Metaheuristic Approaches to Lexical Substitution and Simplification

Sallam Abualhaija
Institute of Computer Technology
Hamburg University of Technology
sallam.abualhaija@tu-harburg.de

Tristan Miller and **Judith Eckle-Kohler** and **Iryna Gurevych**
Ubiquitous Knowledge Processing Lab (UKP-TUDA/UKP-DIPF/AIPHES)
Department of Computer Science
Technische Universität Darmstadt
https://www.ukp.tu-darmstadt.de/

Karl-Heinz Zimmermann
Institute of Computer Technology
Hamburg University of Technology
K.Zimmermann@tu-harburg.de

Abstract

In this paper, we propose using metaheuristics—in particular, simulated annealing and the new D-Bees algorithm—to solve word sense disambiguation as an optimization problem within a knowledge-based lexical substitution system. We are the first to perform such an extrinsic evaluation of metaheuristics, for which we use two standard lexical substitution datasets, one English and one German. We find that D-Bees has robust performance for both languages, and performs better than simulated annealing, though both achieve good results. Moreover, the D-Bees–based lexical substitution system outperforms state-of-the-art systems on several evaluation metrics. We also show that D-Bees achieves competitive performance in lexical simplification, a variant of lexical substitution.

1 Introduction

Lexical substitution is a special case of automatic paraphrasing in which the goal is to provide contextually appropriate replacements for a given word, such that the overall meaning of the context is maintained. The task has applications in question answering, text summarization, sentence compression, information extraction, machine translation, and natural language generation (Androutsopoulos and Malakasiotis, 2010). It is also frequently employed as an *in vivo* evaluation of word sense disambiguation (WSD) systems (McCarthy and Navigli, 2009; Toral, 2009; Miller et al., 2015), because while lexical substitution requires words to be sense-disambiguated, it does not impose use of a predefined sense inventory.

Past work in WSD, whether or not it forms part of a lexical substitution system, has employed a wide range of approaches (Agirre and Edmonds, 2007). Supervised methods usually achieve the best results, but at the tremendous cost of producing manually annotated training data specific to the language and domain. Knowledge-based and unsupervised methods rely only on pre-existing resources such as machine-readable dictionaries and raw corpora. Though generally less accurate, they have the advantage of being more flexible and more adaptable to new languages and domains. For knowledge-based methods, this has been especially true since the advent of large, multilingual, collaboratively constructed resources such as Wikipedia and Wiktionary (Zesch et al., 2008).

In this paper, we present two novel approaches to lexical substitution which are knowledge-based, generally language-independent, and use a combination of traditional wordnets and Wiktionary. The first approach uses simulated annealing (Kirkpatrick et al., 1983), which was first proposed for use in WSD by Cowie et al. (1992) but has attracted relatively little attention since then. The second approach uses D-Bees (Abualhaija and Zimmermann, 2016), a relatively new, biologically inspired disambiguation algorithm that models swarm intelligence. Both algorithms are *metaheuristic* (Talbi, 2009) in that they treat WSD as an optimization problem and modify heuristic (approximate) solu-

tions to avoid entrapment in local optima. Ours is the first extrinsic evaluation of any metaheuristic approaches to WSD in a lexical substitution setting.

We evaluate and compare both approaches on two lexical substitution datasets, one English and one German. We find that both approaches perform well, with D-Bees in particular exceeding state-of-the-art performance in many tasks. We also apply the systems to lexical simplification, a variant of lexical substitution in which the goal is to provide substitutes which are easier to understand. Here, too, we find that D-Bees performs near or above the state of the art.

2 Background

2.1 Lexical Substitution and Simplification

In lexical substitution, a system is given a word in context and tasked with producing a list of words that could be substituted for the word without altering the overall meaning. For example, given the word "bright" in the sentence "Einstein was a bright man," valid substitutes would include "sharp" and "intelligent", but not "shiny" or "luminous", even though the latter two are synonymous with "bright" in other contexts. It is generally expected that the list of substitutes be ordered by acceptability. Most lexical substitution systems therefore comprise two distinct phases: *generation*, in which the system assembles a set of suitable substitutes for the target word, and *ranking*, in which the system orders them according to how well they fit the context.

There have been a number of organized evaluation campaigns for lexical substitution systems, including the English-language task at SemEval-2007 (McCarthy and Navigli, 2009) and the German task at GermEval 2015 (Miller et al., 2015). These campaigns provide standardized datasets where a large number of word–context combinations have been manually annotated with acceptable substitutes. Systems are evaluated by comparing their output to this gold standard, using any or all of three scoring methodologies:

- In the *best* methodology (McCarthy and Navigli, 2009), systems are allowed to suggest as many substitutes as they wish. However, the credit for each guess is normalized by the total number of guesses. The best guess should be placed first in the list. Across the entire dataset, four metrics are calculated: recall (R), mode recall (R_m), precision (P), and mode precision (P_m).[1]
- In *out of ten (OOT)* (McCarthy and Navigli, 2009), systems suggest up to ten substitutes, though neither the exact number nor the order of these is important. This methodology uses minor variations of *best*'s R, R_m, P, and P_m.
- *Generalized average precision (GAP)* (Kishida, 2005) uses a single metric to score a fully ranked list of substitutes. Unlike *OOT*, *GAP* is sensitive to the relative positions of the correct and incorrect substitutes in the list.

