ISSN 1000-9825, CODEN RUXUEW E-mail: jos@iscas.ac.cn
Journal of Software, Vol.20, No.7, July 2009, pp.1746−1755 http://www.jos.org.cn
doi: 10.3724/SP.J.1001.2009.00560 Tel/Fax: +86-10-62562563

基于网络挖掘的上下文相关词汇级复述研究*

赵世奇+, 张 宇, 赵 琳, 刘 挺, 李 生

(哈尔滨工业大学 计算机科学与技术学院,黑龙江 哈尔滨 150001)

Context-Dependent Lexical Paraphrasing Based on Web Mining

ZHAO Shi-Qi+, ZHANG Yu, ZHAO Lin, LIU Ting, LI Sheng

(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

+ Corresponding author: E-mail: zhaosq@ir.hit.edu.cn

Zhao SQ, Zhang Y, Zhao L, Liu T, Li S. Context-Dependent lexical paraphrasing based on Web mining. *Journal of Software*, 2009,20(7):1746−1755. http://www.jos.org.cn/1000-9825/560.htm

Abstract: Lexical paraphrasing is the task of extracting word-level paraphrases. Lexical paraphrases should be context dependent since a word may have different paraphrases in distinct contexts. This paper investigates a framework for acquiring context-dependent lexical paraphrases, in which a web mining method is developed for extracting candidate paraphrases and a classification method is introduced in paraphrase validation. Evaluations are carried out on the People's Daily corpus and the results show that: (1) the web mining method performs well in candidate paraphrase extraction, which extracts 2.3 correct paraphrases on average for each test word in each given context sentence; (2) the classifier for paraphrase validation is effective, which achieves an *f*-measure of 0.6023; (3) 75.11% and 98.31% of the paraphrases extracted by our method cannot be recognized by the two widely used context-independent methods, i.e., the thesaurus-based and clustering-based methods respectively. This indicates that the presented context-dependent method is a considerable supplement to the context-independent ones.

Key words: lexical paraphrasing; context dependent; web mining; paraphrase extraction; paraphrase validation

摘 要: 词汇级复述研究旨在为词汇获取复述.词汇级复述是上下文相关的,即对同一个词在不同上下文中应获取不同的复述词.提出了一种获取上下文相关词汇级复述的方法.该方法包括两部分:基于网络挖掘的候选复述词获取以及基于二元分类的复述词确认.在《人民日报》语料库上的实验结果表明:(1) 基于网络挖掘的候选复述词获取方法是切实可行的,平均为每个待复述词在每个给定的上下文句子中获取2.3个正确复述词;(2) 利用二元分类的方法进行复述确认是有效的,其*F*值达到0.6023;(3) 利用此方法抽取得到的复述中,有75.11%和98.31%无法通过两种常用的上下文无关方法,即基于辞典和基于聚类的方法来获得.这证明了所提出的上下文相关复述方法可以有效地补充传统的上下文无关方法.

关键词: 词汇级复述;上下文相关;网络挖掘;复述抽取;复述确认

中图法分类号: TP181 文献标识码: A

* Supported by the National Natural Science Foundation of China under Grant Nos.60503072, 60575042 (国家自然科学基金)

Received 2007-05-09; Accepted 2008-03-14

1 Introduction

Paraphrases are alternative ways of expressing the same meaning. The research on paraphrasing can be classified as lexical (word-level), phrase-level and sentence-level. Examples of each level are shown in Table 1.

Table 1 Examples of paraphrases at each level

Level	Example
Lexical	(a) Coach (b) Bus (note that these two words can be regarded as paraphrases only in certain contexts)
Phrase-Level	(a) Take care of (b) Look after
Sentence-Level	(a) Bell invented the telephone. (b) Telephone was the invention of Bell.

Lexical paraphrasing, which is also known as lexical substitution, is different from synonyms. Synonyms may not be regarded as paraphrases in certain contexts. Meanwhile, paraphrases in some contexts may not be synonyms. Lexical paraphrasing is important in many Natural Language Processing (NLP) applications. For example, in Natural Language Generation (NLG), lexical paraphrasing can help to generate more varied and fluent sentences[1]. In Machine Translation, lexical paraphrases can be used to solve the data sparseness problem[2] and improve the automatic evaluation of the translation results[3]. In Question Answering (QA), lexical paraphrasing can be used to reformulate the terms of a question so as to make it easier to find answers[4].

