A Comparative Evaluation of Word Sense Disambiguation Algorithms for German

Verena Henrich, Erhard Hinrichs

University of Tübingen, Department of Linguistics
Wilhelmstr. 19, 72074 Tübingen, Germany
{verena.henrich,erhard.hinrichs}@uni-tuebingen.de

Abstract

The present paper explores a wide range of word sense disambiguation (WSD) algorithms for German. These WSD algorithms are based on a suite of semantic relatedness measures, including path-based, information-content-based, and gloss-based methods. Since the individual algorithms produce diverse results in terms of precision and thus complement each other well in terms of coverage, a set of combined algorithms is investigated and compared in performance to the individual algorithms. Among the single algorithms considered, a word overlap method derived from the Lesk algorithm that uses Wiktionary glosses and GermaNet lexical fields yields the best F-score of 56.36. This result is outperformed by a combined WSD algorithm that uses weighted majority voting and obtains an F-score of 63.59. The WSD experiments utilize the German wordnet GermaNet as a sense inventory as well as WebCAGe (short for: *Web-Harvested Corpus Annotated with GermaNet Senses*), a newly constructed, sense-annotated corpus for this language. The WSD experiments also confirm that WSD performance is lower for words with fine-grained sense distinctions compared to words with coarse-grained senses.

Keywords: Word sense disambiguation, German, combined classifiers

1. Introduction

Word sense disambiguation (WSD) has been a very active area of research in computational linguistics. Most of the work has focused on English. One of the factors that has hampered WSD research for other languages has been the lack of appropriate resources, particularly in the form of sense-annotated corpus data. The present paper focuses on WSD for German and utilizes the German wordnet GermaNet (Kunze and Lemnitzer, 2002; Henrich and Hinrichs, 2010) as a sense inventory as well as WebCAGe (Henrich et al., 2012) (short for: *Web-Harvested Corpus Annotated with GermaNet Senses*), a newly constructed, sense-annotated corpus for this language.

Due to the lack of sense-annotated corpora for German prior to the construction of WebCAGe, there has been relatively little research on WSD for this language.[1] The purpose of this paper is to help close this gap. More specifically, it has the following three goals:

1. To apply a much wider range of WSD algorithms to German since the range of methods that has thus far been applied to German is rather limited.

2. To study the combination of WSD methods in either majority or weighted majority voting schemes as well as in a Borda count setup.

3. To study the effect of linking the GermaNet knowledge base with other web-based lexical resources such as the German version of Wiktionary[2] (Henrich et al., 2011). This mapping was performed for the purpose of including Wiktionary sense definitions into GermaNet and is particularly relevant for word overlap methods, which have previously been shown to perform well for English WSD by Pedersen et al. (2005).

The remainder of this paper is structured as follows: Section 2 provides a brief overview of the German wordnet resource GermaNet. Related work is discussed in Section 3. Section 4 introduces the methods and algorithms for WSD that are used in this paper. The performance of the different WSD algorithms is evaluated in Section 5. Finally, there are concluding remarks and an outlook to future work in Section 6.

2. GermaNet Resource

GermaNet (Kunze and Lemnitzer, 2002; Henrich and Hinrichs, 2010) is a lexical semantic network that is modeled after the Princeton WordNet for English (Fellbaum, 1998). It partitions the lexical space into a set of concepts that are interlinked by semantic relations. A semantic concept is represented as a *synset*, i.e., as a set of words whose individual members (referred to as *lexical units*) are taken to be (near) synonyms. Thus, a synset is a set-representation of the semantic relation of synonymy. There are two types of semantic relations in GermaNet. *Conceptual relations*, e.g., hypernymy, part-whole relations, entailment, or causation, hold between two semantic concepts, i.e. synsets. *Lexical relations*, e.g., antonymy, hold between two individual lexical units. GermaNet covers the three word categories of adjectives, nouns, and verbs, each of which is hierarchically structured in terms of the hypernymy relation of synsets. GermaNet's version 6.0 covers 93407 lexical units, which are grouped into 69594 synsets.

Sense definitions are a crucial component for wordnets. Until recently, GermaNet included sense definitions only for a small number of lexical units. However, comprehensive sense definitions are badly needed in order to enhance its usability for a wide variety of NLP applications,

[1] For more discussion, see Section 3. below.

[2] http://www.wiktionary.org/

including word sense disambiguation. In order to add sense descriptions to GermaNet, a semi-automatic mapping was performed that enriches GermaNet's lexical units with Wiktionary sense definitions. This mapping is the result of a two-stage process: an automatic alignment, which achieves 93.8% accuracy, followed by a manual post-correction – see Henrich et al. (2011).

