Natural Language Engineering **15** (4): 503–526. © Cambridge University Press 2009
doi:10.1017/S1351324909990155

Extracting paraphrase patterns from bilingual parallel corpora

SHIQI ZHAO[1], HAIFENG WANG[2], TING LIU[1] and SHENG LI[1]

[1]*Harbin Institute of Technology, No. 27 Jiaohua Street, Nangang District, Harbin 150001, China*
[2]*Toshiba (China) Research and Development Center, No. 1, East Chang An Ave., Dongcheng District, Beijing 100738, China*
e-mails: {zhaosq, tliu, lisheng}@ir.hit.edu.cn, wanghaifeng@rdc.toshiba.com.cn

(*Received 16 November 2007; revised 20 August 2008; accepted 6 February 2009*)

Abstract

Paraphrase patterns are semantically equivalent patterns, which are useful in both paraphrase recognition and generation. This paper presents a pivot approach for extracting paraphrase patterns from bilingual parallel corpora, whereby the paraphrase patterns in English are extracted using the patterns in another language as pivots. We make use of log-linear models for computing the paraphrase likelihood between pattern pairs and exploit feature functions based on maximum likelihood estimation (MLE), lexical weighting (LW), and monolingual word alignment (MWA). Using the presented method, we extract more than 1 million pairs of paraphrase patterns from about 2 million pairs of bilingual parallel sentences. The precision of the extracted paraphrase patterns is above 78%. Experimental results show that the presented method significantly outperforms a well-known method called discovery of inference rules from text (DIRT). Additionally, the log-linear model with the proposed feature functions are effective. The extracted paraphrase patterns are fully analyzed. Especially, we found that the extracted paraphrase patterns can be classified into five types, which are useful in multiple natural language processing (NLP) applications.

1 Introduction

Paraphrases are alternative ways to convey the same information (Barzilay and Mckeown 2001). The research of paraphrasing is important in a number of natural language processing (NLP) applications. In question answering (QA), both questions and answer extraction patterns can be paraphrased so as to solve the word mismatch problem between questions and answers (Lin and Pantel 2001; Hermjakob, Echihabi and Marcu 2002; Ravichandran and Hovy 2002; Zhao, Zhou and Liu 2007). In machine translation (MT), paraphrases can be used to alleviate the sparseness of training data (Callison-Burch, Koehn and Osborne 2006) and expand the reference translations in automatic MT evaluation (Lepage and Denoual 2005; Kauchak and Barzilay 2006). In multidocument summarization (MDS), paraphrases are helpful in sentence clustering so as to produce better summaries (Mckeown *et al.* 2002;

Barzilay 2003). In natural language generation (NLG), paraphrases may help to generate more varied and fluent texts (Iordanskaja, Kittredge and Polguère 1991). In recognizing textual entailment (RTE), paraphrase patterns can be used as inference rules when deciding whether a hypothesis *H* can be inferred from a text *T* (Clark *et al.* 2007; Iftene and Balahur-Dobrescu 2007; Marsi, Krahmer and Bosma 2007).

Paraphrase patterns are sets of semantically equivalent patterns. A pattern generally contains two parts, namely, the pattern words and slots. For instance, '*X solves Y*' is a pattern, in which '*solves*' is the pattern word, while '*X*' and '*Y*' are slots that can be instantiated. In some cases, the pattern slots are constrained to be specified parts-of-speech (POS) or named entities (NE). For example, in the pattern '*the* [*JJ*] *man solves* [*NN*],' the slots '[*JJ*]' and '[*NN*]' can only be filled with an adjective and a noun, respectively.

Paraphrase patterns are useful in both paraphrase recognition and generation. In paraphrase recognition, if two text units (phrases or sentences) match a pair of paraphrase patterns and the corresponding slot-fillers are identical, they can be identified as paraphrases. In paraphrase generation, a text unit that matches a pattern *P* can be rewritten using *P*'s paraphrase patterns.

A variety of methods have been proposed on paraphrase patterns extraction (Lin and Pantel 2001; Ravichandran and Hovy 2002; Shinyama, Sekine and Sudo 2002; Barzilay and Lee 2003; Ibrahim, Katz and Lin 2003; Pang, Knight and Marcu 2003; Szpektor *et al.* 2004). However, these methods have some shortcomings. Especially, the precisions of the paraphrase patterns extracted with these methods are limited.

In this paper, we extract paraphrase patterns from bilingual parallel corpora using a pivot approach. We assume that if two English patterns are aligned with the same pattern in another language (Chinese in this paper), they are likely to be paraphrase patterns. It is an extension of the assumption presented in (Bannard and Callison-Burch 2005), which was originally used for learning phrasal paraphrases. In detail, we present a novel technique for inducing English patterns using the dependency trees of the English sentences and extracting aligned pivot patterns from the Chinese sentences. We then propose three models for extracting paraphrase patterns. Specifically, Model 1 is our basic model for computing the paraphrase likelihood between two patterns, which is based on a log-linear framework and uses both maximum likelihood estimation (MLE) based and lexical weighting (LW) based features. Model 2 is an improved model of Model 1, which looses the constraints of Model 1 and extracts more paraphrase patterns. Model 3 is based on the previous two models, which further exploits monolingual word alignment (MWA) based features for paraphrase patterns filtering. We have reported some preliminary results of Model 1 in a previous paper (Zhao *et al.* 2008). This paper is a much extended one, which improves the model and presents more experiments.

Experimental results show that Model 3 is the most effective among the three proposed models, which extracts more than 1 million pairs of paraphrase patterns from about 2 million pairs of bilingual parallel sentences. The precision of the extracted paraphrase patterns exceeds 78%. The results also show that the proposed models outperform the model introduced by Bannard and Callison-Burch (2005).

Table 1. *Paraphrase patterns extracted by (a) Lin and Pantel (2001), (b) Ibrahim* et al. *(2003), (c) Ravichandran and Hovy (2002), (d) Szpektor* et al. *(2004), and (e) Shinyama* et al. *(2002)*

(a) *X solves Y*	(d) *X provides protection against Y*
Y is solved by X	*X reduces Y*
X finds a solution to Y	*X decreases the risk of Y*
. . . . .	*X be cure for Y*
(b) $A_1 \leftrightarrow^S$ *put* $\rightarrow^K$ *on* $\rightarrow^O A_2$	*X a day keeps Y away*
$A_1 \leftrightarrow^S$ *wear* $\rightarrow^O A_2$	*X to combat Y*
. . . . .	. . . . .
(c) *born in <ANSWER> , <NAME>*	(e) *ORGANIZATION decides* ϕ
<NAME> was born on <ANSWER>,	*ORGANIZATION confirms* ϕ
<NAME> (<ANSWER> -	. . . . .
. . . . .	

Additionally, the pivot approach presented in this paper is evidently more effective than a well-known method DIRT (Lin and Pantel 2001). Further analysis of the results indicates that the extracted paraphrase patterns can be classified into five different types, which are useful in various NLP applications.

The proposed method poses two advantages: (a) We are the first to use a pivot approach for extracting paraphrase patterns. Furthermore, we improve the conventional pivot approach so that a lot more paraphrase patterns can be extracted. (b) We propose log-linear models for computing paraphrase likelihood between pattern pairs. We exploit feature functions based on MLE, LW, and MWA, all of which prove effective in the experiments.

The remainder of this paper is organized as follows. In Section 2, we review previous work on paraphrase patterns extraction and the pivot approach for paraphrasing. In Section 3, we present our method in detail, which includes paraphrase patterns extraction (Model 1), augmentation (Model 2), and filtering (Model 3). Section 4 describes the experiments and results. Section 5 draws the conclusion and discusses future work.

2 Related work

Paraphrase patterns have been learned and used in information extraction (IE) and answer extraction of QA. For example, Lin and Pantel (2001) proposed a method called DIRT, which extracted paraphrase patterns from a parsed monolingual corpus based on an extended distributional hypothesis. They assumed that if two paths in dependency trees tend to link the same set of words, their meanings are similar. An example of the obtained paraphrase patterns can be seen in Table 1(a).

Ibrahim *et al.* (2003) presented a similar method, which extracted paraphrase patterns using parsed monolingual parallel sentences. Their basic assumption is that paths between identical 'anchors' in parallel sentences tend to be paraphrases. An example of the learned paraphrase patterns is shown in Table 1(b).