The most commonly used methods in the acquisition of lexical paraphrases can be categorized as thesaurus-based and clustering-based. In thesaurus-based methods, thesauri such as WordNet[5] in English and TongyiciCilin[6] in Chinese are used for extracting paraphrases[7,8]. Specifically, some researchers extract synonyms as paraphrases while some others use looser definitions, such as hypernyms and hyponyms in WordNet.

In clustering-based methods, the similarity of words is measured. If two words are similar enough, they are clustered and identified as paraphrases. Specifically, various types of similarity measures have been exploited. For instance, in Lin’s method[9], sentences were first parsed with a dependency parser. If two words had dependency relations with similar words, they were clustered as paraphrases. Similarly, Wu and Zhou measured the translation similarity, where if two words had similar translations in another language, they were recognized as paraphrases[10].

There are other methods in lexical and phrase-level paraphrasing. Barzilay and McKeown extracted paraphrases from multiple translations of the same source text using a co-training algorithm[11]. Bannard and Callison-Burch acquired paraphrases from a bilingual corpus, in which a phrase was “translated” to its paraphrases using a foreign language phrase as a pivot[12].

The methods above are effective in constructing paraphrase resources. However, they can only recognize paraphrases that have similar meanings in general. If two words are paraphrases only in specific contexts, these methods may fail. For example, words “培养(train)” and “输送(provide)” have quite different meanings, which cannot be recognized as paraphrases with the above methods. However, they can be regarded as paraphrases in the sentence “她为国家培养了一批批优秀人才(she has trained many qualified personnel for the country)”. In the above methods, context information is generally used in the filtering stage when conducting lexical paraphrasing. To paraphrase a word *w* in sentence *S*, these methods generally first extract *w*’s candidate paraphrases from a pre-constructed paraphrase resource. They then filter the candidates that are not admissible in *S*. Contextual features that are typically used in word sense disambiguation (WSD) are used here[13]. These methods cannot generate new paraphrases according to the given contexts. Therefore, they can be viewed as context-independent methods.

This paper addresses the problem of context-dependent lexical paraphrasing, which aims at extracting different paraphrases of a word according to each given context (in this paper, a context of word *w* is a certain sentence that *w* occurs in). We propose a novel two-stage framework. In the first stage, candidate paraphrases of a word in a given sentence are extracted based on a web mining approach. In the second stage, candidate paraphrases are validated by using a binary classifier. The flowchart of the presented method together with an example can be seen in Fig.1.

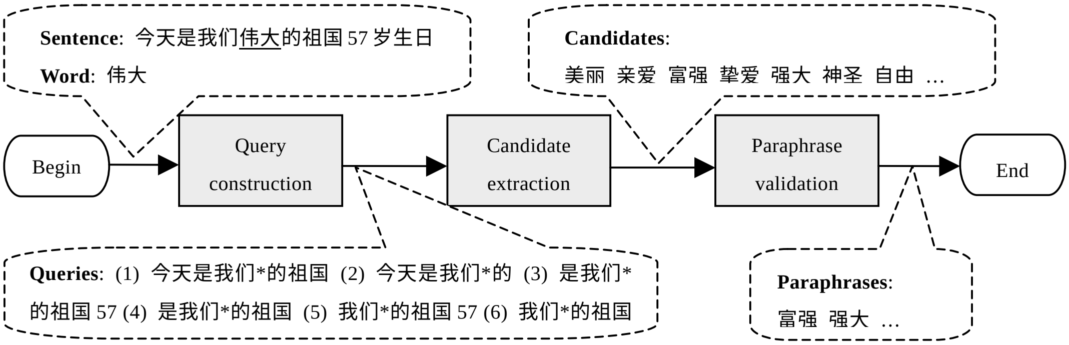
Sentence: 今天是我们伟大的祖国 57 岁生日
Word: 伟大
Candidates:
美丽 亲爱 富强 挚爱 强大 神圣 自由 …
Begin
Query construction
Candidate extraction
Paraphrase validation
End
Queries: (1) 今天是我们*的祖国 (2) 今天是我们*的 (3) 是我们*的祖国 57 (4) 是我们*的祖国 (5) 我们*的祖国 57 (6) 我们*的祖国
Paraphrases:
富强 强大 …

Fig. 1 Flowchart of the method

We conducted experiments on a corpus of texts from People's Daily. Experimental results show that both stages of the method are effective. We also compare our method with the context-independent paraphrasing methods, including the thesaurus-based and clustering-based ones. Results show that our method can extract many context-dependent paraphrases that cannot be acquired by the context-independent methods. Although our method is evaluated in Chinese, it is easy to extend it to other languages.

