Web Mining for Lexical Context-Specific Paraphrasing

Shiqi Zhao, Ting Liu, Xincheng Yuan, Sheng Li, Yu Zhang

Information Retrieval Laboratory, School of Computer Science and Technology,
Box 321, Harbin Institute of Technology, Harbin, P.R. China, 150001,
{zhaosq, tliu, xcyuan, lis, yzhang}@ir.hit.edu.cn

Abstract. In most applications of paraphrasing, contextual information should be considered since a word may have different paraphrases in different contexts. This paper presents a method that automatically acquires lexical context-specific paraphrases from the web. The method includes two main stages, candidate paraphrase extraction and paraphrase validation. Evaluations were conducted on a news title corpus whereby the context-specific paraphrasing method was compared with the Chinese synonymous thesaurus. Results show that the precision of our method is above 60% and the recall is above 55%, which outperforms the thesaurus significantly.

Keywords: context-specific paraphrasing, web mining.

1 Introduction

Lexical paraphrasing aims to acquire paraphrases of words, which is elementary but very important in many NLP applications. For instance, in Question Answering (QA), paraphrases should be detected in question and answer sentences so that the exact answers can be pinpointed. In automatic evaluation of Machine Translation (MT), lexical paraphrases need to be recognized in order to evaluate the systems' translation results more accurately. In Information Extraction (IE), paraphrases of the words in IE patterns should be identified so as to extract the required information from texts.

Two broad approaches to lexical paraphrasing have dominated the literature. One approach acquires paraphrases from dictionaries, such as WordNet in English [2], [6] and Tongyici Cilin in Chinese [7]. The other approach collects lexical paraphrases from monolingual or bilingual corpora. Lin identified words with similar meaning by measuring the similarity of the contextual words [8]. Barzilay and McKeown extracted paraphrases from a corpus of multiple English translations of the same source text [3]. Bannard and Callison-Burch derived paraphrases using bilingual parallel corpora [1]. Wu and Zhou extracted lexical paraphrases with multiple resources, including a monolingual dictionary, a bilingual corpus, and a large monolingual corpus [9].

These methods facilitate the acquisition of paraphrases. However, none of them specify the contexts in which the derived paraphrases can be adapted. Recently, topic adaptation for paraphrasing has been researched. For example, Kaji and Kurohashi selected lexical paraphrases according to different topics [5]. However, the topics are limited and predefined rather than any given context.

This paper addresses the problem of context-specific paraphrasing. Here, a specific context means a sentence in which a word occurs. A new web mining method is presented to extract lexical context-specific paraphrases. In our method, if a word occurs in different sentences, different paraphrases should be extracted according to each sentence.

2 Method

2.1 Candidate Paraphrase Extraction

Two stages are included: candidate paraphrase extraction and paraphrase validation. The method for candidate paraphrase extraction is based on two principles. The first is authors on the web create information independently, thus their "vocabularies" vary greatly [4]. In other words, if a concept is widely discussed on the web, then various expressions (lexical paraphrases) will be found. The other principle is that lexical paraphrases play similar syntactic roles in sentences, which indicates that paraphrases of a given word *w* in sentence *S* can be derived by extracting words whose syntactic roles are similar with *w*. Three main steps are included in candidate paraphrase extraction:

Step1: Query *S* on the web and retrieve similar sentences. In this step, the sentence *S* is searched on the web using Baidu. From the retrieved snippets, sentences whose similarities with *S* exceed a predefined threshold T_{CE} are retained as candidate sentences. Word overlapping rate (WOR) is used here for computing the similarity between *S* and any candidate sentence *S'*:

$$WOR(S,S') = \frac{|WS(S) \cap WS(S')|}{\max(|WS(S)|,|WS(S')|)} \qquad \textbf{(1)}$$

where "*WS(.)*" denotes the set of words in a sentence. "|.|" denotes the cardinality of a set.

Step2: Extract candidates according to syntactic similarity. In this step, sentence *S* and all the candidate sentences are first parsed by a Chinese dependency parser. In a dependency result, two words and their dependency relation are represented as a triple. For example, "<他, SBV, 喜欢>" is a triple. The criterion shown in Fig. 1 is used for extracting candidate paraphrases.

Step3: Filter candidates using ECilin. HIT IR-Lab Tongyici Cilin (Extended) (ECilin for short), [1] is utilized here for filtering incorrect candidates. ECilin is a Chinese synonym dictionary. In ECilin, each word has a sense code and all sense codes are organized into a hierarchy that contains five levels. At the first level, words

[1] ECilin is an extended version of TongyiciCilin, which was developed by Information Retrieval Lab of Harbin Institute of Technology (http://www.ir-lab.org/).

are classified into 12 classes, while at the fifth level thousands of classes are formed with synonyms put in the same class. In this stage, the first level of ECilin is used for candidate filtering. In detail, if *w'* and *w* are not in the same class defined in ECilin's first level, then *w'* is filtered. In addition, if a candidate is a stopword, then it is filtered.

