MELB-MKB: Lexical Substitution System based on Relatives in Context

David Martinez, Su Nam Kim and Timothy Baldwin
LT Group, CSSE
University of Melbourne
Victoria 3010 Australia
{davidm,snkim,tim}@csse.unimelb.edu.au

Abstract

In this paper we describe the MELB-MKB system, as entered in the SemEval-2007 lexical substitution task. The core of our system was the "Relatives in Context" unsupervised approach, which ranked the candidate substitutes by web-lookup of the word sequences built combining the target context and each substitute. Our system ranked third in the final evaluation, performing close to the top-ranked system.

1 Introduction

This paper describes the system we developed for the SemEval lexical substitution task, a new task in SemEval-2007. Although we tested different configurations on the trial data, our basic system relied on WordNet relatives (Fellbaum, 1998) and Google queries in order to identify the most plausible substitutes in the context.

The main goal when building our system was to study the following factors: (i) substitution candidate set, (ii) settings of the relative-based algorithm, and (iii) syntactic filtering. We analysed these factors over the trial data provided by the organisation, and used the BEST metric to tune our system. This metric accepts multiple answers, and averages the score across the answers. We did not experiment with the OOT (top 10 answers) and MULTIWORD metrics.

In the remainder of this paper we briefly introduce the basic Relatives in Context algorithm in Section 2. Next we describe our experiments on the trial data in Section 3. Our final system and its results are described in Section 4. Finally, our conclusions are outlined in Section 5.

2 Algorithm

Our basic algorithm is an unsupervised method presented in Martinez et al. (2006). This technique makes use of the WordNet relatives of the target word for disambiguation, by way of the following steps: (i) obtain a set of close relatives from WordNet for each sense of the target word; (ii) for each test instance define all possible word sequences that include the target word; (iii) for each word sequence, substitute the target word with each relative, and then query Google; (iv) rank queries according to the following factors: length of the query, distance of the relative to the target word, and number of hits; and (v) select the relative from the highest ranked query.[1]

For the querying step, first we tokenise each target sentence, and then we apply sliding windows of different sizes (up to 6 tokens) that include the target word. For each window and each relative in the pool, we substitute the target word for the relative, and query Google. The algorithm stops augmenting the window for the relative when one of its substrings returns zero hits. The length of the query is measured as the number of words, and the distance of the relative to the target words gives preference to synonyms over hypernyms, and immediate hypernyms over further ones.

One important parameter in this method is the candidate set. We performed different experiments to measure the expected score we could achieve

[1] In the case of WSD we would use the relative to chose the sense it relates to.

Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), pages 237–240,
Prague, June 2007. ©2007 Association for Computational Linguistics

from WordNet relatives, and the contribution of different types of filters (syntactic, frequency-based, etc.) to the overall result. We also explored other settings of the algorithm, such as the ranking criteria, and the number of answers to return. These experiments and some other modifications of the basic algorithm are covered in Section 3.

3 Development on Trial data

In this section we analyse the coverage of WordNet over the data, the basic parameter exploration process, a syntactic filter, and finally the extra experiments we carried out before submission. The trial data consisted on 300 instances of 34 words with gold-standard annotations.

3.1 WordNet coverage

The most obvious resource for selecting substitution candidates was WordNet, due to its size and availability. We used version 2.0 throughout this work. In our first experiment, we tried to determine which kind of relationships to use, and the coverage of the gold-standard annotations that we could expect from WordNet relations only. As a basic set of relations, we used the following: SYNONYMY, SIMILAR-TO, ENTAILMENT, CAUSE, ALSO-SEE, and INSTANCE. We created two extended candidate sets using immediate and 2-step hypernyms (hype and hype2, respectively, in Table 1).

Given that we are committed to using WordNet, we set out to measure the percentage of gold-standard substitutes that were "reachable" using different WordNet relations. Table 1 shows the coverage for the three sets of candidates. Instance-coverage indicates the percentage of instances that have at least one of the gold-standard instances covered from the candidate set. We can see that the percentage is surprisingly low.

Any shortcoming in coverage will have a direct impact on performance, suggesting the need for alternate means to obtain substitution candidates. One possibility is to extend the candidates from WordNet by following links from the relatives (e.g. collect all synonyms of the synonymous words), but this could add many noisy candidates. We can also use other lexical repositories built by hand or automatically, such as the distributional theusauri built in Lin (1998). A different approach that we are testing for future work is to adapt the algorithm to work with wildcards instead of explicit candidates. Due to time constraints, we only relied on WordNet for our submission.

