# Pivot Approach for Extracting Paraphrase Patterns from Bilingual Corpora

**Shiqi Zhao**[1], **Haifeng Wang**[2], **Ting Liu**[1], **Sheng Li**[1]
[1]Harbin Institute of Technology, Harbin, China
{zhaosq,tliu,lisheng}@ir.hit.edu.cn
[2]Toshiba (China) Research and Development Center, Beijing, China
wanghaifeng@rdc.toshiba.com.cn

## Abstract

Paraphrase patterns are useful in paraphrase recognition and generation. In this paper, we present a pivot approach for extracting paraphrase patterns from bilingual parallel corpora, whereby the English paraphrase patterns are extracted using the sentences in a foreign language as pivots. We propose a log-linear model to compute the paraphrase likelihood of two patterns and exploit feature functions based on maximum likelihood estimation (MLE) and lexical weighting (LW). Using the presented method, we extract over 1,000,000 pairs of paraphrase patterns from 2M bilingual sentence pairs, the precision of which exceeds 67%. The evaluation results show that: (1) The pivot approach is effective in extracting paraphrase patterns, which significantly outperforms the conventional method DIRT. Especially, the log-linear model with the proposed feature functions achieves high performance. (2) The coverage of the extracted paraphrase patterns is high, which is above 84%. (3) The extracted paraphrase patterns can be classified into 5 types, which are useful in various applications.

## 1 Introduction

Paraphrases are different expressions that convey the same meaning. Paraphrases are important in plenty of natural language processing (NLP) applications, such as question answering (QA) (Lin and Pantel, 2001; Ravichandran and Hovy, 2002), machine translation (MT) (Kauchak and Barzilay, 2006; Callison-Burch et al., 2006), multi-document summarization (McKeown et al., 2002), and natural language generation (Iordanskaja et al., 1991).

Paraphrase patterns are sets of semantically equivalent patterns, in which a pattern generally contains two parts, i.e., the pattern words and slots. For example, in the pattern "*X solves Y*", "*solves*" is the pattern word, while "*X*" and "*Y*" are slots. One can generate a text unit (phrase or sentence) by filling the pattern slots with specific words. Paraphrase patterns are useful in both paraphrase recognition and generation. In paraphrase recognition, if two text units 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 the paraphrase patterns of *P*.

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

In this paper, we extract paraphrase patterns from bilingual parallel corpora based on a pivot approach. We assume that if two English patterns are aligned with the same pattern in another language, they are likely to be paraphrase patterns. This assumption is an extension of the one presented in (Bannard and Callison-Burch, 2005), which was used for deriving phrasal paraphrases from bilingual corpora. Our method involves three steps: (1) corpus preprocessing, including English monolingual dependency

*Proceedings of ACL-08: HLT*, pages 780–788,
Columbus, Ohio, USA, June 2008. ©2008 Association for Computational Linguistics

parsing and English-foreign language word alignment, (2) aligned patterns induction, which produces English patterns along with the aligned pivot patterns in the foreign language, (3) paraphrase patterns extraction, in which paraphrase patterns are extracted based on a log-linear model.

Our contributions are as follows. Firstly, we are the first to use a pivot approach to extract paraphrase patterns from bilingual corpora, though similar methods have been used for learning phrasal paraphrases. Our experiments show that the pivot approach significantly outperforms conventional methods. Secondly, we propose a log-linear model for computing the paraphrase likelihood. Besides, we use feature functions based on maximum likelihood estimation (MLE) and lexical weighting (LW), which are effective in extracting paraphrase patterns.

Using the proposed approach, we extract over 1,000,000 pairs of paraphrase patterns from 2M bilingual sentence pairs, the precision of which is above 67%. Experimental results show that the pivot approach evidently outperforms DIRT, a well known method that extracts paraphrase patterns from monolingual corpora (Lin and Pantel, 2001). Besides, the log-linear model is more effective than the conventional model presented in (Bannard and Callison-Burch, 2005). In addition, the coverage of the extracted paraphrase patterns is high, which is above 84%. Further analysis shows that 5 types of paraphrase patterns can be extracted with our method, which can by used in multiple NLP applications.

