idea that C2 is related to the power-law-like distribution of verb frequencies in corpora, thus selecting the most frequent verbs will quickly capture the 80% majority of verb-related tokens according to the Pareto principle (Newman 2005). In fact, the distributions of verb occurrences in a number of reference corpus candidates are heavy-tailed, see Table 2.[12]

Since verbs carry content as well as serve auxiliary functions, we distinguish the distribution of all verbs from that of verbs excluding modal and auxiliary verbs (that is, verbs mainly indicating possibility or necessity). The latter are usually the most frequent verbs by some distance. In order to achieve distributional profiles we compared a power law fit against a lognormal fit. Since R is negative or null in all cases, a lognormal distribution is the preferred fit. However, a lognormal fit is significant (i.e. $p \leq 0.05$) only for `GVSD`[13], `Wikipedia`, `Gutenberg`[14], `German Parliamentary Corpus`

[12] We apply the toolbox of Alstott *et al.* (2014) according to Clauset *et al.* (2009): power laws (first) are compared to lognormal distributions (second): "R is the loglikelihood ratio between the two candidate distributions. This number will be positive if the data is more likely in the first distribution, and negative if the data is more likely in the second distribution. The significance value for that direction is p." (Alstott *et al.* 2014, p. 5).

[13] German Verb Subcategorisation Database (GSDV), see Scheible *et al.* 2013.

[14] A free digital library with over 60,000 eBooks, including classics, for download or online reading; `https://www.gutenberg.org/`.

Table 2: Power law goodness-of-fit tests for the rank frequency distributions of verbs with and without modals (Mod.) in terms of the coefficient of (adjusted) determination (R resp. R^2) and the Kolmogorow-Smirnow test (test value KSstat and p-value KSp)

Name	Mod.	alpha	x-min	R	P	R^2	Adj. R^2	KSstat	KSp
COW	no	2.30	1,032,974.00	−0.46	0.52	0.90	0.90	0.03	0.97
COW	yes	2.04	1,464,713.00	0.00	0.95	0.97	0.97	0.04	0.99
deCOW16B	no	2.29	819,801.00	−0.42	0.54	0.91	0.91	0.03	0.96
deCOW16B	yes	2.09	723,889.00	−0.16	0.16	0.97	0.97	0.03	0.93
DTA	no	2.12	4,567.00	−1.12	0.33	0.86	0.86	0.02	0.96
DTA	yes	2.02	4,031.00	−0.01	0.95	0.98	0.98	0.03	0.87
GVSD	no	1.50	5.00	−13.53	0.00	0.91	0.91	0.03	0.84
GVSD	yes	1.50	5.00	−12.49	0.00	0.93	0.93	0.03	0.93
Gutenberg	no	1.52	8.00	−20.03	3.34×10^{-05}	0.91	0.91	0.03	0.67
Gutenberg	yes	1.52	8.00	−17.09	0.00	0.99	0.99	0.02	0.98
Leipzig	no	2.21	17,156.00	−1.35	0.30	0.95	0.95	0.04	0.82
Leipzig	yes	2.06	15,889.00	0.00	0.90	0.95	0.95	0.03	0.97
Parlament	no	1.40	3.00	−40.98	4.68×10^{-09}	0.93	0.93	0.04	0.85
Parlament	yes	2.03	17,683.00	0.00	0.80	0.95	0.95	0.03	0.97
SZ	no	1.43	5.00	−50.87	2.19×10^{-11}	0.94	0.94	0.03	0.95
SZ	yes	2.10	33,646.00	−1.04	0.14	0.96	0.96	0.02	1.00
Textbooks	no	2.24	233.00	−3.55	0.06	0.83	0.83	0.05	0.64
Textbooks	yes	2.11	219.00	−0.19	0.77	0.90	0.90	0.04	0.87
Tüba-D/Z	no	2.43	145.00	−0.33	0.58	0.92	0.92	0.03	0.99
Tüba-D/Z	yes	2.19	104.00	−1.16	0.11	0.95	0.95	0.03	0.80
Wikipedia	no	1.45	5.00	−6.81	0.01	0.81	0.81	0.04	0.54
Wikipedia	yes	1.44	6.00	−19.61	3.28×10^{-05}	0.90	0.90	0.03	0.70
ZEIT	no	2.17	6,472.00	−0.77	0.41	0.87	0.87	0.03	0.95
ZEIT	yes	2.04	7,123.00	0.00	0.93	0.97	0.97	0.02	1.00

(GerParCor) corpus[15] (Abrami *et al.* 2022) and SZ[16] (both without modal verbs).

[15] A corpus of historical German parliamentary protocols from three centuries, covering four countries and processed for NLP research in political communication.

[16] Süddeutsche Zeitung 1992–2014

For this reason, we determined the goodness-of-fit values for fitting the distributions to a power law. Results are collected in Table 2. The (adjusted) coefficient of determination was calculated by using the curve fitting toolbox `cftool` from MATLAB (The MathWorks, Inc. 2012). The Kolmogorow-Smirnow test was carried out by using the `igraph` library (Csárdi and Nepusz 2006). The results vary from weaker fits ($R^2 = 0.81$) to strong fits ($R^2 = 0.99$), reflecting the distribution tests from Table 2. Furthermore, we observe no p-value smaller than 0.05 for the Kolmogorow-Smirnow goodness-of-fit test (in which case a power law distribution hypothesis would have to be rejected). Hence, although there is some distributional heterogeneity in the verb frequencies, they are nonetheless all heavy-tailed.

The question then is which of these corpora to use as a reference for determining C2. This can be answered with the help of Table 3, which shows verb token overlap among several reference corpora.[17] The table shows coverage of lemmas of different corpora with respect to one another, weighted by the frequency of the lemmas. A coverage of >75 % is indicated by green cell color (max. ■), a coverage of <25 % by red color (max. ■). Relative coverage in between (i.e., 25–75 %) is colored gray (■). We treat the set of lemmas as a multiset, that is, the coverage of corpus A by corpus B for a lemma $v \in V$ with frequency x_v in A and y_v in B is given by $\sum_{v \in V} \min(x_v, y_v)/|A|$, where $|A|$ is the number of tokens in A of all lemmas in V. The number in brackets indicates the coverage of the lemmas, ignoring frequency. For a given row, the columns show how many of the lemma occurrences in that row corpus are covered by the column corpus. Note that for reference dictionaries such as `GermaNet` the number of occurrences per lemma is always 1 and token coverage is reduced to lemma coverage. It turns out that the largest freely available German corpus `COW` (Schäfer and Bildhauer 2012; Schäfer 2015), best covers all resources displayed in this heatmap. Thus we choose it as the reference for C2, selecting verbs according to their rank frequency distribution.

[17] Whenever needed, corpora were preprocessed with TextImager (Hemati *et al.* 2016), e.g., regarding POS tagging.

Table 3: Verb lemma frequency coverage of annotated verbs in `TGVCorp` with respect to German reference corpora. See Appendix B for version information