For reasons of space, we do not provide detailed explanations and formulas for the nine metrics, but refer readers to the cited papers.

Lexical simplification is a variant of lexical substitution in which the correct ranking is determined not just by the substitutes' contextual fitness but also by their simplicity. (For example, rare words are generally considered to be more complex, as readers are less likely to be familiar with their meanings.) As with other types of text simplification, lexical simplification can be used to make complex texts understandable by a wider range of readers, such as children or second language learners.

To date there has been one shared task in lexical simplification (Specia et al., 2012). Its main evaluation metric is based on Cohen's (1960) κ. Two post-hoc evaluation metrics are also used. The first, *top-ranked (TRnk)*, evaluates the simplest set of substitutes that is ranked first by the system, compared with the top-ranked set of substitutes in the gold standard. This represents the intersection between the first substitute set found by the system with the first set in the gold standard. The intersection should include at least one substitute. The second metric, *recall at n (R@n)* is the ratio of candidates from the top n sets of substitutes to those in the gold standard, where $1 \leq n \leq 3$. For a given n, the contexts with at least $n+1$ substitutes in the gold standard are considered.

2.2 Word Sense Disambiguation, Optimization, and Metaheuristics

Word sense disambiguation, the task of determining which of a word's meanings is the one intended in a given context, is a prerequisite for generating substitutes in knowledge-based lexical substitution.

[1] These metrics are inspired by, but distinct from, the traditional recall and precision metrics from information retrieval.

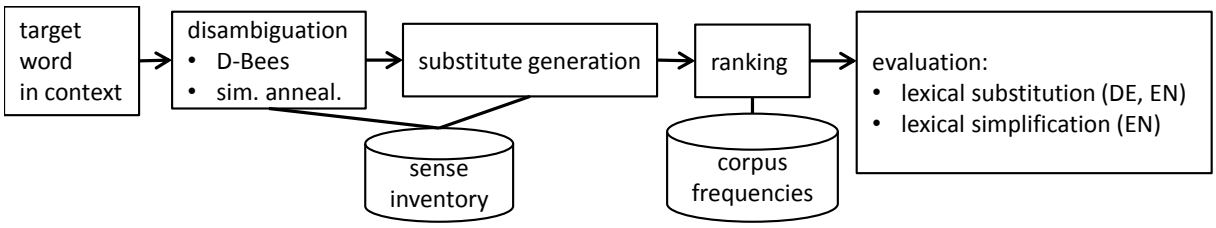

Figure 1: Our approach to lexical substitution and simplification of a target word in context.

There are many different approaches to WSD; for our purposes it is convenient to define it as an optimization problem where the aim is to disambiguate a sequence of words simultaneously (Abualhaija and Zimmermann, 2016): Let $W = (w_1, w_2, \ldots, w_n)$ be a sequence of n words to be disambiguated, and $\sigma = (s_1, s_2, \ldots, s_n)$ the corresponding sequence of senses for each word. Let $\mathscr{S} = \{\sigma_1, \ldots, \sigma_m\}$ be the set of all sequences of senses that represent sense combinations of the words in W. Then the objective function is $\arg\max_{\sigma \in \mathscr{S}} \ell(\sigma)$, where ℓ is the score assigned to a sequence of senses according to some measure of semantic similarity, such as those surveyed by Zesch and Gurevych (2010).

WSD as an optimization problem is NP-hard. This can be worked around by using metaheuristics, which are approximate, tractable algorithms that find near-optimal solutions. Metaheuristics can be *single-solution* and *population-based* search methods. The former manipulate and transform a single solution, giving more focus to the promising regions. Population-based methods work on multiple solutions, distributing their focus and exploring several regions of the search space simultaneously.

3 Approach

We investigate two knowledge-based, language-independent approaches to lexical substitution, whose main difference lies in the metaheuristic WSD component preceding the generation phase. Both approaches use a top-down generation process, in which the target word is first disambiguated in context with respect to a particular sense inventory, and then used to suggest a list of substitutes.[2] In the following subsections, we describe the two disambiguation components and the common substitute generation and ranking components. (See overview in Figure 1.)

[2]This contrasts with a bottom-up approach, where a list of all possible substitutes for the target word is first generated and then filtered to suit the context.