The rest of this paper is structured as follows. Section 2 reviews related work on paraphrase patterns extraction. Section 3 presents our method in detail. We evaluate the proposed method in Section 4, and finally conclude this paper in Section 5.

## 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 (DIRT), in which they obtained paraphrase patterns from a parsed monolingual corpus based on an extended distributional hypothesis, where if two paths in dependency trees tend to occur in similar contexts it is hypothesized that the meanings of the paths are similar. The examples of obtained paraphrase patterns are shown in Table 1 (1).

| | |
|---|---|
| (1) | *X solves Y*<br>*Y is solved by X*<br>*X finds a solution to Y*<br>...... |
| (2) | *born in <ANSWER> , <NAME>*<br>*<NAME> was born on <ANSWER> ,*<br>*<NAME> ( <ANSWER> -*<br>...... |
| (3) | *ORGANIZATION decides* $\phi$<br>*ORGANIZATION confirms* $\phi$<br>...... |

Table 1: Examples of paraphrase patterns extracted with the methods of Lin and Pantel (2001), Ravichandran and Hovy (2002), and Shinyama et al. (2002).

Based on the same hypothesis as above, some methods extracted paraphrase patterns from the web. For instance, Ravichandran and Hovy (2002) defined a question taxonomy for their QA system. They then used hand-crafted examples of each question type as queries to retrieve paraphrase patterns from the web. For instance, for the question type "*BIRTHDAY*", The paraphrase patterns produced by their method can be seen in Table 1 (2).

Similar methods have also been used by Ibrahim et al. (2003) and Szpektor et al. (2004). The main disadvantage of the above methods is that the precisions of the learned paraphrase patterns are relatively low. For instance, the precisions of the paraphrase patterns 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 paraphrase patterns extracted using their method. However, the performance of their method is dependent on the hand-crafted queries for web mining.

Shinyama et al. (2002) presented a method that extracted paraphrase patterns from multiple news articles about the same event. Their method was based on the assumption that NEs are preserved across paraphrases. Thus the method acquired paraphrase patterns from sentence pairs that share comparable NEs. Some examples can be seen in Table 1 (3).

The disadvantage of this method is that it greatly relies on the number of NEs in sentences. The preci-

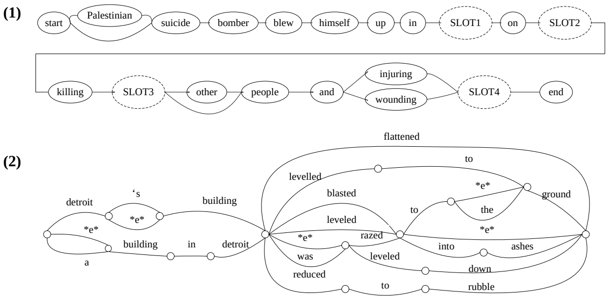

Figure 1: Examples of paraphrase patterns extracted by Barzilay and Lee (2003) and Pang et al. (2003).

sion of the extracted patterns may sharply decrease if the sentences do not contain enough NEs.

Barzilay and Lee (2003) applied multi-sequence alignment (MSA) to parallel news sentences and induced paraphrase patterns for generating new sentences (Figure 1 (1)). 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 1 (2)). Obviously, it is difficult for a sentence to match such complicated patterns, especially if the sentence is not from the same domain in which the patterns are extracted.

Bannard and Callison-Burch (2005) first exploited 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:

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

In Equation (1), $p_{MLE}(c|e_1)$ and $p_{MLE}(e_2|c)$ are the probabilities of translating $e_1$ to $c$ and $c$ to $e_2$, which are computed based on MLE:

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

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.