The rest of the paper is organized as follows: the details of our method, including web mining for candidate paraphrase extraction and binary classification for paraphrase validation, are introduced in Sections 2 and 3. Experiments and results are presented in Section 4. Section 5 draws the conclusions and discusses future work.

2 Mining Candidate Paraphrases from the Web

Firstly, we use an example to explain the feasibility of web mining in paraphrase extraction. Suppose we want to extract paraphrases of the word "伟大(great)" in the sentence "今天是我们伟大的祖国 57 岁生日". We can query "我们*的祖国" in Google (http://www.google.com/). Since Google supports fuzzy search, the retrieved snippets contain the following fragments: "我们富强的祖国", "我们强大的祖国", "我们亲爱的祖国", etc. The words "富强(prosperous and strong)", "强大(powerful)", and "亲爱(dear)" are candidate paraphrases of "伟大(great)" in the given sentence. From the candidates, we can further choose the correct paraphrases.

The web mining approach has been used in answer pattern paraphrasing in QA[14], in which patterns expressing the same relation are mined on the web. The difference between our method and theirs is that we extract lexical paraphrases while they extract paraphrasing patterns. To extract candidate paraphrases of w in S, the web mining approach can be divided into two steps: the first is constructing query Q_{w_S} using w and S. The second is querying Q_{w_S} in Google and retrieving candidate paraphrases.

2.1 Query construction

The paraphrases of w in S are determined by the context words in S. However, not all context words are useful in determining w's paraphrases. The key problem is how to choose useful context words and construct queries for web mining. In this work, given a word/sentence pair (w,S), we construct queries based on two hypotheses:

H1: Context words that are close to w in S are more useful than that far from w.

H2: Among the contextual words, content words (such as none, verb, and adjective) are more useful than function words (such as preposition and conjunction).

Accordingly, we use a rule-based approach in constructing queries. The constructing rules include:

R1: A query Q_{w_S} must be a continuous word sequence in *S* that contains *w*.

R2: Q_{w_S} must contain at least *N*1 content words besides *w*.

R3: Q_{w_S} must contain at most *N*2 words besides *w*, in which at most *N*3 are content words.

R4: If *w* is not the first (or last) word in *S*, *w* should have at least one word in front of (or behind) it in Q_{w_S}.

R5: Replace *w* with wildcard "*" in Q_{w_S}.

R6: Put Q_{w_S} in quotes.

Here we explain the rules briefly. R1 is based on H1, which specifies that a query is a window containing *w*. The contextual words outside the window are not considered. R2 is based on H2, which requires that enough content words should be contained in a query. R3 constrains the length of a query since a too-long query can retrieve too few results. R4 is based on an observation that a query having *w* in the middle generally retrieves more accurate results than that having *w* at both ends. R5 is introduced because Google supports fuzzy search and candidate paraphrases can be extracted from the fragments matched with "*". R6 is used to enhance accuracy by restricting the retrieved snippets to those containing all words in the query. In our experiments, we tuned the parameters *N*1, *N*2, and *N*3 using a development set that will be described in Section 4.3. We finally set *N*1=2, *N*2=5, and *N*3=3, since the method can extract most correct candidate paraphrases under this parameter setting.

For a word/sentence pair (*w*,*S*), there may be more than one query constructed using the rules above. In this work, we keep all the constructed queries and treat them equally. For the example above, i.e., the word is "伟大" and the sentence is "今天是我们伟大的祖国 57 岁生日", six queries are constructed, including "今天是我们*的祖国", "今天是我们*的", "是我们*的祖国 57", "是我们*的祖国", "我们*的祖国 57", and "我们*的祖国".

2.2 Candidate paraphrase extraction

For each word/sentence pair (*w*,*S*), we query Google using the queries formulated as described above. If *N* queries are constructed, we need to query Google *N* times, using one query each time. For each query, candidate paraphrases are extracted from the top 100 retrieved snippets (if the total of retrieved snippets is smaller than 100, then all snippets are retained).