	COW	COW16b	DeReKo (1/16)	Die ZEIT	DTA	Gutenberg	Leipziger WS	Parlament	SZ	EU Bookshop	Textbooks	Wikipedia	GVSD	Duden	Wiktionary	GermaNet	BabelNet	E-VALBU	TüBa-D/Z	WebCAGe	deWaC	TTVC	TTVC*
COW	—	83,4 (2,5)	17,9 (4,6)	1,1 (4,9)	1,0 (4,4)	3,3 (9,0)	2,3 (6,0)	1,5 (3,3)	4,6 (8,0)	1,7 (1,7)	0,0 (0,7)	4,2 (6,3)	1,6 (7,3)	0,0 (3,1)	0,0 (2,3)	0,0 (1,8)	0,0 (0,6)	0,0 (0,1)	0,0 (0,0)	0,0 (0,2)	0,0 (0,0)	0,0 (0,3)	0,0 (1,8)
COW16b	100,0 (100,0)	—	21,5 (97,2)	1,3 (81,8)	1,2 (65,0)	3,9 (81,0)	2,8 (83,8)	1,8 (70,2)	5,5 (92,3)	2,0 (65,8)	0,0 (20,6)	5,0 (82,2)	2,0 (90,4)	0,0 (75,1)	0,0 (61,0)	0,0 (60,4)	0,0 (19,6)	0,0 (3,8)	0,0 (0,6)	0,0 (6,6)	0,0 (0,1)	0,0 (10,8)	0,0 (59,6)
DeReKo (1/16)	98,8 (92,1)	98,6 (48,5)	—	6,0 (55,9)	5,2 (43,6)	17,3 (63,0)	12,7 (59,7)	8,4 (43,5)	25,2 (76,5)	9,0 (32,8)	0,1 (10,7)	22,0 (60,7)	9,0 (71,6)	0,0 (49,6)	0,0 (38,3)	0,0 (33,9)	0,0 (10,4)	0,0 (1,9)	0,0 (0,3)	0,0 (3,3)	0,0 (0,1)	0,0 (5,4)	0,0 (33,4)
Die ZEIT	94,7 (44,7)	94,3 (18,6)	94,4 (25,5)	—	68,5 (26,6)	87,3 (34,0)	98,3 (37,8)	80,0 (28,5)	94,7 (37,1)	68,3 (14,0)	1,4 (5,5)	85,8 (30,6)	90,2 (32,9)	0,1 (20,6)	0,1 (16,3)	0,1 (15,0)	0,0 (4,7)	0,0 (0,9)	0,1 (0,1)	0,0 (1,5)	0,0 (0,0)	0,3 (2,5)	0,1 (14,8)
DTA	92,2 (46,8)	91,5 (17,5)	90,5 (23,6)	75,1 (31,5)	—	91,1 (46,1)	86,5 (30,3)	71,4 (23,5)	87,4 (32,5)	62,9 (14,1)	1,5 (5,7)	79,2 (31,4)	78,3 (31,6)	0,1 (22,6)	0,1 (17,1)	0,1 (15,4)	0,0 (5,1)	0,0 (1,0)	0,1 (0,2)	0,0 (1,8)	0,0 (0,0)	0,3 (2,8)	0,1 (15,2)
Gutenberg	95,0 (40,3)	94,1 (9,1)	91,7 (14,2)	29,2 (16,8)	27,7 (19,2)	—	50,5 (16,6)	34,0 (12,0)	77,1 (31,3)	28,8 (6,7)	0,5 (2,9)	56,1 (26,7)	38,4 (25,1)	0,0 (11,8)	0,0 (8,8)	0,0 (7,3)	0,0 (2,3)	0,0 (0,4)	0,0 (0,1)	0,0 (0,8)	0,0 (0,0)	0,1 (1,2)	0,0 (7,2)
Leipziger WS	94,9 (66,6)	94,7 (23,5)	94,7 (33,5)	46,4 (46,6)	37,2 (31,6)	71,3 (41,4)	—	56,1 (34,2)	94,5 (50,8)	47,6 (17,3)	0,7 (7,0)	77,2 (39,8)	60,0 (44,0)	0,1 (25,8)	0,0 (20,7)	0,0 (18,7)	0,0 (5,9)	0,0 (1,1)	0,0 (0,2)	0,0 (1,9)	0,0 (0,0)	0,1 (3,0)	0,0 (18,4)
Parlament	94,2 (63,7)	94,0 (34,0)	93,8 (42,3)	56,5 (60,7)	45,9 (42,3)	71,8 (51,7)	84,0 (59,2)	—	89,5 (58,0)	68,0 (28,1)	1,0 (11,3)	74,5 (49,0)	70,4 (52,8)	0,1 (35,6)	0,1 (28,7)	0,1 (28,2)	0,0 (9,4)	0,0 (1,9)	0,1 (0,3)	0,0 (3,2)	0,0 (0,1)	0,2 (5,2)	0,1 (27,8)
SZ	98,0 (51,7)	97,7 (14,9)	97,8 (24,8)	23,2 (26,3)	19,5 (19,5)	56,6 (45,1)	49,1 (29,3)	31,1 (19,3)	—	30,2 (10,4)	0,4 (4,7)	62,0 (40,1)	34,7 (38,9)	0,0 (17,3)	0,0 (13,3)	0,0 (11,3)	0,0 (3,5)	0,0 (0,6)	0,0 (0,1)	0,0 (1,1)	0,0 (0,0)	0,1 (1,8)	0,0 (11,2)
EU Bookshop	100,0 (100,0)	100,0 (100,0)	98,8 (99,9)	47,5 (93,4)	39,9 (79,4)	59,9 (90,5)	70,2 (93,9)	67,0 (88,0)	85,8 (97,4)	—	1,0 (31,1)	83,7 (95,0)	66,7 (97,1)	0,1 (88,9)	0,0 (75,6)	0,0 (79,3)	0,0 (27,9)	0,0 (5,8)	0,1 (0,9)	0,0 (10,0)	0,0 (0,2)	0,2 (16,3)	0,0 (78,3)
Textbooks	94,5 (68,7)	88,1 (47,7)	88,2 (49,5)	89,5 (56,2)	85,3 (48,7)	88,4 (59,8)	89,8 (58,2)	89,6 (54,2)	90,5 (67,2)	88,0 (47,3)	—	90,7 (64,8)	89,7 (65,1)	1,5 (48,3)	1,4 (45,4)	1,5 (47,1)	0,7 (23,4)	0,2 (7,7)	2,7 (1,3)	1,1 (11,9)	0,0 (0,2)	13,2 (19,0)	1,4 (46,6)
Wikipedia	97,7 (42,1)	97,3 (13,9)	93,0 (20,5)	22,9 (22,7)	19,3 (19,7)	44,8 (40,1)	43,7 (24,0)	28,2 (17,1)	67,5 (41,9)	32,1 (10,5)	0,4 (4,7)	—	34,0 (33,3)	0,0 (16,9)	0,0 (13,1)	0,0 (11,5)	0,0 (3,7)	0,0 (0,6)	0,0 (0,1)	0,0 (1,1)	0,0 (0,0)	0,1 (1,8)	0,0 (11,4)
GVSD	97,4 (42,9)	96,7 (13,3)	96,9 (21,1)	61,7 (21,3)	48,8 (17,3)	78,6 (33,0)	87,0 (23,1)	68,1 (16,0)	96,9 (35,4)	65,6 (9,4)	1,0 (4,1)	87,2 (29,0)	—	0,1 (15,2)	0,1 (11,7)	0,1 (10,2)	0,0 (3,2)	0,0 (0,6)	0,1 (0,1)	0,0 (1,0)	0,0 (0,0)	0,2 (1,6)	0,1 (10,0)
Duden	90,8 (90,8)	55,8 (55,8)	73,8 (73,8)	67,4 (67,4)	62,3 (62,3)	78,3 (78,3)	68,4 (68,4)	54,5 (54,5)	79,4 (79,4)	43,4 (43,4)	15,5 (15,5)	74,4 (74,4)	76,4 (76,4)	—	61,9 (61,9)	51,5 (51,5)	15,8 (15,8)	2,9 (2,9)	0,4 (0,4)	4,9 (4,9)	0,1 (0,1)	8,0 (8,0)	50,8 (50,8)
Wiktionary	89,8 (89,8)	59,6 (59,6)	75,1 (75,1)	70,1 (70,1)	62,1 (62,1)	76,6 (76,6)	72,4 (72,4)	57,8 (57,8)	80,7 (80,7)	48,6 (48,6)	19,2 (19,2)	75,9 (75,9)	77,8 (77,8)	81,5 (81,5)	—	57,6 (57,6)	20,0 (20,0)	3,8 (3,8)	0,6 (0,6)	6,5 (6,5)	0,1 (0,1)	10,3 (10,3)	56,7 (56,7)
GermaNet	95,3 (95,3)	80,3 (80,3)	90,3 (90,3)	87,9 (87,9)	76,2 (76,2)	86,9 (86,9)	88,6 (88,6)	77,2 (77,2)	93,4 (93,4)	69,3 (69,3)	27,0 (27,0)	90,9 (90,9)	91,9 (91,9)	92,2 (92,2)	78,4 (78,4)	—	26,5 (26,5)	5,2 (5,2)	0,8 (0,8)	8,9 (8,9)	0,1 (0,1)	14,5 (14,5)	98,6 (98,6)
BabelNet	69,3 (69,3)	61,4 (61,4)	65,3 (65,3)	65,2 (65,2)	59,9 (59,9)	64,1 (64,1)	65,7 (65,7)	60,6 (60,6)	68,0 (68,0)	57,5 (57,5)	31,7 (31,7)	68,0 (68,0)	67,2 (67,2)	66,6 (66,6)	64,1 (64,1)	62,6 (62,6)	—	10,2 (10,2)	1,4 (1,4)	15,0 (15,0)	0,3 (0,3)	20,3 (20,3)	61,9 (61,9)
E-VALBU	96,1 (96,1)	95,8 (95,8)	95,9 (95,9)	96,1 (96,1)	95,8 (95,8)	96,1 (96,1)	96,1 (96,1)	96,1 (96,1)	96,1 (96,1)	95,8 (95,8)	84,0 (84,0)	96,1 (96,1)	96,1 (96,1)	96,8 (96,8)	98,4 (98,4)	98,6 (98,6)	82,2 (82,2)	—	7,6 (7,6)	55,0 (55,0)	2,1 (2,1)	67,0 (67,0)	98,1 (98,1)
TüBa-D/Z	100,0 (100,0)	100,0 (100,0)	100,0 (100,0)	100,0 (100,0)	99,6 (100,0)	100,0 (100,0)	100,0 (100,0)	100,0 (100,0)	100,0 (100,0)	100,0 (100,0)	58,5 (93,9)	100,0 (100,0)	100,0 (100,0)	0,9 (98,8)	0,9 (98,8)	0,9 (98,8)	0,7 (76,8)	0,5 (52,4)	—	1,7 (62,2)	0,0 (0,0)	2,2 (17,1)	0,8 (93,9)
WebCAGe	99,9 (99,8)	99,7 (99,6)	99,9 (99,9)	99,8 (99,8)	98,9 (99,1)	99,5 (99,7)	99,9 (99,9)	99,2 (99,4)	100,0 (100,0)	98,3 (98,5)	75,4 (77,4)	99,9 (100,0)	99,9 (99,9)	32,3 (99,2)	32,6 (100,0)	32,6 (100,0)	23,3 (71,5)	10,7 (32,7)	5,4 (5,4)	—	0,4 (1,3)	68,5 (67,0)	32,5 (99,8)
deWaC	100,0 (100,0)	100,0 (100,0)	100,0 (100,0)	100,0 (100,0)	100,0 (100,0)	100,0 (100,0)	100,0 (100,0)	100,0 (100,0)	100,0 (100,0)	100,0 (100,0)	100,0 (100,0)	100,0 (100,0)	100,0 (100,0)	93,3 (93,3)	100,0 (100,0)	100,0 (100,0)	93,3 (93,3)	80,0 (80,0)	0,0 (0,0)	80,0 (80,0)	—	100,0 (100,0)	100,0 (100,0)
TTVC	100,0 (100,0)	99,9 (99,6)	100,0 (99,9)	99,8 (99,6)	98,1 (96,7)	99,7 (99,1)	99,8 (99,7)	99,7 (99,0)	100,0 (100,0)	99,6 (98,4)	67,7 (75,2)	100,0 (99,8)	100,0 (100,0)	4,0 (99,2)	3,9 (96,6)	4,0 (99,7)	2,4 (59,5)	1,0 (24,4)	0,5 (0,9)	5,1 (41,0)	0,0 (1,0)	—	4,0 (99,7)
TTVC*	95,3 (95,3)	80,3 (80,3)	90,4 (90,4)	87,9 (87,9)	76,0 (76,0)	86,9 (86,9)	88,6 (88,6)	77,3 (77,3)	93,4 (93,4)	69,4 (69,4)	27,1 (27,1)	90,8 (90,8)	91,9 (91,9)	92,2 (92,2)	78,3 (78,3)	100,0 (100,0)	26,6 (26,6)	5,2 (5,2)	0,7 (0,7)	9,0 (9,0)	0,1 (0,1)	14,7 (14,7)	—

`COW` is a web-crawled corpus containing 807,782,354 sentences. Due to its automatic pre-processing, it contains a considerable number of lemmatization and POS tagging errors. This explains the unusually high number of verb lemmas found in `COW` (see Table 4). To fix these errors, we apply four heuristics to the selection of verb lemmas output

Table 4: COW-based statistics of verb lemmas and their tokens

	Plain	Filtered
# verb lemmas	368,677	41,316
# verb tokens	939,732,595	880,670,918
% verb hapax legomena	50 %	35 %

by the lemmatization of COW:

1. The lemma candidate must be in present infinitive and thus end in *-n*.
2. It has to consist of at least 2 characters.
3. It must be in lower case.
4. Modal and auxiliary verbs are excluded.

Using these heuristics, 88 % of verb lemmas in COW are removed, but only 6 % of verb tokens (see Table 4).

The frequencies of the remaining verb lemmas are plotted in Figure 4 as a cumulative rank frequency distribution.

We observe that a small number of verbs covers a large number of verb tokens. More specifically, the 945 most frequent verbs cover 80 % of COW's verb tokens. A corpus disambiguating a sufficient number of examples for each of these lemmas would thus satisfy C2 and C3.

However, not all of these verbs are ambiguous, and some have already been annotated. And while we prioritize C2 over C1, we would

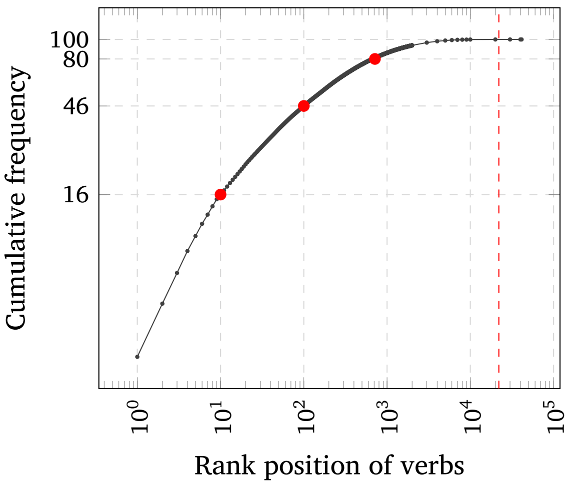

Figure 4: The cumulative distribution of the token frequencies of the verbs in the COW corpus. The 945 most common verb lemmas cover 80 % of the verb tokens in COW

still like to satisfy C1 to the largest degree allowed by our resources. Thus, we select verbs to disambiguate, in descending order of their frequency according to the following criteria:

1. The lemma candidate has at least two senses in GermaNet.
2. It is not already annotated in TüBa-D/Z.
3. It is not a modal verb and not an auxiliary verb.

The result is a set of 1,560 ambiguous verbs with a COW coverage of 78%.

The third condition, C3, concerns the selection of a corpus to be sense-annotated based on our reference set of verbs. Here we started from TüBa-D/Z, a German newspaper corpus, which is annotated semi-automatically at several linguistic levels (Telljohann *et al.* 2012). Parts of TüBa-D/Z are also already sense-annotated. We thus "filled out" an existing corpora instead of starting from scratch.

We also added sentences from other resources to fill in gaps in lemma coverage. More specifically, we included sentences from E-VALBU and the SALSA 2.0 Corpus (Burchardt *et al.* 2006) that are linked to semantic annotations in Berkeley FrameNet (Ruppenhofer *et al.* 2016) format. In this way, future work will gain access to relations between verb-related frames and the verb senses we annotate.

TGVCorp is thus generated as a union of three corpora: TüBa-D/Z, Salsa and E-VALBU – see Table 5 for the corpus statistics. Multiple

Table 5: TGVCorp breakdown

Sources	TüBa-D/Z, Salsa, E-VALBU
Total # of sentences	31,650
Total # of annotated word lemmas	1,560
Total # of tagged word tokens	39,241
Frequency range (occurrences/lemma)	1–261
Average frequency (occurrences/lemma)	25
Polysemy range in GermaNet (senses in GermaNet/lemma)	1-26
Average polysemy in GermaNet (senses in GermaNet/lemma)	3.27
Polysemy range of occurring words (occurring senses/lemma)	1–18
Average occurring polysemy of lemmas (occurring senses/lemma)	2.34
Average occurring polysemy of words (occurring senses/word)	3.77

Table 6: Verb lemmas and tokens in various corpora and their coverage with respect to COW

	TüBa-D/Z	WebCAGe	deWaC	TGVCorp
# verb lemmas	82	959	15	1,560
# verb tokens	9,290	3,186	608	39,241
average frequency	113	3	41	25
average polysemy	2.5	3.7	7.9	2.34
COW coverage (lemma-based)	6.2 %	66.4 %	6.4 %	78.02 %

other corpora are also annotated with GermaNet senses. These are the sense-annotated sections of TüBa-D/Z itself, WebCAGe (Henrich *et al.* 2012) and deWaC (Raileanu *et al.* 2002). Table 6 compares our target corpus to these, demonstrating that only TGVCorp offers a high COW coverage with a large number of lemmas and at the same time a sufficiently high number of example sentences per lemma. This closes the gap left by its competitors.

2.3 *Annotating TGVCorp*

We developed VerbSenseAnnotator[18] to disambiguate TGVCorp at the sense level, and conducted this annotation in two stages. As in related approaches (Henrich 2015; Kilgarriff 1998; Fellbaum *et al.* 2001; Saito *et al.* 2002; Passonneau *et al.* 2012), VerbSenseAnnotator shows sentences in which the occurrences of target verbs are to be disambiguated on the level of lemmas. Sentences are preprocessed by TextImager to capture lemma, POS, and dependency structure information, and to present verbs with corresponding senses from GermaNet. For each target sense of each target verb, the corresponding synonyms, hyponyms, and hypernyms are listed, as well as sense descriptions and example sentences where available, so that annotators can disambiguate more easily. Ideally, exactly one meaning should be selected for each occurrence of each target verb, but when in doubt, more than one is possible. Occurrences of target verbs for which the annotator cannot find a sense in VerbSenseAnnotator can be marked. If multiple senses or no appropriate sense are selected for

[18] https://textimager.hucompute.org/VSD/

a verb occurrence, this indicates that the verb's sense definitions are problematic. Commonly, this was a problem with very fine-grained sense definitions, which are indistinguishable for annotators that have to rely on short sense descriptions and example sentences. Other problematic cases were metaphorical usages or hierarchical senses, such as *laufen* in the sense of movement on foot in general, 'to move' vs. *laufen* in the sense of a fast, running movement, 'to run'. Following the approach of Palmer *et al.* (2007), these senses with very low inter-annotator agreement were manually reviewed and merged if required. A list of all senses merged in this fashion is shown in Appendix A.