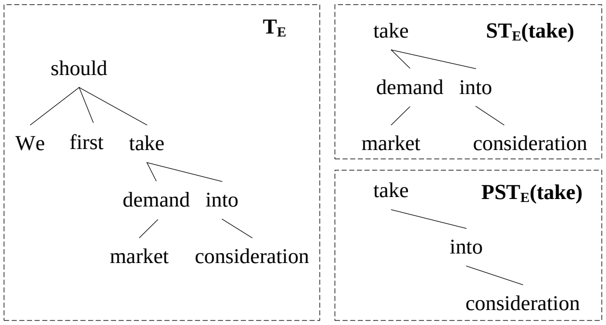

Figure 2: Examples of a subtree and a partial subtree.

# 3 Proposed Method

## 3.1 Corpus Preprocessing

In this paper, we use English paraphrase patterns extraction as a case study. An English-Chinese (E-C) bilingual parallel corpus is employed for training. The Chinese part of the corpus is used as pivots to extract English paraphrase patterns. We conduct word alignment with Giza++ (Och and Ney, 2000) in both directions and then apply the grow-diag heuristic (Koehn et al., 2005) for symmetrization.

Since the paraphrase patterns are extracted from dependency trees, we parse the English sentences in the corpus with 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 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*", $ST_E(take)$ and $PST_E(take)$ are shown in Figure 2[1].

## 3.2 Aligned Patterns Induction

To induce the aligned patterns, we first induce the English patterns using the subtrees and partial subtrees. Then, we extract the pivot Chinese patterns aligning to the English patterns.

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

**Algorithm 1:** Inducing an English pattern

```
Input: words in $ST_E(e)$ : $w_i w_{i+1} ... w_j$
Input: $P_E(e) = \phi$
For each $w_k$ $(i \leq k \leq j)$
    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: $P_C = \phi$
For each $t_l$ $(1 \leq l \leq n)$
    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
```

**Step-1 Inducing English patterns.** In this paper, an English pattern $P_E(e)$ is a string comprising words and part-of-speech (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*". Therefore, we may extract "*take X into consideration*" as a pattern. In addition, the words that are in $ST_E(e)$ but not in $PST_E(e)$ (denoted as $ST_E(e)/PST_E(e)$) are also useful for inducing patterns, since they can constrain the pattern slots. In the example in Figure 2, the word "*demand*" indicates that a noun can be filled in the slot *X* and the pattern may have the form "*take NN into consideration*". 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*". Note that the patterns induced in this way are quite specific, since the 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 of "*demand*", whose POS also forms a slot. The simplified pattern is "*take NN into consideration*".

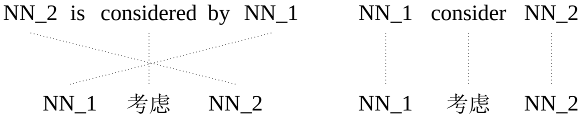

Figure 3: Aligned patterns with numbered slots.

**Step-2 Extracting pivot patterns.** For each English pattern $P_E(e)$, we extract an aligned Chinese pivot pattern $P_C$. Let a Chinese sentence $S_C$ be the translation of the English sentence $S_E$, $P_E(e)$ a pattern induced from $S_E$, we extract the pivot pattern $P_C$ aligning to $P_E(e)$ as in Algorithm 2. Note that the Chinese patterns are not extracted from parse trees. They are only sequences of Chinese words and POSes that are aligned with 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...), while each 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.

### 3.3 Paraphrase Patterns Extraction

As mentioned above, if patterns $e_1$ and $e_2$ are aligned with the same pivot pattern *c*, $e_1$ and $e_2$ may be paraphrase patterns. The paraphrase likelihood can be computed using Equation (1). However, we find that using only the 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:

$$score(e_2|e_1) = \sum_c \exp[\sum_{i=1}^{N} \lambda_i h_i(e_1, e_2, c)] \quad (3)$$

where $h_i(e_1, e_2, c)$ is a feature function and $\lambda_i$ is the

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

weight. In this paper, 4 feature functions are used in our log-linear model, 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. $score_{MLE}(c|e)$ is computed as:

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

$score_{MLE}(e|c)$ is computed in the same way.