Candidate Set	Subs. Cov.	Inst. Cov.
basic	344/1152 (30%)	197 / 300 (66%)
hype	404/1152 (35%)	229/300 (76%)
hype2	419/1152 (36%)	229/300 (76%)

Table 1: WordNet coverage for different candidate sets, based on substitute (Subs.) and instance (Inst.) coverage.

3.2 Parameter Tuning

In this experiment we tuned different parameters of the basic algorithm. First, we observed the data in order to identify the most relevant variables for this task. We tried to avoid including too many parameters and overfitting the system to the trial dataset. At this point, we separated the instances by PoS, and studied the following parameters:

Candidate set: From WordNet, we tested four possible datasets for each target word: basic-set, 1st-sense (basic relations from the first sense only), hype (basic set and immediate hypernyms), and hype2 (basic set and up to two-step hypernyms).

Semcor-based filters: Semcor provides frequency information for WordNet senses, and can be used to identify rare senses. As each candidate is obtained via WordNet semantic relations with the target word, we can filter out those candidates that are related with unfrequent senses in Semcor. We tested three configurations: (1) no filter, (2) filter out candidates when the *candidate*-sense in the relation does not occur in Semcor, (3) and filter out candidates when the *target*-sense in the relation does not occur in Semcor. The filters can potentially lead to the removal of all candidates, in which case a back-off is applied (see below).

Relative-ranking criteria: Our algorithm ranks relatives according to the length in words of their context-match. In the case of ties, the number of returned hits from Google is applied. The length can be different depending on whether we count punctuation marks as separate tokens, and whether the word-length of substitute multiwords is included.

We tested three options: including the target word, not including the target word (multiwords count as a single word), and not counting punctuation marks.

Back-off: We need a back-off method in case the basic algorithm does not find any matches. We tested the following: sense-ordered synonyms from WordNet (highest sense first, randomly breaking ties), and most frequent synonyms from the first system (using two corpora: Semcor and BNC).

Number of answers: We also measured the performance for different numbers of system outputs (1, 2, or 3).

All in all, we performed 324 (4x3x3x3x3) runs for each PoS, based on the different combinations. The best scores for each PoS are shown in Table 2, together with the baselines. We can see that the precision is above the official WordNet baseline, but is still very low. The results illustrate the difficulty of the task. In error analysis, we observed that the performance and settings varied greatly depending on the PoS of the target word. Adverbs produced the best performance, followed by nouns. The scores were very low for adjectives and verbs (the baseline score for verbs was only 2%).

We will now explain the main conclusions extracted from the parameter analysis. Regarding the candidate set, we observed that using synonyms only was the best approach for all PoS, except for verbs, where hypernyms helped. The option of limiting the candidates to the first sense only helped for adjectives, but not for other PoS.

For the Semcor-based filter, our results showed that the target-sense filter improved the performance for verbs and adverbs. For nouns and adjectives, the candidate-sense filter worked best. All in all, applying the Semcor filters was effective in removing rare senses and improving performance.

The length criteria did not affect the results significantly, and only made a difference in some extreme cases. Not counting the length of the target word helped slightly for nouns and adverbs, and removing punctuation improved results for adjectives. Regarding the back-off method, we observed that the count of frequencies in Semcor was the best approach for all PoS except verbs, which reached their best performance with BNC frequencies.

PoS	Relatives in Context	WordNet Baseline
Nouns	18.4	14.9
Verbs	6.7	2.0
Adjectives	9.6	7.5
Adverbs	31.1	29.9
Overall	14.4	10.4

Table 2: Experiments to tune parameters on the trial data, based on the BEST metric. Scores correspond to precision (which is the same as recall).

Finally, we observed that the performance for the BEST score decreased significantly when more than one answer was returned, probably due to the difficulty of the task.