3. Related Work

Word sense disambiguation has been a widely studied natural language processing task in recent years. It goes beyond the scope of the present paper to provide a comprehensive account of the state of the art in WSD. See Agirre and Edmonds (2006) as well as Navigli (2009) for a more in-depth survey and discussion of WSD methods. In this section, we will focus on existing work on word sense disambiguation for German, so as to be able to relate the research reported in the present paper to previous research on this language. All research on word sense disambiguation work for German (Steffen et al., 2003; Widdows et al., 2003; Broscheit et al., 2010) has focussed on the lexical sample task for WSD, i.e., the disambiguation of a fixed set of polysemous target words. The number of ambiguous target words used in these studies has been rather small, ranging from 24 (Widdows et al., 2003) to 40 (Steffen et al., 2003; Broscheit et al., 2010) nouns. By contrast, the all-words WSD task consists of disambiguating each and every word that appears in a test corpus. If a lexical token contained in the corpus is not ambiguous, this is of course trivial. For ambiguous words, the assumption is that the surrounding corpus context will aid in identifying the correct sense of the given target word. In recent work, Navigli et al. (2007; 2010) have successfully applied graph-based disambiguation methods to this all-words disambiguation task. We agree that the all-words disambigation task is ultimately more meaningful and more realistic from a cognitive perspective, relating computational approaches to language processing by humans. However, the all-words task has higher prerequisites in terms of sense-annotated corpora that do not currently hold for German. Nevertheless, the present study significantly goes beyond the previous studies on German word sense disambiguation in that a total of 1499 nouns are included in the lexical sample disambiguation task compared to the much smaller samples used by Steffen et al. (2003), Widdows et al. (2003), and Broscheit et al. (2010).

The earliest German WSD studies that we are aware of are those of Steffen et al. (2003) and Widdows et al. (2003) who apply unsupervised methods for domain-specific WSD on medical texts. Steffen et al. (2003) apply two separate WSD methods: One method automatically determines domain-specific senses of ambiguous target words on the basis of their relative statistical relevance across several domain specific corpora. The other method is instance-based and uses k-nearest neighbor classification.

Widdows et al. (2003) perform automatic WSD for English and German and derive their sense inventory for these two languages from the Medical Subject Headings thesaurus (MeSH) contained in the Unified Medical Language System (UMLS). They apply two unsupervised WSD methods: i) bilingual disambiguation based on parallel corpus of English-German medical scientific abstracts obtained from the Springer Link web site[3], and ii) collocational disambiguation as introduced by Yarowsky (1995) which uses multi-word expressions and collocations from UMLS as seed examples for Yarowsky's algorithm.

The studies of Steffen et al. (2003) and Widdows et al. (2003) both focus on the medical domain and are thus domain-dependent. A recent study that performs WSD on a domain-independent German corpus is that of Broscheit et al. (2010). It uses GermaNet as a knowledge base and the graph-based algorithm Personalized PageRank (PPR) of Agirre and Soroa (2009) as well as a voting approach that combines PPR with the unsupervised algorithms of Lapata and Keller (2007) and McCarthy et al. (2004) for determining the most frequent sense. The best results reported by Broscheit et al. (2010) are not obtained by the voting approach but rather by the PPR algorithm alone. In order to compare the performance of the algorithms described in the present paper with the results of Broscheit et al. (2010), the Personalized PageRank algorithm is included in the set of experiments described in Section 5.

4. Word Sense Disambiguation Algorithms

4.1. Semantic Relatedness Measures

In order to be able to apply a wide range of WSD algorithms to German, we have reimplemented the same suite of semantic relatedness algorithms for German that were previously used by Pedersen et al. (2005) for English WSD[4]. Following their terminology, these relatedness algorithms can be grouped into path-based, information-content-based, and gloss-based. The algorithms used in the present work in each of these categories are summarized in the following list. See Budanitsky and Hirst (2006) for a detailed description.

4.1.1. Path-Based Measures

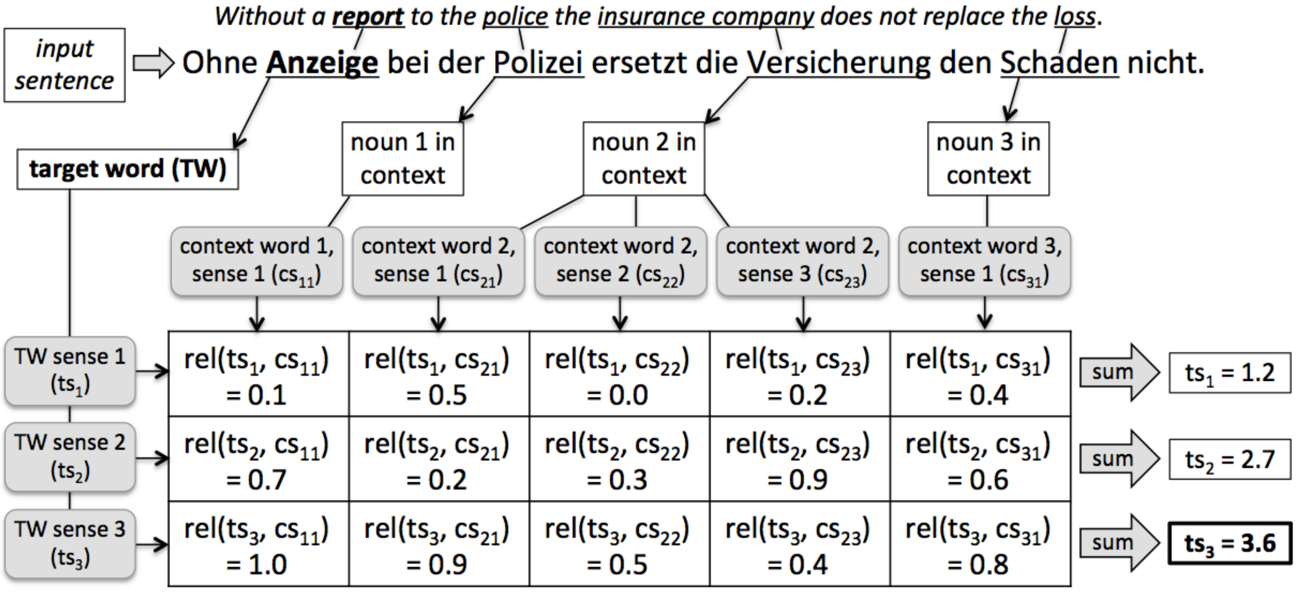
The following path-based measures all use the GermaNet graph-structure and compute the shortest path between two concepts contained in the graph.

