HIT: Web based Scoring Method for English Lexical Substitution

Shiqi Zhao, Lin Zhao, Yu Zhang, Ting Liu, Sheng Li
Information Retrieval Laboratory, School of Computer Science and Technology,
Box 321, Harbin Institute of Technology
Harbin, P.R. China, 150001
{ zhaosq, lzhao, zhangyu, tliu, lisheng }@ir.hit.edu.cn

Abstract

This paper describes the HIT system and its participation in SemEval-2007 English Lexical Substitution Task. Two main steps are included in our method: candidate substitute extraction and candidate scoring. In the first step, candidate substitutes for each target word in a given sentence are extracted from WordNet. In the second step, the extracted candidates are scored and ranked using a web-based scoring method. The substitute ranked first is selected as the best substitute. For the multiword subtask, a simple WordNet-based approach is employed.

1 Introduction

Lexical substitution aims to find alternative words that can occur in given contexts. It is important in many applications, such as query reformulation in question answering, sentence generation, and paraphrasing. There are two key problems in the lexical substitution task, the first of which is candidate substitute extraction. Generally speaking, synonyms can be regarded as candidate substitutes of words. However, some looser lexical relationships can also be considered, such as *Hypernyms* and *Hyponyms* defined in WordNet (Fellbaum, 1998). In addition, since lexical substitution is context dependent, some words which do not have similar meanings in general may also be substituted in some certain contexts (Zhao et al., 2007). As a result, finding a lexical knowledge base for substitute extraction is a challenging task.

The other problem is candidate scoring and ranking according to given contexts. In the lexical substitution task of SemEval-2007, context is constrained as a sentence. The system therefore has to score the candidate substitutes of each target word using the given sentence. The following questions should be considered here: (1) What words in the given sentence are "useful" context? (2) How to combine the context words and use them in ranking candidate substitutes? For the first question, we can use all words of the sentence, words in a window, or words having syntactic relations with the target word. For the second question, we can regard the context words as "bag of words", n-grams, or syntactic structures.

In HIT, we extract candidate substitutes from WordNet, in which both synonyms and hypernyms are investigated (Section 3.1). After that, we score the candidates using a web-based scoring method (Section 3.2). In this method, we first select fragments containing the target word from the given sentence. Then we construct queries by replacing the target word in the fragments with the candidate substitute. Finally, we search Google using the constructed queries and score each candidate based on the counts of retrieved snippets.

The rest of this paper is organized as follows: Section 2 reviews some related work on lexical substitution. Section 3 describes our system, especially the web-based scoring method. Section 4 presents the results and analysis.

2 Related Work

Synonyms defined in WordNet have been widely used in lexical substitution and expansion (Smeaton et al., 1994; Langkilde and Knight, 1998; Bol-

Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), pages 173–176,
Prague, June 2007. ©2007 Association for Computational Linguistics

shakov and Gelbukh, 2004). In addition, a lot of methods have been proposed to automatically construct thesauri of synonyms. For example, Lin (1998) clustered words with similar meanings by calculating the dependency similarity. Barzilay and McKeown (2001) extracted paraphrases using multiple translations of literature works. Wu and Zhou (2003) extracted synonyms with multiple resources, including a monolingual dictionary, a bilingual corpus, and a monolingual corpus. Besides the handcrafted and automatic synonym resources, the web has been exploited as a resource for lexical substitute extraction (Zhao et al., 2007).

As for substitute scoring, various methods have been investigated, among which the classification method is the most widely used (Dagan et al., 2006; Kauchak and Barzilay, 2006). In detail, a binary classifier is trained for each candidate substitute, using the contexts of the substitute as features. Then a new contextual sentence containing the target word can be classified as 1 (the candidate is a correct substitute in the given sentence) or 0 (otherwise). The features used in the classification are usually similar with that in word sense disambiguation (WSD), including bag of word lemmas in the sentence, n-grams and parts of speech (POS) in a window, etc. There are other models presented for candidate substitute scoring. Glickman et al. (2006) proposed a Bayesian model and a Neural Network model, which estimate the probability of a word may occur in a given context.

3 HIT System

3.1 Candidate Substitute Extraction

In HIT, candidate substitutes are extracted from WordNet. Both synonyms and hypernyms defined in WordNet are investigated. Let *w* be a target word, *pos* the specified POS of *w*. *n* the number of *w*'s synsets defined in WordNet. Then the system extracts *w*'s candidate substitutes as follows:

- Extracts all the synonyms in each synset under *pos*[1] as candidate substitutes.
- If *w* has no synonym for the *i-th* synset ($1 \leq i \leq n$), then extracts the synonyms of its nearest hypernym.
- If *pos* is *r* (or *a*), and no candidate substitute can be extracted as described above, then extracts candidate substitutes under the POS *a* (or *r*).