Besides the above methods, some researchers acquired paraphrase patterns from the web. For instance, Ravichandran and Hovy (2002) learned paraphrase patterns

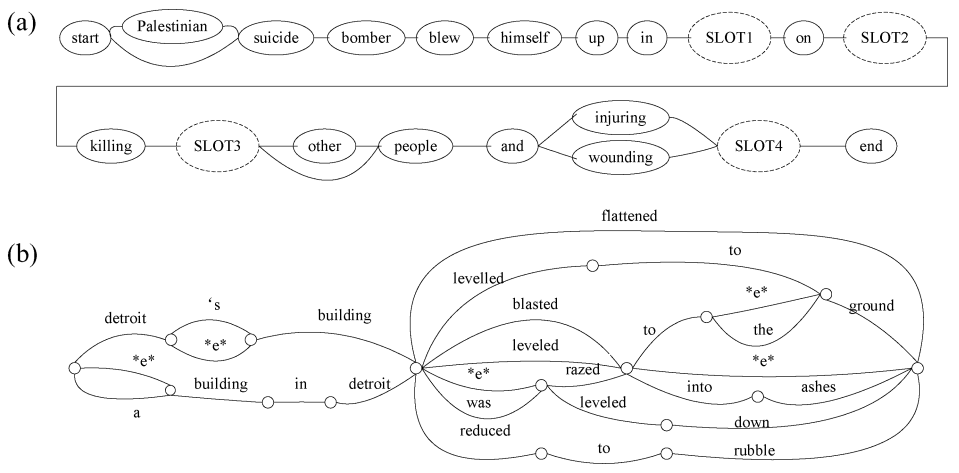

Fig. 1. Paraphrase patterns extracted by Barzilay and Lee (2003) and Pang *et al.* (2003).

for their QA system. They first defined a question taxonomy. Then for each question type, they used hand-crafted 'seeds' as queries to retrieve paraphrase patterns from the web. For example, for the question type '*birthday*,' they used seeds like '*Mozart, 1756*' and retrieved paraphrase patterns as shown in Table 1(c).

Szpektor *et al.* (2004) also exploited the web to obtain paraphrase patterns. Its major difference from Ravichandran and Hovy's method (Ravichandran and Hovy 2002) is that it did not need hand-crafted seeds as queries. Instead, the method automatically learned a set of seeds for each given pattern, and then used the learned seeds as queries to retrieve paraphrases of the given pattern. For example, for the pattern '*X prevent Y*,' the extracted paraphrase patterns are depicted in Table 1(d).

The main disadvantage of the above methods is that the precision of the learned paraphrase patterns is relatively low. Specifically, the precisions reported in Lin and Pantel (2001), Ibrahim *et al.* (2003), and Szpektor *et al.* (2004) are lower than 50%. Ravichandran and Hovy (2002) did not directly evaluate the precision of the extracted paraphrase patterns. However, the performance of their method is heavily dependent on the hand-crafted queries for web mining.

Shinyama *et al.* (2002) presented a method that extracted paraphrase patterns from multiple news articles reporting the same event on the same day. Their method was based on the assumption that NEs should be preserved across paraphrases. Thus the method acquired paraphrase patterns from sentence pairs that share comparable NEs. An example is shown in Table 1(e). The main disadvantage of this method is that it greatly relies on the number of NEs contained in the sentences. The precision of the extracted patterns may sharply decrease if the sentences do not contain enough NEs.

Barzilay and Lee (2003) applied multisequence alignment (MSA) to parallel news sentences and induced paraphrase patterns for generating new sentences (Figure 1a). Pang *et al.* (2003) built finite state automata (FSA) from semantically equivalent translation sets based on syntactic alignment. The learned FSAs could be used in

paraphrase representation and generation (Figure 1b). Obviously, it is difficult for a sentence to match such complicated patterns, especially if the sentence is not sampled from the same domain in which the patterns are extracted.

There are also studies on learning entailment rules in RTE (Bar-Haim *et al.* 2007; Chambers *et al.* 2007; Roth and Sammons 2007), which are similar to paraphrase patterns. Nonetheless, textual entailment is a kind of directional relation, thus the entailment rules are also directional. In other words, the text unites before and after applying the rules may not have the same meaning. For example, when applying the 'relative clauses' rule defined in (Bar-Haim *et al.* 2007), the sentence '*The assailants fired six bullets at the car, which carried Vladimir Skobtsov*' can be transformed into '*The car carried Vladimir Skobtsov*.' These two sentences are clearly not paraphrases. In contrast, most of the paraphrase patterns learned with our method are not directional. Experimental results show that over 96% of the learned paraphrase patterns are symmetric.

Bannard and Callison-Burch (2005) first utilized bilingual corpora for phrasal paraphrase extraction. They assumed that if two English phrases e_1 and e_2 are aligned with the same phrase c in another language, these two phrases may be paraphrases. Specifically, they computed the paraphrase probability in terms of the translation probabilities, as shown in (1):

$$p(e_2|e_1) = \sum_c p_{MLE}(c|e_1)p_{MLE}(e_2|c) \tag{1}$$

Here, $p_{MLE}(c|e_1)$ and $p_{MLE}(e_2|c)$ are the probabilities of translating e_1 to c and c to e_2, which can be computed using the bilingual parallel corpus based on MLE:

$$p_{MLE}(c|e_1) = \frac{count(c, e_1)}{\sum_{c'} count(c', e_1)} \tag{2}$$

where $count(c, e_1)$ is the frequency count that phrases c and e_1 are aligned in the corpus. $p_{MLE}(e_2|c)$ is computed in the same way. This method proved effective in extracting high-quality phrasal paraphrases. As a result, we extend it to paraphrase pattern extraction in this paper.

3 The proposed method

In this paper, we extract English paraphrase patterns using an English–Chinese (E-C) bilingual parallel corpus. In detail, we first induce aligned bilingual patterns from the parallel sentences and then identify English paraphrase patterns using the aligned Chinese patterns as pivots.

The preprocessing of the bilingual parallel corpus includes word alignment and dependency parsing. We conduct word alignment with Giza++ (Och and Ney 2000) in both directions using its default setting and then apply the grow-diag heuristic (Koehn *et al.* 2005) for symmetrization. In this paper, the paraphrase patterns are extracted from dependency trees, thus we parse the English sentences in the corpus with a dependency parser – MaltParser (Nivre *et al.* 2007). Let S_E be an English sentence, T_E the parse tree of S_E, e a word of S_E, we define the subtree and partial subtree following the definitions in (Ouangraoua *et al.* 2007). In detail, a

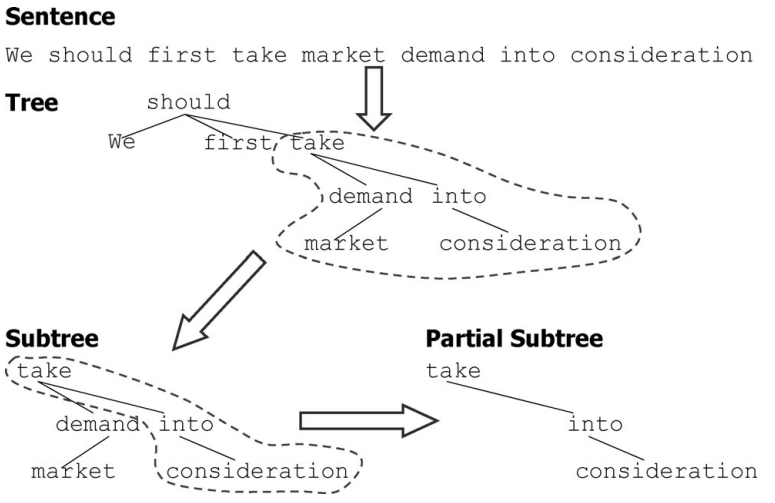

Fig. 2. An example of the parse tree, subtree, and partial subtree for the given sentence.

subtree $ST_E(e)$ is a particular connected subgraph of the tree T_E, which is rooted at e and includes all the descendants of e. A partial subtree $PST_E(e)$ is a connected subgraph of the subtree $ST_E(e)$, which is rooted at e but does not necessarily include all the descendants of e. For instance, for the sentence '*We should first take market demand into consideration*,' the parse tree T_E, subtree $ST_E(take)$, and a partial subtree $PST_E(take)$ are shown in Figure 2.[1]

3.1 Aligned bilingual patterns induction

The induction of the aligned bilingual patterns involves two steps. In Step 1, we induce the English patterns using the subtrees and partial subtrees. Then in Step 2, we extract the pivot Chinese patterns aligning to the English patterns.