3.1 Disambiguation with Simulated Annealing

Simulated annealing (Kirkpatrick et al., 1983) is a single-solution algorithm in which a randomly created solution is iteratively modified until a "good-enough" solution is found. To apply it to WSD, we use essentially the same setup as Cowie et al. (1992). We start with a randomly initialized *sense combination* $\sigma_0 = (s_1, s_2, \ldots, s_n)$ from a given sense inventory, for each word in the context. We then retrieve the glosses for each sense, preprocess them via lemmatization and stop word removal, and give each remaining term a score of $n-1$ if it appears n times. We calculate the configuration's *redundancy*, R_0, by summing up all the scores. In other words, R_0 is the lexical overlap between sense definitions. The aim of simulated annealing is to maximize this overlap, or more precisely to minimize the *energy* function $E_i = 1/(1+R_i)$ in each iteration i.

In this iterative process, each iteration makes a random change on the configuration σ_i to produce σ_{i+1}, on which the corresponding E_i is computed. If $E_{i+1} < E_i$ (i.e., $\Delta E < 0$), then the new configuration replaces the old configuration for the next iteration. Otherwise, the new configuration might still be accepted with probability $\Pr = \exp(-\Delta E/T)$, where T is initially set to 1 but replaced with $0.9T$ for each subsequent iteration. This way, the algorithm risks exploring poor-looking paths that might nonetheless yield better results in the long run, and the earlier the iterations are, the greater the probability that a poor path is followed. In our experiments we iterate up to 30 times.

3.2 Disambiguation with D-Bees

D-Bees (Abualhaija and Zimmermann, 2016) is a population-based algorithm inspired by bee colony optimization (BCO) (Teodorović, 2009). BCO models the foraging behaviour of honey bees, where thousands of individuals with limited knowledge collaborate to maximize their collective bene-

fit. In nature, bees fly around their hive to look for nectar and pollen. When they find it, they return to the hive and perform a dance to advertise its location and quality to the others. The observers then decide whether to remain committed to their own path or to abandon it in favour of one of the advertised paths. BCO simulates this method through a multi-agent decentralized system.

D-Bees starts by choosing one of the target words as the hive, which spawns bee agents and sends them to other words in the context. The number of bee agents equals the number of candidate senses of the hive; each bee agent starts off with one of these senses in its memory. For each word it visits, the bee disambiguates it by randomly selecting a candidate sense, building up a path of senses and maintaining a running total similarity score. This *forward pass* continues until a set number of moves is reached.

The bee then makes a *backward pass* to the hive and exchanges its partial solution with the other agents on the virtual dancing floor. Each bee then determines whether it should stick to its path or adopt that of another bee; this is accomplished through *loyalty* and *recruiting probability* functions that depend mainly on the quality of the partial solutions. On the next forward pass, the bees resume their searches from the ends of their chosen paths. The forward and backward passes are alternated until there are no more words to be disambiguated. The bee agent with the best solution determines the final sense labelling of all words in the context.

In experiments on separate tuning datasets, we determined the number of moves in the forward pass to be one-third the number of context words. For the calculation of semantic similarity, we use a variant of the adapted Lesk algorithm (Banerjee and Pedersen, 2002). For each sense, we build a textual representation by concatenating its gloss with those of its hyper- and hyponyms. We then calculate the lexical overlap between the two texts.

3.3 Substitute Generation

Once the target word is disambiguated with respect to a particular sense inventory, we generate an unordered list of substitutes (to be subsequently ordered by the ranking module). The sense inventory we use for disambiguation is WordNet 3.1 (Fellbaum, 1998) for our English tasks, and GermaNet 10.0 (Hamp and Feldweg, 1997; Henrich and Hinrichs, 2010) for the German one. These are expert-built resources in which words representing the same concept are grouped together into *synsets*; synsets are in turn linked into a network by semantic relations such as hypernymy and meronymy.

In preliminary experiments on generating substitutes, we varied two independent parameters: which lexical-semantic resources to use as the source of substitutes, and which semantic relations to follow from the disambiguated synset.

With respect to the first parameter, we tried drawing substitutes from the disambiguation inventory (WordNet or GermaNet) alone, and also drawing additional substitutes from Wiktionary. Our use of Wiktionary as a complementary resource is motivated by Meyer and Gurevych (2012), who found its coverage to be complementary to those of expert-built resources, and by Henrich and Hinrichs (2012), who found that using information from both GermaNet and Wiktionary improved WSD performance. We used a relatively simple, Lesk-like method for mapping senses from WordNet/GermaNet to Wiktionary.

For the second parameter, we tried one setup in which we took all synonyms found in the disambiguated synset and in its hypernyms, and one in which we additionally pulled in synonyms from the hyponyms and all other related synsets (except antonyms). The first setup was informed by the annotation guidelines of the lexical substitution datasets, which indicate that it is permissible to suggest substitute terms that are more generic but not more specific. The second setup was informed by the analyses of Kremer et al. (2014) and Miller et al. (2016), which found, contrarily, that other semantic relations, including hyponyms, were a fruitful source of substitutes.

We obtained the best overall results when using both WordNet/GermaNet and Wiktionary, and when following semantic relations of all types (other than antonymy), to build the substitute list. We therefore used this setup for all our lexical substitution and simplification experiments.