Given:

S: original sentence;
S': candidate sentence;
$DT(S)$: dependency tree of S;
$DT(S')$: dependency tree of S';
$<w_1, rel, w_2>$: a triple in $DT(S)$;
$<w_1', rel', w_2'>$: a triple in $DT(S')$.

Criterion:

If $rel=rel'$ and $w_2=w_2'$, then w_1' is extracted as a candidate paraphrase of w_1.
If $w_1=w_1'$ and $rel=rel'$, then w_2' is extracted as a candidate paraphrase of w_2.

Fig. 1. Criterion for candidate paraphrase extraction

2.2 Paraphrase Validation

Though the obtained candidates are filtered, there still remain a lot of incorrect candidates. Therefore, a method for validating candidate paraphrases is necessary. Let $w_1, \cdots, w_n$ be n candidate paraphrases of word w in sentence S. We generate n new sentences $S_1, \cdots, S_n$ first, in which S_i $(1 \leq i \leq n)$ is generated by replacing w in S with candidate w_i. Intuitively, if searching S and S_i using a search engine can retrieve similar results (snippets), then one can say that S and S_i are similar in meaning. Accordingly, word w and w_i are similar in this specific context.

2.2.1 Assumption for Paraphrase Validation

For word w in sentence S (and each candidate w_i in S_i), a pseudo document $PD_S(w)$ ($PD_{Si}(w_i)$) is constructed. First, Search S (S_i) in Baidu, and obtain top 100 retrieved snippets. Then, sentences containing w (w_i) are extracted from the snippets. $PD_S(w)$ ($PD_{Si}(w_i)$) is constructed using these extracted sentences. In this work, paraphrases are validated based on the following assumption:

Assumption: Given w in S, and a candidate paraphrase w_i in S_i, if the similarity between their pseudo documents $PD_S(w)$ and $PD_{Si}(w_i)$ exceeds a predefined threshold T, then w_i is validated as w's paraphrase within the specific sentence S.

2.2.2 Similarity Measurements for Pseudo Documents

According to the assumption, a similarity measurement is needed for computing similarities between pseudo documents. Here, two different similarity measurements are investigated: VSM-based similarity and syntactic similarity.

VSM-based similarity (VSMSim): Given two pseudo documents $PD_{S1}(w_1)$ and $PD_{S2}(w_2)$. In VSM, they are represented as vectors V_1 and V_2, in which the weight of each word is calculated using a *tf·itf* heuristic:

$$tf \cdot itf(w, PD) = tf(w, PD) \times \log \frac{\max(tf(w', C_{CD}))}{tf(w, C_{CD})} \qquad (2)$$

where *tf(w, PD)* denotes the term frequency of word *w* in pseudo document *PD*. $tf(w, C_{CD})$ is *w*'s term frequency counted on a China Daily Corpus (C_{CD}). $max(tf(w', C_{CD}))$ is the largest term frequency obtained on the corpus. The VSM-based similarity is calculated as the cosine similarity between V_1 and V_2:

$$VSMSim(PD_{S1}(w_1), PD_{S2}(w_2)) = \cos(V_1, V_2) = \frac{V_1 \bullet V_2}{\|V_1\| \|V_2\|} \qquad (3)$$

where "$\bullet$" denotes inner product. "$\|.\|$" denotes the length of a vector.

Syntactic similarity (SYNSim): In order to compute syntactic similarity, $PD_{S1}(w_1)$ and $PD_{S2}(w_2)$ are first parsed using the dependency parser described above. The syntactic similarity of pseudo documents is calculated with the same method as described in [8], as is rewritten in Equation (4). The similarity is calculated through the surrounding contextual words which have dependency relations with the investigated words according to the parsing results.

$$SYNSim(PD_{S1}(w_1), PD_{S2}(w_2)) = \frac{\sum_{(rel,w) \in T(w_1) \cap T(w_2)} (I(w_1, rel, w) + I(w_2, rel, w))}{\sum_{(rel,w) \in T(w_1)} I(w_1, rel, w) + \sum_{(rel,w) \in T(w_2)} I(w_2, rel, w)} \qquad (4)$$

where $T(w_i)$ denotes the set of words that have the dependency relation *rel* with w_i.

$$I(w_i, rel, w_j) = \log \frac{p(w_i, rel, w_j)}{p(w_i \mid rel) p(w_j \mid rel) p(rel)} \qquad (5)$$

3 Results and Analysis

3.1 Data and Metrics

In the experiments, a corpus of news titles is chosen as test data. That is mainly because in many applications, such as QA, IE, and multi-document summarization, the words and sentences to be paraphrased are usually from news articles. The news titles are collected from "sina news (http://news.sina.com.cn/)". All titles in the "important news" section from March 15, 2006 to April 5, 2006 are downloaded. 257 titles are left after removing duplications.