Step 1 Inducing English patterns. In this paper, an English pattern $P_E(e)$ is a string comprising words and POS tags. Our intuition for inducing an English pattern is that a partial subtree $PST_E(e)$ can be viewed as a unit that conveys a definite meaning, though the words in $PST_E(e)$ may not be continuous. For example, $PST_E(take)$ in Figure 2 contains words '*take …into consideration*,' which has the meaning of 'considering something.' We may extract '*take X into consideration*' as a pattern, in which '*X*' is a slot. In addition, we can use the words that are in $ST_E(e)$ but not in $PST_E(e)$ (denoted as $ST_E(e)/PST_E(e)$) to constrain the pattern slots. In the example in Figuer 2, the word '*demand*' indicates that the slot X can be filled with a noun. Based on this intuition, we induce an English pattern $P_E(e)$ as in Algorithm 1.[2]

For the example in Figure 2, the generated pattern $P_E(take)$ is '*take* [*NN*] [*NN*] *into consideration*.' The patterns induced in this way are quite specific, since the

[1] A subtree may contain several partial subtrees. In this paper, all the possible partial subtrees are considered when extracting paraphrase patterns.

[2] POS(w_k) in Algorithm 1 denotes the POS tag of w_k.

Algorithm 1: Inducing an English pattern

```
Input: words in ST_E(e) : w_i w_{i+1} ... w_j
Input: P_E(e) = φ
For each w_k (i ≤ k ≤ j)
    If w_k is in PST_E(e)
        Append w_k to the end of P_E(e)
    Else
        Append POS(w_k) to the end of P_E(e)
End For
```

Algorithm 2: Inducing an aligned pivot pattern

```
Input: S_C = t_1 t_2 ... t_n
Input: P_C = φ
For each t_l (1 ≤ l ≤ n)
    If t_l is aligned with w_k in S_E
        If w_k is a word in P_E(e)
            Append t_l to the end of P_C
        If POS(w_k) is a slot in P_E(e)
            Append POS(w_k) to the end of P_C
End For
```

POS of each word in $ST_E(e)/PST_E(e)$ forms a slot. Such patterns are difficult to be matched in applications. We therefore take an additional step to simplify the patterns. Let e_i and e_j be two words in $ST_E(e)/PST_E(e)$, whose POS pos_i and pos_j are slots in $P_E(e)$. If e_i is a descendant of e_j in the parse tree, we remove pos_i from $P_E(e)$. For the example above, the POS of '*market*' is removed, since it is the descendant (modifier) of '*demand*,' whose POS also forms a slot. The simplified pattern is '*take* [*NN*] *into consideration*.'

Step 2 Inducing pivot patterns. For each English pattern $P_E(e)$ induced from the sentence S_E, we extract an aligned Chinese pivot pattern P_C from S_E's Chinese translation S_C. The details can be found in Algorithm 2. Note that the Chinese patterns are not extracted from parse trees. They are only sequences of Chinese words and POS that are aligned with the English patterns.

A pattern may contain two or more slots sharing the same POS. To distinguish them, we assign a number to each slot in the aligned E-C patterns. In detail, the slots having identical POS in P_C are numbered incrementally (i.e., $1, 2, 3\ldots$), while a slot in $P_E(e)$ is assigned the same number as its aligned slot in P_C. The examples of the aligned patterns with numbered slots are illustrated in Figure 3. We can find from the example that two English patterns are aligned with the same Chinese pattern, which can be used as a pivot when extracting English paraphrase patterns.

3.2 Model 1: paraphrase patterns extraction

As mentioned above, if English patterns e_1 and e_2 are aligned with the same pivot pattern c, e_1 and e_2 are likely to be paraphrase patterns. The paraphrase likelihood

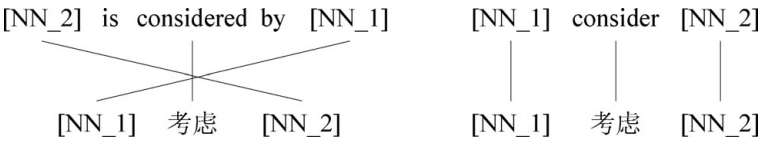

Fig. 3. Aligned bilingual patterns with numbered slots.

can be computed using (1). However, we find that using only the maximum likelihood estimation (MLE) based probabilities can suffer from data sparseness. In order to exploit more and richer information to estimate the paraphrase likelihood, we propose a log-linear model in this section:

$$score(e_2|e_1) = \sum_{c} \exp\left[\sum_{i=1}^{4} \lambda_i h_i(e_1, e_2, c)\right] \tag{3}$$

where $h_i(e_1, e_2, c)$ is a feature function and λ_i is the weight. Four feature functions are used in Model 1, which include

$$h_1(e_1, e_2, c) = score_{MLE}(c|e_1)$$
$$h_2(e_1, e_2, c) = score_{MLE}(e_2|c)$$
$$h_3(e_1, e_2, c) = score_{LW}(c|e_1)$$
$$h_4(e_1, e_2, c) = score_{LW}(e_2|c)$$

Feature functions $h_1(e_1, e_2, c)$ and $h_2(e_1, e_2, c)$ are based on MLE, in which the score $score_{MLE}(c|e)$ is computed as follows:

$$score_{MLE}(c|e) = \log p_{MLE}(c|e) \tag{4}$$

where $p_{MLE}(c|e)$ is the MLE-based paraphrase probability defined in (2). $score_{MLE}(e|c)$ is computed in the same way.

Feature functions $h_3(e_1, e_2, c)$ and $h_4(e_1, e_2, c)$ are based on LW. LW was originally used to validate the quality of a phrase translation pair in MT (Koehn, Och and Marcu 2003). It checks how well the words of the phrases translate to each other. This paper uses LW to measure the quality of aligned bilingual patterns. We define $score_{LW}(c|e)$ as the logarithm of the lexical weight:[3]

$$score_{LW}(c|e) = \frac{1}{n} \sum_{i=1}^{n} \log\left(\frac{1}{|\{j|(i,j) \in a\}|} \sum_{\forall(i,j)\in a} w(c_i|e_j)\right) \tag{5}$$

where a denotes the word alignment between c and e. n is the number of words in c. c_i and e_j are words of c and e. $w(c_i|e_j)$ is computed as follows:

$$w(c_i|e_j) = \frac{count(c_i, e_j)}{\sum_{c_i'} count(c_i', e_j)} \tag{6}$$

where $count(c_i, e_j)$ is the frequency count of the aligned word pair (c_i, e_j) in the bilingual parallel corpus. $score_{LW}(e|c)$ is computed in the same way.

[3] The logarithm of the lexical weight is divided by n so as not to penalize long patterns.

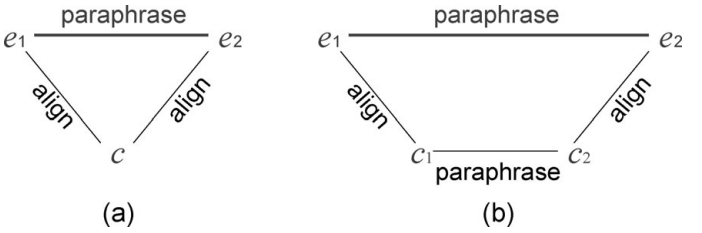

Fig. 4. The illustration of the assumptions in the pivot approach.

In addition, we set a threshold T. If the paraphrase likelihood between e_1 and e_2 exceeds T, e_2 is extracted as the paraphrase of e_1.

3.3 Model 2: paraphrase patterns augmentation

In the conventional pivot approach, only the English patterns that share at least one identical pivot pattern are considered when extracting paraphrases (as shown in Figure 4a). This constraint makes it fail to extract quite a lot of paraphrase patterns. For example, e_1 ('*improve* [*NNS_1*] *'s livelihood*') and e_2 ('*help improve* [*NNS_1*] *'s living standards*') can be viewed as paraphrase patterns, but they do not share identical pivot patterns. Thus Model 1 proposed in Section 3.2 cannot identify them as paraphrases.