- *lch*: Similarity between two concepts is computed as the negative logarithm of the length of the shortest path between the concepts (limited to hypernymy/hyponymy relations only) over the path length of the overall depth of the wordnet – as introduced by Leacock and Chodorow (1998).

- *wup*: Conceptual similarity between two concepts is computed as the shortest path length (limited to hypernymy/hyponymy relations) between the two concepts, normalized by the depth of their lowest common subsumer (LCS) – as introduced by Wu and Palmer (1994).

[3] http://link.springer.de/

[4] The publication of this paper will be accompanied by making this suite of semantic relatedness algorithms freely available for academic research – see http://www.sfs.uni-tuebingen.de/GermaNet/tools.shtml


Without a report to the police the insurance company does not replace the loss.
input sentence
Ohne Anzeige bei der Polizei ersetzt die Versicherung den Schaden nicht.
target word (TW)
noun 1 in context
noun 2 in context
noun 3 in context
context word 1, sense 1 (cs11)
context word 2, sense 1 (cs21)
context word 2, sense 2 (cs22)
context word 2, sense 3 (cs23)
context word 3, sense 1 (cs31)
TW sense 1 (ts1)
TW sense 2 (ts2)
TW sense 3 (ts3)
sum
ts1 = 1.2
ts2 = 2.7
ts3 = 3.6

TW sense 1 (ts_1)	$rel(ts_1, cs_{11}) = 0.1$	$rel(ts_1, cs_{21}) = 0.5$	$rel(ts_1, cs_{22}) = 0.0$	$rel(ts_1, cs_{23}) = 0.2$	$rel(ts_1, cs_{31}) = 0.4$
TW sense 2 (ts_2)	$rel(ts_2, cs_{11}) = 0.7$	$rel(ts_2, cs_{21}) = 0.2$	$rel(ts_2, cs_{22}) = 0.3$	$rel(ts_2, cs_{23}) = 0.9$	$rel(ts_2, cs_{31}) = 0.6$
TW sense 3 (ts_3)	$rel(ts_3, cs_{11}) = 1.0$	$rel(ts_3, cs_{21}) = 0.9$	$rel(ts_3, cs_{22}) = 0.5$	$rel(ts_3, cs_{23}) = 0.4$	$rel(ts_3, cs_{31}) = 0.8$

Figure 1: Algorithm for disambiguating a target word applied for each relatedness measure.

- *hso*: For computing the semantic relatedness between two concepts, the length of the shortest path between the concepts (not limited to the hypernymy/hyponymy relations) and the change of ”direction” (i.e., the relations in a wordnet can be grouped into upwards, downwards, and horizontal) are considered – as introduced by Hirst and St. Onge (1998).

- *path*: Relatedness between two concepts is computed as a function of the distance between two nodes and the longest possible ’shortest path’ between any two nodes in GermaNet.

4.1.2. Information-content-based Measures

These measures rely on the information content (IC) of a concept in the GermaNet graph that is estimated by the relative frequency of this word in a large corpus. In the present case, these frequencies were obtained from the TüPP-D/Z[5], a German newspaper corpus of 200 Mio. words.

- *res*: Similarity between two concepts is computed as the IC of their LCS in the graph – as introduced by Resnik (1995).

- *jcn*: Similarity between two concepts is computed as the inverse of their distance (measured as the sum of the ICs of the concepts minus double the IC of their LCS) – as introduced by Jiang and Conrath (1997).

- *lin*: Similarity between two concepts is measured as the IC (multiplied by two) of their LCS over the sum of the ICs of the concepts – as introduced by Lin (1998).

4.1.3. Gloss-based Measures

The following measures, based on the idea of Lesk (1986), use paraphrases from GermaNet and Wiktionary for counting word overlaps. The Wiktionary paraphrases are linked to corresponding GermaNet senses via the automatic mapping between GermaNet and Wiktionary described in Henrich et al. (2011). Furthermore, the use of lexical fields, i.e., bags of words from surrounding synsets, can be examined for calculating word overlaps. By the mapping from GermaNet to Wiktionary, such lexical fields can also be obtained for Wiktionary.

- *lesk-Gg*: Use glosses from GermaNet
- *lesk-Gw*: Use glosses from Wiktionary
- *lesk-Lg*: Use lexical fields from GermaNet
- *lesk-Lw*: Use lexical fields from Wiktionary
- *lesk-Ggw-Lgw*: Use glosses and lexical fields from both GermaNet and Wiktionary
- *lesk-Gw-Lg*: Use glosses from Wiktionary and lexical fields from GermaNet

4.2. WSD Using Semantic Relatedness Measures

For each relatedness measure taken in isolation, the word sense disambiguation of a polysemous target word in a given sentence starts with a calculation of semantic relatedness for all sense combinations in question as illustrated in Figure 1. That is, a relatedness measure $rel(ts, cs)$ is calculated for each sense ts_i of the target word to each sense cs_{jk} of each word j in the context window. The computed relatedness values are illustrated in the table in Figure 1.