Words in the snippets that match "*" are extracted as candidate paraphrases. In detail: (1) if "*" occurs at the beginning of the query, the nearest word on the left is extracted; (2) if "*" occurs at the end of the query, the nearest word on the right is extracted; (3) if "*" occurs in the middle of the query, all words taking the place of "*" are extracted. After that, all candidate paraphrases retrieved using different queries are combined. The frequency (The frequency of a candidate *w*′ means the number of *w*′ in the retrieved snippets) of each is counted and recorded. Here, we filter the candidates using some simple constraints, including: (1) filter the candidates whose parts-of-speech (POS) are different from that of the original word *w*; (2) filter the candidates that are single-character words. This is based on our observation that most of the words that contain only one character are due to errors of word segmentation; (3) filter the candidates that occur only once in the snippets.

3 Binary Classification for Paraphrase Validation

The web mining approach guarantees that an extracted candidate *w*′ and the original word *w* occur in some similar contexts specified by the sentence *S*. Thus many context-specific paraphrases can be acquired. However, there are also plenty of non-paraphrases in the candidates. For instance, in the example of Section 2, "亲爱(dear)" is an incorrect candidate paraphrase of "伟大(great)" within the given sentence.

Although the candidates are filtered as described in the last section, the precision is still fairly low. The manually labeling result shows that only about 15% of the candidates are correct paraphrases. Therefore, a validation stage is necessary. In this work, paraphrase validation is viewed as a binary classification problem, in which an input triple (w,w',S) is classified as "true" if w' is a correct paraphrase of w in S. Otherwise it is classified as "false". Please note that we do not train a classifier for each word as proposed by Dagan *et al.*[13]. Instead, we train a uniform classifier that can classify any input triple.

3.1 Feature selection

In the paraphrase validation based on binary classification, we aim at classifying a pair of words as "paraphrase" or "non-paraphrase". Therefore, we need to exploit features that can reflect the relation or similarity between two words rather than the simple features such as bag-of-words or part-of-speech. In our experiments, we have exploited multiple features and found that the following four features are the most effective in classification.

- F_{CF} (candidate frequency feature): It is easy to understand that a candidate that occurs frequently in the snippets is more likely to be a correct paraphrase. Hence, the candidate frequency value is selected as a feature, which is defined by

$$F_{CF}(w')=\frac{freq(w')}{\max_{w''} freq(w'')} \tag{1}$$

where $freq(w')$ is the frequency of candidate w' in a retrieved snippets. w'' denotes any candidate paraphrase of w. $\max_{w''}freq(w'')$ denotes the maximum frequency over all the candidates of w.

- F_{SS} (semantic similarity feature): The Extended TongyiciCilin (http://ltp.ir-lab.org/Sharing_Plan.htm) (ECilin for short) is a Chinese thesaurus that organizes synonyms as synsets. Each synset is assigned a sense code. All the synsets are organized as a hierarchy of five levels. Given two synsets, the lower their common ancestor node is, the more similar their senses are. In this work, we compute the semantic similarity between w and w' based on ECilin and select it as a feature. The value of F_{SS} feature is computed as follows[16]:

$$F_{SS}(w,w')=\max_{w\in C,w'\in C'}\frac{2\times\log P(C_0)}{\log P(C)+\log P(C')} \tag{2}$$

where C and C' are the synsets containing w and w'. C_0 is the lowest ancestor node that subsumes both C and C'. $P(C)$ is the probability that a randomly selected word belongs to synset C. Here, $P(C)$ can be computed as $P(C)=\sum_{w\in C}P(w)$, where $P(w)$ is the probability of word w that is estimated over the People's Daily corpus (Section 4.1) based on Maximum Likelihood Estimation. Since a word may have more than one sense, the maximum semantic similarity over all the senses is computed as the F_{SS} feature value.