However, for the example above, we find that e_1 and e_2 are aligned with c_1 ('改善 [*NNS_1*] 生活') and c_2 ('改善 [*NNS_1*] 的生活'), respectively. Here, c_1 and c_2 exactly mean the same thing.[4] Based on this observation, we can assume that if two English patterns e_1 and e_2 are aligned with a pair of paraphrase pivot patterns c_1 and c_2, e_1 and e_2 are also likely to be paraphrase patterns (Figure 4b). This may be a useful supplement to the original assumption. Based on both the original assumption and the new one, we present Model 2:

$$score(e_2|e_1) = \sum_{c_1,c_2} \exp\left[\sum_{i=1}^{5} \lambda_i h_i(e_1, e_2, c_1, c_2)\right] \tag{7}$$

where c_1 and c_2 are two Chinese patterns, which are either identical or paraphrases. The five feature functions in Model 2 include:

$$\begin{aligned}
h_1(e_1, e_2, c_1, c_2) &= score_{MLE}(c_1|e_1)\\
h_2(e_1, e_2, c_1, c_2) &= score_{MLE}(e_2|c_2)\\
h_3(e_1, e_2, c_1, c_2) &= score_{LW}(c_1|e_1)\\
h_4(e_1, e_2, c_1, c_2) &= score_{LW}(e_2|c_2)\\
h_5(e_1, e_2, c_1, c_2) &= score_{PP}(c_2|c_1)
\end{aligned}$$

The first four feature functions are defined as in Model 1, while $h_5(e_1, e_2, c_1, c_2)$ is the feature function that estimates the paraphrase likelihood of c_1 and c_2. In

[4] The only difference between c_1 and c_2 is the word '的,' a Chinese stopword that can be ignored in this example.

this work, we compute $score_{PP}(c_2|c_1)$ based on Model 1. The process of extracting paraphrase pivot patterns from the aligned bilingual patterns is almost the same as described in Section 3.2. The only difference is that the Chinese pivot patterns become the targets while the English patterns are used as pivots. We set a fairly high threshold when identifying paraphrase pivot patterns so as not to bring in too much noise. In addition, we set $score_{PP}(c_2|c_1) = 0$ when $c_1 = c_2$.[5] More details about extracting paraphrase pivot patterns can be found in Section 4.

3.4 Model 3: paraphrase patterns filtering

As mentioned before, the main disadvantage of the previous paraphrase patterns extraction methods is that the precision is relatively low. Our experiments show that the precision of the paraphrase patterns extracted using the pivot approach is above 60%, which is higher than most of the existing methods. However, it is necessary to further improve the precision if we want to use the paraphrase patterns in NLP applications. We therefore need to introduce new features to filter the noise from the extracted paraphrase patterns.

In this work, we regard the corpus of the extracted paraphrase patterns as a monolingual parallel corpus, on which we can conduct word alignment. We believe that, although the corpus contains noise, the pattern pairs with larger alignment scores are more likely to be correct paraphrase patterns. We thus present Model 3:

$$score(e_2|e_1) = \sum_{c_1,c_2} \exp\left[\sum_{i=1}^{7} \lambda_i h_i(e_1, e_2, c_1, c_2)\right] \tag{8}$$

In Model 3, seven feature functions are used. The first five are the same as defined in Model 2, while $h_6(e_1, e_2, c_1, c_2)$ and $h_7(e_1, e_2, c_1, c_2)$ are two feature functions that estimate the monolingual word alignment (MWA) likelihood between e_1 and e_2. We conduct word alignment on the corpus with Giza++ in both directions and compute the alignment likelihood respectively:

$$h_6(e_1, e_2, c_1, c_2) = score_{MWA}(e_2|e_1)$$
$$h_7(e_1, e_2, c_1, c_2) = score_{MWA}(e_1|e_2)$$

where, $score_{MWA}(e_2|e_1)$ is the logarithm of the word alignment score provided by Giza++. $score_{MWA}(e_1|e_2)$ is computed in the same way. The word alignment model is trained on the corpus of paraphrase patterns extracted using Model 2.

3.5 Parameter estimation

In each model presented above, we need to estimate a set of parameters, which include the weights of the feature functions, i.e., $\lambda_1, \ldots, \lambda_i, \ldots, \lambda_N$, and the threshold T. Since the parameter estimation methods are quite similar for the models, we only introduce the estimation for Model 1 in detail.

[5] Note that, the score computed using the log-linear model is smaller than 0.

In Model 1, there are five parameters need to estimate, i.e., λ_1, λ_2, λ_3, λ_4 in (3) and the threshold T. To estimate the parameters, we first construct a development set. In detail, we randomly sample 7,086 groups of aligned E-C patterns that are obtained as described in Section 3.1. The English patterns in each group are all aligned with the same Chinese pivot pattern. We then extract paraphrase patterns from the aligned patterns as described in Section 3.2. In this process, we set $\lambda_i = 1$ ($i = 1, \ldots, 4$) and assign T a minimum value, so as to obtain all possible paraphrase patterns.

A total of 4,162 pairs of candidate paraphrase patterns were extracted and manually labeled as '1' (correct paraphrase patterns) or '0' (incorrect). Here, two patterns are regarded as paraphrase patterns if for many instantiations of the patterns' slots, the resulting text units are paraphrases. In other words, a pair of patterns is considered as correct paraphrase patterns if the judges could think of reasonable contexts under which it holds. We admit that the annotation and evaluation methodology employed here needs to be improved, as we found that it is hard for judges to think of reasonable contexts. In the future work, we will follow the method presented by Szpektor, Shnarch and Dagan (2007), in which instances are provided when assessing rules (paraphrases or entailment rules). Specifically, the judges assess whether a rule holds under each specific instance. A rule is considered correct only if the percentage of instances assessed as correct is sufficiently high. A similar method has also been proposed in (Pantel *et al.* 2007).

We use gradient descent algorithm (Press *et al.* 1992) to estimate the parameters. For each set of parameters, we compute the precision P, recall R, and f-measure F as: $P = \frac{|set1 \cap set2|}{|set1|}$, $R = \frac{|set1 \cap set2|}{|set2|}$, $F = \frac{2PR}{P+R}$, where $set1$ denotes the set of paraphrase patterns extracted under the current parameters. $set2$ denotes the set of manually labeled correct paraphrase patterns in the development set. We select the parameters that can maximize the f-measure on the development set.

For Model 2, one more parameter needs to be estimated, i.e., λ_5. The estimation is almost the same as above, except that when we construct the development set by sampling groups of aligned bilingual patterns, the English patterns in each group are aligned with either the same Chinese pivot pattern or paraphrase pivot patterns. For Model 3, two more parameters are added in the estimation, namely, λ_6 and λ_7. However, the estimation process is the same as for Model 2.

4 Experiments and analysis

The E-C parallel corpus used in our experiments was constructed by combining several bilingual corpora for machine translation from Linguistic Data Consortium (LDC). The corpora include:

LDC2000T46, LDC2000T47, LDC2002E18, LDC2002T01, LDC2003E07, LDC2003E14, LDC2003T17, LDC2004E12, LDC2004T07, LDC2004T08, LDC2005E83, LDC2005T06, LDC2005T10, LDC2006E24, LDC2006E34, LDC2006E85, LDC2006E92, LDC2006T04, LDC2007T02, LDC2007T09

After filtering sentences that are too long (more than 40 words) or too short (less than 5 words), 2,048,009 pairs of parallel sentences were retained. Statistics show

Table 2. *Comparing Model 1 with B&C Model*

Model	#PP (pairs)	Weighted #PP	Precision
B&C Model	1,015,533	624,248	61.47%
Model 1	1,058,624	740,296	69.93%

that the corpus contains over 47 million English running words and 44 million Chinese running words.

We used two constraints in the experiments to improve the efficiency of computation. First, only subtrees containing no more than 10 words were kept to induce English patterns. Second, although any POS tag can form a slot in the induced patterns, we focused only on three kinds of POS in the experiments, i.e., nouns (tags include NN, NNS, NNP, NNPS), verbs (VB, VBD, VBG, VBN, VBP, VBZ), and adjectives (JJ, JJS, JJR). In addition, we constrained that a pattern must contain at least one content word so as to filter patterns like '*the* [*NN_1*].'

4.1 Evaluation of Model 1

As previously mentioned, Model 1, a log-linear model with MLE-based and LW-based feature functions, is our basic model for paraphrase patterns extraction. In our experiments, we first evaluated Model 1 and the proposed feature functions. In addition, since Model 1 can be regarded as an extension to the model presented by Bannard and Callison-Burch (2005) (B&C Model for short), it is necessary to directly compare these two models in the experiments.[6]

We extracted paraphrase patterns using Model 1 and B&C Model, respectively. From the results of each model, we randomly picked 3,000 pairs of paraphrase patterns to evaluate the precision. The 6,000 pairs were mixed and presented to the human judges, so that the judges cannot know by which model each pair was produced. The sampled patterns were then manually labeled and the precision was computed as described in Section 3.5.