3.4 Ranking

The final step of lexical substitution is to rank the substitutes. Our method, like those employed in previous lexical substitution tasks, assumes that a substitute's suitability depends on the type of its semantic relation to the target word. We therefore order the substitutes as follows: synonyms, hypernyms, hyponyms, other relations. Within

each semantic relation type, we sort the substitutes first by source (first WordNet/GermaNet, then Wiktionary), and then secondarily by reverse frequency in a large corpus. In preliminary experiments, we found that this method was generally better than simply sorting the entire substitute list by reverse frequency. To determine lemma frequency, we use the same frequency lists used to construct the original datasets: WaCky (Baroni et al., 2009) for German, and BNC (Burnard, 2007) for English.

4 Lexical Substitution for German

4.1 Dataset and Baselines

In our experiments, we use the data from GermEval 2015 (Miller et al., 2015), a shared task for German-language lexical substitution. It is split into a training and a test set of 1040 and 1000 sentences from the German edition of Wikipedia. Each sentence in the dataset contains one of 75 unique target words (25 nouns, 25 verbs, and 25 adjectives); in the test set, ten sentences are provided for each of the nouns and adjectives, and twenty for each verb.

Miller et al. (2015) report results of several naïve baselines, the best-performing of which are *weighted sense* (Toral, 2009) and *top-ranked synonym* (McCarthy and Navigli, 2009). Neither baseline makes any attempt to disambiguate the target word; rather, they build a substitute list by gathering synonyms of all possible senses of the target, as well as synonyms of closely related senses such as hypernyms, and then ranking these words by their frequency (either within the list itself or in a large corpus). We consider these two naïve baselines as reasonable lower bounds.

The more challenging baseline performance comes from the best-performing participating systems at GermEval 2015, which represent the state of the art in German-language lexical substitution. One of these systems (Hintz and Biemann, 2015) is a supervised, bottom-up approach inspired by previous English-language work by Szarvas et al. (2013a). It first retrieves a list of substitutes from various lexicons, then applies a maxent classifier to determine whether each substitute fits the context. The second system (Jackov, 2015) is based on techniques from machine translation. It first disambiguates the input text by mapping German words to concepts represented by WordNet synsets. It then produces and scores various parsing hypotheses, and selects the synonyms and hypernyms of the target in the best-scoring hypothesis.

4.2 Results and Analysis

Table 1 shows the results of the baselines described above, along with those of our basic D-Bees- and simulated annealing–based systems, and an enhanced version of the D-Bees system that we describe below.[3] Both our basic systems outperform the prior state of the art for the four *OOT* metrics, with the D-Bees–based system performing slightly better than the one using simulated annealing. However, neither system was able to beat Hintz and Biemann (2015) for the *GAP* and *best* metrics.

In light of this gap, we modified the D-Bees–based system to account for some idiosyncrasies of our German-language resources:

- Where GermaNet provided additional spellings of a synonym (e.g., "*wacklig*" for "*wackelig*"), we placed the variant spellings at the end of the substitute list. This prevented the top ranks of the list from being overloaded with nearly identical terms.

- Where our resources provided gender-specific variants of a synonym, we filtered out those that did not match the gender of the target. For example, when building the substitute list for "*Meisterin*" (female champion), we exclude "*Meister*" (male champion), even though GermaNet lists it as a synonym.

- To control for Wiktionary's lack of consistency, we filtered out Wiktionary-derived synonyms where the synonymy relation was not symmetric. For example, the Wiktionary entry for "*Likör*" gives "*Crème*" as a synonym, but the entry for "*Crème*" does not give "*Likör*", so when building a substitute list for "*Likör*", we do not include "*Crème*".

With these resource-specific enhancements, the D-Bees system achieves state-of-the-art performance not only for *OOT* but also for *GAP*, and performs only slightly worse than Hintz and Biemann (2015) for *best*. (This is an impressive result considering that Hintz and Biemann (2015) is a supervised system while ours is based solely on external knowledge bases and does not require any training data.) We also examined its performance by part of speech. We found that it remains the

[3] Here, as well as in §5, we report results on the test split and used the training split for tuning our algorithms.

Table 1: System performance on the GermEval 2015 lexical substitution dataset.

System	Best				OOT				GAP
	P	*R*	P_m	R_m	*P*	*R*	P_m	R_m	
D-Bees	7.66	7.66	14.85	14.85	20.68	20.68	37.73	37.73	12.94
D-Bees (enhanced)	10.39	10.39	22.39	22.39	**21.88**	**21.88**	**39.64**	**39.64**	**16.40**
simulated annealing	9.40	9.40	19.67	19.67	19.95	19.95	36.16	36.16	14.34
Hintz and Biemann (2015)	**11.20**	**11.10**	**24.28**	**24.21**	19.49	19.31	33.99	33.89	15.96
Jackov (2015)	6.73	6.45	13.36	12.86	20.14	19.32	33.18	31.92	11.26
top-ranked synonyms	10.04	10.04	19.82	19.82	15.21	15.21	27.99	27.99	12.25
weighted sense	7.50	7.50	13.46	13.46	20.54	20.54	35.55	35.55	14.28

best-performing system for *GAP* across all parts of speech, and for nouns and verbs is able to match or exceed Hintz and Biemann (2015) on some *best* metrics.