- F_{DS} (distributional similarity feature): There have been researchers that tried to compute words' similarity based on Distributional Hypothesis. The idea is that words that tend to occur in the same contexts tend to have similar meanings. Lin improved this idea by utilizing syntactic contexts, in which the corpus is first parsed using a dependency parser and the context of a given word only contains words having dependency relations with the target word[9]. In our work, the distributional similarity between w and w' is calculated by Lin's definition:

$$F_{DS}(w,w')=\frac{\sum\limits_{(rel,t)\in T(w)\cap T(w')}(I(w,rel,t)+I(w',rel,t))}{\sum\limits_{(rel,t)\in T(w)}I(w,rel,t)+\sum\limits_{(rel,t)\in T(w')}I(w',rel,t)} \tag{3}$$

where t denotes any word. $T(w)$ denotes the set of words which have dependency relation rel with word w. $I(w,rel,t)$ is the weighted mutual information defined by

$$I(w,rel,t)=P(w,rel,t)\log\frac{P(w,rel,t)}{P(w\mid rel)P(t\mid rel)P(rel)} \tag{4}$$

where $P(w,rel,t)$, $P(w|rel)$, $P(t|rel)$, and $P(rel)$ are also estimated over the People’s Daily corpus. The parser used in our experiment is the dependency parser developed by Liu, *et al*.[17].

- F_{TS} (translation similarity feature): Wu and Zhou’s work has demonstrated that translation information is useful for detecting paraphrases[10]. Roughly speaking, if two words have similar translations in another language, they are likely to be paraphrases. Here we use a Chinese-English translation dictionary to compute the translation similarity of two words. Because the dictionary contains no translation probability information, we use a simplified calculation method. Precisely, let *tran*(w) and *tran*(w') be the sets of possible translations of w and w', respectively. If they have at least one translation in common, the translation similarity is 1, otherwise 0. This gives the following:

$$F_{TS}(w,w') = \begin{cases} 1, & \text{if } tran(w) \cap tran(w') \neq \varnothing \\ 0, & \text{if } tran(w) \cap tran(w') = \varnothing \end{cases} \tag{5}$$

3.2 PAUM classifier

As mentioned above, only about 15% of the candidates are correct paraphrases. This makes the input data for the classifier quite unbalanced. The methods dealing with classification with unbalanced data mainly include the one-class SVMs[18], and Perceptron Algorithm with Uneven Margins (PAUM)[19]. We use PAUM as classifier since it outperforms one-class SVM in our experiments. PAUM is an extension of the perceptron algorithm, which is specifically designed to cope with two class problems where positive examples are rare compared with negative ones, as is the case in paraphrase validation.

4 Evaluation

Three experiments were carried out in the evaluation. The first is to evaluate the web mining approach in candidate extraction. The second evaluates the classifier for paraphrase validation. The third compares our method with the context-independent ones, including both thesaurus-based and clustering-based methods.

4.1 Experimental setting

Evaluation of paraphrasing is highly problematic, because no technique or data set is generally recognized. In our previous work[15], we have evaluated a similar method using a test set of news titles. However, news titles are different from normal sentences. Therefore, in this paper, the evaluation is conducted on the People’s Daily corpus (PDC), which consists of People’s Daily news articles from January to June of 1998. Words in PDC have been segmented manually and POS tags have been annotated by human annotators. The PDC corpus contains a total of 6 527 085 tokens and 799 240 sentences (here, the comma is also regarded as a punctuation of sentence end). To evaluate our context-dependent paraphrasing method, we have constructed a test set of word/sentence pairs as (w,S), in which paraphrases of each word w are extracted and evaluated according to the context sentence S. The construction involves two steps:

- **Step 1**. Select test words. We randomly select 300 content words from PDC whose frequencies are higher than 50, comprising 175 verbs, 71 nouns, and 54 adjectives;
- **Step 2**. Select context sentences. For each test word w, we randomly select 4 sentences containing w from PDC. We then filter the sentences in which w’s meaning is ambiguous or unclear. Furthermore, the sentences too similar to others were also filtered. After filtering, 502 sentences were left, with each test word of 1~4 context sentences. Accordingly, we construct 502 word/sentence pairs.

4.2 Evaluation of candidate extraction

In this section, we first evaluate the query construction approach. The result shows that, for 501 of the 502 word/sentence pairs in the test set, our approach can construct at least one query. The only case that query construction failed is because the sentence is too short to contain enough content words. On average, 3.6 queries are constructed for each word/sentence pair.

We queried Google using the constructed queries. From the retrieved snippets, we extracted 7 568 candidate paraphrases. Two annotators were asked to label the candidates. Finally, 1 141 (15.1%) out of the 7 568 were recognized as correct paraphrases. On average, 2.3 correct candidates were extracted for each word/sentence pair.