In a next step, all calculated values per target word are summed and the target word yielding the highest sum is defined to be the overall disambiguation result returned by that relatedness measure. Formally, the overall WSD result of a single relatedness measure is defined as:

$$result_{rel} := \max_{ts \in T} \sum_{ts \in T, cs \in C} rel(ts, cs)$$

where T is the set of all target word senses and C the set containing all senses of all words in the context.

[5] http://www.sfs.uni-tuebingen.de/en/tuepp.shtml

4.3. Combined WSD Algorithms

It has been shown for various NLP tasks, including part-of-speech tagging (van Halteren et al., 2001; Henrich et al., 2009) and word sense disambiguation (Florian and Yarowsky, 2002), that multiple classifier systems outperform single decision systems. Further, the performance of such methods is usually better the more diverse the individual systems are (Polikar, 2006). Since the WSD algorithms used in the present paper are based on rather different underlying ideas, they are likely to produce diverse results. Therefore, combining them into a joint classifier appears like an obvious direction to pursue. In order to be able to combine the values of the individual algorithms into a joint overall score, the values returned by the single relatedness measures were normalized from zero to one.

In order to further boost the performance of a combined algorithm, we experimented with maximizing the precision of the single algorithms. To this end, modified variants of the single algorithms were developed by introducing thresholds that allow the algorithms to yield non-zero values only if their scores are above a certain value and therefore, in effect, introduce minimal levels of confidence. This, of course, leads to a loss in coverage and recall of the single algorithms. However, a combined algorithm can compensate for this loss, as long as the coverage of the single modified algorithms complement one another.

The following list summarizes some of the combined algorithms that we have experimented with and whose results will be shown in the Evaluation section (Polikar, 2006):

- *Majority voting*: Each relatedness measure votes for a candidate sense of the target word. The votes are summed with weight 1; and the target word sense with the highest count(s) win(s).

- *Weighted majority voting*: Each relatedness measure votes for a candidate sense of the target word. The votes are summed with a specified weight (per relatedness measure); and the target word sense with the highest count(s) win(s).

- *Borda count*: Each relatedness measure rankorders the candidate senses of the target word. The first ranked sense receives a value of $N-1$ (where N is the amount of senses the target word has), the second ranked sense gets $N-2$ votes, etc. Thus, the last ranked candidate sense receives 0. The values are summed for each target word sense; and the sense with the highest value(s) win(s).

5. Evaluation

5.1. The WebCAGe Sense-Annotated Corpus

All experiments recorded in this paper use WebCAGe[6] (Henrich et al., 2012), a web-harvested corpus annotated with GermaNet senses, which is based on the sense alignment of GermaNet senses with Wiktionary senses. As described in Section 2, the original purpose of this mapping was to automatically add sense descriptions to GermaNet. However, the alignment of these two resources opens up a much wider range of possibilities for data mining community-driven resources such as Wikipedia and web-generated content more generally. It is precisely this potential that was fully exploited for the creation of the WebCAGe sense-annotated corpus. Wiktionary senses are frequently illustrated by one or more example sentences, which in turn are often linked to external references, including Wikipedia articles[7], sentences contained in the Gutenberg project[8], and other textual web sources. Therefore, the GermaNet-Wiktionary alignment and the various pointers contained in Wiktionary example sentences make it possible to automatically assemble a corpus annotated with GermaNet senses. More specifically, WebCAGe contains a total of 1499 sense-annotated polysemous nouns (on average 2.6 senses in GermaNet) that occur 6847 times in 6413 sentences. These occurrences are distributed over the different subcorpora as follows: 4103 occurrences in the Wiktionary example sentences themselves, 1643 in the Wikipedia articles, 655 in the texts from the Gutenberg project, and 446 in the texts harvested from other web materials.

In order to evaluate the performance of the various algorithms, a 2:1 ratio between the training and testing data was used.[9] Thus, we randomly picked every third file from the text types of Wikipedia, Gutenberg, and external web sources, as well as every third example sentence from Wiktionary to be included in the test corpus. The remaining files/sentences constitute the training corpus.

5.2. Profiling the Suite of WSD Algorithms

In order to evaluate the above-described WSD setup, an extensive set of experiments using many different measures of relatedness and various algorithms for combining those individual results was performed. All experiments were run on the test set of WebCAGe (as described in Section 5.1.) and are restricted to the word class of nouns with a context window of 51 words. The reason for only considering nouns is due to the fact that – with the exception of *hso* and *lesk-** – all relatedness measures were intended for this word class only. Table 1 shows the results for the individual disambiguation results using one relatedness measure at a time as illustrated in Figure 1. The coverage (column *Cov.*) is calculated as the number of sentences, where the measure returns a result, compared to the overall number of sentences used for the evaluation.