The number of extracted paraphrase patterns (#PP), weighted #PP, and the precision for B&C Model and Model 1 are summarized in Table 2. Here, 'weighted #PP' was defined as #PP multiplied by the precision. We believe that it is more reasonable to compare the weighted #PP among different models, since it approximates the number of correct paraphrase patterns in the corresponding corpus. We can see from Table 2 that when #PPs are comparable for the two models, the precision of the paraphrase patterns extracted using Model 1 is significantly higher than that of B&C Model.

Actually, B&C Model can be viewed as a special case of Model 1. In particular, Model 1 will be equivalent to B&C Model if we set $\lambda_1 = \lambda_2 = 1$ and $\lambda_3 = \lambda_4 = 0$ (i.e., using only the MLE-based features). We therefore believe that the enhancement

[6] We also estimated a threshold T' for B&C Model using the development set. The pattern pairs whose scores based on (1) exceed T' were extracted as paraphrase patterns.

of the precision is mainly due to the contribution of the LW-based features. It is not surprising, since Bannard and Callison-Burch (2005) have pointed out that word alignment error is the major factor influencing the performance of the pivot approach. The LW-based features validate the quality of word alignment and assign low scores to those aligned bilingual pattern pairs that contain incorrect alignments. Therefore, the precision can be enhanced.

As described in Section 3.2, we constrain that the pattern words of an English pattern e must be extracted from a partial subtree. However, we do not have such constraint on the Chinese pivot patterns. In this section, we also investigate whether the performance can be improved if we constrain that the pattern words of a pivot pattern c must also be extracted from a partial subtree.

To conduct the evaluation, we parsed the Chinese sentences of the corpus with a Chinese dependency parser (Liu *et al.* 2006). We then induced English patterns and extracted aligned pivot patterns. For a pair of aligned patterns e and c, we constrain that the pattern words of c must be extracted from a Chinese partial subtree. After that, we extracted paraphrase patterns, from which we sampled 3,000 pairs for evaluating the precision.

The evaluation results show that 736,161 pairs of paraphrase patterns were extracted when we constrained pivot patterns extraction. The precision of the extracted paraphrase patterns is 65.77%. Compared with Model 1, #PP gets much smaller due to the constraint. Meanwhile, the precision of the extracted paraphrase patterns also gets lower. The results suggest that the performance of the method cannot be improved by constraining the extraction of pivot patterns.

4.2 Evaluation of Model 2

As mentioned in Section 3.3, Model 2 improves the original assumption of the pivot approach. Specifically, Model 2 assumes that two English patterns are likely to be paraphrases if they share identical pivot patterns or paraphrase pivot patterns.

To implement Model 2, we first constructed a corpus of paraphrase pivot patterns based on Model 1. In order not to bring in too much noise, the corpus only contained 53,252 pairs of paraphrase pivot patterns with high paraphrase scores. We then sampled 500 pairs from the corpus and manually labeled the data. Labeling results show that the precision is 84.4%. We used this corpus to detect paraphrase pivots when extracting English paraphrase patterns based on Model 2.

Using Model 2, we extracted 1,399,407 pairs of paraphrase patterns, from which 3,000 pairs were sampled for evaluation. The comparison between Model 1 and Model 2 can be found in the first two lines of Table 3. We can see that, compared with Model 1, the weighted #PP of Model 2 increases by 26.94% while the precision merely decreases by 3.98%. This comparison indicates that the extracted paraphrase patterns can be significantly augmented by using paraphrase pivots. It therefore demonstrates that Model 2 is an improvement over the original pivot approach.

In addition, we separately evaluated the augmented paraphrase patterns. The results can be found in the third line of Table 3. It can be seen that the precision of the augmented paraphrase patterns is 58.87%, which is evidently lower than the

Table 3. *The performance of Model 2 and the comparison with Model 1. 'A' means paraphrase patterns augmented using Model 2, 'A_{-T}' means augmenting by simply lowering the threshold*

Model	#PP	Weighted #PP	Precision
Model 1	1,058,624	740,296	69.93%
Model 2	1,399,407	939,702 (+26.94% than Model 1)	67.15% (−3.98% than Model 1)
Model 2 (A)	340,783	200,619	58.87%
Model 1 (A_{-T})	345,178	149,462	43.30%

overall precision. This result is not surprising, as the paraphrase pivots used for augmentation are not totally correct, the precision of which is 84.4%. As expect, we found that the augmented paraphrase patterns are not so obvious as those extracted using Model 1. In other words, the augmented paraphrase patterns generally contain more variations.

We can also increase #PP by simply lowering the threshold for paraphrase patterns extraction. Hence it is necessary to investigate whether Model 2 can outperform this simple option. In our experiments, we lowered the threshold in Model 1 so that the #PP of the augmented paraphrases is comparable with that of Model 2. We then evaluated the augmented paraphrase patterns augmented in this way. The result can be found in the last line of Table 3. We can see that, when the #PPs are comparable, the precision of the paraphrase patterns augmented by lowering the threshold of Model 1 is significantly lower than that augmented using Model 2. This result implies that Model 2 is essential for paraphrase patterns augmentation.

4.3 Evaluation of Model 3

Model 3 filters incorrect paraphrase patterns by introducing feature functions based on monolingual word alignment. In particular, we assume that the pattern pairs with low alignment scores are likely to be incorrect and should be filtered. As mentioned above, the monolingual alignment model was trained using the paraphrase patterns extracted based on Model 2.

Using Model 3, we extracted 1,049,036 pairs of paraphrase patterns, from which we sampled 3,000 pairs for evaluating the precision. The evaluation results are depicted in Table 4. It can be seen that, compared with Model 1, the weighted #PP and the precision are enhanced by 10.94% and 11.95%, respectively. It suggests that Model 3 significantly outperforms Model 1. In addition, compared with Model 2, the weighted #PP of Model 3 is 12.60% lower because of the MWA-based filtering. However, the precision of Model 3 is 16.59% higher. Given that the precision is more important than the coverage in many NLP applications, we can conclude that Model 3 performs better than Model 2. Note that, the corpus of paraphrase patterns on which we trained the monolingual word alignment model is not a clean parallel corpus, as only about 67% of the pattern pairs are correct paraphrases.

Table 4. *The performance of Model 3 and the comparison with Model 1 and Model 2. 'F_{+T}' means filtering the paraphrase patterns by simply increasing the threshold*

Model	#PP	Weighted #PP	Precision
Model 1	1,058,624	740,296	69.93%
Model 2	1,399,407	939,702 (+26.94% than Model 1)	67.15% (−3.98% than Model 1)
Model 3	1,049,036	821,290 (+10.94% than Model 1) (−12.60% than Model 2)	78.29% (+11.95% than Model 1) (+16.59% than Model 2)
Model 2 (F_{+T})	1,051,198	738,467	70.25%

Nevertheless, the experiment results show that the MWA-based feature functions are effective for filtering noise from the extracted paraphrase patterns.

We can also filter the noise and enhance the precision by increasing the threshold for paraphrase patterns extraction. Therefore, we evaluated whether the MWA-based features are more effective than the simple threshold-tuning method. To this end, we increased the paraphrase extraction threshold in Model 2 to an extent that the number of the extracted paraphrase patterns based on Model 2 is comparable with that of Model 3. We then sampled 3,000 pairs and evaluated the precision. The evaluation result is shown in the last line of Table 4. We can see that, after increasing the threshold, the precision is merely improved to 70.25%, which is evidently lower than that achieved by Model 3 (78.29%). This result indicates that the MWA-based feature functions can greatly enhance the precision, which cannot be achieved by simply tuning the threshold.

Note that, all the sampled paraphrase patterns used for evaluation in the experiments above (Section 4.1, 4.2, and 4.3) were split into two parts and labeled by two judges, from which 1,000 pairs were randomly selected and cross annotated for agreement measurement. The agreement between the two judges is 85.4% and the Kappa value is 0.62, which can be regarded as 'substantial agreement.'

4.4 Comparison with DIRT

It is necessary to compare our method with another paraphrase patterns extraction method. In this paper, we compare our method with DIRT (Lin and Pantel 2001), which has been extensively used in many NLP applications, especially in RTE (Clark *et al.* 2007; Iftene and Balahur-Dobrescu 2007; Marsi *et al.* 2007).