To evaluate the quality of verb-sense annotation, each target sentence was annotated independently by several annotators in two stages. The first stage comprised the bulk of annotation work, in which a total of 19 annotators participated, including undergraduates, graduate students, doctoral students, and postdoctoral fellows in computer science and computational linguistics. The second stage involved 7 annotators. The procedure was the same for both stages, with two exceptions. The first difference was in the choices annotators had. In the first stage, they could select multiple senses for a single instance. This was not possible in the second stage, where the annotators had to select a single sense. In addition, they could mark sentences that were ambiguous or incomprehensible due to a lack of context. The second difference relates to the selection of the gold label in situations where annotators disagreed. To address this issue during the first stage, we developed a method that compares the inter-annotator agreement between each annotator and the original `TüBa-D/Z` annotation to prefer the annotator with the highest agreement.[19] Therefore, in order to be consistent with the `TüBa-D/Z` interpretations, we decided to prefer the annotator who agreed in the majority of cases. Given this approach, we do not know with certainty the reliability of our annotations. However, by selecting the annotator this way, and manually checking senses with low agreement between annotators, we guarantee at least a strong orientation towards `TüBa-D/Z`, even if this is certainly not the only authoritative resource. In the second

[19] This approach is motivated by the fact that annotators often agreed on the distinction of senses, but not on their interpretations (i.e. they agreed that a verb has n different senses, but not on what these senses are).

stage, each disagreement was checked and a gold label was manually selected. During this process, we discovered many senses with very low inter-annotator agreement.

3 A SIMPLE METHOD FOR AUTOMATIC VSD

Using `TGVCorp`, we train a supervised system for VSD by elaborating the approach of Hemati (2020). We follow approaches that use human-annotated training data to learn to assign senses from predefined lexical resources to ambiguous lexical text occurrences (Hemati 2020; Henrich 2015; Papandrea *et al.* 2017; Luo *et al.* 2018; Peters *et al.* 2018; Melamud *et al.* 2016; Uslu *et al.* 2018). One of the most elaborate early approaches to WSD in German is that of Henrich (2015), who uses `GermaNet` as a sense inventory to train supervised and knowledge-based systems. A problem faced by these and related approaches is that the underlying annotated corpora usually only contain a few lemmas or have very few annotated instances per lemma. Although `TGVCorp` is one step ahead in filling this gap, sense compression must be performed for tackling the latter bottleneck, as will be explained below. To perform VSD, we train `TTvSense`, a supervised classifier based on `fastSense` (Uslu *et al.* 2018), which in turn is based on `fastText` (Joulin *et al.* 2017; Bojanowski *et al.* 2016). `TTvSense` is a feed-forward network that includes sense compression according to Vial *et al.* 2019. We compare `TTvSense` with `EWISER` (Bevilacqua and Navigli 2020), a state-of-the-art approach to WSD, and show how to circumvent the data bottleneck problem in VSD using language models. To compare `EWISER` and `TTvSense`, we reproduce the method of Henrich (2015) using the *TüBa-D/Z Gold Standard for Supervised WSD* corpus, focusing on verbs (see Table 7 for its statistics). We split this data to maintain the following ratio per lemma (Henrich 2015; Botev and Ridder 2017; Witten *et al.* 2011): 60 % for training, 20 % for validation and 20 % for testing. For methods that do not require validation sets, this part was omitted to keep training and test sets comparable.

	GermaNet	WordNet Subset
Total # of annotated word lemmas	82	68
Total # of tagged word tokens	9,290	5,765
Frequency range (occurrences/lemma)	1–822	2–280
Average frequency (occurrences/lemma)	113.3	84.8
Polysemy range in GermaNet (senses in GermaNet/lemma)	1–14	—
Average polysemy in GermaNet (senses in GermaNet/lemma)	2.9	—
Polysemy range of occurring words (occurring senses/lemma)	1–9	1–4
Average occurring polysemy of lemmas (occurring senses/lemma)	2.45	1.74
Average occurring polysemy of words (occurring senses/word)	3.16	1.97

Table 7: TüBa-D/Z sense annotation subset for supervised WSD Henrich (2015), verbs only

3.1 *TTvSense*

TTvSense represents a word as a sum of *n*-gram vectors, where the word itself is one of the *n*-grams initialized from previously trained word embeddings. These word representations are fine-tuned during the training. A sentence is encoded by averaging the word representations for all words contained in it. This sentence encoding forms the input for a single fully connected layer, which produces output scores for all senses of all lemmas. Finally the output senses are filtered to remove all which do not belong to the current target lemma. The list of valid senses for the target lemma is obtained from the training corpus as part of the training process. To extend this model, we performed sense compression on GermaNet according to Vial *et al.* (2019). In this process, all senses for a given lemma are removed from their original synset and reassigned to be just below the last common ancestor in the hyperonymy hierarchy. The procedure is explained in detail in Section 3.5.

TTvSense uses information about the target word only after the scores have been calculated. Furthermore, it does not process position or word order information. This is a problem when a sentence manifests several disambiguation-relevant contexts due to its clause structure. For example, the first half of the sentence *Er lief ins Büro und machte den Rechner an.* 'He ran into the office and turned on the computer' indicates a motion sense of *lief* 'ran' that is not matched by the second half which might indicate another sense of that verb (*Der Computer lief* 'The computer was running'). Without position and target information, the classifier cannot distinguish these contexts, thus accuracy suffers. To deal with this problem, we split sentences along conjunctions and punctuation marks and processed only the segment that contained the target word.

3.2 EWISER

EWISER (Bevilacqua and Navigli 2020) sums the last four layers of BERT (Devlin *et al.* 2018) and normalizes them to a context vector H_0, which is fed into a two-layer fully-connected network to produce output values Z:

$$H_1 = \text{swish}(H_0 W + b)$$
$$Z = H_1 O$$

The first layer is a traditional, fully connected layer with a Swish (Ramachandran *et al.* 2017) activation function and is used to re-encode H_0 from BERT to have the same dimensionality as the pretrained sense embeddings O. The weights of the second layer are initialized with O to produce logits for each sense in the inventory. Finally, these logits are modified based on the graph structure of the given WordNet to produce "structured logits". For a given synset s with logit z_s and n_s related synsets z_i a new structured logit q_s is computed by adding the logits of all related synsets: $q_s = z_s + \sum_i z_i / n_s$. This takes the form of a residual layer where the weights are initialized by an adjacency matrix A in which the entries of each row sum up to 1:

$$Q = ZA^T + Z$$

During training the underlying BERT model is kept frozen while the weights A are fine-tuned. The sense embeddings follow a freeze-and-thaw training scheme where they are kept frozen for the first n epochs before being unfrozen and fine-tuned during the remaining epochs.

Experimentation 3.3

We conducted a series of experiments with German and English data and performed comparisons on English verbs from Navigli *et al.* (2017). Since EWISER requires WordNet or BabelNet (Navigli and Ponzetto 2012) labels, we experimented on the subset of TüBa-D/Z for which there are mappings from GermaNet to WordNet. The experiments are repeated for TGVCorp. The GermaNet senses in texts were mapped to WordNet using EuroWordNet's (Vossen 1998) Inter-Lingual Index. This mapping is not complete and does not ensure a one-to-one relation, so we removed all instances for which there is no mapping. In cases with multiple relevant labels we only considered the first one provided by the mapping, discarding any others. The resulting WordNet subset is considerably smaller than the original corpus, with fewer examples per lemma and significantly lower polysemy. See Table 7 above for a comparison. The mapping from WordNet to BabelNet is done in EWISER itself, but requires updating multiple dictionary files. EWISER operates only on a subset of the BabelNet-WordNet mapping that matches entries in these files. These dictionaries limit the lemmas and the labels for each lemma which the system will produce. The pretrained checkpoint comes with multilingual dictionaries based on SemEval tasks. Testing the pretrained checkpoint on TüBa-D/Z, EWISER achieves only 53 % with these dictionaries, 69 % if we update the dictionaries to include the labels in the test set, and 78 % if we additionally remove all labels which do not occur in the test set. Accurate dictionaries are critical to achieving good results in practice.

For EWISER we tested three different models. One was trained only on the training section of TüBa-D/Z and one on both the TüBa-D/Z training section and the WordNet Glosses and Examples corpora. Due to time and computational restraints we chose the best performing hyperparameters from Bevilacqua and Navigli 2020 for training. We also tested the pretrained multilingual model provided by Bevilacqua and Navigli 2020.

For TTvSense we examine the impact of the sentence fragmentation and sense compression over the baseline classifier. Hyperparameters were optimized on the validation set of TüBa-D/Z using Tree-structured Parzen Estimator (TPE) (Bergstra and Bengio 2012)

Table 8: Hyperparameters of training TTvSense

Epochs	40
Initial learning rate	0.2
Hidden dim	100
Window size	3
Loss	softmax
Pretrained embeddings	Mikolov embeddings computed by means of the Süddeutsche Zeitung corpus (1992–2014)

Table 9: EWISER hyperparameters. Training takes place in two stages where the sense embeddings are kept frozen during the first stage and fine-tuned during the second

Epochs first stage	50
Epochs second stage	20
Initial learning rate first stage	10^{-4}
Initial learning rate second stage	10^{-5}
BERT model	bert base multilingual cased
Hidden dim	512
Sense embeddings	SensEmBERT + LMMS
Structured logits	hypernyms, derivational, verb group, similarity

as implemented by hyperopt (Bergstra *et al.* 2013). The hyperparameters for TTvSense and EWISER are shown in Tables 8 and 9.

Both EWISER and our classifier use dictionaries to limit output senses for each lemma. These essentially form another hyperparameter. For our experiments, these dictionaries were computed before the training process, excluding all senses that did not appear in the training corpora. Results are shown in Table 10. We outperform EWISER in all German tests, but perform significantly worse on the English corpora. However, our fastText-based classifier trains and evaluates much faster despite not using a GPU. Training on our machine with an AMD FX-8350 and GTX 1070 on TüBa-D/Z only, our classifier took about 4 minutes on the CPU, while EWISER took about 30 minutes despite also using the GPU. This is repeated during evaluation, with TTvSense evaluating the entire test set in less than one second, compared to about 45 seconds for EWISER. In times of problematic CO_2 emissions by NLP (Bender *et al.* 2021), this is a relevant finding.

Table 10: VSD results on TüBa-D/Z sense annotation subset for supervised WSD. For EWISER the subscripts indicate the source/training corpora. For TTvSense the subscripts indicate sentence fragmentation (sf) and sense compression (sc)

System	Base Corpus	Micro F1 score
Most frequent sense		71.75
Context2Vec		76.04
Best of Henrich (2015)	TüBa-D/Z with	80.74
Flair	GermaNet Labels	83.13
TTvSense		80.93 ± 0.39
$\text{TTvSense}_{\text{sf}}$		87.39 ± 0.81
$\text{TTvSense}_{\text{sf+sc}}$		89.14
Most frequent sense		87.24
$\text{EWISER}_{\text{tueba}}$		88.43 ± 0.63
$\text{EWISER}_{\text{tueba + WNGC}}$		90.94 ± 0.37
$\text{EWISER}_{\text{multilingual pretrained}}$	WordNet subset	78.13
TTvSense	of TüBa-D/Z	88.79 ± 0.14
$\text{TTvSense}_{\text{sf}}$		93.13 ± 0.85
$\text{TTvSense}_{\text{sf+sc}}$		93.52 ± 0.29

Table 11: VSD results on SemCor and SENSEVAL

System	Micro F1 score
$\text{TTvSense}_{\text{sc}}$	43.91
$\text{TTvSense}_{\text{sf+sc}}$	46.94
$\text{TTvSense}_{\text{sf+sc on SemCor only}}$	55.67
EWISER	69.40

We also ran comparisons on English verbs using SemCor (Miller *et al.* 1994; Navigli *et al.* 2017) as training data and the concatenation of English WSD SENSEVAL tasks as test data. We tried to determine generalization errors of our classifier by also training and testing on SemCor verbs only, using the same splitting as for TüBa-D/Z. The results are shown in Table 11 and discussed below. We then tested TTvSense on TGVCorp. The results are shown in Table 12.