The metrics are precision, recall, and f-measure. Let $M_1, \ldots, M_T$ be T paraphrasing methods to be compared. N is the number of sentences in test data. n_i is the number of words in the *i-th* sentence that can be paraphrased by all the T methods. nt_{ij} is the number of acquired paraphrases for the *j-th* paraphrased word in the *i-th* sentence using method M_t ($1 \leq t \leq T$). mt_{ij} is the number of correct paraphrases (judged manually) in the nt_{ij} paraphrases. Precision of method M_t is defined as:

$$precision(M_t) = \frac{\sum_{i=1}^{N}\sum_{j=1}^{n_i}\frac{mt_{ij}}{nt_{ij}}}{\sum_{i=1}^{N} n_i} \qquad (1 \leq t \leq T) \qquad \textbf{(6)}$$

Recall is difficult to calculate since it is impossible to enumerate all paraphrases that a word has within a context. Therefore, an approximate approach is used to calculate recall of each method. Specifically, for the *j-th* paraphrased word in the *i-th* sentence, all its correct paraphrases acquired by the T methods are put together (with duplication removed). Let m_{ij} be the total number. Then we assume that m_{ij} is the number of paraphrases that the word can really have within this specific sentence. Recall of method M_t is defined as:

$$recall(M_t) = \frac{\sum_{i=1}^{N}\sum_{j=1}^{n_i}\frac{mt_{ij}}{m_{ij}}}{\sum_{i=1}^{N} n_i} \qquad (1 \leq t \leq T) \qquad \textbf{(7)}$$

The f-measure of method M_t is defined as:

$$f - measure(M_t) = \frac{2 \times precision(M_t) \times recall(M_t)}{precision(M_t) + recall(M_t)} \qquad (1 \leq t \leq T) \qquad \textbf{(8)}$$

3.2 Experimental Results

In the experiments, four methods are completed and compared, including: (1) M_{ECilin}: the method that extracts paraphrases using the fifth level (synonyms) of ECilin. (2) $M_{CSP\text{-}Candi}$: the context-specific paraphrasing (CSP) method that extracts candidate paraphrases as described in Section 2.1 without paraphrase validation. (3) $M_{CSP\text{-}VSM}$: CSP method using VSMSim in paraphrase validation. (4) $M_{CSP\text{-}SYN}$: CSP method using SYNSim in paraphrase validation.

Three thresholds are used in the methods: (1) T_{CE}: threshold for candidate extraction; (2) T_{VSM}: threshold for VSMSim in validation; (3) T_{SYN}: threshold for SYNSim in validation. In the experiments, these thresholds are empirically set 0.30, 0.60, and 0.08 respectively. The comparing results are shown in Table 1:

Table 1. Comparing results of four methods

Method	Precision (%)	Recall (%)	F-measure (%)
M_{ECilin}	12.66	44.27	19.69
$M_{CSP\text{-}Candi}$	48.94	63.41	55.24
$M_{CSP\text{-}VSM}$	63.34	56.06	59.48
$M_{CSP\text{-}SYN}$	66.10	55.28	60.21

3.3 Analysis

3.3.1 Comparison with Method Using ECilin

As can be seen from Table 1, all the three CSP methods, i.e. $M_{CSP\text{-}Candi}$, $M_{CSP\text{-}VSM}$, and $M_{CSP\text{-}SYN}$ outperform M_{ECilin} significantly in precision, recall, and f-measure. Specifically, precision of M_{ECilin} is quite low, which shows that most synonyms defined in ECilin are not paraphrases in specific contexts. For example, the Chinese words "逝世(die)" and "毙命(die)" are synonyms. However, they can never be used in the same context, as the former expresses the death of a personage while the latter is usually used to express the death of an evil person. In contrast, in the CSP methods, these kinds of synonyms cannot be extracted as paraphrases, which makes precision much higher. On the other hand, recall of M_{ECilin} is also much lower than that of the CSP methods, which demonstrates that paraphrases in specific contexts are not necessarily synonyms.