As mentioned in Section 2, DIRT learns paraphrase patterns from a parsed monolingual corpus based on an extended distributional hypothesis. In DIRT, a pattern has two slots, i.e., *slotX* and *slotY*. The similarity of two patterns e_1 and e_2 is defined as the geometric average of the similarities of their *slotX* and *slotY* slots:

$$sim(e_1, e_2) = \sqrt{sim(slotX_1, slotX_2) \times sim(slotY_1, slotY_2)} \tag{9}$$

Here, the similarity between a pair of slots is defined as follows:

$$sim(slot_1, slot_2) = \frac{\sum_{w \in T(e_1,s) \cap T(e_2,s)} mi(e_1, s, w) + mi(e_2, s, w)}{\sum_{w \in T(e_1,s)} mi(e_1, s, w) + \sum_{w \in T(e_2,s)} mi(e_2, s, w)} \tag{10}$$

Table 5. *The top-n precision of DIRT*

Method	Top-1	Top-5	Top-10	Top-20
DIRT	80.00%	64.20%	50.40%	39.36%

where s is a slot, w is a filler, $T(e_i, s)$ is the set of words that fill in the s slot of pattern e_i. $mi(e, s, w)$ is the mutual information between e, s, and w:

$$mi(e, s, w) = log \frac{p(e, s, w)}{p(e|s)p(w|s)p(s)} \tag{11}$$

Lin and Pantel (2001) extracted paraphrase patterns from about 1 GB newspaper texts. The extracted paraphrase patterns can be obtained from Pantel's demo.[7] In this paper, we randomly selected 50 input patterns and retrieved at most top 20 paraphrase patterns for each input pattern from the demo. A total of 987 paraphrase patterns were obtained in this way for all the 50 input patterns. The retrieved paraphrase patterns were manually labeled as correct or incorrect by two judges. The criterion for annotation is the same as described in Section 3.5. The agreement between the judges is 95% and the Kappa value is 0.90. We computed top-n precision for the DIRT results as:

$$precision_{top-n} = \frac{\sum_{i=1}^{N} n_i}{N \times n} \tag{12}$$

where N is the number of input patterns having n paraphrase patterns. n_i is the number of correct paraphrase patterns in the top-n paraphrase patterns of the i-th input pattern. The top-1 top-5, top-10, and top-20 precisions are shown in Table 5. We can find that only the top-1 precision is higher than Model 3 (see Table 4). The top-5, top-10, and top-20 precisions are all evidently lower than Model 3.

Here, we also informally compare our evaluation results of DIRT with those reported in previous papers. In particular, in the original paper of Lin and Pantel (2001), the average precision of DIRT (top-40 paraphrase patterns were kept for each input pattern) is 38.39%. In (Szpektor *et al.* 2007), the upper bound and lower bound precisions of DIRT (at most top-20 paraphrase patterns were kept for each input pattern) are 30.5% and 18.6%, respectively.

Although the precisions reported in the papers are different, all of them are fairly low. We believe the main reason is that the extended distributional hypothesis does not hold in many cases. Patterns sharing similar slot-fillers do not necessarily have the same meaning. They may even have the opposite meanings. For example, '*X worsens Y*' and '*X solves Y*' were extracted as paraphrase patterns by DIRT.

There are three main differences between the paraphrase patterns learned using DIRT and our method: (a) in DIRT, a pattern must have two slots, while in our method, the number of pattern slots is not limited; (b) in DIRT, the slot fillers must be nouns, while in our method the slot fillers can be nouns, verbs, and adjectives;

[7] http://demo.patrickpantel.com/Content/LexSem/paraphrase.htm

Table 6. *The statistics and examples of each type of paraphrase patterns*

(1)	Trivial change: 88 (17.60%) *the* [*NNP_1*] *and the president* ↔ [*NNP_1*] *and president* *the development of* [*NN_1*] ↔ *the development in* [*NN_1*] *a limited number of* [*NNS_1*] ↔ *the limited number of* [*NNS_1*] . . . . .
(2)	Phrase replacement: 268 (53.60%) [*NN_1*] *and e-commerce* ↔ [*NN_1*] *and electronic commerce* *to take* [*JJ_1*] *steps* [*VB_1*] ↔ *to adopt* [*JJ_1*] *measures* [*VB_1*] [*NNS_1*] *and other interested parties* ↔ [*NNS_1*] *and other stakeholders* . . . . .
(3)	Phrase reordering: 58 (11.60%) *the* [*NN_1*] *part* ↔ *part of* [*NN_1*] [*NNP_1*] *'s* [*NN_1*] *growth* ↔ *the* [*NN_1*] *growth of* [*NNP_1*] *to strengthen further* [*NN_1*] ↔ *to strengthen* [*NN_1*] *further* . . . . .
(4)	Structural paraphrase: 65 (13.00%) *the* [*NN_1*] *'s residents* ↔ *the inhabitants of* [*NN_1*] *a decline in* [*NN_1*] ↔ [*NN_1*] *reduction* *continue to watch closely* [*NN_1*] ↔ *continue to monitor* [*NN_1*] *closely* . . . . .
(5)	Information +/−: 21 (4.2%) *in promoting the development of* [*NN_1*] ↔ *in promoting* [*NN_1*] *the parties* [*VBN_1*] ↔ *the various parties* [*VBN_1*] *changes that have taken place in* [*NN_1*] ↔ *the changes in* [*NN_1*] . . . . .

(c) in DIRT, the relation between two slots must be between a verb and an object-noun or a small clause, while in our method, there is no constraint of the relation between the slots.

4.5 Analysis of the paraphrase patterns

4.5.1 Analyzing the types of the paraphrase patterns

We sampled 500 pairs of correct paraphrase patterns extracted using Model 3 and analyzed the types. We found that the paraphrase patterns can be classified into five types, which include: (a) trivial change, such as changes of prepositions and articles; (b) phrase replacement; (c) phrase reordering; (d) structural paraphrase, which involves both phrase replacement and phrase reordering; (e) adding or reducing (+/−) information that does not change the meaning. Some statistics and examples of each type are shown in Table 6. Paraphrase patterns in type 2–5 are all useful in NLP applications:

(1) First, over 50% of the paraphrase patterns are in the type of phrase replacement, which can be used in IE pattern reformulation and sentence-level paraphrase generation. Compared with phrasal paraphrases, the phrase replacements in patterns are more accurate due to the constraints of the slots.
(2) The paraphrase patterns in the type of phrase reordering can also be used in IE pattern reformulation and sentence paraphrase generation. Especially, in sentence paraphrase generation, this type of paraphrase patterns can reorder

the phrases in a sentence, which can hardly be achieved by the conventional MT-based generation method (Quirk, Brockett and Dolan 2004).

(3) The structural paraphrase patterns have the advantages of both phrase replacement and phrase reordering. Thus they can be used to generate paraphrases that are more complex and valuable.

(4) The paraphrase patterns in the type of 'information +/−' are useful in sentence compression and expansion. A sentence matching a long pattern can be compressed by paraphrasing it using shorter patterns. Similarly, a short sentence can be expanded by paraphrasing it using longer patterns.

In this experiment, we also labeled and analyzed the correct paraphrase patterns retrieved from the DIRT demo (see Section 4.4). We found that the paraphrase patterns learned with DIRT can also be classified into the 5 types described above. The statistics are as follows: (a) trivial change: 7.38%, (b) phrase replacement: 61.48%, (c) phrase reordering: 10.93%, (d) structural paraphrase: 18.03%, (e) information +/−: 2.19%. We can find that, comparing with Table 6, the distributions of different types of paraphrase patterns are similar. This finding is interesting, as it perhaps reveals the distribution of different types of paraphrases in the real world.

4.5.2 Analyzing the symmetry of the paraphrase patterns

It is interesting to find out how many of the learned paraphrase patterns are symmetric and how many are directional. To this end, we manually labeled the 500 correct paraphrase patterns sampled above and found that 485 (97%) are symmetric and only 15 (3%) are directional. For example, '[*NNP_1*] *reunification* → [*NNP_1*] *unity*' is directional, as the former pattern entails the latter one, but the latter cannot entail the former. This result suggests that the overwhelming majority of the learned paraphrase patterns are symmetric, which is different from entailment rules.

4.5.3 Analyzing the slots of the paraphrase patterns

For the extracted paraphrase patterns using Model 3, we investigate the number and type of the pattern slots. The results are summarized in Table 7 and 8. From Table 7, we can see that more than 93% of the paraphrase patterns contain only one slot. In addition, 6.45% of the paraphrase patterns contain two slots, such as '*give* [*NN_1*] [*NN_2*] ↔ *give* [*NN_2*] *to* [*NN_1*].' Less than 1% of the extracted paraphrase patterns contain three or four slots. This result suggests that our method tends to extract short paraphrase patterns, which is mainly because the data sparseness problem is more serious when extracting long patterns.