5 Lexical Substitution for English

5.1 Dataset and Baselines

Our English-language data is taken from the SemEval-2007 shared task (McCarthy and Navigli, 2009). That task uses a sample of 201 target words (nouns, verbs, adjectives and adverbs); for each word, ten context sentences are selected from the English Internet Corpus (Sharoff, 2006). Five human annotators provided up to three substitutes for each target. The dataset is split into a training set (300 sentences) and a test set (1710 sentences).

McCarthy and Navigli (2009) provide results for the aforementioned "top-ranked synonyms" algorithm as a lower bound on performance. State-of-the-art performance across the nine evaluation metrics is represented by the top-performing systems at SemEval-2007 (Giuliano et al., 2007; Hassan et al., 2007; Yuret, 2007; Zhao et al., 2007) and by several later systems (Biemann and Riedl, 2013; Melamud et al., 2015).[4] Of these systems, only Yuret (2007) is supervised.

5.2 Results and Analysis

Table 2 shows the results for the state-of-the-art and naïve baselines, along with results of our two basic systems and, as before, an enhanced version of the D-Bees system. Our systems' performance is generally much lower here than on the German-language data, with D-Bees failing to exceed the state of the art.

As with our German experiments, we tried modifying the D-Bees–based system to work around the language-specific problems we observed. The most significant of these adaptations are as follows:

- Our analysis suggested that WordNet's notoriously fine sense granularity was adversely affecting the WSD process. We therefore modified D-Bees to perform "soft" WSD (Ramakrishnan et al., 2004), meaning that we allow it to select several different senses as the correct ones—in our case, up to five. To compensate for the larger number of substitution candidates, we limit the ranked list of substitutes to 20. (This hearkens back to the bottom-up approaches defined in §3.) Substitutes generated from the best disambiguation solution are ranked highest.

- In contrast to German, English lexical substitutes are often drawn from indirect hypernyms (Kremer et al., 2014; Miller et al., 2016). (This too may be an artifact of WordNet's fine granularity.) We therefore extended our substitute search to two levels of hypernyms.

- The glosses provided by WordNet sometimes consist of a list of equivalent terms which do not appear in the list of synonyms. For example, WordNet defines one sense of the adverb "right" as "precisely, exactly", though it does not actually list those words as synonyms. We therefore include as the lowest-ranked substitutes those words from the target's gloss that match its part of speech.

[4] We are aware of several further lexical substitution systems (Moon and Erk (2013), Ó Séaghdha and Korhonen (2014), Roller and Erk (2016), Sinha and Mihalcea (2011), Szarvas et al. (2013b), and Thater et al. (2010) as reimplemented by Kremer et al. (2014)), though they do not report results on the full SemEval-2007 test set, or else do not report any of the same metrics we do, or else are concerned only with ranking but not generating substitutes.

Table 2: System performance on the SemEval-2007 lexical substitution dataset.

System	Best				OOT				GAP
	P	*R*	P_m	R_m	*P*	*R*	P_m	R_m	
D-Bees	8.73	8.73	14.88	14.88	24.88	24.88	35.53	35.53	13.25
D-Bees (enhanced)	11.77	11.77	19.35	19.35	34.68	34.68	47.80	47.80	**17.93**
simulated annealing	5.87	5.87	9.84	9.84	18.44	18.44	25.53	25.53	9.19
Zhao et al. (2007)	11.35	11.35	18.86	18.86	33.88	33.88	46.91	46.91	—
Giuliano et al. (2007)	6.95	6.94	20.33	20.33	**69.03**	**68.90**	58.54	58.54	—
Yuret (2007)	**12.90**	**12.90**	20.65	20.65	46.15	46.15	61.30	61.30	—
Hassan et al. (2007)	12.77	12.77	**20.73**	**20.73**	49.19	49.19	**66.26**	**66.26**	—
Melamud et al. (2015)	8.09	8.09	13.41	13.41	27.65	27.65	39.19	39.19	—
Biemann and Riedl (2013)	—	—	—	—	27.48	27.48	37.19	37.19	—
top-ranked synonyms	9.95	9.95	15.28	15.28	29.70	29.35	40.57	40.57	—

- As WordNet contains no hypernymy relations for adjectives, for our purposes we use its "similar-to" relation instead.
- For word frequency, we generally prefer the counts provided by WordNet, since they are sense-disambiguated. (This use of manually sense-annotated data makes our approach weakly supervised.) In other cases, such as when ranking substitutes from Wiktionary, we use Web 1T (Brants and Franz, 2006) instead of BNC. Web 1T is a much larger, more modern, Web-derived corpus that may better reflect the lemma distributions in the Web-derived SemEval-2007 dataset.