3.3.2 Evaluation of Paraphrase Validation

In this section, the effectiveness of paraphrase validation stage is analyzed. It can be seen from Table 1 that both $M_{CSP\text{-}VSM}$ and $M_{CSP\text{-}SYN}$ outperform $M_{CSP\text{-}Candi}$ greatly in precision, which suggests that the validation methods using distinct similarity measurements are both effective in filtering incorrect candidates. At the same time, it can be found that recall decreases after the validation stage. The decrease indicates that some correct paraphrases are filtered in the validation by mistake. Nevertheless, the increases in f-measure demonstrate the effectiveness of paraphrase validation.

3.3.3 Comparison of Similarity Measurements

This section compares $M_{CSP\text{-}VSM}$ and $M_{CSP\text{-}SYN}$ so as to evaluate the two similarity measurements. It can be seen from Table 1 that $M_{CSP\text{-}VSM}$ and $M_{CSP\text{-}SYN}$ produce similar results. However, $M_{CSP\text{-}SYN}$ is better in precision while $M_{CSP\text{-}VSM}$ is better in recall. The reason why $M_{CSP\text{-}SYN}$ achieves a higher precision is that syntactic information is helpful in filtering incorrect candidates. For example, the sentence "巴林 客轮 沉没 48 人 遇难 (Tourist boat sinks off Bahrain, at least 48 died)" is from our test data.

For the word "客轮(tourist boat)", "海岸(coast)" is extracted as a candidate paraphrase mistakenly. In $M_{CSP\text{-}VSM}$, this incorrect candidate cannot be filtered in validation, since their PDs share a lot of identical words, which makes them quite similar when represented as vectors in VSM. Nevertheless, these two words play different syntactic functions in sentences and have dependency relations with quite different words in PDs. Therefore, in the validation of $M_{CSP\text{-}SYN}$, this incorrect candidate can be easily filtered.

4 Conclusions

This paper proposes a web mining method to automatically acquire context-specific paraphrases. There are three main contributions. First, this work focuses on the problem of context-specific paraphrasing, which has seldom been addressed before. Second, a novel two-stage web mining method is presented. Third, a novel assumption is introduced in paraphrase validation and two different similarity measurements are investigated.

For the presented CSP methods, $M_{CSP\text{-}VSP}$ and $M_{CSP\text{-}SYN}$, precisions are 63.34% and 66.10%, and recalls are 56.06% and 55.28% respectively. The results significantly outperform the method using ECilin.

In the future work, paraphrase validation will be improved. Especially, different similarity measurements will be combined so as to get an optimal compromise of precision and recall.

Acknowledgments. This research was supported by National Natural Science Foundation of China (60435020, 60575042, 60503072). We thank Wanxiang Che and Weigang Li for useful discussions and their valuable comments on this paper.

References

1. Bannard, C., Callison-Burch, C.: Paraphrasing with Bilingual Parallel Corpora. In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL2005), pages 597-604, 2005.
2. Barzilay, R., Elhadad, M.: Using Lexical Chains for Text Summarization. In Proceeding of the Intelligent Scalable Text Summarization Workshop (ISTS'97), pages 10-17, 1997.
3. Barzilay, R., McKeown, K. R.: Extracting Paraphrases from a Parallel Corpus. In Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL2001), pages 50-57, 2001.
4. Cui, H., Wen, J.-R., Nie, J.-Y., Ma, W.-Y.: Probabilistic Query Expansion Using Query Logs. In Proceedings of Eleventh International World Wide Web Conference (WWW2002), pages 325-332, 2002.
5. Kaji, N., Kurohashi, S.: Lexical Choice via Topic Adaptation for Paraphrasing Written Language to Spoken Language. In Proceedings of the Second International Joint Conference on Natural Language Processing (IJCNLP2005), pages 981-992, 2005.
6. Langkilde, I., Knight, K.: Generation that Exploits Corpus-based Statistical Knowledge. In Proceedings of the 17th International Conference on Computational Linguistics and the 36th

Annual Meeting of the Association for Computational Linguistics (COLING-ACL), pages 704-710, 1998.

7. Li, W., Liu, T., Zhang, Y., Li, S., He, W.: Automated Generalization of Phrasal Paraphrases from the Web. In Proceedings of the Third International Workshop on Paraphrasing (IWP2005), pages 49-56, 2005.
8. Lin, D.: Automatic Retrieval and Clustering of Similar Words. In Proceedings of the 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics (COLING-ACL), pages 768-774, 1998.
9. Wu, H., Zhou, M.: Optimizing Synonym Extraction Using Monolingual and Bilingual Resources. In Proceedings of the Second International Workshop on Paraphrasing (IWP2003), pages 72-79, 2003.