We have examined some queries that retrieved no correct paraphrases and found two reasons. The first is that the context information provided by the sentence is inadequate, which is not able to constrain the words occurring in the place of “*”. The other is just the opposite. The context is too specific, few candidates can be retrieved.

4.3 Evaluation of paraphrase validation

The inputs to the classifier in paraphrase validation are triples of the form (w,w',S). If w' is a correct paraphrase of w in S, the triple is a positive example. Otherwise, it is negative. Using the test word/sentence pairs and annotated candidates, 7 568 triples were formed, in which 1 141 were positive and 6 427 were negative. The data was divided into three parts, namely, the training, development, and test data. Some statistics of the data are shown in Table 2.

Table 2 Statistics of the data

	Positive	Negative
Training	564	3 169
Test	392	2 146
Development	185	1 112

The metrics used in the evaluation are *precision*, *recall*, and *f-measure*, which are defined by

$$precision = |S_{rp} \cap S_{ap}| / S_{rp} \tag{6}$$

$$recall = |S_{rp} \cap S_{ap}| / S_{ap} \tag{7}$$

$$f\text{-}measure = \frac{2 \times precision \times recall}{precision + recall} \tag{8}$$

where S_{rp} is the set of correct paraphrases recognized by the classifier and S_{ap} is the set of paraphrases manually recognized by the annotators.

The development data is used to optimize parameters of the classifier. In the PAUM classifier, four parameters need to be optimized i.e. the iteration times IT, learning rate LR, positive margin PM, and negative margin NM. In this experiment, we selected the parameters that achieved the highest f-measure on the development data. Specifically, we set IT=100, LR=0.001, PM=5, NM=−0.14. The result on the test data using these parameters is shown in Table 3.

Table 3 Performance of the binary classifier

Precision (%)	Recall (%)	*f*-Measure (%)
60.00	60.46	60.23

4.4 Feature contributions

To investigate the contributions of the four features, we trained and tested the classifier four times, omitting one feature each time (the parameters are the same as above). The results are shown in Table 4.

Table 4 Effect of eliminating each feature

	Precision (%)	Recall (%)	*f*-Measure (%)
No F_{CF}	61.11	44.90	51.76
No F_{SS}	61.27	44.39	51.48
No F_{DS}	61.27	44.39	51.48
No F_{TS}	44.66	57.65	50.33

Comparing the results in Tables 3 and 4, we can find that all the four features are effective for classification. In detail, eliminating F_{CF}, F_{SS}, or F_{DS} resulted in a notable decrease in recall, while eliminating F_{TS} has a large impact on precision. This result is expected since the four features validate the candidate paraphrases from different aspects, which are all crucial for paraphrase validation.

4.5 Error analysis

By analyzing the results, especially the errors, we find the following issues. The first is, although our method can recognize context-dependent paraphrases that have different meanings in general, those paraphrases too dependent on contexts are still difficult to be recognized. For instance, "力量(force)" and "人员(staff)" are paraphrases in sentence "…组织专门力量出版了…". However, the meanings of these two words are too far away. It greatly influences the recall of the classifier. To solve this problem, we can improve the calculation method of the distributional similarity. For the example above, the distributional similarity between "力量" and "人员" is small. However, we can compute the distributional similarity between "专门力量" and "专门人员", which can be larger.

The second issue is that some non-paraphrases have quite similar meanings, which makes it difficult to distinguish them using given contexts. For example, "逮捕(arrest)" and "起诉(charge)" are not paraphrases in sentence "…被检察机关逮捕…". However, their semantic similarity and distributional similarity are large. This problem decreases the effectiveness of the F_{SS} and F_{DS} features and influences the precision of classification. This problem can be solved by computing the substitution probability of two words and using it as a feature. In this example, "逮捕" and "起诉" are similar words, but they can seldom be substituted by each other. The substitution probabilities can be calculated using a monolingual parallel corpus, such as the one used by Barzilay and McKeown[11]. We can compute the probability that the given two words are aligned with each other in the corpus.

Another issue is that although the translation similarity feature F_{TS} proves effective in classification, the feature value computation should be improved. Right now, the F_{TS} feature has Boolean values, which means that the translation similarity of two words are "1" if they share one common translation word. This is problematic. For example, words "处理(process)" and "前进(go ahead)" have 47 and 39 translations respectively. Their translation similarity is "1" just because they have only two common translations "go" and "make". In the future work, we will compute the F_{TS} feature values based on vector space model (VSM), in which translations of a word are represented using a vector and the cosine similarity is computed as the F_{TS} value.