Table 12:
VSD results on TGVCorp

System	Micro F1 score
TTvSense	63.2 ± 0.4
$\texttt{TTvSense}_{\mathrm{sf}}$	69.8 ± 0.1
$\texttt{TTvSense}_{\mathrm{sf+sc}}$	65.5 ± 0.2

3.4 Discussion

TTvSense outperforms EWISER on both TüBa-D/Z and TGVCorp, even when taking the WordNet Gloss Corpus as additional training data for EWISER. Interestingly, this result is not repeated in English, where our classifier performs much worse. We think that this could be due to two main factors: In the German experiments, we obtained training and test data from TüBa-D/Z based on a single newspaper. SemCor, on the other hand, is based on the Brown Corpus, which contains various newspapers, books, and other sources. SENSEVAL comes mainly from articles in the Washington Post. The improvement when testing and training only on SemCor might indicate that our classifier overfits on the training data and generalizes worse than EWISER. At the same time, the increase is too small to explain the whole performance gap between German and English. The second effect is language-specific. Our classifier uses averaged word form embeddings as the context vector. This approach might work better for German than for English, since the morphology in German is more extensive, reducing the importance of positional information. However, positional information is still relevant due to sentence-internal contexts belonging to different verbs. TTvSense reflects this through its simple sentence segmentation algorithm, which performs worse on English data due to different punctuation rules. The sentence segmentation reduces error rates by around a third in all German tests, but only by about 5 % in English tests. In any case, TTvSense, which we trained to disambiguate 1,560 German high-relevance verbs (see above), is a classifier for VSD that represents a new state of the art for German verbs.

3.5 An experiment in sense compression

Supervised systems rely on annotated training data and cannot directly disambiguate senses which they have not seen. Sense compression is a method of extending the coverage of existing annotations by exploiting the hyperonymy structure. For this, we adapt the algorithm of Vial *et al.* (2019) for GermaNet. We consider GermaNet as a graph $G = (V, E)$, where the set of vertices consists of synsets S and senses (GermaNet LexUnits) L with $V = S \cup L$ and

(1) $$\begin{aligned} E = \big\{ & (u, v) : (u,\, v \in S, u \text{ is hypernym of } v) \\ & \vee (v \in S, u \in L, u \text{ is member sense of } v) \big\} \end{aligned}$$

G is directed and acyclic, where each vertex in L is a leaf node and only vertices in L are leaves. Using G, a graph variant G' is created as follows: pick a lemma v and select the set of vertices

(2) $$L_v = \{ l \in L : l \text{ belongs to lemma } v \}$$

which corresponds to the set of senses which belong to lemma v. Then mark all vertices which are ancestors of more than one $l \in L_v$. Finally, add an edge for every $l \in L_v$ between l and the child of its first marked ancestor and remove the edge between l and its original synset. This ensures that only one sense per lemma per synset exists without violating the hyperonymy structure of the graph. Repeat this process for every lemma. Finally, remove any synsets that do not have any attached senses.

For a given sense $l \in L$ the new label is determined by its direct parent. Given a target lemma and a compressed synset s one can convert back to the original sense label by searching the direct children of s for the one sense belonging to the target lemma. This procedure – see Algorithm 1 – guarantees that each synset contains only one sense per lemma, provided that the original graph fulfills the same condition. The statistics for Algorithm 1 operating on GermaNet are listed in Table 13. To quantify the effectiveness of sense compression, we performed an out-of-sample test by removing lemmas from the dataset such that there were at least 10 training instances left for each of the compressed synsets. The instances belonging to the removed lemmas formed the test set. Note that synsets can have less than 10 training instances, in which case the associated lemmas are not taken into account for removal. The results for this test are shown in Table 14.

This out-of-sample test shows that we achieve about 60 % F1 score on TGVCorp (ca. 70 % on TüBa-D/Z) from scratch with the compression algorithm – the alternative, of course, would be 0 %.

Algorithm 1: Algorithm for sense compression

```
for each verb v do
    /* Mark descendants of more than one sense */
    for each vertex l in L_v do
        while l is not null do
            if l.mark is not 'unmarked' then
                l.mark = 'conflict';
            else
                l.mark = 'visited';
            end
            l = parent of l;
        end
    end
    /* Reattach senses */
    for each vertex l in L_v do
        current = l;
        while mark of parent of current is not 'conflict' do
            current = parent of current;
        end
        Remove edge between l and parent of l;
        Add edge between l and current;
    end
end
/* Cleanup of empty synsets */
for vertex v in S do
    if v has no children in L then
        Reattach children of v to parent of v;
        Remove v from graph;
    end
end
```

Table 13: Results of compressing GermaNet

	Pre-compression	Post-compression
# Synsets	14,179	1,633
Average # senses per synset	1.29	11.89
Average depth of senses	6.71	2.85
Highest depth	16	14

	TGVCorp	TüBa-D/Z
F1 Score	60.62 ± 0.69	69.53 ± 0.18
Size of train set	≈ 18700	≈ 6000
Size of test set	≈ 17500	≈ 3100
# Lemmas removed	803	37/38

Table 14: Results for the out-of-sample tests using the sense compression algorithm

3.6 *Trying to leverage language models*

WSD is challenged by the data bottleneck problem (Navigli 2009). We attempt to address this problem beyond costly annotation by using language models (Devlin *et al.* 2018) that can be fine-tuned for downstream tasks (Zhou and Srikumar 2022) – here language generation (Rothe *et al.* 2020). That is, we use BERT (Devlin *et al.* 2018) to extend TGVCorp by generating new sentences starting from manually annotated ones. Following Ravfogel *et al.* (2020), we iteratively mask and replace words in sentences from left to right by sampling from the top k suggestions provided by BERT. Unlike Ravfogel *et al.* (2020), we do not only sample content words like nouns. German is less analytical than English, so substituting nouns alone easily leads to ungrammatical sentences due to agreement errors. We address this issue by processing sentences in two passes. In the first pass, nouns, adjectives, substitution pronouns, and adverbial adjectives are substituted; in the second pass, all other words are processed, leaving annotated verbs and punctuation untouched. Note that we do not try to maintain the POS of the source word, nor the original number of BERT tokens. For words consisting of multiple WordPiece tokens (Wu *et al.* 2016), we mask all tokens and replace them from left to right. To minimize morphological inconsistencies, however, only the first of them is sampled using BERT and then the top suggestions are selected for the remaining tokens (dependent selection). For example, after replacing the first token in "Schaff ##ner" with "Kell [MASK]", the only viable option for "##ner" is identity substitutions; if this were excluded and one were to sample independently from the top k BERT suggestions, the result would likely be a non-word. The whole procedure serves to ensure both semantic variability and a certain degree of grammatical correctness. Table 15 exemplifies our procedure.

Table 15: Left: Source sentences in which words to be replaced are in italics. Right: sentence candidate in which the italicized word is predicted by BERT for the masked word in the source sentence

Source sentence	Generated sentence candidate
Der *Schaffner* läuft zum *Bahnhof*.	Der *junge Mann* läuft zum *Flughafen*. Der *Bursche* läuft zum *Metzger*. Der *Fünfjährige* läuft durchs *Tor*.
Die *Diskussion* hat mein *Denken* zu diesem *Thema* verändert.	Die *Diagnose* hat mein *Vertrauen* zu dem *Institut* verändert. Die *Vergangenheit* hat meine *Einstellung* zu dem *Job* verändert. Die *Debatte* hat mein *Fazit* zu meinem *Amt* verändert.
Das *Gerät* läuft *einwandfrei*.	Das *Program* läuft *jetzt bis 2020*. Das *Geschäft* läuft *im Moment gut*. Das *Haus* läuft *immer noch leer*.

Table 16:
F1 scores when training our classifier with additional sentences from BERT. Baseline score is 87.3 %

	k	3	30	100
	1	86.3	86.4	86.0
n	3	85.9	85.7	85.4
	10	—	84.1	83.9

We evaluate this approach of generating new, similar sentences from annotated seed sentences, by extending TüBa-D/Z using this method and training TTvSense on the new training data. We have two new hyperparameters in this approach: (1) the number of new sentences n for each seed sentence and (2) the depth k to which we sample content words. Only sentences from the training subset were selected as seed sentences. We trained with sentence fragmentation but without sense compression. The results are shown in Table 16.

It is obvious that forming new sentences in this way did not improve the results. The reason could be that our sentence generator interpolated only in the range of sentence patterns already observed in the training corpus, introducing errors that made training more difficult. While this is disappointing in light of increasingly better and more diverse text generators, it points to a general problem of poor extrapolation capabilities of such approaches, which requires far more research to overcome. Although scores did not improve they also did not meaningfully degrade even with deep sampling. This suggests that this method could be used to create "look-alike" corpora.

Optimising `TTvSense` for VSD on `TGVCorp` 3.7

This section explains how `TTvSense` was optimized for `TGVCorp`. Since it is a sequence classifier that does not receive information about the target lemma, `TTvSense` has difficulties with longer sentences. To improve it, the aforementioned sentence segmenter was used in both training and testing. Table 17 shows that it improves VSD significantly.

	`TüBa-D/Z`	`TGVCorp`
w/o splitting	78.97 %	62.07 %
with splitting	86.16 %	71.38 %

Table 17:
Micro-F1 scores of `TTvSense` for VSD with and without sentence splitting

`TTvSense`, which is based on `fastSense`, has several parameters that must be learned based on the training data. This process of fitting model parameters to existing data is called *model training*. Another class of parameters, called hyperparameters, cannot be learned directly from the training process. Hyperparameters are variables that control the training process itself. They must be set beforehand and are configuration variables of the training process that are kept constant during training. They define higher-level concepts for the model, such as complexity, convergence rate, or penalty (Bergstra and Bengio 2012). We perform hyperparameter optimization to find optimal hyperparameter configurations for `TTvSense` on `TGVCorp` that maximize the prediction accuracy. For this task, we use TPE (Bergstra and Bengio 2012) implemented by `hyperopt` (Bergstra *et al.* 2013). Table 18 shows the parameter space of hyperparameter optimization. Figure 5 shows the results of each trial during the optimization process. The difference between the best and worst performer is 23 %. This shows that optimizing the hyperparameters can be crucial.

Table 18: Parameter space of TTvSense used in our experiments. The column *Possible Values* describes the range of values of the parameters. The parameter setting with the best value is highlighted in bold

Parameter	**Possible Values**
epoch	[5,10,...,**40**,...,250]
wordNgram	[1,2,...,**10**]
minCount	[**1**,2,3]
learning rate	[0.1,...,**0.2**,...,1)]
loss	[**softmax**,hs,ns]
pretrainedVectors	[**true**,false]

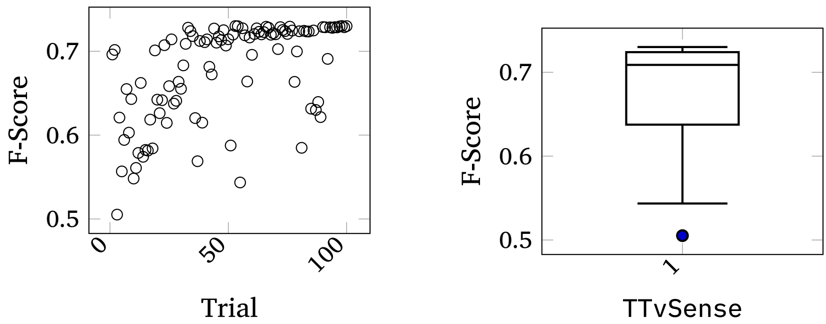

Figure 5: The figure shows the results of optimizing TTvSense on TGVCorp by means of TPE. The scatter plot on the left side shows the results of each trial. The boxplot shows in which area the results are located and how they are distributed over this area. The difference between the best and the worst performing setting is 23 %

4 CONCLUSION

In this paper, we have (further) developed an essentially three-part pipeline for VSD in German (1) starting from the constraint-based selection of a part of a sense inventory (i.e. GermaNet) via (2) the annotation of a sense-disambiguated corpus (TGVCorp) to (3) a classifier (TTvSense) trained on it. We also optimized our classifier in three ways: (A) in terms of compressing the selected sense inventory, (B) in terms of obtaining additional training sentences, and

(C) – quasi-traditionally – in terms of hyperparameter optimization. (A) was used to obtain training examples by transfer for senses for which there are not enough annotations in the training corpus. (B) was used to extend our training corpus by generating new sentences. While (A) directly addresses the data bottleneck problem in WSD (Navigli 2009), this does not necessarily apply to (B). The reason for this is probably that sentence generation as we have implemented it only intensifies existing imbalances in the training data (virtually by interpolating along sufficiently confirmed sentence patterns): sentence generation based on our implementation is not creative enough, so to speak. Another outcome of our work is that we disambiguated the occurrences of 1,560 verbs from `GermaNet` in a corpus based on `TüBa-D/Z` (see Table 5). As a result, we currently have the largest corpus-based sense-disambiguated set of verbs, for which we simultaneously provide a classifier that outperforms the `BERT`-based `EWISER` system in German. This is worth highlighting in two respects: on the one hand, we show a potential for energy saving by relying on a simpler ML architecture to support green NLP (cf. Bender *et al.* 2021). On the other hand, we extend the list of approaches that do not rely on large transformer-based architectures, but instead on simpler resources for solving NLP tasks with comparable quality (cf. Henlein and Mehler 2022, for similar findings).

