UNT: SubFinder: Combining Knowledge Sources for Automatic Lexical Substitution

Samer Hassan, Andras Csomai, Carmen Banea, Ravi Sinha, Rada Mihalcea*
Department of Computer Science and Engineering
University of North Texas
samer@unt.edu, csomaia@unt.edu, carmenb@unt.edu, rss0089@unt.edu, rada@cs.unt.edu

Abstract

This paper describes the University of North Texas SUBFINDER system. The system is able to provide the most likely set of substitutes for a word in a given context, by combining several techniques and knowledge sources. SUBFINDER has successfully participated in the *best* and *out of ten (oot)* tracks in the SEMEVAL lexical substitution task, consistently ranking in the first or second place.

1 Introduction

Lexical substitution is defined as the task of identifying the most likely alternatives (substitutes) for a target word, given its context (McCarthy, 2002). Many natural language processing applications can benefit from the availability of such alternative words, including word sense disambiguation, lexical acquisition, machine translation, information retrieval, question answering, text simplification, and others.

The task is closely related to the problem of word sense disambiguation, with the substitutes acting as synonyms for the input word meaning. Unlike word sense disambiguation however, lexical substitution is not performed with respect to a given sense inventory, but instead candidate synonyms are generated "on the fly" for a given word occurrence. Thus, lexical substitution can be regarded in a way as a hybrid task that combines word sense disambiguation and distributional similarity, targeting the identification of *semantically similar* words that *fit the context*.

2 A system for lexical substitution

SUBFINDER is a system able to provide the most likely set of substitutes for a word in a given context. In SUBFINDER, the lexical substitution task is carried out as a sequence of two steps. First, candidates are extracted from a variety of knowledge sources; so far, we experimented with WordNet (Fellbaum, 1998), Microsoft Encarta encyclopedia, Roget, as well as synonym sets generated from bilingual dictionaries, but additional knowledge sources can be integrated as well. Second, provided a list of candidates, a number of ranking methods are applied in a weighted combination, resulting in a final list of lexical substitutes ranked by their semantic fit with both the input target word and the context.

3 Candidate Extraction

Candidates are extracted using several lexical resources, which are combined into a larger comprehensive resource.

WordNet: WordNet is a large lexical database of English, with words grouped into synonym sets called *synsets*. A problem we encountered with this resource is that often times the only candidate in the synset is the target word itself. Thus, to enlarge the set of candidates, we use both the synonyms and the hypernyms of the target word. We also remove the target word from the synset, to ensure that only viable candidates are considered.

Microsoft Encarta encyclopedia: The Microsoft Encarta is an online encyclopedia and thesaurus resource, which provides for each word the part of speech and a list of synonyms. Using the part of speech as identified in the context, we are able to extract synsets for the target word. An important feature in the Encarta Thesaurus is that the first word in the synset acts as a definition for the synset, and therefore disambiguates the target word. This definition is maintained as a separate entry in the com-

*Contact author.

Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), pages 410–413,
Prague, June 2007. ©2007 Association for Computational Linguistics

prehensive resource, and it is also added to its corresponding synset.

Other Lexical Resources: We have also experimented with two other lexical resources, namely the Roget thesaurus and a thesaurus built using bilingual dictionaries. In evaluations carried out on the development data set, the best results were obtained using only WordNet and Encarta, and thus these are the resources used in the final SUBFINDER system.

All these resources entail different forms of synset clustering. In order to merge them, we use the largest overlap among them. It is important to note that the choice of the first resource considered has a bearing on the way the synsets are clustered. In experiments ran on the development data set, the best results were obtained using a lexical resource constructed starting with the Microsoft Encarta Thesaurus and then mapping the WordNet synsets to it.

4 Candidate Ranking

Several ranking methods are used to score the candidate substitutes, as described below.

Lexical Baseline (LB): In this approach we use the pre-existing lexical resources to provide a ranking over the candidate substitutes. We rank the candidates based on their occurrence in the two selected lexical resources WordNet and Encarta, with those occurring in both resources being assigned a higher ranking. This technique emphasizes the resources annotators' agreement that the candidates belong indeed to the same synset.

Machine Translation (MT): We use machine translation to translate the test sentences back-and-forth between English and a second language. From the resulting English translation, we extract the replacement that the machine translation engine provides for the target word. To locate the translated word we scan the translation for any of the candidates (and their inflections) as obtained from the comprehensive resource, and score the candidate synset accordingly.