The enhanced D-Bees–based system performs significantly better than the base system, though in common with the two post–SemEval-2007 systems, it still fails to surpass the state of the art for *best* and *OOT*. The two knowledge-based systems that outperform our system by a large margin, Giuliano et al. (2007) and Hassan et al. (2007), employ particularly strong substitute generation components that use a combination of WordNet with a rich thesaurus resource—the *Oxford American Writer Thesaurus* and the *Microsoft Encarta* encyclopedia, respectively. Both resources outperform Wiktionary in terms of coverage of synonyms and semantically related words. However, as these resources are proprietary, they were not available to us.

Our system's performance is roughly on par with Zhao et al. (2007), another bottom-up approach. Our enhanced system does achieve the highest known *GAP* score, though this is largely because most prior work does not use this metric, or else applies it only to the ranking of gold-standard substitutes.

6 Lexical Simplification

6.1 Experimental Setup

Our experiments use the dataset from the SemEval-2012 English lexical simplification task (Specia et al., 2012). It uses the same contexts and target words as the SemEval-2007 dataset, but the gold-standard substitutes, which include the original target words, have been manually re-ranked according to their perceived simplicity. Unlike SemEval-2007, the SemEval-2012 task is concerned exclusively with ranking substitutes; all the original participating systems were given the gold-standard substitutes and simply asked to put them in the correct order. However, to score our own systems we use their own substitute lists, removing only those substitutes that do not also appear in the gold-standard list. This puts us at somewhat of a disadvantage, since our substitute lists often contain only a subset of the gold-standard substitutes. It also makes use of the κ metric problematic, since κ expects the system and gold-standard lists to contain the same set of substitutes. We therefore report only *TRnk* and *R@n* scores.

Specia et al. (2012) report scores for two lower-bound baselines: one puts the substitute lists in random order, and the other orders them by inverse frequency of occurrence in Web 1T.[5] The state of the art is represented by Jauhar and Specia (2012),

[5] A third baseline leaves the lists in their original order (i.e., by inverse number of annotators who chose them). We ignore it here as it relies entirely on manual labelling.

Table 3: System performance on the SemEval-2012 lexical simplification dataset.

System	TRnk	$R@1$	$R@2$	$R@3$
D-Bees (enhanced) (original ordering)	37.5	71.6	**75.5**	76.4
D-Bees (enhanced) (unigram ordering)	50.9	**72.8**	75.2	76.3
D-Bees (enhanced) (n-gram ordering)	47.1	71.3	74.5	75.7
Jauhar and Specia (2012)	**60.2**	57.5	68.9	76.9
unigram ordering baseline	58.5	55.9	68.1	76.0
random ordering baseline	34.0	32.1	61.2	**82.5**

a supervised system that classifies substitutes using a context-sensitive n-gram frequency model, a bag-of-words model, and psycholinguistic features. At SemEval-2012 it achieved the best performance for every metric except $R@3$, where it was beaten only by the random baseline.

We first calculated the proportion of instances for which our systems suggested at least one substitute appearing in the gold standard (other than the target word itself). For the simulated annealing system, the percentage was 45.7%, for the D-Bees system it was 58.7%, and for the enhanced D-Bees system, it was 81.6%. We tentatively conclude that the soft WSD of enhanced D-Bees is necessary to generate sufficient numbers of substitutes in common with the gold standard, and exclude our other two systems from further consideration.

Since the SemEval-2012 lexical simplification task is concerned only with ranking, we test three different rankings of the enhanced D-Bees substitute list. First, we preserve the original order of the system. Second, we order by unigram frequency in Web 1T, as in the SemEval-2012 baseline. Our third ranking is an n-gram ordering approach that we found to work well ($\kappa = 0.461$) on the full gold-standard substitute lists. Here the substitutes are sorted according to the summation of the combined frequency of the substitute and context words. More formally, let W be the set of all unique words in the context window, excluding the target w_t, and let S be the set of substitutes for w_t. Then each substitute $s \in S$ is given a score

$$F(s) = \sum_{w \in W} f(s,w),$$

where $f(s,w)$ is the Web 1T co-occurrence frequency for s and w. The list of substitutes is then sorted by descending score.

6.2 Results and Analysis

Table 3 shows the published results for our baselines, along with the results from the enhanced D-Bees–based system from §5.2 using various ranking methods. While none of our configurations scored particularly well on *TRnk*, all of them surpassed the state of the art for $R@1$ and $R@2$, and performed about as well as Jauhar and Specia (2012) for $R@3$. These results are particularly impressive in light of the fact that the SemEval-2012 systems had access to the gold-standard substitutes, whereas our systems did not.

The good $R@n$ scores when using the original ordering indicate that the D-Bees–based system is (quite serendipitously) predisposed to selecting simple substitutes and ranking them relatively highly. We note that there is relatively little difference between our three system configurations, suggesting that all three ranking methods are doing more or less the same thing, at least for the first few substitutes. This result is somewhat surprising in light of Specia et al.'s (2012) assumption that the notion of simplicity is context-dependent. (It is this notion that our n-gram–based ranking model was attempting to capture.) It could be that, for our systems, the context (including text complexity) is already sufficiently accounted for during WSD.