3.3 Syntactic Filter

After the basic parameter analysis, we studied the contribution of a syntactic filter to remove those candidates that, when substituted, generate an ungrammatical sentence. Intuitively, we would expect this to have a high impact for verbs, which vary considerably in their subcategorisation properties. For example, in the case of the (reduced) target *If we **order** our lives well ...*, the syntactic filter should ideally disallow candidates such as *If we **range** our lives well ...*

In order to apply this filter, we require a parser which has an explicit notion of grammaticality, ruling out the standard treebank parsers. We experimented briefly with RASP, but found that the English Resource Grammar (ERG: Flickinger (2002)), combined with the PET run-time engine, was the best fit for out needs. Unfortunately we could not get unknown word handling working within the ERG for our submission, such that we get a meaningful output for a given input string only in the case that the ERG has full lexical coverage over that string (we will never get a spanning parse for an input where we are missing lexical entries). As such, the syntactic filter is limited in coverage only to strings where the ERG has lexical coverage.

Ideally, we would have tested this filter on trial data, but unfortunately we ran out of time. Thus, we simply eyeballed a sample of examples, and we decided to include this filter in our final submission. As we will see in Section 4, its effect was minimal. We plan to perform a complete evaluation of this module in the near future.

3.4 Extra experiments

One of the limitations of the "Relatives in Context" algorithm is that it only relies on the local context. We wanted to explore the contribution of other words in the context for the task, and we performed an experiment including the Topical Signatures resource (Agirre and Lopez de Lacalle, 2004). We simply counted the overlapping of words shared between the context and the different candidates. We only tested this for nouns, for which the results were below baseline. We then tried to integrate the topic-signature scores with the "Relatives in Context" algorithm, but we did not improve our basic system's results on the trial data. Thus, this approach was not included in our final submission.

Another problem we observed in error analysis was that the Semcor-based filters were too strict in some cases, and it was desirable to have a way of penalising low frequency senses without removing them completely. Thus, we weighted senses by the inverse of their sense-rank. As we did not have time to test this intuition properly, we opted for applying the sense-weighting only when the candidates had the same context-match length, instead of using the number of hits. We will see the effect of this method in the next section.

4 Final system

The test data consisted of 1,710 instances. For our final system we applied the best configuration for each PoS as observed in the development experiments, and the syntactic filter. We also incorporated the sense-weighting to solve ties. The results of our system, the best competing system, and the best baseline (WordNet) are shown in Table 3 for the BEST metric. Precision and recall are provided for all the instances, and also for the "Mode" instances (those that have a single preferred candidate).

Our method outperforms the baseline in all cases, and performs very close to the top system, ranking third out of eight systems. This result is consistent in the "further analysis" tables provided by the task organisers for subsets of data, where our system always performs close to the top score. The overall scores are below 13% recall for all systems when targeting all instances. This illustrates the difficulty of the task, and the similarity of the top-3 scores suggests that similar resources (i.e. WordNet) have been used in the development of the systems.

System	All instances P	All instances R	Mode P	Mode R
Best	12.90	12.90	20.65	20.65
Relat. in Context	**12.68**	**12.68**	**20.41**	**20.41**
WordNet baseline	9.95	9.95	15.28	15.28

Table 3: Official results based on the BEST metric.

After the release of the gold-standard data, we tested two extra settings to measure the effect of the syntactic filter and the sense-weighting in the final score. We observed that our application of the syntactic filter had almost no effect in the performance, but sense-weighting increased the overall recall by 0.4% (from 12.3% to 12.7%).

5 Conclusions

Although the task was difficult and the scores were low, we showed that by using WordNet and the local context we are able to outperform the baselines and achieve close to top performance. For future work, we would like to integrate a parser with unknown word handling in our system. We also aim to adapt the algorithm to match the target context with wildcards, in order to avoid explicitly defining the candidate set.

Acknowledgments

This research was carried out with support from Australian Research Council grant no. DP0663879.

References

Eneko Agirre and Oier Lopez de Lacalle. 2004. Publicly available topic signatures for all WordNet nominal senses. In *Proc. of the 4rd International Conference on Languages Resources and Evaluations (LREC 2004)*, pages 1123–6, Lisbon, Portugal.

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

Dan Flickinger. 2002. On building a more efficient grammar by exploiting types. In Stephan Oepen, Dan Flickinger, Jun'ichi Tsujii, and Hans Uszkoreit, editors, *Collaborative Language Engineering*. CSLI Publications, Stanford, USA.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of COLING-ACL*, pages 768–74, Montreal, Canada.

David Martinez, Eneko Agirre, and Xinglong Wang. 2006. Word relatives in context for word sense disambiguation. In *Proc. of the 2006 Australasian Language Technology Workshop*, pages 42–50, Sydney, Australia.