3.2 Candidate Substitute Scoring

As mentioned above, all words in the given sentence can be used as contextual information in the scoring of candidate substitutes. However, it is obvious that not all context words are really useful when determining a word's substitutes. An example can be seen from Figure 1.

> *She turns eyes <head>***bright***</head> with excitement towards Fiona , still tugging on the string of the minitiature airship-cum-dance card she has just received at the door .*

Figure 1. An example of a context sentence.

In the example above, words *turns*, *eyes*, *with*, and *excitement* are useful context words, while the others are not. The useless contexts may even be noise if they are used in the scoring. As a result, it is important to select context words carefully.

In HIT, we select context words based on the following assumption: useful context words for lexical substitute are those near the target word in the given sentence. In other words, the words that are far from the target word are not taken into consideration. Obviously, this assumption is not always true. However, considering only the neighboring words can reduce the risk of bringing in noise. Besides, Edmonds (1997) has also demonstrated in his paper that short-distance collocations with neighboring words are more useful in lexical choice than long ones.

Let *w* be the target word, *t* a candidate substitute, *S* the context sentence. Our basic idea is that: One can substitute *w* in *S* with *t*, which generates a new sentence *S'*. If *S'* can be found on the web, then the substitute is admissible. The more times *S'* occurs on the web, the more probable the substitute is. In practice, however, it is difficult to find a whole sentence *S'* on the web due to sparseness. Instead, we use fragments of *S'* which contains *t* and several neighboring context words (based on the assumption above). Then the question is how to obtain one (or more) fragment of *S'*.

A window with fixed size can be used here. Suppose *p* is the position of *t* in *S'*, for instance, we can construct a fragment using words from position *p-r* to *p+r*, where *r* is the radius of window.

[1] In this task, four kinds of POS are specified: *n* - noun, *v* - verb, *a* - adjective, *r* - adverb.

However, a fixed *r* is difficult to set, since it may be too large for some sentences, which makes the fragments too specific, while too small for some other sentences, which makes the fragments too loose. An example can be seen in Table 1.

1(a) *But when Daniel turned <head>**blue**</head> one time and he totally stopped breathing.* 1(b) *Daniel turned **t** one time*
2(a) *We recommend that you <head>**check**</head> with us beforehand.* 2(b) *that you **t** with us*

Table 1. Examples of fragments with fixed size.

In Table1, 1(a) and 2(a) are two sentences from the test data of SemEval-2007Task10. 1(b) and 2(b) are fragments constructed according to 1(a) and 2(a), where the window radius is 2 and *t* denotes any candidate substitute of the target word. It is obvious that 1(b) is a rather strict fragment, which makes it difficult to find sentences containing it on the web, while 2(b) is quite loose, which can hardly constrain the semantics of *t*.

Having considered the problem above, we propose a rule-based method that constructs fragments with varied lengths. Let F_t be a fragment containing *t*, the construction rules are as follows:

Rule-1: F_t must contain at least two words besides *t*, at least one of which is non-stop word.

Rule-2: F_t does not cross sub-sentence boundary (“,”).

Rule-3: F_t should be the shortest fragment that satisfies Rule-1 and Rule-2.

According to the rules above, we construct at most three fragments for each *S’*: (1) *t* occurs at the beginning of F_t, (2) *t* occurs in the middle of F_t, and (3) *t* occurs at the end of F_t. Here we have another constraint: if one constructed fragment *F1* is the substring of *F2*, then *F2* is removed. Please note that the morphology is not taken into account when we construct queries.

For the sentence 1(a) and 2(a) in Table 1, the constructed fragments are as follows:

For 1(a): *Daniel turned **t***; ***t** one time*; *turned **t** one*
For 2(a): *recommend that you **t***; ***t** with us beforehand*

Table 2. Examples of the constructed fragments

To score a candidate substitute, we replace “*t*” in the fragments with each candidate substitute and use them as queries, which are then fed to Google. The score of *t* is computed according to the counts of retrieved snippets:

$$Score_{WebMining}(t) = \frac{1}{n}\sum_{i=1}^{n} count(Snippet(F_t i)) \qquad (1)$$

where *n* is the number of constructed fragments, $F_t i$ is the *i-th* fragment (query) corresponding to *t*, and *count*(*Snippet*($F_t i$)) is the count of snippets retrieved by $F_t i$.

All candidate substitutes with scores larger than 0 are ranked and the first 10 substitutes are retained for the *oot* subtask. If the number of candidates whose scores are larger than 0 is less than 10, the system ranks the rest of the candidates by their frequencies using a word frequency list. The spare capacity is filled with those candidates with largest frequencies. For the *best* subtask, we simply output the substitute that ranks first in *oot*.

3.3 Detection of Multiwords

The method used to detect multiword in the HIT system is quite similar to that employed in the baseline system. We also use WordNet to detect if a multiword that includes the target word occurs within a window of 2 words before and 2 words after the target word.