So far, so traditional our approach. But what about resources that have access to large portions of the web to train the largest possible language models currently available? Don't these methods make NLP efforts like the one shown here seem anachronistic by potentially leveraging access to every online dictionary, every online text that can be linked to it, and every NLP resource that can be used to enhance the database? More precisely, why not just use a large language model such as `ChatGPT` (OpenAI 2023) as a ready-made tool for NLP including VSD? Why all the effort and tiny technical details when it is so much easier with a tool that seems to have direct access to an all-encompassing resource suitable for almost any NLP task? And indeed, `ChatGPT` is apparently a ready-made tool also for German VSD. See Figure 6 (left) for a chat in which we embedded `Wiktionary`'s sense inventory for the verb *abtragen* in the sense of *abbauen* 'remove' into a question to `ChatGPT` that

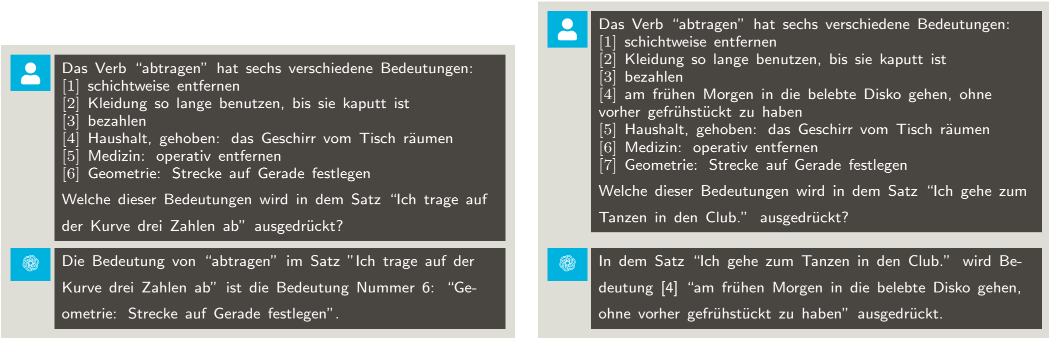

Figure 6: VSD with ChatGPT 3.5 using the Wiktionary entry for the verb *abtragen* 'to dismantle'. We have added an additional fake sense on the right (namely sense [4]), demonstrating that ChatGPT hallucinates (download Wiktionary data/ChatGPT: January 27, 2023 – graphically customized)

answers correctly. One might now assume, and the current discussion suggests, that `ChatGPT` solves many of the good old computational linguistic tasks for which a large community of researchers has developed so much in the past. Indeed, this could be a realistic scenario if `ChatGPT` were completely open so that one could reconstruct its responses algorithmically, extend the underlying algorithm as needed, or modify its training resources to adapt it for further research. This apparent gap leaves a third scenario: using `ChatGPT` to generate training corpora with which to train simple classifiers such as the one presented here, to obtain systems that are at least algorithmically open and that the scientific community can independently develop and adapt for its purposes. Research based on machine reading comprehension (Wang *et al.* 2022) aims in such a direction: it could help public research benefit from the increasingly powerful language models that have themselves benefited from decades of work by a wide range of researchers. In terms of lexical resources, such an open NLP would follow the third and the fifth of the seven theses of Storrer (2001, p. 63, 65) on digital dictionaries: these resources should be transparent (as well as reconstructable or reproducible) and comprehensible for their users, but also expandable according to their own scientific goals. Along this line of thinking, we could add an eighth thesis, namely that NLP resources should be algorithmically controllable and algorithmically extensible by their users. Last but not least, we return to Figure 6: on the right side, one can see almostw the same chat, except that we have inserted a "nonsense" sense (number 4), which is "correctly" recognized by `ChatGPT` for an appropriately phrased example sentence without any occurrence of the verb *abtragen*. Such a scenario – which exposes certain capabilities of `ChatGPT` as an illusion in the minds of its users – brings us back to Section 1 and the question of sense identification: If we believe in the existence, identifiability, and separability of, e.g., word senses (unlike, e.g., Kilgarriff 1997), this task seems to remain a human one, unless we trust the validity of cluster algorithms (or related approaches) operating on, say, vector representations of words (see Schütze 1998 for a seminal work in this regard) to solve this task on a human level. According to this reading, interpretation – and thus, for instance, the determination of relevant word senses – remains a task that cannot yet be automated given the state-of-the-art in ML, not even by resorting to the huge amount of digitized data.

APPENDICES

TABLE OF MERGED SENSES A

The following table shows merged senses, where merging follows one of these decision criteria (C.):

- Senses not distinguishable
- Circular Senses
- Senses/distinctions are missing
- Obsolete or dialectical meanings
- Metaphor

LexIds	maps to	lemma	C.
78225	76100	ablehnen	🟦
79173	78279	ablehnen	🟥
78263	78279	ablehnen	🟥
83482	83480	abschließen	🟦
144567	144566	abspielen	🟦
75468	75463	abstimmen	🟦
77711	74980	agieren	🟥
75668	74980	agieren	🟥
79573	74040	anbieten	🟥
75755	74040	anbieten	🟥
76330	83407	anfangen	🟦
83272	78924	anführen	🟦
79800	79740	angehen	🟦
79517	78181	anlocken	⬛
76490	74114	annehmen	🟦
75163	74114	annehmen	🟦
77336	77249	annehmen	🟦
79535	78077	anordnen	🟥
83780	75422	anpassen	⬛
82446	82402	ansehen	🟦
82445	82402	ansehen	🟦
75659	144803	ansiedeln	⬛
80564	76263	anwenden	🟦
77735	76263	anwenden	🟦
144832	75543	anzeigen	🟥
77955	77709	arbeiten	🟦
79738	79207	attackieren	🟦
75850	83145	aufbauen	🟦
78434	85400	aufdecken	🟦
79554	76194	auferlegen	🟦
83470	79874	aufgeben	🟦
83497	85392	aufheben	🟦
83504	73727	aufhören	🟦
77580	77882	aufklären	🟥
78832	77882	aufklären	🟦
82438	77430	aufpassen	🟦

LexIds	maps to	lemma	C.
74690	74898	aufregen	🟦
144916	74898	aufregen	🟥
82315	77888	aufspüren	🟥
80824	80818	aufstellen	🟦
83259	78652	aufstellen	🟦
82739	81866	auftauchen	🟦
77554	81866	auftauchen	⬛
85538	75835	aufteilen	🟦
75671	75667	auftreten	🟥
83814	82740	auftreten	🟥
82725	74394	aufweisen	🟦
84888	84886	ausbauen	🟦
84887	84886	ausbauen	🟦
83156	78555	ausdenken	🟦
77474	74521	aushalten	🟦
77462	74521	aushalten	🟦
83426	83190	auslösen	🟦
145113	76111	ausschalten	🟦
78829	78613	aussprechen	🟦
145187	84768	austauschen	🟥
145195	83519	ausweichen	⬛
73494	73491	auszeichnen	🟦
82930	82896	bauen	🟥
77382	79034	beanspruchen	🟦
74672	74678	bedauern	🟥
82700	80406	bedecken	🟦
74853	73640	beeindrucken	🟦
84840	78080	beeinflussen	🟥
84870	79663	beeinträchtigen	🟦
145236	80003	befestigen	🟦
76443	76256	befriedigen	🟦
82286	77712	begegnen	🟦
82320	75176	begegnen	⬛
83406	145239	beginnen	🟥
81169	75945	begleiten	🟦
109526	79013	begründen	🟥

LexIds	maps to	lemma	C.
79021	78337	beharren	■
79766	77478	behaupten	■
109404	79094	bekräftigen	■
145263	79803	bekämpfen	■
76219	73964	belohnen	■
77420	75553	bemühen	■
78041	75368	benennen	■
85957	76270	benutzen	■
77750	78343	berücksichtigen	■
74239	75567	beschaffen	■
76509	77950	beschäftigen	■
109437	79935	besetzen	■
109435	79935	besetzen	■
75566	75031	besorgen	■
145311	78029	bestimmen	■
109454	75372	bestimmen	■
78328	78324	bestätigen	■
79082	78324	bestätigen	■
77483	75262	besuchen	■
141358	76528	betreffen	■
75802	75324	betreiben	■
80757	80753	bewegen	■
78441	82734	beweisen	■
78598	82734	beweisen	■
109317	73988	bezahlen	■
109316	73988	bezahlen	■
77734	79049	beziehen	■
76533	79049	beziehen	■
74039	75746	bieten	■
83873	75746	bieten	■
75779	75746	bieten	■
79585	77993	billigen	■
79164	75057	binden	■
82299	82303	blicken	■
85323	76113	blockieren	■
85315	76113	blockieren	■

LexIds	maps to	lemma	C.
77378	85724	brauchen	■
85727	85724	brauchen	■
84250	83725	brechen	■
76300	83725	brechen	■
81248	73921	bringen	■
78032	73765	charakterisieren	■
78975	78552	darlegen	■
73766	73304	darstellen	■
78976	78551	darstellen	■
78954	78551	darstellen	■
109332	78593	demonstrieren	■
77708	77789	denken	■
83258	78596	dokumentieren	■
82808	82055	drehen	■
81914	82055	drehen	■
83349	79622	drucken	■
81188	80691	drängen	■
75872	75023	durchführen	■
75866	75023	durchführen	■
79887	76367	durchsetzen	■
76240	73457	eignen	■
78345	73551	einbeziehen	■
77752	73551	einbeziehen	■
77963	75164	eingehen	■
77373	77361	einrichten	■
85175	74094	einräumen	■
76493	76492	einsetzen	■
77362	75462	einstellen	■
144378	74209	empfangen	■
82487	74485	empfinden	■
83548	83535	enden	■
82306	77588	entdecken	■
83174	78984	entfalten	■
78044	76437	entscheiden	■
76222	73963	entschädigen	■
76442	73437	entsprechen	■

LexIds	maps to	lemma	C.
83158	78543	entwerfen	■
83036	78535	entwickeln	■
84008	83834	entwickeln	■
83882	83834	entwickeln	■
109986	74318	erarbeiten	■
74571	74547	erfreuen	■
73413	76454	erfüllen	■
78581	73745	ergeben	■
74434	73745	ergeben	■
84937	77818	ergänzen	■
83883	78308	erheben	■
74724	77109	erholen	■
84039	84038	erhöhen	■
82264	82262	erkennen	■
78970	78895	erklären	■
89997	74211	erlangen	■
76088	78311	erlauben	■
77545	75260	erleben	■
77541	75260	erleben	■
79714	74515	erleiden	■
74657	74515	erleiden	■
77886	82321	ermitteln	■
82764	76087	ermöglichen	■
79193	79923	erobern	■
110251	78567	erschließen	■
100797	74609	erschrecken	■
77454	74518	ertragen	■
77331	77396	erwarten	■
74237	74322	erwerben	■
78960	78959	erzählen	■
83450	75849	eröffnen	■
144397	83148	etablieren	■
81239	81559	fahren	■
81634	81559	fahren	■
87060	73571	fehlen	■
87224	84801	festigen	■
78740	75095	festlegen	■
82261	77584	feststellen	■
77892	77584	feststellen	■
82307	77891	finden	■
81546	81620	fliegen	■
141265	81350	fliegen	■
79030	77376	fordern	■
112657	78321	freigeben	■
74620	74602	fürchten	■
75118	73801	geben	■
81724	81356	gehen	■
130725	73519	gehen	■
73387	73375	geschehen	■
78313	76090	gestatten	■
78313	76090	gestatten	■
77245	77229	glauben	■
82690	82239	glänzen	■
78194	73600	halten	■
77745	73600	halten	■
77593	73600	halten	■
77652	76286	halten	■
74370	73671	halten	■
73856	73815	handeln	■
83800	77800	heben	■
83793	84749	heilen	■
82323	77583	herausfinden	■
78781	78775	hervorheben	■
79668	76127	hindern	■
75265	75216	hingehen	■
77991	74519	hinnehmen	■
82728	78787	hinweisen	■
82450	82447	hören	■
77481	82447	hören	■
74870	78174	inspirieren	■
77244	77241	kennen	■
77242	77241	kennen	■

LexIds	maps to	lemma	C.
74728	82603	klagen	🟦
80318	80310	klopfen	🟦
78529	78522	klären	🟦
84083	73789	kommen	🟦
77713	79643	konfrontieren	🟦
78129	75814	kontrollieren	🟦
83243	85706	kopieren	🟦
82863	85706	kopieren	🟦
141069	79789	kämpfen	🟦
81843	81834	landen	🟧
81449	81357	laufen	🟦
83806	73401	laufen	🟦
109367	76423	lauten	🟦
73265	76674	leben	🟥
83944	74723	legen	🟦
86971	75707	lehren	🟦
79287	77523	lesen	🟦
82677	82207	leuchten	🟦
79516	74501	locken	🟦
78179	74501	locken	⬛
140156	77196	locken	🟦
78509	76298	lösen	🟦
78426	76298	lösen	🟦
77579	76298	lösen	🟦
83092	83110	malen	🟦
86797	86794	melden	🟦
86796	86794	melden	🟦
110714	80694	mischen	🟦
77600	82281	mitbekommen	🟦
75241	75250	mitmachen	🟦
74249	81171	mitnehmen	🟦
140604	80058	montieren	🟦
74626	73584	mögen	🟦
77590	78574	nehmen	🟥
85914	74109	nehmen	🟥
80339	74109	nehmen	🟦