4.6 Comparison with context-independent methods

In this section, we compare our method with two context-independent paraphrasing methods. The purpose is to find out how many paraphrases acquired by our method are context-dependent, which cannot be obtained by the context-independent methods. The methods to be compared include:

Method 1 (**thesaurus-based method**). Method 1 extracted paraphrases from ECilin. In detail, for a test word *w*, we extracted its synonyms in ECilin as its paraphrases. Here, WSD can be conducted to enhance precision if a word has more than one sense. Nonetheless, since we only care about the coverage of the methods in this comparison, WSD was not conducted. If *w* has several senses, all its possible synonyms were got as its paraphrases.

Method 2 (**clustering-based method**). Method 2 followed the method presented by Lin[9]. The PDC corpus was first parsed using a dependency parser. From the parsed corpus, we extracted nouns, verbs, and adjectives that occurred at least 50 times. Then we computed the pairwise similarity between all nouns, all verbs, and all adjectives using the similarity measure in Eq.(3). Two words are defined as a pair of respective nearest neighbors (RNNs) if each is the other's most similar word. Here a pair of RNNs can be regarded as a pair of paraphrases.

From the 395 manually annotated correct paraphrases in the test data, 237 were identified by the binary classifier in the validation stage. Experiments showed that: (1) 178 of the 237 (75.11%) cannot be obtained by Method 1, suggesting that most paraphrases obtained by our method are not synonyms defined in thesauri; (2) 233 of the 237 (98.31%) cannot be obtained by Method 2. It is not surprising since the definition of RNNs is fairly strict. Therefore, the majority of paraphrases extracted by our method are not RNNs. From the results, we find that most paraphrases extracted by our context-dependent method cannot be recognized by the context-independent methods. Table 5 depicts some examples of context-dependent paraphrases obtained by our method.

Table 5 Examples of the extracted context-dependent lexical paraphrases

Word	Paraphrase	Contextual sentence
地方 (Room)	问题 (Problem)	但还有需要改进的地方. (But there is plenty of **room** for improvement).
长大 (Grow up)	生活 (Live)	我的女儿从小跟我的父母一起长大. (My daughter has **grown up** with my parents from childhood).
居住 (Living)	使用 (Usable)	教职工家庭人均居住面积由 1993 年底的 6.9 平方米提高到 1997 年底的 8.45 平方米. (The per-capita **living** area for teachers increased from 6.9 square meters in 1993 to 8.45 square meters in 1997).
培养 (Train)	输送 (Provide)	她为国家培养了一批批优秀人才. (She has **trained** many qualified personnel for the country).
前进 (Go forward)	奋斗 (Struggle)	为后人继续**前进**提供更好更有利的物质和思想条件. (Provide good material conditions, to ensure that successors can continue to **go forward**).
议题 (Issue)	内容 (Content)	会议的主要议题是研讨长江上游水利、环保、能源、生态、经济社会可持续发展等一系列问题. (The main **issues** of the meeting include irrigation works, environmental protection, and zoology, etc).
誉为 (Laud)	视为 (Regard)	被誉为“绿色生命通道”. (It is **lauded** as “Green Life Passage”).

We guess that there must be paraphrases that can be extracted by the context-independent methods but cannot by ours, since in some cases the given contexts are insufficient or ambiguous. However, the experimental results demonstrate that our method is a considerable supplement to the context-independent ones. In the future work, we shall conduct lexical paraphrasing by combining the context-independent and context-dependent methods.

5 Conclusions and Future Work

This paper proposes a method for acquiring context-dependent lexical paraphrases. Our contribution is that, given a word *w* and its context sentence *S*, we can extract *w*’s paraphrases according to *S* by using a web mining approach. In addition, we recast paraphrase validation as a binary classification problem, in which multiple features are combined. The experiment results show that: (1) The web mining approach is effective for extracting candidate paraphrases. Especially the rule-based query construction approach is effectual; (2) The classifier for paraphrase validation can achieve an f-measure of 0.6023 and all the four features are useful; (3) Most paraphrases obtained by our method are context dependent, which cannot be obtained by context-independent methods.