A difference from the baseline system lies in that our system looks up WordNet using longer multiword candidates first. If a longer one is found in WordNet, then its substrings will be ignored. For example, if we find “*get along with*” in WordNet, we will output it as a multiword and will not check “*get along*” any more.

4 Results

Our system is the only one that participates all the three subtasks of Task10, i.e., *best*, *oot*, and *mw*. The evaluation results of our system can be found in Table 3 to Table 5. Our system ranks the fourth in the *best* subtask and seventh in the *oot* subtask.

We have analyzed the results from two aspects, i.e., the ability of the system to extract candidate substitutes and the ability to rank the correct substitutes in front. There are a total of 6,873 manual substitutes for all the 1,710 items in the gold standard, only 2,168 (31.54%) of which have been extracted as candidate substitutes by our system. This result suggests that WordNet is not an appropriate source for lexical substitute extraction. In the future work, we will try some other lexical resources, such as the Oxford American Writer Thesaurus and Encarta. In addition, we will also try the method that automatically constructs lexical resources, such as the automatic clustering method.

Further analysis shows that, 1,388 (64.02%) out of the 2,168 extracted correct candidates are ranked in the first 10 in the *oot* output of our system. This suggests that there is a big space for our system to improve the candidate scoring method. In the future work, we will consider more and richer features, such as the syntactic features, in candidate substitute scoring. Furthermore, A disadvantage of this method is that the web mining process is quite inefficient. Therefore, we will try to use the Web 1T 5-gram Version 1 from Google (LDC2006T13) in the future.

	P	R	ModeP	ModeR
OVERALL	11.35	11.35	18.86	18.86
Further Analysis				
NMWT	11.97	11.97	19.81	19.81
NMWS	12.55	12.38	19.93	19.65
RAND	11.81	11.81	20.03	20.03
MAN	10.81	10.81	17.53	17.53
Baselines				
WORDNET	9.95	9.95	15.58	15.58
LIN	8.84	8.53	14.69	14.23

Table 3. *best* results.

	P	R	ModeP	ModeR
OVERALL	33.88	33.88	46.91	46.91
Further Analysis				
NMWT	35.60	35.60	48.48	48.48
NMWS	36.63	36.63	49.33	49.33
RAND	33.95	33.95	47.25	47.25
MAN	33.81	33.81	46.53	46.53
Baselines				
WORDNET	29.70	29.35	40.57	40.57
LIN	27.70	26.72	40.47	39.19

Table 4. *oot* results.

	Our System		WordNet BL	
	P	R	P	R
detection	45.34	56.15	43.64	36.92
identification	41.61	51.54	40.00	33.85

Table 5. *mw* results.

Acknowledgements

This research was supported by National Natural Science Foundation of China (60575042, 60503072, 60675034).

References

Barzilay Regina and McKeown Kathleen R. 2001. Extracting paraphrases from a Parallel Corpus. In *Proceedings of ACL/EACL*.

Bolshakov Igor A. and Gelbukh Alexander. 2004. Synonymous Paraphrasing Using WordNet and Internet. In *Proceedings of NLDB*.

Dagan Ido, Glickman Oren, Gliozzo Alfio, Marmorshtein Efrat, Strapparava Carlo. 2006. Direct Word Sense Matching for Lexical Substitut*ion. In Proceedings of ACL*.

Edmonds Philip. 1997. Choosing the Word Most Typical in Context Using a Lexical Co-occurrence Network. In *Proceedings of ACL*.

Fellbaum Christiane. 1998. WordNet: An Electronic Lexical Database. MIT Press, Cambridge, MA.

Glickman Oren, Dagan Ido, Keller Mikaela, Bengio Samy. 2006. Investigating Lexical Substitution Scoring for Subtitle Generation. In *Proceedings of CoNLL*.

Kauchak David and Barzilay Regina. 2006. Paraphrasing for Automatic Evaluation. In *Proceedings of HLT-NAACL*.

Langkilde I. and Knight K. 1998. Generation that Exploits Corpus-based Statistical Knowledge. *In Proceedings of the COLING-ACL*.

Lin Dekang. 1998. Automatic Retrieval and Clustering of Similar Words. In *Proceedings of COLING-ACL*.

Smeaton Alan F., Kelledy Fergus, and O'Donell Ruari. 1994. TREC-4 Experiments at Dublin City University: Thresholding Posting Lists, Query Expansion with WordNet and POS Tagging of Spanish. *In Proceedings of TREC-4*.

Wu Hua and Zhou Ming. 2003. Optimizing Synonym Extraction Using Monolingual and Bilingual Resources. In *Proceedings of IWP*.

Zhao Shiqi, Liu Ting, Yuan Xincheng, Li Sheng, and Zhang Yu. 2007. Automatic Acquisition of Context-Specific Lexical Paraphrases. In *Proceedings of IJCAI-07*.