LexIds	maps to	lemma	C.
73793	73792	neigen	🟦
79449	76412	nennen	🟦
74900	74688	nerven	🟦
78357	75851	organisieren	🟦
75569	74255	organisieren	🟦
141981	80361	packen	⬛
112508	112507	probieren	🟦
82766	78517	produzieren	🟦
142056	75742	promovieren	🟦
142072	75735	qualifizieren	🟦
86970	85872	rauchen	🟦
110711	75589	regeln	🟦
141611	82907	rekonstruieren	🟦
85174	75822	räumen	🟦
82749	78518	schaffen	🟥
82781	78518	schaffen	🟦
129735	79300	schimpfen	🟥
85814	129775	schmecken	🟦
87037	74801	schreien	🟦
141668	84827	schwächen	🟦
79748	76018	schützen	🟦
74386	74371	sparen	🟦
83017	74363	speichern	🟥
79286	78950	sprechen	🟦
81463	80765	springen	🟦
82488	74489	spüren	🟦
80952	80958	stammen	🟥
80957	80958	stammen	🟦
83441	75871	starten	🟦
130045	74497	staunen	🟦
79999	80440	stecken	🟦
80446	80440	stecken	🟦
89378	89380	stecken	⬛
84903	77806	steigern	🟥
80844	80813	stellen	🟦
82666	76837	stinken	🟦

LexIds	maps to	lemma	C.
81861	83502	stoppen	🟦
81201	81093	stoßen	🟦
82679	82208	strahlen	🟦
145181	83179	strahlen	🟦
141822	79772	streiten	🟦
83764	84804	stärken	🟦
89400	79649	stören	🟦
73751	75953	stützen	🟦
81986	75683	tanzen	🟦
75197	77276	trauen	🟥
75273	75175	treffen	🟦
89423	89422	treten	🟦
130357	82360	umsehen	🟦
83973	75159	unterbringen	🟦
130381	75863	unternehmen	🟦
86282	79915	unterwerfen	🟦
78656	76196	urteilen	🟥
78729	75108	verabschieden	🟦
130400	85386	verbergen	🟦
84688	83789	verbessern	🟦
82970	85720	verbrauchen	🟦
110875	74215	verbuchen	🟦
81361	77414	verfolgen	🟦
79560	74337	verfügen	🟥
78031	74337	verfügen	🟥
74405	74337	verfügen	🟦
130457	75012	vergewaltigen	🟦
73296	73645	verhalten	🟦
130471	78674	verhandeln	🟦
78804	75070	verheiraten	🟦
84852	84067	verkürzen	🟦
75925	77318	verlangen	🟦
83938	74423	verlieren	🟦
84003	84923	verlängern	🟦
112505	75571	vermitteln	🟦
111004	76022	vernachlässigen	🟦
84659	79919	vernichten	🟦
84262	79919	vernichten	🟦
131539	78078	verordnen	🟦
112413	75223	verpassen	🟥
112409	75223	verpassen	🟥
79171	75099	verpflichten	🟦
78744	78812	verraten	🟦
81224	81074	verschieben	🟦
75762	79159	versprechen	🟦
89447	78850	verständigen	🟦
79792	79744	verteidigen	🟦
76011	79011	verteidigen	🟦
78361	85576	verteilen	🟦
132277	75434	vertragen	🟥
82963	76271	verwenden	🟦
77400	79042	vorbehalten	🟦
132404	79808	vordringen	🟦
112510	82326	vorfinden	🟦
78653	75694	vorgeben	🟦
77366	77365	vorsehen	🟦
78570	77596	vorstellen	🟥
76414	76413	vorstellen	🟦
132715	78967	vortragen	🟦
82721	73940	vorweisen	🟦
109707	109708	wachen	🟦
84007	76735	wachsen	🟦
83859	84024	wachsen	🟥
80590	84998	wachsen	🟦
77275	75194	wagen	🟦
83556	73391	wandeln	🟦
78918	78913	warnen	🟦
89494	73656	warten	🟦
76055	73656	warten	🟦
73824	73823	wechseln	🟥
89501	84143	wechseln	🟦
85060	74157	wegnehmen	🟥

LexIds	maps to	lemma	C.
132876	82251	wehen	🟥
112234	79746	wehren	🟦
133237	79595	weiterleiten	🟦
133293	133286	wenden	🟦
109333	79199	werben	🟦
84147	77510	wiederholen	🟦
113289	82738	wiederspiegeln	🟦
73643	73637	wirken	🟦
83180	73637	wirken	🟦
73329	73312	wohnen	🟦
74008	73967	zahlen	🟥
89629	83077	zeichnen	🟦
83101	83077	zeichnen	🟦
73628	78592	zeigen	🟦
113100	78428	zerlegen	🟦
81203	81075	ziehen	🟦

LexIds	maps to	lemma	C.
79069	78227	zugeben	🟦
78315	76091	zulassen	🟦
139606	78532	zurückführen	🟦
74848	73307	zusammenhängen	🟦
75845	74281	zusammenstellen	🟦
78533	78004	zuschreiben	🟦
139871	77996	zustimmen	🟦
84160	78231	ändern	🟥
78608	78742	äußern	🟥
83970	85366	öffnen	🟦
83965	85376	öffnen	🟦
73739	73831	überlassen	🟦
74111	74110	übernehmen	🟥
130392	73677	übersehen	🟦
82436	76079	überwachen	🟦
139979	76299	überwinden	🟦

B RESOURCE VERSIONS

This appendix lists the details on the corpora we used, in particular the version or date accessed.

1. **BabelNet** – Version 4.0.1
2. **Bundestag Corpus** – Full texts of the plenary minutes and printed papers of the German Bundestag from the 1st to the 18th legislative period (1949–2017)
3. **COW** – decow16ax (DE stands for German, COW for "COrpus from the Web", 16 for 2016 (major technology version), A for the first release built using 2016 technology. The following X indicates that the corpus is a sentence shuffle)
4. **COW16b** – decow16bx (DE stands for German, COW for "COrpus from the Web", 16 for 2016 (major technology version), B for the second release built using 2016 technology. The following X indicates that the corpus is a sentence shuffle)
5. **DeReKo** – We did not have access to this corpus directly, due to licensing issues. Instead, the *Institut für Deutsche Sprache* (IDS) kindly sent us a summary of frequency, lemma and POS information for tokens occurring in a section (DeReKo-2020-I subcorpus) of the full corpus
6. **deWaC** – https://wacky.sslmit.unibo.it (Baroni *et al.* 2009)
7. **DTA** – *Deutsches Textarchiv*. Core and supplementary texts, version released on July 21, 2017
8. **Duden** – *Deutsches Universalwörterbuch* 2003; for exemplification we additionally consulted the Duden online version (download: 2024-02-14)
9. **EU Bookshop** – Release v2 (Tiedemann 2012)
10. **E-VALBU** – final version
11. **Gutenberg** – Edition 13
12. **GermaNet** – Version 14
13. **GVSD** – The *German Verb Subcategorisation Database*. Accessed on February 15, 2021
14. **Leipziger Wortschatz** – volumes 1995–1997 (Goldhahn *et al.* 2012)
15. **Textbooks** – A collection of 14 German textbooks on economics, published between 2014 and 2020. The textbooks have been used in the study by Lücking *et al.* (2021) and are listed in their appendix B
16. **SALSA** – SALSA 2.0
17. **Süddeutsche Zeitung** – 1992–2014
18. **TüBa-D/Z** – Version 10.0
19. **WebCAGe** – Version 3.0
20. **Wikipedia** – German version, accessed on February 3, 2016.
21. **Wiktionary** – German version, accessed on May 1, 2019.
22. **Die ZEIT** – 1946–2007

REFERENCES

Giuseppe ABRAMI, Mevlüt BAGCI, Leon HAMMERLA, and Alexander MEHLER (2022), German Parliamentary Corpus (GerParCor), in *Proceedings of the Language Resources and Evaluation Conference (LREC 2022)*, pp. 1900–1906, European Language Resources Association, Marseille, France.

Jeff ALSTOTT, Ed BULLMORE, and Dietmar PLENZ (2014), powerlaw: a Python package for analysis of heavy-tailed distributions, *PLoS ONE*, 9(4):e95816, doi:10.1371/journal.pone.0095816.

Nicholas ASHER, Márta ABRUSÁN, and Tim VAN DE CRUYS (2017), Types, meanings and co-composition in lexical semantics, in Stergios CHATZIKYRIAKIDIS and Zhaohui LUO, editors, *Modern perspectives in type-theoretical semantics*, number 98 in Studies in Linguistics and Philosophy, pp. 135–161, Springer International Publishing AG, Cham, Switzerland, doi:10.1007/978-3-319-50422-3_6.

Collin F. BAKER, Charles J. FILLMORE, and John B. LOWE (1998), The Berkeley FrameNet project, in *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, COLING-ACL '98, August 10–14, 1998, Université de Montréal, Montréal, Quebec, Canada. Proceedings of the Conference.*, pp. 86–90, `http://aclweb.org/anthology/P/P98/P98-1013.pdf`.

Marco BARONI, Silvia BERNARDINI, Adriano FERRARESI, and Eros ZANCHETTA (2009), The WaCky wide web: A collection of very large linguistically processed web-crawled corpora, *Language Resources & Evaluation*, 43:209–226, doi:10.1007/s10579-009-9081-4.

Emily M. BENDER, Timnit GEBRU, Angelina MCMILLAN-MAJOR, and Shmargaret SHMITCHELL (2021), On the dangers of stochastic parrots: Can language models be too big?, in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, p. 610–623, Association for Computing Machinery, New York, NY, USA, ISBN 9781450383097, doi:10.1145/3442188.3445922.

James BERGSTRA and Yoshua BENGIO (2012), Random search for hyper-parameter optimization, *Journal of Machine Learning Research*, 13:281–305, `http://dl.acm.org/citation.cfm?id=2188395`.

James BERGSTRA, Daniel YAMINS, and David D. COX (2013), Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures, in *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pp. 115–123, `http://proceedings.mlr.press/v28/bergstra13.html`.

Michele BEVILACQUA and Roberto NAVIGLI (2020), Breaking through the 80% glass ceiling: Raising the state of the art in word sense disambiguation by incorporating knowledge graph information, in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2854–2864, Association for Computational Linguistics, Online, doi:10.18653/v1/2020.acl-main.255, `https://aclanthology.org/2020.acl-main.255`.

Piotr BOJANOWSKI, Edouard GRAVE, Armand JOULIN, and Tomás MIKOLOV (2016), Enriching word vectors with subword information, *CoRR*, abs/1607.04606, `http://arxiv.org/abs/1607.04606`.

Claire BONIAL, Julia BONN, Kathryn CONGER, Jena HWANG, Martha PALMER, and Nicholas REESEM (2015), English PropBank annotation guidelines, Technical report, Center for Computational Language and Education Research, Institute of Cognitive Science, University of Colorado at Boulder, `http://propbank.github.io/`.

Zdravko BOTEV and Ad RIDDER (2017), *Variance reduction*, pp. 1–6, American Cancer Society, ISBN 9781118445112, doi:10.1002/9781118445112.stat07975, `https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118445112.stat07975`.

Sabine BRANTS, Stefanie DIPPER, Peter EISENBERG, Silvia HANSEN, Esther KÖNIG, Wolfgang LEZIUS, Christian ROHRER, George SMITH, and Hans USZKOREIT (2004), TIGER: Linguistic interpretation of a German corpus, *Journal of Language and Computation*, 2:597–620.

Aljoscha BURCHARDT, Katrin ERK, Anette FRANK, Andrea KOWALSKI, Sebastian PADÓ, and Manfred PINKAL (2006), The SALSA corpus: A German corpus resource for lexical semantics, in *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006)*, European Language Resources Association (ELRA), Genoa, Italy, `http://www.lrec-conf.org/proceedings/lrec2006/pdf/339_pdf.pdf`.

Gennaro CHIERCHIA and Sally MCCONNELL-GINET (2000), *Meaning and grammar – an introduction to semantics*, MIT Press, Cambridge, 2 edition.

Aaron CLAUSET, Cosma Rohilla SHALIZI, and Mark E. J. NEWMAN (2009), Power-law distributions in empirical data, *SIAM Review*, 51(4):661–703, doi:10.1137/070710111, Society of Industrial and Applied Mathematics.

Robin COOPER (2011), Copredication, quantification and frames, in Sylvain POGODALLA and Jean-Philippe PROST, editors, *Logical aspects of computational linguistics*, number 6736 in Lecture Notes in Computer Science, pp. 64–79, Springer, Berlin and Heidelberg, doi:10.1007/978-3-642-22221-4_5.

D. Alan CRUSE (2000), *Meaning in language*, Oxford University Press, New York.

Gábor Csárdi and Tamás Nepusz (2006), The igraph software package for complex network research, *InterJournal*, Complex Systems:1695, `https://igraph.org`.