$h_3(e_1, e_2, c)$ 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 et al., 2003). It checks how well the words of the phrases translate to each other. This paper uses LW to measure the quality of aligned patterns. We define $score_{LW}(c|e)$ as the logarithm of the lexical weight[3]:

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

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:

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

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

In our experiments, we set a threshold $T$. If the score between $e_1$ and $e_2$ based on Equation (3) exceeds $T$, $e_2$ is extracted as the paraphrase of $e_1$.

### 3.4 Parameter Estimation

Five parameters need to be estimated, i.e., $\lambda_1$, $\lambda_2$, $\lambda_3$, $\lambda_4$ in Equation (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 3.2. 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 3.3. In this process, we set $\lambda_i = 1$ $(i = 1, ..., 4)$ and assign $T$ a minimum value, so as to obtain all possible paraphrase patterns.

A total of 4,162 pairs of paraphrase patterns have been extracted and manually labeled as "1" (correct paraphrase patterns) or "0" (incorrect). Here, two patterns are regarded as paraphrase patterns if they can generate paraphrase fragments by filling the corresponding slots with identical words. 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. We select the parameters that can maximize the F-measure on the development set[4].

## 4 Experiments

The E-C parallel corpus in our experiments was constructed using several LDC bilingual corpora[5]. After filtering sentences that are too long ($>$ 40 words) or too short ($<$ 5 words), 2,048,009 pairs of parallel sentences were retained.

We used two constraints in the experiments to improve the efficiency of computation. First, only subtrees containing no more than 10 words were used to induce English patterns. Second, although any POS tag can form a slot in the induced patterns, we only focused on three kinds of POSes 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

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

[4] The parameters are: $\lambda_1 = 0.0594137$, $\lambda_2 = 0.995936$, $\lambda_3 = -0.0048954$, $\lambda_4 = 1.47816$, $T = -10.002$.

[5] The corpora include LDC2000T46, LDC2000T47, LDC2002E18, LDC2002T01, LDC2003E07, LDC2003E14, LDC2003T17, LDC2004E12, LDC2004T07, LDC2004T08, LDC2005E83, LDC2005T06, LDC2005T10, LDC2006E24, LDC2006E34, LDC2006E85, LDC2006E92, LDC2006T04, LDC2007T02, LDC2007T09.

| Method | #PP (pairs) | Precision |
|---|---|---|
| LL-Model | 1,058,624 | 67.03% |
| MLE-Model | 1,015,533 | 60.60% |
| DIRT top-1 | 1,179 | 19.67% |
| DIRT top-5 | 5,528 | 18.73% |

Table 2: Comparison of paraphrasing methods.

so as to filter patterns like "*the [NN_1]*".

### 4.1 Evaluation of the Log-linear Model

As previously mentioned, in the log-linear model of this paper, we use both MLE based and LW based feature functions. In this section, we evaluate the log-linear model (LL-Model) and compare it with the MLE based model (MLE-Model) presented by Bannard and Callison-Burch (2005)[6].

We extracted paraphrase patterns using two models, respectively. From the results of each model, we randomly picked 3,000 pairs of paraphrase patterns to evaluate the precision. The 6,000 pairs of paraphrase patterns 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 3.4.

The number of the extracted paraphrase patterns (#PP) and the precision are depicted in the first two lines of Table 2. We can see that the numbers of paraphrase patterns extracted using the two models are comparable. However, the precision of LL-Model is significantly higher than MLE-Model.

Actually, MLE-Model is a special case of LL-Model and the enhancement of the precision is mainly due to the use of LW based features. It is not surprising, since Bannard and Callison-Burch (2005) have pointed out that word alignment error is the major factor that influences the performance of the methods learning paraphrases from bilingual corpora. The LW based features validate the quality of word alignment and assign low scores to those aligned E-C pattern pairs with incorrect alignment. Hence the precision can be enhanced.

### 4.2 Comparison with DIRT

It is necessary to compare our method with another paraphrase patterns extraction method. However, it is difficult to find methods that are suitable for comparison. Some methods only extract paraphrase patterns using news articles on certain topics (Shinyama et al., 2002; Barzilay and Lee, 2003), while some others need seeds as initial input (Ravichandran and Hovy, 2002). In this paper, we compare our method with DIRT (Lin and Pantel, 2001), which does not need to specify topics or input seeds.