From Table 8, we can find that more than 76% of the paraphrase patterns contain noun slots, while over 9% and 17% contain verb slots and adjective slots.[8] This result implies that nouns are the most typical variables in paraphrase patterns.

[8] A pattern may contain more than one type of slots, thus the sum of the percentages is larger than 1.

Table 7. *The statistics of the numbers of pattern slots*

Slot number	#PP	Percentage
1-slot	980,386	93.46%
2-slots	67,676	6.45%
3-slots	966	<1%
4-slots	8	<1%

Table 8. *The statistics of the type of pattern slots*

Slot type	#PP	Percentage
N-slots	801,003	76.36%
V-slots	102,849	9.80%
J-slots	183,737	17.51%

4.5.4 Error analysis

We have observed the errors from the manually labeled samples and found that the errors can be summarized as follows:

(1) **Word alignment errors.** As mentioned above, the performance of our approach depends on the quality of automatic word alignment. We found that nearly 41% of the erroneous paraphrases are due to word alignment errors on the bilingual corpus. For example, the Chinese word '吸引' should be aligned with '*attract*.' However, it is incorrectly aligned with '*attract foreign investors*' in some sentences. Thus our method extracts '*attract foreign investors* [*VB_1*]' and '*attract* [*VB_1*]' as a pair of paraphrase patterns. Our experiments in Section 4.1 have demonstrated that the use of lexical weighting based features can mitigate the influence of word alignment errors. Nonetheless, word alignment quality is still the main factor that influences the paraphrase extraction.

(2) **The basic assumption of the pivot approach.** In the pivot approach, we assume that if two English patterns are aligned with the same Chinese pattern, they are potential paraphrases. This assumption is useful, but does not always hold. This is because when people translate a sentence into a foreign language, not all words or phrases are exactly translated. The meaning of some words may be changed and some words may even be ignored. People can correctly understand the meaning under the help of the contexts. However, two English patterns aligned with the same pivot pattern may have different meanings without certain contexts. For example, our method extracts '[*JJ_1*] *fees*' and '*the* [*JJ_1*] *school fees*' as paraphrases, as '[学费] (*school fees*)' is translated as '*fees*' in some sentences with certain context constraint. We found that over 25% of the errors are in this type.

(3) **Function word errors.** Plenty of English function words, such as '*of*,' '*to*,' '*for*,' do not have translations in Chinese. Hence they are often lost through translation. As a result, English patterns with different function words may align with the same Chinese pattern, though they may have different meanings. For example, both

'*investments by* [*NNS_1*]' and '*investments for* [*NNS_1*]' align with '[*NNS_1*] 投资,' as '*by*' and '*for*' are lost during translation. Our method incorrectly extracts them as paraphrases. Actually, this type of errors can be viewed as a special case of error type 2. We list it separately since this type is quite common, which makes up 12% of the errors.

(4) **Paraphrase pivots errors.** As introduced in Section 3.3, we extract paraphrase pivots so as to augment the collection of English paraphrase patterns. Obviously, the incorrect paraphrase pivots can produce erroneous English paraphrase patterns. The percentage of this type of errors is about 11%.

(5) **Other errors.** There are other kinds of errors, such as spelling mistakes and translation errors in the bilingual corpus, word segmentation errors of the Chinese sentences, tense mismatches (e.g., '*undertaking* [*JJ_1*] *steps*' and '*taken* [*JJ_1*] *measures*'), and compound errors (which reflect more than one type of errors). 11% of the errors are in this type.

4.6 Evaluation within context sentences

In the previous sections, we have evaluated the precision of the paraphrase patterns without considering context information. In this section, we evaluate the paraphrase patterns within specific context sentences.

The test set used here includes 119 English sentences. We parsed the sentences with MaltParser and induced patterns as described in Section 3.1. For each pattern e in sentence S_E, we searched e's paraphrase patterns from the corpus of the extracted paraphrase patterns. The result shows that 101 of the 119 sentences match at least one pattern that can be paraphrased using the extracted paraphrase patterns, the coverage of which is 84.87%. For each matched pattern, we kept top-5 paraphrase patterns. An example can be seen in Figure 5. The extracted paraphrase patterns were labeled by our judges. A candidate paraphrase pattern will be labeled as correct if it can replace the original pattern in the given context sentence.

The annotation result shows that the top-1 precision is 61.54% while the top-5 precision is 53.53%. We can see that the precision is significantly lower than that shown in Table 4. This is because we consider the constraints of the given context sentences here. For example, the pattern '[*JJ_1*] *numbers*' is a possible paraphrase of '[*JJ_1*] *figures*.' However, in the sentence '*Andrew Tsuei would like to go to various parts of the world to interview key figures for his program, but funding is limited*,' '*key figures*' cannot be rewritten as '*key numbers*.'

To capture the context information in pattern replacement, we exploit a language model (LM) feature for reranking the paraphrase patterns in the given context sentences, which is defined as:

$$score(e'|e, S_E) = \lambda score_{LL}(e'|e) + (1 - \lambda) score_{LM}(e'|S_E) \tag{13}$$

Here, $score_{LL}(e'|e)$ denotes the paraphrase score based on Model 3. $score_{LM}(e'|S_E)$ is the LM based score, which can be computed as:

$$score_{LM}(e'|S_E) = \frac{1}{n} log P_{LM}(S'_E) \tag{14}$$

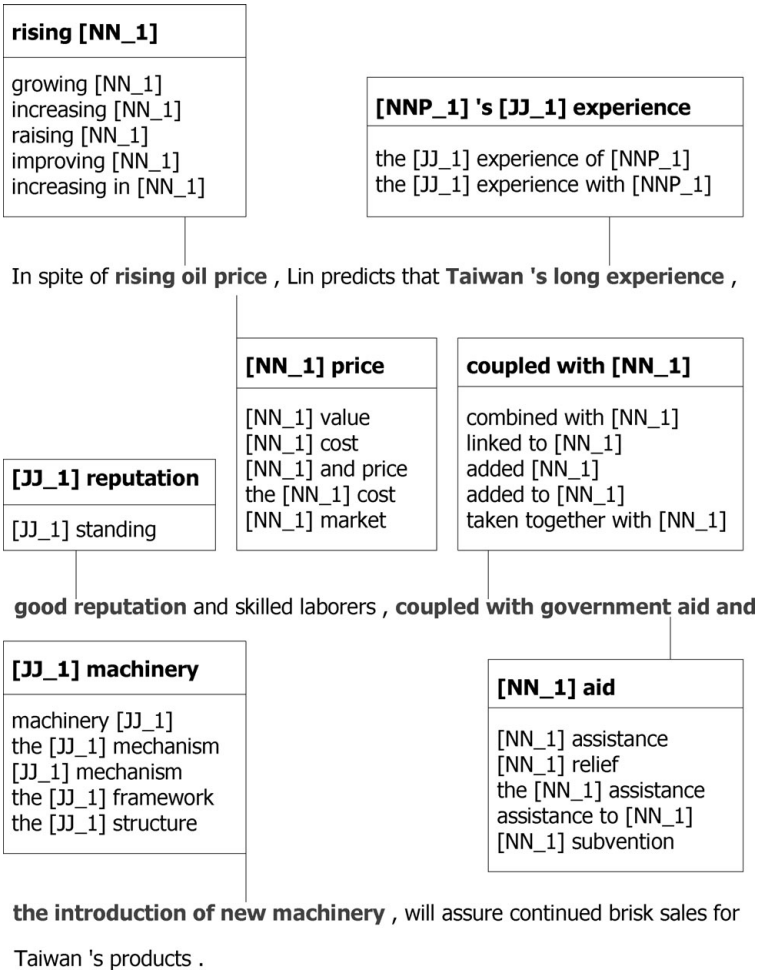

Fig. 5. Pattern replacement for '*In spite of rising oil price, Lin predicts that Taiwan's long experience, good reputation and skilled laborers, coupled with government aid and the introduction of new machinery, will assure continued brisk sales for Taiwan's products.*'

where S'_E is the sentence generated by replacing e in S_E with e'. The language model in the experiment was a tri-gram model trained using the English sentences in the bilingual corpus. We empirically set $\lambda = 0.1$.

The selected best paraphrase patterns in the given context sentences after LM-based reranking were manually labeled. The result shows that the precision of the best paraphrase patterns is 65.00%. It indicates that the LM-based reranking can evidently improve the precision (from 61.54%). Further analysis shows that about 77% of the correct paraphrase substitutes are in the type of phrase replacement.