7 Conclusion

In this paper, we have presented the first extrinsic evaluations of simulated annealing and D-Bees in a lexical substitution setting. We used each algorithm as the WSD component in the same knowledge-based, language-independent lexical substitution system. The systems were tested on German and English datasets, and surpassed state-of-the-art performance on the former. The D-Bees system generally had better results, so we applied some resource-specific adaptations based on our own observations of GermaNet and WordNet, as well as on previ-

ously published studies on German and English lexical substitution. These adaptations led to dramatic improvements in performance on both datasets. We also tested the adapted D-Bees system in a lexical simplification setting, where (in spite of some handicaps) it exceeded state-of-the-art performance on two evaluation metrics. Our findings would seem to validate the utility of metaheuristic approaches for lexical substitution and simplification, with the caveat that optimal performance is achieved only when the systems are adapted to the language or linguistic resources used. This adaptation effort may nonetheless be lower than that required to source annotated training data for supervised approaches.

Regarding future work, there are several issues of interest. The first concerns our use of collaboratively constructed language resources. While our WSD components used only expert-built resources, we found it beneficial to draw additional substitution candidates from Wiktionary. For this we used a very basic sense alignment technique, though a more profound sense mapping between WordNet/GermaNet and Wikitionary, such as those surveyed by Gurevych et al. (2016), might lead to better downstream results. The approach D-Bees uses for calculating sense similarity is also quite basic; though it seemed to work well in practice, we are keen to investigate other methods, such as taking the WordNet/GermaNet graph structure into account, or using other measures of text similarity to compare glosses.

Acknowledgments

This work was supported in part by the research training group Adaptive Preparation of Information from Heterogeneous Sources (AIPHES, GRK 1994/1) and by the German Institute for Educational Research (DIPF).

References

Sallam Abualhaija and Karl-Heinz Zimmermann. 2016. D-Bees: A novel method inspired by bee colony optimization for solving word sense disambiguation. *Swarm and Evolutionary Computation*, 27:188–195.

Eneko Agirre and Philip Edmonds. 2007. *Word Sense Disambiguation: Algorithms and Applications*, volume 33 of *Text, Speech, and Language Technology*. Springer.

Ion Androutsopoulos and Prodromos Malakasiotis. 2010. A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research*, 38(1):135–187, May.

Satanjeev Banerjee and Ted Pedersen. 2002. An adapted Lesk algorithm for word sense disambiguation using WordNet. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing: Third International Conference, CICLing 2002*, volume 2276 of *Lecture Notes in Computer Science*, pages 136–145. Springer.

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 and Evaluation*, 43(3):209–226.

Chris Biemann and Martin Riedl. 2013. Text: Now in 2D! A framework for lexical expansion with contextual similarity. *Journal of Language Modelling*, 1(1):55–95, April.

Thorsten Brants and Alex Franz. 2006. *Web 1T 5-gram Version 1*. Linguistic Data Consortium.

Lou Burnard, 2007. *Reference Guide for the British National Corpus (XML Edition)*. British National Corpus Consortium, February.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, April.

Jim Cowie, Joe Guthrie, and Louise Guthrie. 1992. Lexical disambiguation using simulated annealing. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING 1992)*, volume 1, pages 359–365.

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

Claudio Giuliano, Alfio Gliozzo, and Carlo Strapparava. 2007. FBK-irst: Lexical substitution task exploiting domain and syntagmatic coherence. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 145–148, June.

Iryna Gurevych, Judith Eckle-Kohler, and Michael Matuschek, 2016. *Linked Lexical Knowledge Bases: Foundations and Applications*, volume 34 of *Synthesis Lectures on Human Language Technologies*, chapter 3: Linking Algorithms, pages 29–44. Morgan & Claypool.

Birgit Hamp and Helmut Feldweg. 1997. GermaNet – A lexical-semantic net for German. In *Proceedings of the ACL Workshop on Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, pages 9–15.

Samer Hassan, Andras Csomai, Carmen Banea, Ravi Sinha, and Rada Mihalcea. 2007. UNT: SubFinder: Combining knowledge sources for automatic lexical

substitution. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 410–413, June.

Verena Henrich and Erhard Hinrichs. 2010. GernEdiT – The GermaNet editing tool. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*, pages 2228–2235.

Verena Henrich and Erhard Hinrichs. 2012. A comparative evaluation of word sense disambiguation algorithms for German. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, pages 576–583.

Gerold Hintz and Chris Biemann. 2015. Delexicalized supervised German lexical substitution. In *Proceedings of GermEval 2015: LexSub*, pages 11–16.

Luchezar Jackov. 2015. Lexical substitution using deep syntactic and semantic analysis. In *Proceedings of GermEval 2015: LexSub*, pages 17–20.

Sujay Kumar Jauhar and Lucia Specia. 2012. UOW-SHEF: SimpLex – Lexical simplicity ranking based on contextual and psycholinguistic features. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantic (SemEval 2012)*, volume 2, pages 477–481, June.