As mentioned in Section 2, DIRT learns paraphrase patterns from a parsed monolingual corpus based on an extended distributional hypothesis. In our experiment, we implemented DIRT and extracted paraphrase patterns from the English part of our bilingual parallel corpus. Our corpus is smaller than that reported in (Lin and Pantel, 2001). To alleviate the data sparseness problem, we only kept patterns appearing more than 10 times in the corpus for extracting paraphrase patterns. Different from our method, no threshold was set in DIRT. Instead, the extracted paraphrase patterns were ranked according to their scores. In our experiment, we kept top-5 paraphrase patterns for each target pattern.

From the extracted paraphrase patterns, we sampled 600 groups for evaluation. Each group comprises a target pattern and its top-5 paraphrase patterns. The sampled data were manually labeled and the top-n precision was calculated as $\frac{\sum_{i=1}^{N} n_i}{N \times n}$, where $N$ is the number of groups and $n_i$ is the number of correct paraphrase patterns in the top-n paraphrase patterns of the i-th group. The top-1 and top-5 results are shown in the last two lines of Table 2. Although there are more correct patterns in the top-5 results, the precision drops sequentially from top-1 to top-5 since the denominator of top-5 is 4 times larger than that of top-1.

Obviously, the number of the extracted paraphrase patterns is much smaller than that extracted using our method. Besides, the precision is also much lower. We believe that there are two reasons. First, the extended distributional hypothesis is not strict enough. 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 para-

[6] In this experiment, we also estimated a threshold $T'$ for MLE-Model using the development set ($T' = -5.1$). The pattern pairs whose score based on Equation (1) exceed $T'$ were extracted as paraphrase patterns.

| Type | Count | Example |
|---|---|---|
| trivial change | 79 | $(\mathbf{e_1})$ *all the members of [NNPS_1]* $(\mathbf{e_2})$ *all members of [NNPS_1]* |
| phrase replacement | 267 | $(\mathbf{e_1})$ *[JJ_1] economic losses* $(\mathbf{e_2})$ *[JJ_1] financial losses* |
| phrase reordering | 56 | $(\mathbf{e_1})$ *[NN_1] definition* $(\mathbf{e_2})$ *the definition of [NN_1]* |
| structural paraphrase | 71 | $(\mathbf{e_1})$ *the admission of [NNP_1] to the wto* $(\mathbf{e_2})$ *the [NNP_1] 's wto accession* |
| information + or - | 27 | $(\mathbf{e_1})$ *[NNS_1] are in fact women* $(\mathbf{e_2})$ *[NNS_1] are women* |

Table 3: The statistics and examples of each type of paraphrase patterns.

phrase patterns by DIRT. The other reason is that DIRT can only be effective for patterns appearing plenty of times in the corpus. In other words, it seriously suffers from data sparseness. We believe that DIRT can perform better on a larger corpus.

### 4.3 Pivot Pattern Constraints

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. Hence, it is interesting to 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 the aligned patterns $(e, c)$, if $c$'s pattern words were not extracted from a partial subtree, the pair was filtered. After that, we extracted paraphrase patterns, from which we sampled 3,000 pairs for evaluation.

The results show that 736,161 pairs of paraphrase patterns were extracted and the precision is 65.77%. Compared with Table 2, the number of the extracted paraphrase patterns gets smaller and the precision also gets lower. The results suggest that the performance of the method cannot be improved by constraining the extraction of pivot patterns.

### 4.4 Analysis of the Paraphrase Patterns

We sampled 500 pairs of correct paraphrase patterns extracted using our method and analyzed the types. We found that there are 5 types of paraphrase patterns, which include: (1) trivial change, such as changes of prepositions and articles, etc; (2) phrase replacement; (3) phrase reordering; (4) structural paraphrase, which contain both phrase replacements and phrase reordering; (5) adding or reducing information that does not change the meaning. Some statistics and examples are shown in Table 3.