We experimented with a range of languages such as French, Italian, Spanish, Simplified Chinese, and German, but the best results obtained on the development data were based on the French translations. This could be explained because French is part of the Romance languages family and synonyms to English words often find their roots in Latin. If we consider again the word *bright*, it was translated into French as *intelligent* and then translated back into English as *intelligent* for obvious reasons. In one instance, *intelligent* was the best replacement for *bright* in the trial data. Despite the fact that we also used Italian and Spanish (which are both Latin-based) we can only assume that French worked better because translation engines are better trained on French. From the resulting English translation, we extract the replacement that the machine translation engine provides for the target word. To locate the translated word we scan the translation for any of the candidates (and their inflections) as obtained from the comprehensive resource, and score the candidate synset accordingly. The translation process was carried out using Google and AltaVista translation engines resulting in two systems MTG and MTA respectively. The translation systems feature high precision when a candidate is found (about 20% of the time), at the cost of low recall. The lexical baseline method is therefore used when no candidates are returned by the translation method.

Most Common Sense (MCS): Another method we use for ranking candidates is to consider the first word appearing in the first synset returned by WordNet. When no words other than the target word are available in this synset, the method recursively searches the next synset available for the target word. In order to guarantee a sufficient number of candidates, we use the lexical baseline method as a baseline.

Language Model (LM): We model the semantic fit of a candidate substitute within the given context using a language model, expressed using the conditional probability:

$$P(c|g) = P(c,g)/P(g) \approx Count(c,g) \quad (1)$$

where c represents a possible candidate and g represents the context. The probability $P(g)$ of the context is the same for all the candidates, hence we can ignore it and estimate $P(c|g)$ as the N-gram frequency of the context where the target word is replaced by the proposed candidate. To avoid skewed counts that can arise from the different morphological inflections of the target word or the candidate and the bias that the context might have toward any specific inflection, we generalize $P(c|g)$ to take into account all the inflections of the selected candidate as shown in equation 2.

$$P^n(c|g) \approx \sum_{i=1}^{n} Count(c_i, g) \quad (2)$$

where n is the number of possible inflections for the candidate c.

We use the Google N-gram dataset to calculate the term $Count(c_i\ g)$. The Google N-gram corpus is a

collection of English N-grams, ranging from one to five N-grams, and their respective frequency counts observed on the Web (Brants and Franz, 2006). In order for the model to give high preference to the longer N-grams, while maintaining the relative frequencies of the shorter N-grams (typically more frequent), we augment the counts of the higher order N-grams with the maximum counts of the lower order N-grams, hence guaranteeing that the score assigned to an N-gram of order N is higher than the the score of an N-gram of order $N-1$.

Semantic Relatedness using Latent Semantic Analysis (LSA): We expect to find a strong semantic relationship between a good candidate and the target context. A relatively simple and efficient way to measure such a relatedness is the Latent Semantic Analysis (Landauer et al., 1998). Documents and terms are mapped into a 300 dimensional latent semantic space, providing the ability to measure the semantic relatedness between two words or a word and a context. We use the InfoMap package from Stanford University's Center for the Study of Language and Information, trained on a collection of approximately one million Wikipedia articles. The rank of a candidate is given by its semantic relatedness to the entire context sentence.

Information Retrieval (IR): Although the Language Model approach is successful in ranking the candidates, it suffers from the small N-gram size imposed by using the Google N-grams corpus. Such a restriction is obvious in the following 5-gram example *who was a bright boy* in which the context is not sufficient to disambiguate between *happy* and *smart* as possible candidates. As a result, we adapt an information retrieval approach which uses all the content words available in the given context. Similar to the previous models, the target word in the context is replaced by all the generated inflections of the selected candidate and then queried using a web search engine. The resulting rank represents the sum of the total number of pages in which the candidate or any of its inflections occur together with the context. This also reflects the semantic relatedness or the relevance of the candidate to the context.

Word Sense Disambiguation (WSD): Since previous work indicated the usefulness of word sense disambiguation systems in lexical substitution (Dagan et al., 2006), we use the SenseLearner word sense disambiguation tool (Mihalcea and Csomai, 2005) to disambiguate the target word and, accordingly, to propose its synonyms as candidates.

Final System: Our candidate ranking methods are aimed at different aspects of what constitutes a good candidate. On one hand, we measure the semantic relatedness of a candidate with the original context (the LSA and WSD methods fall under this category). On the other hand, we also want to ensure that the candidate fits the context and leads to a well formed English sentence (e.g., the language model method). Given that the methods described earlier aim at orthogonal aspects of the problem, it is expected that a combination of these will provide a better overall ranking.

We use a voting mechanism, where we consider the reciprocal of the rank of each candidates as given by one of the described methods. The final score of a candidate is given by the decreasing order of the weighted sum of the reciprocal ranks:

$$score\,(c_i) = \sum_{m \in rankings} \lambda_m \frac{1}{r^m_{c_i}}$$

To determine the weight λ of each individual ranking we run a genetic algorithm on the development data, optimized for the *mode* precision and recall. Separate sets of weights are obtained for the *best* and *oot* tasks. Table 1 shows the weights of the individual ranking methods. As expected, for the *best* task, the language model type of methods obtain higher weights, whereas for the *oot* task, the semantic methods seem to perform better.