Benjamin David, Sylvia Springorum, and Sabine Schulte im Walde (2014), German perception verbs: Automatic classification of prototypical and multiple non-literal meanings, in *Proceedings of the 12th Konvens 2014*.

Henriëtte de Swart (2011), Mismatches and coercion, in Claudia Maienborn, Klaus von Heusinger, and Paul Portner, editors, *Semantics: An international handbook of natural language meaning*, volume 1 of *Handbücher zur Sprach- und Kommunikationswissenschaft*, chapter 25, pp. 574–597, De Gruyter Mouton, doi:10.1515/9783110226614.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018), BERT: Pre-training of deep bidirectional transformers for language understanding, *arXiv preprint arXiv:1810.04805*.

Andrea Di Fabio, Simone Conia, and Roberto Navigli (2019), VerbAtlas: A novel large-scale verbal semantic resource and its application to semantic role labeling, in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 627–637, Association for Computational Linguistics, Hong Kong, China, doi:10.18653/v1/D19-1058, `https://www.aclweb.org/anthology/D19-1058`.

Stefanie Dipper, Hannah Kermes, Esther König-Baumer, Wolfgang Lezius, Frank H. Müller, and Tylman Ule (2002), DEREKO – (DEutsches REferenzKOrpus) German Reference Corpus. Final report (Part I), Technical report, IMS: Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart, SfS: Seminar für Sprachwissenschaft, Universität Tübingen.

Konrad Duden, Dieter Berger, and Werner Scholze (1980), *Duden*, volume 2, Bibliographisches Institut.

Veena D. Dwivedi (2013), Interpreting quantifier scope ambiguity: Evidence of heuristic first, algorithmic second processing, *PLoS ONE*, 8(11):e81461, doi:10.1371/journal.pone.0081461.

Philip Edmonds and Scott Cotton (2001), SENSEVAL-2: overview, in *Proceedings of Second International Workshop on Evaluating Word Sense Disambiguation Systems, SENSEVAL@ACL 2001, Toulouse, France, July 5-6, 2001*, pp. 1–5, `https://aclanthology.info/papers/S01-1001/s01-1001`.

Gertrud Faass and Kerstin Eckart (2013), SdeWaC – a corpus of parsable sentences from the web, in Iryna Gurevych, Chris Biemann, and Torsten Zesch, editors, *Language processing and knowledge in the web*, pp. 61–68, Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN 978-3-642-40722-2.

Christiane Fellbaum, editor (1998), *WordNet: An electronic lexical database*, MIT Press, Cambridge.

Christiane FELLBAUM and George A. MILLER (1998), *Lexical chains as representations of context for the detection and correction of malapropisms*, pp. 305–332, MITP, ISBN 9780262272551, https://ieeexplore.ieee.org/document/6287673.

Christiane FELLBAUM, Martha PALMER, Hoa Trang DANG, Lauren DELFS, and Susanne WOLF (2001), Manual and automatic semantic annotation with WordNet, in *Proceedings of the NAACL Workshop on WordNet and Other Lexical Resources: Applications, Customizations*, pp. 1–8, Carnegie Mellon University Pittsburg, PA.

Charles J. FILLMORE and Colin BAKER (2010), A frames approach to semantic analysis, in Bernd HEINE and Heiko NARROG, editors, *The Oxford Handbook of Linguistic Analysis*, pp. 313–340, Oxford University Press, Oxford.

William A. GALE, Kenneth W. CHURCH, and David YAROWSKY (1992), One sense per discourse, in *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*, https://www.aclweb.org/anthology/H92-1045.

Spandana GELLA, Desmond ELLIOTT, and Frank KELLER (2019), Cross-lingual visual verb sense disambiguation, in *Proceedings of NAACL-HLT 2019*, pp. 1998–2004.

Brendan S. GILLON (1990), Ambiguity, generality, and indeterminacy: Tests and definitions, *Synthese*, 85(3):391–416.

Dirk GOLDHAHN, Thomas ECKART, and Uwe QUASTHOFF (2012), Building large monolingual dictionaries at the Leipzig Corpora Collection: From 100 to 200 languages, in Nicoletta CALZOLARI, Khalid CHOUKRI, Thierry DECLERCK, Mehmet Ugur DOGAN, Bente MAEGAARD, Joseph MARIANI, Jan ODIJK, and Stelios PIPERIDIS, editors, *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, European Language Resources Association (ELRA), Istanbul, Turkey, ISBN 978-2-9517408-7-7.

Birgit HAMP and Helmut FELDWEG (1997), GermaNet – a lexical-semantic net for German, in *Proceedings of ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, pp. 9–15.

Wahed HEMATI (2020), *TextImager-VSD: Large scale verb sense disambiguation and named entity recognition in the context of TextImager*, Ph.D. thesis, Goethe-University Frankfurt, http://publikationen.ub.uni-frankfurt.de/frontdoor/index/index/docId/56089.

Wahed HEMATI, Tolga USLU, and Alexander MEHLER (2016), TextImager: A distributed UIMA-based system for NLP, in *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference System Demonstrations, December 11–16, 2016, Osaka, Japan*, pp. 59–63, https://www.aclweb.org/anthology/C16-2013/.

Alexander HENLEIN and Alexander MEHLER (2022), What do toothbrushes do in the kitchen? How transformers think our world is structured, in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5791–5807, Association for Computational Linguistics, Seattle, United States, doi:10.18653/v1/2022.naacl-main.425, https://aclanthology.org/2022.naacl-main.425.

Verena HENRICH (2015), *Word sense disambiguation with GermaNet*, Ph.D. thesis, Universität Tübingen, doi:10.15496/publikation-4706, https://publikationen.uni-tuebingen.de/xmlui/handle/10900/63284.

Verena HENRICH and Erhard HINRICHS (2012), A comparative evaluation of word sense disambiguation algorithms for German, in *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012)*, pp. 576–583, European Language Resources Association (ELRA), Istanbul, Turkey, http://www.lrec-conf.org/proceedings/lrec2012/pdf/164_Paper.pdf.

Verena HENRICH and Erhard W. HINRICHS (2013), Extending the TüBa-D/Z Treebank with GermaNet sense annotation, in *Language processing and knowledge in the web – 25th International Conference, GSCL 2013, Darmstadt, Germany, September 25–27, 2013. Proceedings*, pp. 89–96, doi:10.1007/978-3-642-40722-2_9.

Verena HENRICH, Erhard W. HINRICHS, and Tatiana VODOLAZOVA (2011), Aligning GermaNet senses with Wiktionary sense definitions, in *Human Language Technology Challenges for Computer Science and Linguistics – 5th Language and Technology Conference, LTC 2011, Poznań, Poland, November 25–27, 2011, Revised Selected Papers*, pp. 329–342, doi:10.1007/978-3-319-08958-4_27.

Verena HENRICH, Erhard W. HINRICHS, and Tatiana VODOLAZOVA (2012), Webcage – A web-harvested corpus annotated with GermaNet senses, in *EACL 2012, 13th Conference of the European Chapter of the Association for Computational Linguistics, Avignon, France, April 23–27, 2012*, pp. 387–396, http://aclweb.org/anthology/E/E12/E12-1039.pdf.

Ray JACKENDOFF (1983), *Semantics and cognition*, MIT Press, Cambridge, MA.

Armand JOULIN, Edouard GRAVE, Piotr BOJANOWSKI, and Tomas MIKOLOV (2017), Bag of tricks for efficient text classification, in *Proceedings of the 15th Conference of the EACL: Volume 2, Short Papers*, pp. 427–431, Association for Computational Linguistics, Valencia, Spain, https://www.aclweb.org/anthology/E17-2068.

David KAPLAN (1989), Demonstratives, in Joseph ALMOG, John PERRY, and Howard WETTSTEIN, editors, *Themes from Kaplan*, pp. 481–563, Oxford University Press, New York and Oxford.

Rudi KELLER (1990), *Sprachwandel: von der unsichtbaren Hand in der Sprache*, Francke, Tübingen.

Christopher KENNEDY (2011), Ambiguity and vagueness: An overview, in Claudia MAIENBORN, Klaus VON HEUSINGER, and Paul PORTNER, editors, *Semantics: An international handbook of natural language meaning*, volume 1 of *Handbücher zur Sprach- und Kommunikationswissenschaft*, chapter 23, pp. 507–535, De Gruyter Mouton, doi:10.1515/9783110226614.

Adam KILGARRIFF (1997), “I don’t believe in word senses”, *Computers and the Humanities*, 31(2):91–113, doi:10.1023/A:1000583911091.

Adam KILGARRIFF (1998), Gold standard datasets for evaluating word sense disambiguation programs, *Computer Speech & Language*, 12(4):453–472, doi:10.1006/csla.1998.0108.

Paul R. KROEGER (2019), *Analyzing meaning*, number 5 in Textbooks in Language Sciences, Language Science Press, Berlin, second corrected and slightly revised edition.

Jacqueline KUBCZAK (2009), Hier wird Ihnen geholfen! E-VALBU – Das elektronische Valenzwörterbuch deutscher Verben, *Sprachreport*, 4:17–23.

Claudia KUNZE and Lothar LEMNITZER (2002), GermaNet – representation, visualization, application, in M. RODRIGUEZ GONZÁLEZ and C. PAZ SUÁREZ ARAUJO, editors, *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC 2002)*, pp. 1485–1491, European Language Resources Association, Paris.

Els LEFEVER and Véronique HOSTE (2010), SemEval-2010 task 3: Cross-lingual word sense disambiguation, in *5th International Workshop on Semantic Evaluation (SemEval 2010)*, pp. 15–20, Association for Computational Linguistics (ACL).

Els LEFEVER and Véronique HOSTE (2013), SemEval-2013 task 10: Cross-lingual word sense disambiguation, in *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pp. 158–166.

Beth LEVIN (1993), *English verb classes and alternations: A preliminary investigation*, University of Chicago Press.

Andy LÜCKING, Sebastian BRÜCKNER, Giuseppe ABRAMI, Tolga USLU, and Alexander MEHLER (2021), Computational linguistic assessment of textbooks and online texts by means of threshold concepts in economics, *Frontiers in Education*, 5:578475, doi:10.3389/feduc.2020.578475, `https://www.frontiersin.org/articles/10.3389/feduc.2020.578475/`.

Fuli LUO, Tianyu LIU, Qiaolin XIA, Baobao CHANG, and Zhifang SUI (2018), Incorporating glosses into neural word sense disambiguation, in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15–20, 2018, Volume 1: Long Papers*,

pp. 2473–2482,
https://aclanthology.info/papers/P18-1230/p18-1230.

John LYONS (1977), *Semantics*, volume 1, Cambridge University Press, London.

Alexander MEHLER, Rüdiger GLEIM, Wahed HEMATI, and Tolga USLU (2018), Skalenfreie online soziale Lexika am Beispiel von Wiktionary, in Stefan ENGELBERG, Henning LOBIN, Kathrin STEYER, and Sascha WOLFER, editors, *Proceedings of 53rd Annual Conference of the Institut für Deutsche Sprache (IDS), March 14–16, Mannheim, Germany*, pp. 269–291, De Gruyter, Berlin.

Oren MELAMUD, Jacob GOLDBERGER, and Ido DAGAN (2016), context2vec: Learning generic context embedding with bidirectional LSTM, in *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11–12, 2016*, pp. 51–61,
http://aclweb.org/anthology/K/K16/K16-1006.pdf.

George A. MILLER (1995), Wordnet: A lexical database for English, *Communications of the ACM*, 38(11):39–41, doi:10.1145/219717.219748,
http://doi.acm.org/10.1145/219717.219748.

George A. MILLER, Martin CHODOROW, Shari LANDES, Claudia LEACOCK, and Robert G. THOMAS (1994), Using a semantic concordance for sense identification, in *Human Language Technology: Proceedings of a workshop held at Plainsboro, New Jersey, March 8–11, 1994*,
https://aclanthology.org/H94-1046.

Marc MOENS and Mark STEEDMAN (1988), Temporal ontology and temporal reference, *Computational Linguistics*, 14(2):15–28.

Roberto NAVIGLI (2009), Word sense disambiguation: A survey, *ACM Computing Survey*, 41(2):10:1–10:69, doi:10.1145/1459352.1459355,
https://doi.org/10.1145/1459352.1459355.

Roberto NAVIGLI, José CAMACHO-COLLADOS, and Alessandro RAGANATO (2017), Word sense disambiguation: A unified evaluation framework and empirical comparison, in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3–7, 2017, Volume 1: Long Papers*, pp. 99–110,
https://aclanthology.info/papers/E17-1010/e17-1010.

Roberto NAVIGLI, David JURGENS, and Daniele VANNELLA (2013), SemEval-2013 task 12: Multilingual word sense disambiguation, in *Proceedings of the 7th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2013, Atlanta, Georgia, USA, June 14-15, 2013*, pp. 222–231,
http://aclweb.org/anthology/S/S13/S13-2040.pdf.

Roberto NAVIGLI and Simone Paolo PONZETTO (2012), BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network, *Artificial Intelligence*, 193:217–250, ISSN 0004-3702, doi:10.1016/j.artint.2012.07.001.