The paraphrase patterns are useful in NLP applications. Firstly, 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.

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 et al., 2004).

The structural paraphrase patterns have the advantages of both phrase replacement and phrase reordering. More paraphrase sentences can be generated using these patterns.

The paraphrase patterns in the type of "information + and -" 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.

For the 3,000 pairs of test paraphrase patterns, we also investigate the number and type of the pattern slots. The results are summarized in Table 4 and 5.

From Table 4, we can see that more than 92% of the paraphrase patterns contain only one slot, just like the examples shown in Table 3. In addition, about 7% of the paraphrase patterns contain two slots, such as "*give [NN_1] [NN_2]*" vs. "*give [NN_2] to [NN_1]*". This result suggests that our method tends to extract short paraphrase patterns,

| Slot No. | #PP | Percentage | Precision |
|---|---|---|---|
| 1-slot | 2,780 | 92.67% | 66.51% |
| 2-slots | 218 | 7.27% | 73.85% |
| ≥3-slots | 2 | <1% | 50.00% |

Table 4: The statistics of the numbers of pattern slots.

| Slot Type | #PP | Percentage | Precision |
|---|---|---|---|
| N-slots | 2,376 | 79.20% | 66.71% |
| V-slots | 273 | 9.10% | 70.33% |
| J-slots | 438 | 14.60% | 70.32% |

Table 5: The statistics of the type of pattern slots.

which is mainly because the data sparseness problem is more serious when extracting long patterns.

From Table 5, we can find that near 80% of the paraphrase patterns contain noun slots, while about 9% and 15% contain verb slots and adjective slots[7]. This result implies that nouns are the most typical variables in paraphrase patterns.

### 4.5 Evaluation within Context Sentences

In Section 4.1, 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 open test set includes 119 English sentences. We parsed the sentences with MaltParser and induced patterns as described in Section 3.2. For each pattern $e$ in sentence $S_E$, we searched $e$'s paraphrase patterns from the database of the extracted paraphrase patterns. The result shows that 101 of the 119 sentences contain at least one pattern that can be paraphrased using the extracted paraphrase patterns, the coverage of which is 84.87%.

Furthermore, since a pattern may have several paraphrase patterns, we exploited a method to automatically select the best one in the given context sentence. In detail, a paraphrase pattern $e'$ of $e$ was reranked based on a language model (LM):

$$\begin{aligned} &score(e'|e, S_E) = \\ &\lambda score_{LL}(e'|e) + (1-\lambda) score_{LM}(e'|S_E) \end{aligned} \quad (7)$$

Here, $score_{LL}(e'|e)$ denotes the score based on Equation (3). $score_{LM}(e'|S_E)$ is the LM based score: $score_{LM}(e'|S_E) = \frac{1}{n} logP_{LM}(S'_E)$, 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 $\lambda = 0.7$.

The selected best paraphrase patterns in context sentences were manually labeled. The context information was also considered by our judges. The result shows that the precision of the best paraphrase patterns is 59.39%. To investigate the contribution of the LM based score, we ran the experiment again with $\lambda = 1$ (ignoring the LM based score) and found that the precision is 57.09%. It indicates that the LM based reranking can improve the precision. However, the improvement is small. Further analysis shows that about 70% of the correct paraphrase substitutes are in the type of phrase replacement.

## 5 Conclusion

This paper proposes a pivot approach for extracting paraphrase patterns from bilingual corpora. We use a log-linear model to compute the paraphrase likelihood and exploit feature functions based on MLE and LW. Experimental results show that the pivot approach is effective, which extracts over 1,000,000 pairs of paraphrase patterns from 2M bilingual sentence pairs. The precision and coverage of the extracted paraphrase patterns exceed 67% and 84%, respectively. In addition, the log-linear model with the proposed feature functions significantly outperforms the conventional models. Analysis shows that 5 types of paraphrase patterns are extracted with our method, which are useful in various applications.

In the future we wish to exploit more feature functions in the log-linear model. In addition, we will try to make better use of the context information when replacing paraphrase patterns in context sentences.