5 Conclusion and future work

This paper proposes a new method that extracts paraphrase patterns from bilingual corpora based on a pivot approach. We use log-linear models to compute the

paraphrase likelihood and exploit feature functions based on MLE, LW, and MWA. Experimental results show that the pivot approach is effective, which extracts more than 1M pairs of paraphrase patterns from 2M bilingual sentence pairs, the precision of which is above 78%. In addition, the log-linear model with the proposed feature functions significantly outperforms the conventional models. Analysis shows that the extracted paraphrase patterns can be classified into five classes, which are useful in various applications.

In the future work, we wish to exploit more feature functions in the log-linear model. In addition, we will consider following the method presented in (Szpektor and Dagan 2007). In particular, we can first transform the patterns into their canonical forms, and then extract paraphrase patterns. We believe that, by doing this we can improve the statistics of the patterns and mitigate data sparseness problem. Consequently, more interesting paraphrases may be learned. Also, the output may be more compact due to the removal of duplications.

Acknowledgements

This research was supported by National Natural Science Foundation of China (Nos: 60503072, 60575042). We want to thank Lin Zhao, Xiaohang Qu, and Zhenghua Li for their help in the experiments, including running word alignment and dependency parsing, corpus processing, and experiment data labeling.

References

Bannard, C., and Callison-Burch, C. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of ACL*, pp. 597–604, Ann Arbor, MI.

Bar-Haim, R., Dagan, I., Greental, I., Szpektor, I., and Friedman, M. 2007. Semantic inference at the lexical-syntactic level for textual entailment recognition. In *Proceedings of the Workshop on Textual Entailment and Paraphrasing*, pp. 131–6, Prague, Czech Republic.

Barzilay, R. 2003. *Information Fusion for Multidocument Summarization: Paraphrasing and Generation*. Ph.D. Thesis, Columbia University, New York.

Barzilay, R., and Lee, L. 2003. Learning to paraphrase: an unsupervised approach using multiple-sequence alignment. In *Proceedings of HLT-NAACL*, pp. 16–23, Edmonton, Canada.

Barzilay, R., and McKeown, K. R. 2001. Extracting paraphrases from a parallel corpus. In *Proceedings of ACL*. pp. 50–7. Toulouse, France.

Callison-Burch, C., Koehn, P., and Osborne, M. 2006. Improved statistical machine translation using paraphrases. In *Proceedings of HLT-NAACL*, pp. 17–24, New York.

Chambers, N., Cer, D., Grenager, T., Hall, D., Kiddon, C., MacCartney, B., de Marneffe, M., Ramage, D., Yeh, E., and Manning, C. D. 2007. Learning alignments and leveraging natural logics. In *Proceedings of the Workshop on Textual Entailment and Paraphrasing*, pp. 165–70, Prague, Czech Republic.

Clark, P., Murray, W. R., Thompson, J., Harrison, P., Hobbs, J., and Fellbaum, C. 2007. On the role of lexical and world knowledge in RTE3. In *Proceedings of the Workshop on Textual Entailment and Paraphrasing*, pp. 54–9, Prague, Czech Republic.

Hermjakob, U., Echihabi, A., and Marcu, D. 2002. Natural language based reformulation resource and web exploitation for question answering. In *Proceedings of TREC*, Gaithersburg, MD.

Ibrahim, A., Katz, B., and Lin, J. 2003. Extracting structural paraphrases from aligned monolingual corpora. In *Proceedings of IWP*, pp. 57–64, Sapporo, Japan.

Iftene, A., and Balahur-Dobrescu, A. 2007. Hypothesis transformation and semantic variability rules used in recognizing textual entailment. In *Proceedings of the Workshop on Textual Entailment and Paraphrasing*, pp. 125–30, Prague, Czech Republic.

Iordanskaja, L., Kittredge, R., and Polguère, A. 1991. Lexical selection and paraphrase in a meaning-text generation model. In C. L. Paris, W. R. Swartout, and W. C. Mann (eds.), *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, Kluwer, Norwell, MA, pp. 293–312.

Kauchak, D., and Barzilay, R. 2006. Paraphrasing for automatic evaluation. In *Proceedings of HLT-NAACL*, pp. 455–62, New York.

Koehn, P., Axelrod, A., Mayne, A. B., Callison-Burch, C., Osborne, M., and Talbot, D. 2005. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *Proceedings of IWSLT*, Pittsburgh, PA.

Koehn, P., Och, F. J., and Marcu, D. 2003. Statistical phrase-based translation. In *Proceedings of HLT-NAACL*, pp. 127–33, Edmonton, Canada.

Lepage, Y., and Denoual, E. 2005. Automatic generation of paraphrases to be used as translation references in objective evaluation measures of machine translation. In *Proceedings of IWP*, pp. 57–64, Jeju, Korea.

Lin, D., and Pantel, P. 2001. Discovery of inference rules for question answering. *Natural Language Engineering* **7**(4): 343–60.

Liu, T., Ma, J., Zhu, H., and Li, S. 2006. Dependency parsing based on dynamic local optimization. In *Proceedings of CoNLL-X*, pp. 211–15, New York.

Marsi, E., Krahmer, E., and Bosma, W. 2007. Dependency-based paraphrasing for recognizing textual entailment. In *Proceedings of the Workshop on Textual Entailment and Paraphrasing*, pp. 83–8, Prague, Czech Republic.

Mckeown, K. R., Barzilay, R., Evans, D., Hatzivassiloglou, V., Klavans, J. L., Nenkova, A., Sable, C., Schiffman, B., and Sigelman, S. 2002. Tracking and summarizing news on a daily basis with Columbia's newsblaster. In *Proceedings of HLT*, pp. 280–5, San Diego, CA.

Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S., and Marsi, E. 2007. MaltParser: a language-lndependent system for data-driven dependency parsing. *Natural Language Engineering* **13**(2): 95–135.

Och, F., and Ney, H. 2000. Improved statistical alignment models. In *Proceedings of ACL*, pp. 440–7, Hong Kong, China.

Ouangraoua, A., Ferraro, P., Tichit, L., and Dulucq, S. 2007. Local similarity between quotiented ordered trees. *Journal of Discrete Algorithms* **5**(1): 23–35.

Pang, B., Knight, K., and Marcu, D. 2003. Syntax-based alignment of multiple translations: extracting paraphrases and generating new sentences. In *Proceedings of HLT-NAACL*, pp. 181–8, Edmonton, Canada.

Pantel, P., Bhagat, R., Coppola, B., Chklovski, T., and Hovy, E. 2007. ISP: Learning inferential selectional preferences. In *Proceedings of HLT-NAACL*, pp. 564–71, Rochester, NY.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. 1992. *Numerical recipes in C: the art of scientific computing*. Cambridge, UK: Cambridge University Press, pp. 412–20.

Quirk, C., Brockett, C., and Dolan, W. 2004. Monolingual machine translation for paraphrase generation. In *Proceedings of EMNLP*, pp. 142–9, Barcelona, Spain.

Ravichandran, D., and Hovy, E. 2002. Learning surface text patterns for a question answering system. In *Proceedings of ACL*, pp. 41–7, Philadelphia, PA.

Roth, D., and Sammons, M. 2007. Semantic and logical inference model for textual entailment. In *Proceedings of the Workshop on Textual Entailment and Paraphrasing*, pp. 107–12, Prague, Czech Republic.

Shinyama, Y., Sekine, S., and Sudo, K. 2002. Automatic paraphrase acquisition from news articles. In *Proceedings of HLT*, pp. 40–6. San Diego, CA.

Szpektor, I., and Dagan, I. 2007. Learning canonical forms of entailment rules. In *Proceedings of RANLP*, Borovets, Bulgaria.

Szpektor, I., Shnarch, E., and Dagan, I. 2007. Instance-based evaluation of entailment rule acquisition. In *Proceedings of ACL*, pp. 456–63, Prague, Czech Republic.

Szpektor, I., Tanev, H., Dagan, I., and Coppola, B. 2004. Scaling web-based acquisition of entailment relations. In *Proceedings of EMNLP*, pp. 41–8, Barcelona, Spain.

Zhao, S.-Q., Wang, H.-F., Liu, T., and Li, S. 2008. Pivot approach for extracting paraphrase patterns from bilingual corpora. In *Proceedings of ACL-08:HLT*, pp. 780–8, Columbus, OH.

Zhao, S.-Q., Zhou, M., and Liu, T. 2007. Learning question paraphrases for QA from encarta logs. In *Proceedings of IJCAI*, pp. 1795–1800, Hyderabad, India.