Mark E. J. Newman (2005), Power laws, Pareto distributions and Zipf's law, *Contemporary Physics*, 46:323–351.

Geoffrey Nunberg (1995), Transfers of meaning, *Journal of Semantics*, 12(2):109–132, doi:10.1093/jos/12.2.109.

OpenAI (2023), ChatGPT (version 3.5), `https://github.com/openai/gpt-3`.

Martha Palmer, Hoa Trang Dang, and Christiane Fellbaum (2007), Making fine-grained and coarse-grained sense distinctions, both manually and automatically, *Natural Language Engineering*, 13(2):137–163, doi:10.1017/S135132490500402X.

Martha Palmer, Daniel Gildea, and Nianwen Xue (2010), *Semantic role labeling*, Morgan & Claypool Publishers.

Simone Papandrea, Alessandro Raganato, and Claudio Delli Bovi (2017), SupWSD: A flexible toolkit for supervised word sense disambiguation, in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9–11, 2017 – System Demonstrations*, pp. 103–108, `https://aclanthology.info/papers/D17-2018/d17-2018`.

Rebecca J. Passonneau, Collin F. Baker, Christiane Fellbaum, and Nancy Ide (2012), The MASC word sense corpus, in *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012), Istanbul, Turkey, May 23–25, 2012*, pp. 3025–3030, `http://www.lrec-conf.org/proceedings/lrec2012/summaries/589.html`.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer (2018), Deep contextualized word representations, in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pp. 2227–2237, `https://aclanthology.info/papers/N18-1202/n18-1202`.

Mohammad Taher Pilehvar and Jose Camacho-Collados (2021), *Embeddings in natural language processing: Theory and advances in vector representations of meaning*, Morgan & Claypool Publishers.

Sameer Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer (2007), Semeval-2007 task 17: English lexical sample, SRL and all words, in *Proceedings of the 4th International Workshop on Semantic Evaluations, SemEval@ACL 2007, Prague, Czech Republic, June 23–24, 2007*, pp. 87–92, `http://aclweb.org/anthology/S/S07/S07-1016.pdf`.

James Pustejovsky (1991), The generative lexicon, *Computational Linguistics*, 17:409–441.

James Pustejovsky (1995), *The generative lexicon*, MIT Press, Cambridge, MA.

Diana RAILEANU, Paul BUITELAAR, Spela VINTAR, and Jörg BAY (2002), Evaluation corpora for sense disambiguation in the medical domain, in *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002), May 29–31, 2002, Las Palmas, Canary Islands, Spain*, http://www.lrec-conf.org/proceedings/lrec2002/sumarios/166.htm.

Prajit RAMACHANDRAN, Barret ZOPH, and Quoc V. LE (2017), Searching for activation functions, *CoRR*, abs/1710.05941, http://arxiv.org/abs/1710.05941.

Shauli RAVFOGEL, Yanai ELAZAR, Jacob GOLDBERGER, and Yoav GOLDBERG (2020), Unsupervised distillation of syntactic information from contextualized word representations, *arXiv preprint arXiv:2010.05265*.

Burghard B. RIEGER (1989), *Unscharfe Semantik: Die empirische Analyse, quantitative Beschreibung, formale Repräsentation und prozedurale Modellierung vager Wortbedeutungen in Texten*, Peter Lang, Frankfurt a. M.

Burghard B. RIEGER (2001), Computing granular word meanings. A fuzzy linguistic approach in computational semiotics, in Paul WANG, editor, *Computing with words*, pp. 147–208, Wiley, New York.

Sascha ROTHE, Shashi NARAYAN, and Aliaksei SEVERYN (2020), Leveraging pre-trained checkpoints for sequence generation tasks, *Transactions of the Association for Computational Linguistics*, 8:264–280, doi:10.1162/tacl_a_00313, https://aclanthology.org/2020.tacl-1.18.

Josef RUPPENHOFER, Michael ELLSWORTH, Myriam SCHWARZER-PETRUCK, Christopher R. JOHNSON, and Jan SCHEFFCZYK (2016), FrameNet II: Extended theory and practice, Technical report, International Computer Science Institute.

Jahn-Takeshi SAITO, Joachim WAGNER, Graham KATZ, P. D. Gerson REUTER, Michael B. BURKE, and Sabine REINHARD (2002), Evaluation of GermaNet: Problems using GermaNet for automatic word sense disambiguation, in *LREC Workshop on WordNet Structure and Standardization and How these Affect WordNet Applications and Evaluation*.

Silke SCHEIBLE, Sabine SCHULTE IM WALDE, Marion WELLER, and Max KISSELEW (2013), A compact but linguistically detailed database for German verb subcategorisation relying on dependency parses from a web corpus: Tool, guidelines and resource, in *Proceedings of the 8th Web as Corpus Workshop*, pp. 63–72, Lancaster, UK.

Anne SCHILLER, Simone TEUFEL, Christine STÖCKERT, and Christine THIELEN (1999), Guidelines für das Tagging deutscher Textcorpora mit STTS (Kleines und großes Tagset), Technical report, Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.

Karin Kipper SCHULER (2006), *Verbnet: A broad-coverage, comprehensive verb lexicon*, Ph.D. thesis, University of Pennsylvania, http://verbs.colorado.edu/~kipper/Papers/dissertation.pdf.

Helmut SCHUMACHER, editor (1986), *Verben in Feldern. Valenzwörterbuch zur Syntax und Semantik deutscher Verben*, de Gruyter, Berlin and New York.

Helmut SCHUMACHER, Jacqueline KUBCZAK, Renate SCHMIDT, and Vera DE RUITER (2004), *VALBU – Valenzwörterbuch deutscher Verben*, number 31 in Studien zur Deutschen Sprache, Narr, Tübingen.

Hinrich SCHÜTZE (1998), Automatic word sense discrimination, *Computational Linguistics*, 24(1):97–123.

Roland SCHÄFER (2015), Processing and querying large web corpora with the COW14 architecture, in *Proceedings of Challenges in the Management of Large Corpora 3 (CMLC-3)*, UCREL, IDS, Lancaster, `http://rolandschaefer.net/?p=749`.

Roland SCHÄFER and Felix BILDHAUER (2012), Building large corpora from the web using a new efficient tool chain, in *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC 2012)*, pp. 486–493, European Language Resources Association (ELRA), Istanbul, Turkey, ISBN 978-2-9517408-7-7, `http://rolandschaefer.net/?p=70`.

Benjamin SNYDER and Martha PALMER (2004), The English all-words task, in *Proceedings of the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, SENSEVAL@ACL 2004, Barcelona, Spain, July 25-26, 2004*, `https://aclanthology.info/papers/W04-0811/w04-0811`.

Jan-Philipp SOEHN (2005), Selectional restrictions in HPSG: I'll eat my hat!, in Stefan MÜLLER, editor, *Proceedings of the HPSG05 Conference*, pp. 343–353, Department of Informatics, University of Lisbon, CSLI Publications, Stanford.

John F. SOWA (2000), *Knowledge representation: Logical, philosophical, and computational foundations*, Brooks/Cole.

Luc STEELS (2011–12), Modeling the cultural evolution of language, *Physics of Life Reviews*, 8(4):339–356, doi:10.1016/j.plrev.2011.10.014, `http://groups.lis.illinois.edu/amag/langev/paper/steels2011REVIEW.html`.

Angelika STORRER (2001), Digitale Wörterbücher als Hypertexte: Zur Nutzung des Hypertextkonzepts in der Lexikographie, in *Chancen und Perspektiven computer-gestützter Lexikographie: Hypertext, Internet und SGML/XML für die Produktion und Publikation digitaler Wörterbücher*, pp. 53–70, Max Niemeyer Verlag, Tübingen.

Angelika STORRER (2010), Deutsche Internet-Wörterbücher: Ein Überblick, *Lexicographica*, 27(1):155–164.

Heike TELLJOHANN, Erhard W. HINRICHS, Sandra KÜBLER, Heike ZINSMEISTER, and Kathrin BECK (2012), *Stylebook for the Tübingen Treebank of Written German (TüBa-D/Z)*, Seminar für Sprachwissenschaft, Wilhelmstr. 19, D-72074 Tübingen, `http://www.sfs.uni-tuebingen.de/fileadmin/static/ascl/resources/tuebadz-stylebook-1201.pdf`.

THE MATHWORKS, INC. (2012), MATLAB and curve fitting toolbox release 2012, Natick, MA.

Jörg TIEDEMANN (2012), Parallel data, tools and interfaces in OPUS, in Nicoletta CALZOLARI, Khalid CHOUKRI, Thierry DECLERCK, Mehmet Ugur DOGAN, Bente MAEGAARD, Joseph MARIANI, Jan ODIJK, and Stelios PIPERIDIS, editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012)*, European Language Resources Association (ELRA), Istanbul, Turkey, ISBN 978-2-9517408-7-7.

Tolga USLU, Alexander MEHLER, Daniel BAUMARTZ, Alexander HENLEIN, and Wahed HEMATI (2018), fastsense: An efficient word sense disambiguation classifier, in *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan.

Loïc VIAL, Benjamin LECOUTEUX, and Didier SCHWAB (2019), Sense vocabulary compression through the semantic knowledge of WordNet for neural word sense disambiguation, *CoRR*, abs/1905.05677, `http://arxiv.org/abs/1905.05677`.

Piek VOSSEN (1998), Introduction to EuroWordNet, *Computers and the Humanities*, 32(2-3):73–89, doi:10.1023/A:1001175424222.

Nan WANG, Jiwei LI, Yuxian MENG, Xiaofei SUN, Han QIU, Ziyao WANG, Guoyin WANG, and Jun HE (2022), An MRC framework for semantic role labeling, in *Proceedings of the 29th International Conference on Computational Linguistics*, pp. 2188–2198, International Committee on Computational Linguistics, Gyeongju, Republic of Korea, `https://aclanthology.org/2022.coling-1.191`.

Stephen WECHSLER, Jean-Pierre KOENIG, and Anthony DAVIS (2021), Argument structure and linking, in Stefan MÜLLER, Anne ABEILLÉ, Robert D. BORSLEY, and Jean-Pierre KOENIG, editors, *Head-Driven Phrase Structure Grammar: The handbook*, Language Science Press, Berlin, `https://langsci-press.org/catalog/book/259`, prepublished book chapter.

WIKTIONARY (2019), Free dictionary, `https://www.wiktionary.org/`, accessed: 2019-09-23.

Ian H. WITTEN, Eibe FRANK, and Mark A. HALL (2011), *Data mining: Practical machine learning tools and techniques, 3rd edition*, Morgan Kaufmann, Elsevier, ISBN 9780123748560, `http://www.worldcat.org/oclc/262433473`.

Yonghui WU, Mike SCHUSTER, Zhifeng CHEN, Quoc V. LE, Mohammad NOROUZI, Wolfgang MACHEREY, Maxim KRIKUN, Yuan CAO, Qin GAO, Klaus MACHEREY, Jeff KLINGNER, Apurva SHAH, Melvin JOHNSON, Xiaobing LIU, Lukasz KAISER, Stephan GOUWS, Yoshikiyo KATO, Taku KUDO, Hideto KAZAWA, Keith STEVENS, George KURIAN, Nishant PATIL, Wei WANG, Cliff

YOUNG, Jason SMITH, Jason RIESA, Alex RUDNICK, Oriol VINYALS, Greg CORRADO, Macduff HUGHES, and Jeffrey DEAN (2016), Google's neural machine translation system: Bridging the gap between human and machine translation, *CoRR*, abs/1609.08144, http://arxiv.org/abs/1609.08144.

Yichu ZHOU and Vivek SRIKUMAR (2022), A closer look at how fine-tuning changes BERT, in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1046–1061, Association for Computational Linguistics, Dublin, Ireland, doi:10.18653/v1/2022.acl-long.75, https://aclanthology.org/2022.acl-long.75.

Arnold M. ZWICKY and Jerrold M. SADOCK (1975), Ambiguity tests and how to fail them, in *Syntax and Semantics volume 4*, pp. 1–36, Academic Press, New York.

Dominik Mattern

Text Technology Lab
Goethe University Frankfurt
Frankfurt am Main, Germany

Wahed Hemati
0000-0002-5477-2538

Shikenso GmbH
Frankfurt am Main, Germany

Andy Lücking
0000-0002-5070-2233

Text Technology Lab
Goethe University Frankfurt
Frankfurt am Main, Germany

Alexander Mehler
0000-0003-2567-7539
mehler@em.uni-frankfurt.de

Text Technology Lab
Goethe University Frankfurt
Frankfurt am Main, Germany
(Corresponding author)

Dominik Mattern, Wahed Hemati, Andy Lücking, and Alexander Mehler (2024), *On German verb sense disambiguation: A three-part approach based on linking a sense inventory (GermaNet) to a corpus through annotation (TGVCorp) and using the corpus to train a VSD classifier (TTvSense)*, Journal of Language Modelling, 12(1):155–212
https://dx.doi.org/10.15398/jlm.v12i1.356

This work is licensed under the *Creative Commons Attribution 4.0 Public License*.
http://creativecommons.org/licenses/by/4.0/