In future work, we will improve the query construction of web mining so as to enhance the accuracy of the extracted candidate paraphrases. We will try to use the words having syntactic relations with the target word to construct queries. Additionally, more features will be investigated in the classification for paraphrase validation. For example, we will compute the language model based score of the sentence that is generated by substituting the paraphrase word and use the score as a feature in paraphrase validation.

References:

[1] Granville R. Controlling lexical substitution in computer text generation. In: Proc. of the COLING. 1984. 381−384.

[2] Callison-Burch C, Koehn P, Osborne M. Improved statistical machine translation using paraphrases. In: Proc. of the HLT-NAACL. 2006. 17−24.

[3] Kauchak D, Barzilay R. Paraphrasing for automatic evaluation. In: Proc. of the HLT-NAACL. 2006. 455−462.

[4] Hermjakob U, Echihabi A, Marcu D. Natural language based reformulation resource and web exploitation for question answering. In: Proc. of the TREC-2002 Conf. 2002. 801−809.

[5] http://wordnet.princeton.edu/

[6] Mei JJ, Zhu YM, Gao YQ, Yin HX. Tong Yi Ci Ci Lin. 2nd ed., Shanghai: Shanghai Lexicographical Publishing House, 1996 (in Chinese).

[7] Langkilde I, Knight K. Generation that exploits corpus-based statistical knowledge. In: Proc. of the COLING/ACL. 1998. 704−710.

[8] Li WG, Liu T, Zhang Y, Li S, He W. Automated generalization of phrasal paraphrases from the web. In: Proc of the IWP. 2005. 49−56.

[9] Lin DK. Automatic retrieval and clustering of similar words. In: Proc. of the COLING/ACL. 1998. 768−774.

[10] Wu H, Zhou M. Optimizing synonym extraction using monolingual and bilingual resources. In: Proc. of the IWP. 2003. 72−79.

[11] Barzilay R, McKeown KR. Extracting paraphrases from a parallel corpus. In: Proc. of the ACL/EACL. 2001. 50−57.

[12] Bannard C, Callison-Burch C. Paraphrasing with bilingual parallel corpus. In: Proc. of the ACL. 2005. 597−604.

[13] Dagan I, Glickman O, Gliozzo A, Marmorshtein E, Strapparava C. Direct word sense matching for lexical substitution. In: Proc. of the COLING/ACL. 2006. 449−456.

[14] Ravichandran D, Hovy E. Learning surface text patterns for a question answering system. In: Proc. of the ACL. 2002. 41−47.

[15] Zhao SQ, Liu T, Yuan XC, Li S, Zhang Y. Automatic acquisition of context-specific lexical paraphrases. In: Proc. of the IJCAI. 2007. 1789−1794.

[16] Lin DK. An information-theoretic definition of similarity. In: Proc. of the ICML. 1998. 296−304.

[17] Liu T, Ma JS, Li S. Chinese dependency parsing model based on lexical governing degree. Journal of Software, 2006,17(9): 1876−1883 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/17/1876.htm

[18] Manevitz LM, Yousef M. One-Class SVMs for document classification. Journal of Machine Learning Research, 2001,2(12):139−154.

[19] Li YY, Zaragoza H, Herbrich R, Shawe-Taylor J, Kandola J. The perceptron algorithm with uneven margins. In: Proc. of the ICML. 2002. 379−386.

附中文参考文献:

[6] 梅家驹,竺一鸣,高蕴琦,殷鸿翔.同义词词林.上海:上海辞书出版社,1996.

[17] 刘挺,马金山,李生.基于词汇支配度的汉语依存分析模型.软件学报,2006,17(9):1876−1883. http://www.jos.org.cn/1000-9825/17/1876.htm

ZHAO Shi-Qi was born in 1981. He is a Ph.D. candidate in Harbin Institute of Technology. His current research area is natural language processing.

ZHANG Yu was born in 1972. He is an associate professor in Harbin Institute of Technology. His research areas are information retrieval and natural language processing.

ZHAO Lin was born in 1983. He is a master candidate in Harbin Institute of Technology. His current research area is in natural language processing.

LIU Ting was born in 1972. He is a professor and Ph.D. supervisor in Harbin Institute of Technology. His research areas are natural language processing and information retrieval.

LI Sheng was born in 1943. He is a professor and Ph.D. supervisor in Harbin Institute of Technology. His research areas are natural language processing, information retrieval and machine translation.