Table 1 shows that, with the exception of *hso*, the path-based measures (rows 1, 2, and 4) and the information-content-based measures (rows 5 to 7) yield good results in terms of coverage ranging from 87.7% to 95.7%. By comparison, the gloss-based Lesk algorithms which use only GermaNet glosses (*lesk-Gg*) or Wiktionary glosses (*lesk-Gw*) score considerably lower in coverage and therefore also yield a recall which is below all of the path- and information-content-based measures (again with the exception of *hso*). This can only be due to the fact that the glosses do not contain enough lexical material. This defect can be remedied by including lexical material in the lexical fields

[6] See http://www.sfs.uni-tuebingen.de/en/webcage.shtml

[7] http://www.wikipedia.org/

[8] http://gutenberg.spiegel.de/

[9] The use of a 2:1 ratio is recommended by Agirre and Edmonds (2006, page 77) for evaluations of the WSD tasks.

Method	**Cov.**	**Recall**	**Prec.**	**F1**
lch	92.51%	47.17%	50.99%	49.01
wup	91.31%	49.16%	53.84%	51.40
hso	63.43%	41.35%	**65.20**%	50.61
path	95.70%	49.48%	51.71%	50.57
res	90.52%	45.66%	50.44%	47.93
jcn	93.86%	46.14%	49.15%	47.60
lin	87.73%	46.69%	53.22%	49.75
lesk-Gg	14.26%	5.90%	41.34%	10.32
lesk-Gw	76.33%	43.75%	57.31%	49.62
lesk-Lg	78.57%	49.40%	**62.88**%	55.33
lesk-Lw	97.93%	53.31%	54.43%	53.86
lesk-Ggw-Lgw	**99.76**%	53.71%	53.83%	53.77
lesk-Gw-Lg	92.83%	**54.34**%	58.54%	**56.36**

Table 1: Evaluation results for semantic relatedness measures.

obtained for both resources (*lesk-Lg* and *lesk-Lw*). There are considerable jumps in coverage from 14.26% (*lesk-Gg*) to 78.57% (*lesk-Lg*) and from 76.33% (*lesk-Gw*) to 97.93% (*lesk-Lw*). An almost complete coverage (99.76%) can be achieved by *lesk-Ggw-Lgw* which makes use of lexical fields and glosses from both resources. These dramatic jumps in coverage underscore the usefulness of enriching GermaNet with sense descriptions from Wiktionary for such gloss-based algorithms. This answers in the affirmative one of the leading questions for the research reported here, namely, to study the effect of the GermaNet-Wiktionary mapping on word overlap WSD algorithms.

Among the different variants of the Lesk algorithm discussed so far, the *lesk-Gw* and *lesk-Lg* stand out from the rest in terms of precision (57.31% and 62.88%, respectively). For this reason, we included a final variant *lesk-Gw-Lg* which achieves the overall best F-score of 56.36.[10] Notice also that two algorithms (*hso* and *lesk-Lg*) considerably outperform this best overall algorithm in terms of precision and another algorithm (*lesk-Ggw-Lgw*) in terms of coverage. This apparent heterogeneity of performance provides good motivation for investigating a combined WSD algorithm that can exploit the relative virtues of the best performing single algorithms. In order to further boost the performance of the combined algorithms, the precision of the single algorithms is maximized by introducing thresholds that allow the algorithms to yield non-zero values only if their scores are above a certain value – as explained in Section 4.

Method	**Cov.**	**Recall**	**Prec.**	**F1**
lch w. threshold	29.96%	20.00%	66.76%	30.78
wup w. thresh.	56.97%	35.78%	62.80%	45.58
hso w. threshold	39.20%	26.61%	67.89%	38.24
path w. thresh.	29.96%	20.00%	66.76%	30.78
res w. threshold	53.86%	30.76%	57.10%	39.98
jcn w. threshold	23.03%	12.75%	55.36%	20.73
lin w. threshold	39.60%	25.42%	64.19%	36.42

Table 2: Evaluation results for semantic relatedness measures with thresholds.

Table 2 shows the performance of those modified variants of the single algorithms.[11] The precision of these modified algorithms is much higher (6% to 16%) compared to the precision of the corresponding algorithms in Table 1. Only for *hso* the precision has not increased significantly (2.7% only). Further, there is an apparent loss in coverage, and therefore also in recall, which depends on coverage. This loss in coverage and recall of the single algorithms can be compensated in a combined algorithm as long as the coverage of the single algorithms complement one another.

The best results are achieved by combining all the single algorithms in Tables 1 and 2. Table 3 (rows 1 to 3) shows the results of the experiments with combined algorithms. The best overall result with an F-score of 63.59 is achieved by weighted majority voting (abbreviated as *wmv* in Table 3). The training set was used to obtain the optimal weights for the individual algorithms used in this wmv algorithm.

Method	**Cov.**	**Recall**	**Prec.**	**F1**
Majority voting	100%	56.73%	56.73%	56.73
wmv	100%	63.59%	63.59%	**63.59**
Borda count	100%	59.92%	59.92%	59.92
PPR	100%	49.96%	49.96%	49.96
Random	70.92%	18.96%	26.74%	22.19

Table 3: Evaluation results for combinatory algorithms and baselines.

As mentioned in Section 3, the state-of-the-art Personalized PageRank algorithm (Agirre and Soroa, 2009) (see row *PPR* in Table 3), which yielded the best results for German WSD reported by Broscheit et al. (2010), was also run on the test data. As shown in Table 3, the PPR algorithm performs considerably lower than any of the combined algorithms and many of the single algorithms described in Table 1. Finally, a random sense baseline (again, see Table 3) was outperformed by a wide margin by all algorithms listed in Tables 1 and 3, with the exception of *lesk-Gg*, in terms of recall and precision.