Scott Kirkpatrick, Charles D. Gelatt, Jr., and Mario P. Vecchi. 1983. Optimization by simulated annealing. *Science*, 220(4598):671–680, May.

Kazuaki Kishida. 2005. Property of average precision and its generalization: an examination of evaluation indicator for information retrieval experiments. Technical Report NII-2005-014E, National Institute of Informatics, Tokyo, October.

Gerhard Kremer, Katrin Erk, Sebastian Padó, and Stefan Thater. 2014. What substitutes tell us – Analysis of an "all-words" lexical substitution corpus. In *14th Conference of the European Chapter of the Association for Computational Linguistics: Proceedings of the Conference (EACL 2014)*, pages 540–549, April.

Diana McCarthy and Roberto Navigli. 2009. The English lexical substitution task. *Language Resources and Evaluation*, 43(2):139–159, June.

Oren Melamud, Omer Levy, and Ido Dagan. 2015. A simple word embedding model for lexical substitution. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 1–7, June.

Christian M. Meyer and Iryna Gurevych. 2012. Wiktionary: A new rival for expert-built lexicons? Exploring the possibilities of collaborative lexicography. In Sylviane Granger and Magali Paquot, editors, *Electronic Lexicography*, pages 259–291. Oxford University Press.

Tristan Miller, Darina Benikova, and Sallam Abualhaija. 2015. GermEval 2015: LexSub – A shared task for German-language lexical substitution. In *Proceedings of GermEval 2015: LexSub*, pages 1–9.

Tristan Miller, Mohamed Khemakhem, Richard Eckart de Castilho, and Iryna Gurevych. 2016. Sense-annotating a lexical substitution data set with Ubyline. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, pages 828–835, May.

Taesun Moon and Katrin Erk. 2013. An inference-based model of word meaning in context as a paraphrase distribution. *ACM Transactions on Intelligent Systems and Technology*, 4(3):42:1–42:28, June.

Diarmuid Ó Séaghdha and Anna Korhonen. 2014. Probabilistic distributional semantics with latent variable models. *Computational Linguistics*, 40(3):587–631, September.

Ganesh Ramakrishnan, B. P. Prithviraj, A. Deepa, Pushpak Bhattacharyya, and Soumen Chakrabarti. 2004. Soft word sense disambiguation. In *Proceedings of the 2nd Global WordNet Conference (GWC 2004)*, pages 291–298.

Stephen Roller and Katrin Erk. 2016. PIC a different word: A simple model for lexical substitution in context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL–HLT 2016)*, pages 1121–1126, June.

Serge Sharoff. 2006. Open-source corpora: Using the Net to fish for linguistic data. *International Journal of Corpus Linguistics*, 11(4):435–462.

Ravi Sinha and Rada Mihalcea. 2011. Explorations in lexical sample and all-words lexical substitution. *Natural Language Engineering*, 1(1):1–27.

Lucia Specia, Sujay Kumar Jauhar, and Rada Mihalcea. 2012. Semeval-2012 task 1: English lexical simplification. In *Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval-2012)*, pages 347–355.

György Szarvas, Chris Biemann, and Iryna Gurevych. 2013a. Supervised all-words lexical substitution using delexicalized features. In *Proceedings of the 10th Conference of the North American Chapter of the Association for Computational Linguistics and the 18th Conference on Human Language Technologies (NAACL–HLT 2013)*, pages 1131–1141.

György Szarvas, Róbert Busa-Fekete, and Eyke Hüllermeier. 2013b. Learning to rank lexical substitutions. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pages 1926–1932, October.

El-Ghazali Talbi. 2009. *Metaheuristics: From Design to Implementation*. Wiley.

Dušan Teodorović. 2009. Bee colony optimization (BCO). In Chee Peng Lim, Lakhmi C. Jain, and Satchidananda Dehuri, editors, *Innovations in Swarm Intelligence*, pages 39–60. Springer.

Stefan Thater, Hagen Fürstenau, and Manfred Pinkal. 2010. Contextualizing semantic representations using syntactically enriched vector models. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2011)*, pages 948–957.

Antonio Toral. 2009. The lexical substitution task at EVALITA 2009. In *Proceedings of the 11th Conference of the Italian Association for Artificial Intelligence*.

Deniz Yuret. 2007. KU: Word sense disambiguation by substitution. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 207–214, June.

Torsten Zesch and Iryna Gurevych. 2010. Wisdom of crowds versus wisdom of linguists – measuring the semantic relatedness of words. *Natural Language Engineering*, 16(1):25–59, January.

Torsten Zesch, Christof Müller, and Iryna Gurevych. 2008. Extracting lexical semantic knowledge from Wikipedia and Wiktionary. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*, volume 8, pages 1646–1652.

Shiqi Zhao, Lin Zhao, Yu Zhang, Ting Liu, and Sheng Li. 2007. HIT: Web based scoring method for English lexical substitution. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 173–176, June.