## Acknowledgments

This research was supported by National Natural Science Foundation of China (60503072, 60575042). We thank Lin Zhao, Xiaohang Qu, and Zhenghua Li for their help in the experiments.

[7] Notice that, a pattern may contain more than one type of slots, thus the sum of the percentages is larger than 1.

## References

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with Bilingual Parallel Corpora. In *Proceedings of ACL*, pages 597-604.

Regina Barzilay and Lillian Lee. 2003. Learning to Paraphrase: An Unsupervised Approach Using Multiple-Sequence Alignment. In *Proceedings of HLT-NAACL*, pages 16-23.

Chris Callison-Burch, Philipp Koehn, and Miles Osborne. 2006. Improved Statistical Machine Translation Using Paraphrases. In *Proceedings of HLT-NAACL*, pages 17-24.

Ali Ibrahim, Boris Katz, and Jimmy Lin. 2003. Extracting Structural Paraphrases from Aligned Monolingual Corpora. In *Proceedings of IWP*, pages 57-64.

Lidija Iordanskaja, Richard Kittredge, and Alain Polguère. 1991. Lexical Selection and Paraphrase in a Meaning-Text Generation Model. In Cécile L. Paris, William R. Swartout, and William C. Mann (Eds.): Natural Language Generation in Artificial Intelligence and Computational Linguistics, pages 293-312.

David Kauchak and Regina Barzilay. 2006. Paraphrasing for Automatic Evaluation. In *Proceedings of HLT-NAACL*, pages 455-462.

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation. In *Proceedings of IWSLT.*

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of HLT-NAACL*, pages 127-133.

De-Kang Lin and Patrick Pantel. 2001. Discovery of Inference Rules for Question Answering. In *Natural Language Engineering* 7(4): 343-360.

Ting Liu, Jin-Shan Ma, Hui-Jia Zhu, and Sheng Li. 2006. Dependency Parsing Based on Dynamic Local Optimization. In *Proceedings of CoNLL-X*, pages 211-215.

Kathleen R. Mckeown, Regina Barzilay, David Evans, Vasileios Hatzivassiloglou, Judith L. Klavans, Ani Nenkova, Carl Sable, Barry Schiffman, and Sergey Sigelman. 2002. Tracking and Summarizing News on a Daily Basis with Columbia's Newsblaster. In *Proceedings of HLT*, pages 280-285.

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülsen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. 2007. MaltParser: A Language-Independent System for Data-Driven Dependency Parsing. In *Natural Language Engineering* 13(2): 95-135.

Franz Josef Och and Hermann Ney. 2000. Improved Statistical Alignment Models. In *Proceedings of ACL*, pages 440-447.

Aïda Ouangraoua, Pascal Ferraro, Laurent Tichit, and Serge Dulucq. 2007. Local Similarity between Quotiented Ordered Trees. In *Journal of Discrete Algorithms* 5(1): 23-35.

Bo Pang, Kevin Knight, and Daniel Marcu. 2003. Syntax-based Alignment of Multiple Translations: Extracting Paraphrases and Generating New Sentences. In *Proceedings of HLT-NAACL*, pages 102-109.

William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. 1992. Numerical Recipes in C: The Art of Scientific Computing. Cambridge University Press, Cambridge, U.K., 1992, 412-420.

Chris Quirk, Chris Brockett, and William Dolan. 2004. Monolingual Machine Translation for Paraphrase Generation. In *Proceedings of EMNLP*, pages 142-149.

Deepak Ravichandran and Eduard Hovy. 2002. Learning Surface Text Patterns for a Question Answering System. In *Proceedings of ACL*, pages 41-47.

Yusuke Shinyama, Satoshi Sekine, and Kiyoshi Sudo. 2002. Automatic Paraphrase Acquisition from News Articles. In *Proceedings of HLT*, pages 40-46.

Idan Szpektor, Hristo Tanev, Ido Dagan and Bonaventura Coppola. 2004. Scaling Web-based Acquisition of Entailment Relations. In *Proceedings of EMNLP*, pages 41-48.