5.3. Profiling WSD for Individual Words

The purpose of this section is to further analyse the experimental results for the single and combined WSD algorithms presented in the previous section. To this end, we have chosen a set of 16 nouns which occur a minimum of 18 times in the test set. For reasons of space, it is not possible to show the performance of these words for all of the single and combined WSD algorithms discussed in the previous section. Therefore, only the best performing algorithm (in terms of F-measure) for each of the categories of *path-based*, *information-content-based*, *gloss-based*, and *com-*

[10] As mentioned above, this best result matches the findings of Pedersen et al. (2005) for English WSD where the Lesk algorithm also yielded the overall best results.

[11] Note that Table 2 does not include threshold variants of the Lesk algorithms, since the Lesk algorithms fail to consistently exhibit more reliable results above a certain threshold.

Word	Occurrences in test corpus	Senses		Methods			
		GN	C-GN	*wup*	*lin*	*lesk*	*wmv*
Brauerei	40	2	1	70.97	70.97	46.67	37.50
Dank	56	2	1	46.15	33.33	51.06	38.30
Schnecke	95	2	2	55.56	66.67	86.27	51.85
Option	124	2	2	52.05	71.43	51.35	75.68
Raclette	25	2	2	75.86	68.97	82.76	80.00
Araber	27	2	2	41.18	35.29	87.50	88.24
Eichel	34	2	2	32.00	32.00	56.00	92.31
Export	24	2	2	92.31	92.31	91.67	92.31
Elf	30	2	2	71.43	69.23	85.71	100
Besetzung	52	3	2	40.58	32.35	28.99	22.86
Steuer	71	3	2	56.10	65.85	35.29	90.70
Atrium	39	3	3	84.21	58.18	72.41	55.17
Ende	18	3	3	14.81	29.63	7.14	71.43
Aspiration	21	3	3	66.67	88.89	66.67	88.89
Bogen	58	4	4	34.48	50.91	34.48	68.97
Feld	79	5	3	48.89	45.45	45.83	4.17

Table 4: Individual evaluation results (F-score) for selected words.

bined are included: *wup*, *lin*, *lesk-Gw-Lg* (abbreviated as *lesk* in Table 4), and weighted majority voting (abbreviated as *wmv*). Table 4 summarizes the performance (F-score) of each of these algorithms for the individual words in question. Table 4 also lists for each word the occurrence count in the test set of WebCAGe (column *occurrences*) and the number of senses in GermaNet (column *GN*).

The column *C-GN* in Table 4 lists the number of coarse-grained GermaNet senses that can be obtained by clustering the existing sense distinctions contained in GermaNet. The motivation for such a clustering is the following: In the past, it has been called into question whether wordnets constitute the right type of resource to provide sense inventories for word sense disambiguation. This critique has been levied particularly against the Princeton WordNet for English since it makes very fine sense distinctions (particularly for words with many wordnet senses) which cannot reliably be replicated by human annotators with sufficient inter-annotator agreement (Palmer et al., 2007). The lack of reliable inter-annotator agreement for such words seems troublesome since it is unreasonable to expect that an automatic WSD algorithm should perform better than humans would on the same task. This insight has led to clustering fine-grained WordNet sense distinction into coarse-grained sense groupings, either by manually reconciling differences in human annotator judgements (Palmer et al., 2007) or by an automatic mapping from WordNet senses to more coarse-grained senses provided by the Oxford Dictionary for English (Navigli, 2006). Moreover, Palmer et al. (2007) have shown for the English Verb Lexical Sample Task of SensEval-2 that the WSD performance improves from an accuracy of 77.1% for fine-grained sense distinctions to an accuracy of 81.4% for coarse-grained sense distinctions. Navigli (2006) showed an improvement from 65% to 78% for the SensEval-3 all words task, using the WSD system Gambl (Decadt et al., 2004), which scored the best for this SensEval-3 shared task.

The average ambiguity rate of 2.6 GermaNet senses for polysemous nouns in WebCAGe is slightly lower compared to the Princeton WordNet (average of 2.79 noun senses for version 3.0). However, manual inspection of the words profiled in Table 4 has shown that their sense distinctions sometimes suffer from the same kinds of deficiencies witnessed by the human annotators when they used the Princeton WordNet for manual sense-annotation of English texts. One such deficiency described in Palmer et al. (2007) concerns *sense subsumption*, which involves the choice between a more general or a more specific sense entry.

An example of this sort is the noun *Steuer*, which has a total of three senses in GermaNet with the following paraphrases that illustrate these senses: (1) *nicht zweckgebundene Abgabe an den Staat* 'general purpose tax paid to the government', (2) *Vorrichtung zum Lenken eines Fahrzeuges* 'device for steering a vehicle', and (3) *Vorrichtung zum Steuern von Fahrzeugen* 'device for driving vehicles'. Sense (1) is clearly distinct from the other two senses. However, senses (2) and (3) have nearly identical paraphrases and are thus closely related. It is therefore sensible to collapse these two senses into one. As a result, *Steuer* then has the two coarse-grained senses described by paraphrases (1) and (2) above.