5 Results and Discussion

The SUBFINDER system participated in the *best* and the *oot* tracks of the lexical substitution task. The *best* track calls for any number of best guesses, with the most promising one listed first. The credit for each correct guess is divided by the number of guesses. The *oot* track allows systems to make up to 10 guesses, without penalizing, and without being of any benefit if less than 10 substitutes are provided. The ordering of guesses in the *oot* metric is unimportant.

For both tracks, the evaluation is carried out using precision and recall, calculated based on the number of matching responses between the system and the human annotators, respectively. A "mode" evaluation is also conducted, which measures the ability of the systems to capture the most frequent response (the "mode") from the gold standard annotations. For details, please refer to the official task description document (McCarthy and Navigli, 2007).

Tables 2 and 3 show the results obtained by SUBFINDER in the *best* and *oot* tracks respectively. The tables also show a breakdown of the results based

on: only target words that were not identified as multiwords (NMWT); only substitutes that were not identified as multiwords (NMWS); only items with sentences randomly selected from the Internet corpus (RAND); only items with sentences manually selected from the Internet corpus (MAN).

	WSD	LSA	IR	LB	MCS	MTA	MTG	LM
best	34	2	64	63	56	69	38	97
oot	6	82	7	28	46	14	32	68

Table 1: Weights of the individual ranking methods

	P	R	Mode P	Mode R
OVERALL	12.77	12.77	20.73	20.73
Further Analysis				
NMWT	13.46	13.46	21.63	21.63
NMWS	13.79	13.79	21.59	21.59
RAND	12.85	12.85	20.18	20.18
MAN	12.69	12.69	21.35	21.35
Baselines				
WORDNET	9.95	9.95	15.28	15.28
LIN	8.84	8.53	14.69	14.23

Table 2: BEST results

	P	R	Mode P	Mode R
OVERALL	49.19	49.19	66.26	66.26
Further Analysis				
NMWT	51.13	51.13	68.03	68.03
NMWS	54.01	54.01	70.15	70.15
RAND	51.71	51.71	68.04	68.04
MAN	46.26	46.26	64.24	64.24
Baselines				
WORDNET	29.70	29.35	40.57	40.57
LIN	27.70	26.72	40.47	39.19

Table 3: OOT results

Compared to other systems participating in this task, our system consistently ranks on the first or second place. SUBFINDER clearly outperforms all the other systems for the "mode" evaluation, showing the ability of the system to find the substitute most often preferred by the human annotators. In addition, the system exceeds by a large margin all the baselines calculated for the task, which select substitutes based on existing lexical resources (e.g., WordNet or Lin distributional similarity).

Separate from the "official" submission, we ran a second experiment where we optimized the combination weights targeting high precision and recall (rather than high *mode*). An evaluation of the system using this new set of weights yields a precision and recall of 13.34 with a *mode* of 21.71 for the *best* task, surpassing the best system according to the anonymous results report. For the *oot* task, the precision and recall increased to 50.30, still maintaining second place.

6 Conclusions

The lexical substitution task goes beyond simple word sense disambiguation. To approach such a task, we first need a good comprehensive and precise lexical resource for candidate extraction. Secondly, we need to semantically filter the highly diverse and ambiguous set of candidates, while taking into account their fitness in the context in order to form a proper linguistic expression. To accomplish this, we built a system that incorporates lexical, semantic, and probabilistic methods to capture both the semantic similarity with the target word and the semantic fit in the context. Compared to other systems participating in this task, our system consistently ranks on the first or second place. SUBFINDER clearly outperforms all the other systems for the "mode" evaluation, proving its ability to find the substitute most often preferred by the human annotators.

Acknowledgments

This work was supported in part by the Texas Advanced Research Program under Grant #003594. The authors are grateful to the Language and Information Technologies research group at the University of North Texas for many useful discussions and feedback on this work.

References

T. Brants and A. Franz. 2006. Web 1t 5-gram version 1. Linguistic Data Consortium.

I. Dagan, O. Glickman, A. Gliozzo, E. Marmorshtein, and C. Strapparava. 2006. Direct word sense matching for lexical substitution. In *Proceedings of the International Conference on Computational Linguistics ACL/COLING 2006*.

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

T. K. Landauer, P. Foltz, and D. Laham. 1998. Introduction to latent semantic analysis. *Discourse Processes*, 25.

D. McCarthy and R. Navigli. 2007. The semeval English lexical substitution task. In *Proceedings of the ACL Semeval workshop*.

D. McCarthy. 2002. Lexical substitution as a task for wsd evaluation. In *Proceedings of the ACL Workshop on Word Sense Disambiguation: Recent Successes and Future Directions*, Philadelphia.

R. Mihalcea and A. Csomai. 2005. Senselearner: Word sense disambiguation for all words in unrestricted text. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics*, Ann Arbor, MI.