Another deficiency of fine-grained sense distinctions discussed by Palmer et al. (2007) concerns *vague contexts*, that is, contexts that are applicable to more than one sense of a target word. An example of this sort is the noun *Besetzung*, which has a total of three senses in GermaNet with the following paraphrases that illustrate these senses: (1) *Zuteilung einer Stelle, eines Postens, einer Rolle an jemanden* 'assignment of a position, post, role to someone', (2) *Militär: die Stationierung von Truppen in einem fremden Gebiet* 'military: stationing of troops in a foreign territory', and (3) *Gesamtheit der Künstler (Schauspieler, Musiker) eines Werkes* 'ensemble of artists (actors, musicians) for the performance of a piece of art'. Here, senses (1) and

(3), while distinct, are at the same time clearly overlapping in coverage. This leads to the dilemma, that for the disambiguation of a target word the contexts will often not sufficiently discriminate between the senses in question. For example, in the sentence *Chris ergänzt die jetzige Besetzung des Theaters optimal.* 'Chris is the ideal addition to the current team composition of the theater.' it is unclear whether *Besetzung* refers to the set of jobs in the theater (sense 1) or to the cast for a particular theater play (sense 3). As in the case of sense subsumption it therefore makes sense to collapse senses (1) and (3) and assign two coarse-grained senses to the word *Besetzung*.

The WSD experiments for the words in Table 4 were carried out with the ordinary GermaNet senses (listed under *GN*). The reason for also including the coarse-grained senses *C-GN* in the table is to be able to distinguish those words where the number of *GN* and *C-GN* senses coincides from the ones where two or more of the fine-grained senses can be collapsed. This distinction turns out to be a good indicator for differences in WSD scores obtained by the overall best scoring *wmv* combined algorithm. With one exception, the words where two or more senses can be collapsed (i.e. *C-GN* < *GN*, for *Brauerei*, *Dank*, *Besetzung*, and *Feld* in Table 4), performed lower for this combined algorithm in comparison with those words where no senses could be collapsed (i.e., *C-GN* = *GN*). The exception is *Steuer* (*C-GN=2* < *GN=3*), for which wmv anomalously performs extremely well. This result corroborates the findings of Navigli (2006) and Palmer et al. (2007) that fine-grained sense distinctions lead to worse results than coarse-grained ones.

The results in Table 4 also corroborate the finding described in Section 5.2 that the individual algorithms exhibit a rather heterogeneous behavior and deviate in their performance on a word-to-word basis, thus confirming the usefulness of applying combined algorithms to the WSD task.

6. Conclusion and Future Work

In summary, the present paper has explored a wide range of WSD algorithms for German. Among the single algorithms considered, a variant of the Lesk algorithm (*lesk-Gw-Lg*) that uses Wiktionary glosses and GermaNet lexical fields yields the best F-score of 56.36. Since the individual algorithms produce diverse results in terms of precision that complement each other well in terms of coverage, a set of combined algorithms outperform the score of the best individual classifier. The best overall result is obtained by a combined WSD algorithm that uses weighted majority voting and yields an F-score of 63.59. This result contradicts the previous finding of Broscheit et al. (2010) who did not obtain better results by combining individual WSD algorithms. The present study also applied the Personalized PageRank individual classifier which performed best overall in the study reported by Broscheit et al. but not in the experiments reported here.

The WSD experiments also confirm that WSD performance is lower for words with fine-grained sense distinctions compared to words with coarse-grained senses. In future work, we plan to rerun the experiments reported here with a coarse-grained GermaNet sense inventory in order to ascertain whether this will lead to improved results for German, as Navigli (2006) and Palmer et al. (2007) reported for English.

7. Acknowledgements

The research reported in this paper was supported by the SFB 833 grant by the German Research Council (DFG) to the University of Tübingen. We would like to thank Chris Culy and three anonymous LREC 2012 reviewers for their helpful comments on earlier versions of this paper. We are very grateful to Anne Brock for her help with the implementation of the semantic relatedness measures and to Tatiana Vodolazova for her significant contributions to WebCAGe, the sense-annotated corpus used in the present paper.

8. References

E. Agirre and P. Edmonds, editors. 2006. *Word Sense Disambiguation: Algorithms and Applications*. Springer.

E. Agirre and A. Soroa. 2009. Personalizing PageRank for Word Sense Disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pages 33–41, Stroudsburg, PA, USA. Association for Computational Linguistics.

S. Broscheit, A. Frank, D. Jehle, S. P. Ponzetto, D. Rehl, A. Summa, K. Suttner, and S. Vola. 2010. Rapid bootstrapping of Word Sense Disambiguation resources for German. In *Proceedings of the 10. Konferenz zur Verarbeitung Natürlicher Sprache*, pages 19–27, Saarbrücken, Germany.

A. Budanitsky and G. Hirst. 2006. Evaluating WordNet-based Measures of Lexical Semantic Relatedness. *Computational Linguistics*, 32:13–47.

B. Decadt, V. Hoste, W. Daelemans, and A.V. den Bosch. 2004. GAMBL, Genetic Algorithm Optimization of Memory-Based WSD. In Rada Mihalcea and Phil Edmonds, editors, *Proceedings of the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (Senseval-3)*, pages 108–112.

C. Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

R. Florian and D. Yarowsky. 2002. Modeling Consensus: Classifier Combination for Word Sense Disambiguation. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, EMNLP '02, pages 25–32, Stroudsburg, PA, USA. Association for Computational Linguistics.

V. Henrich and E. Hinrichs. 2010. GernEdiT - The GermaNet Editing Tool. In *Proceedings of the ACL 2010 System Demonstrations*, pages 19–24, Uppsala, Sweden. Association for Computational Linguistics.

V. Henrich, T. Reuter, and H. Loftsson. 2009. CombiTagger: A System for Developing Combined Taggers. In H. Chad Lane and Hans W. Guesgen, editors, *FLAIRS Conference*. AAAI Press.

V. Henrich, E. Hinrichs, and T. Vodolazova. 2011. Semi-Automatic Extension of GermaNet with Sense Definitions from Wiktionary. In *Proceedings of the 5th Language & Technology Conference: Human Language*

Technologies as a Challenge for Computer Science and Linguistics, LTC '11, pages 126–130, Poznan, Poland.

V. Henrich, E. Hinrichs, and T. Vodolazova. 2012. WebCAGe – A Web-Harvested Corpus Annotated with GermaNet Senses. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, page to appear, Avignon, France.

G. Hirst and D. St-Onge. 1998. Lexical Chains as Representations of Context for the Detection and Correction of Malapropisms. In C. Fellbaum, editor, *WordNet: An Electronic Lexical Database*, pages 305–332. MIT Press, Cambrige, MA.

J.J. Jiang and D.W. Conrath. 1997. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In *Proceedings of the International Conference on Research in Computational Linguistics*, ROCLING X, pages 19–33, Taiwan.

C. Kunze and L. Lemnitzer. 2002. GermaNet representation, visualization, application. In *Proceedings of the 3rd International Language Resources and Evaluation*, LREC '02, pages 1485–1491, Las Palmas, Canary Islands.

M. Lapata and F. Keller. 2007. An Information Retrieval Approach to Sense Ranking. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 348–355, Rochester, New York. Association for Computational Linguistics.

C. Leacock and M. Chodorow. 1998. Combining Local Context and WordNet Similarity for Word Sense Identification. In C. Fellbaum, editor, *WordNet: An Electronic Lexical Database*, pages 305–332. MIT Press.

M. Lesk. 1986. Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. In *Proceedings of the 5th annual international conference on Systems documentation*, SIGDOC '86, pages 24–26, New York, NY, USA. ACM.

D. Lin. 1998. An Information-Theoretic Definition of Similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 296–304, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

D. McCarthy, R. Koeling, J. Weeds, and J. Carroll. 2004. Finding Predominant Word Senses in Untagged Text. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 279–286, Barcelona, Spain.

R. Navigli and M. Lapata. 2010. An Experimental Study of Graph Connectivity for Unsupervised Word Sense Disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:678–692.

R. Navigli. 2006. Meaningful Clustering of Senses. Helps Boost Word Sense Disambiguation Performance. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 105–112, Stroudsburg, PA, USA. Association for Computational Linguistics.

R. Navigli. 2007. *Structural Semantic Interconnections: a Knowledge-Based WSD Algorithm, its Evaluation and Applications*. Ph.D. thesis, University of Rome "La Sapienza".

R. Navigli. 2009. Word Sense Disambiguation: a Survey. *ACM Computing Surveys*, 41(2):1–69.

M. Palmer, H.T. Dang, and C. Fellbaum. 2007. Making fine-grained and coarse-grained sense distinctions, both manually and automatically. *Natural Language Engineering*, 13(2):137–163.

T. Pedersen, S. Banerjee, and S. Patwardhan. 2005. Maximizing Semantic Relatedness to Perform Word Sense Disambiguation. Research Report UMSI 2005/25, University of Minnesota Supercomputing Institute.

R. Polikar. 2006. Ensemble Based Systems in Decision Making. *IEEE Circuits and Systems Magazine*, 6(3):21–45.

P. Resnik. 1995. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 1*, pages 448–453, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

D. Steffen, B. Sacaleanu, and P. Buitelaar. 2003. Domain Specific Sense Disambiguation with Unsupervised Methods. In C. Kunze, L. Lemnitzer, and A. Wagner, editors, *Anwendungen des deutschen Wortnetzes in Theorie und Praxis. Tagungsband des 1. GermaNet-Workshops des GLDV-AK Lexikografie*, pages 79–86.

H. van Halteren, W. Daelemans, and J. Zavrel. 2001. Improving Accuracy in Word Class Tagging through the Combination of Machine Learning Systems. *Computational Linguistics*, 27:199–229.

D. Widdows, S. Peters, S. Cederberg, C.-K. Chan, D. Steffen, and P. Buitelaar. 2003. Unsupervised monolingual and bilingual word-sense disambiguation of medical documents using umls. In *Proceedings of the ACL 2003 workshop on Natural language processing in biomedicine - Volume 13*, BioMed '03, pages 9–16, Stroudsburg, PA, USA. Association for Computational Linguistics.

Z. Wu and M. Palmer. 1994. Verb Semantics And Lexical Selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, Stroudsburg, PA, USA. Association for Computational Linguistics.

D. Yarowsky. 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, ACL '95, pages 189–196, Stroudsburg, PA, USA. Association for Computational Linguistics.