Combining Multiple Resources to Improve SMT-based Paraphrasing Model*

Shiqi Zhao[1], **Cheng Niu**[2], **Ming Zhou**[2], **Ting Liu**[1], **Sheng Li**[1]
[1]Harbin Institute of Technology, Harbin, China
{zhaosq,tliu,lisheng}@ir.hit.edu.cn
[2]Microsoft Research Asia, Beijing, China
{chengniu,mingzhou}@microsoft.com

Abstract

This paper proposes a novel method that exploits multiple resources to improve statistical machine translation (SMT) based paraphrasing. In detail, a phrasal paraphrase table and a feature function are derived from each resource, which are then combined in a log-linear SMT model for sentence-level paraphrase generation. Experimental results show that the SMT-based paraphrasing model can be enhanced using multiple resources. The phrase-level and sentence-level precision of the generated paraphrases are above 60% and 55%, respectively. In addition, the contribution of each resource is evaluated, which indicates that all the exploited resources are useful for generating paraphrases of high quality.

1 Introduction

Paraphrases are alternative ways of conveying the same meaning. Paraphrases are important in many natural language processing (NLP) applications, such as machine translation (MT), question answering (QA), information extraction (IE), multi-document summarization (MDS), and natural language generation (NLG).

This paper addresses the problem of sentence-level paraphrase generation, which aims at generating paraphrases for input sentences. An example of sentence-level paraphrases can be seen below:

S1: The table was ***set up in the carriage shed****.*
S2: The table was ***laid under the cart-shed****.*

Paraphrase generation can be viewed as monolingual machine translation (Quirk et al., 2004), which typically includes a translation model and a language model. The translation model can be trained using monolingual parallel corpora. However, acquiring such corpora is not easy. Hence, data sparseness is a key problem for the SMT-based paraphrasing. On the other hand, various methods have been presented to extract phrasal paraphrases from different resources, which include thesauri, monolingual corpora, bilingual corpora, and the web. However, little work has been focused on using the extracted phrasal paraphrases in sentence-level paraphrase generation.

In this paper, we exploit multiple resources to improve the SMT-based paraphrase generation. In detail, six kinds of resources are utilized, including: (1) an automatically constructed thesaurus, (2) a monolingual parallel corpus from novels, (3) a monolingual comparable corpus from news articles, (4) a bilingual phrase table, (5) word definitions from Encarta dictionary, and (6) a corpus of similar user queries. Among the resources, (1), (2), (3), and (4) have been investigated by other researchers, while (5) and (6) are first used in this paper. From those resources, six phrasal paraphrase tables are extracted, which are then used in a log-linear SMT-based paraphrasing model.

Both phrase-level and sentence-level evaluations were carried out in the experiments. In the former one, phrase substitutes occurring in the paraphrase sentences were evaluated. While in the latter one, the acceptability of the paraphrase sentences was evaluated. Experimental results show that: (1) The

[*] This research was finished while the first author worked as an intern in Microsoft Research Asia.

Proceedings of ACL-08: HLT, pages 1021–1029,
Columbus, Ohio, USA, June 2008.

SMT-based paraphrasing is enhanced using multiple resources. The phrase-level and sentence-level precision of the generated paraphrases exceed 60% and 55%, respectively. (2) Although the contributions of the resources differ a lot, all the resources are useful. (3) The performance of the method varies greatly on different test sets and it performs best on the test set of news sentences, which are from the same source as most of the training data.

The rest of the paper is organized as follows: Section 2 reviews related work. Section 3 introduces the log-linear model for paraphrase generation. Section 4 describes the phrasal paraphrase extraction from different resources. Section 5 presents the parameter estimation method. Section 6 shows the experiments and results. Section 7 draws the conclusion.

2 Related Work

Paraphrases have been used in many NLP applications. In MT, Callison-Burch et al. (2006) utilized paraphrases of unseen source phrases to alleviate data sparseness. Kauchak and Barzilay (2006) used paraphrases of the reference translations to improve automatic MT evaluation. In QA, Lin and Pantel (2001) and Ravichandran and Hovy (2002) paraphrased the answer patterns to enhance the recall of answer extraction. In IE, Shinyama et al. (2002) automatically learned paraphrases of IE patterns to reduce the cost of creating IE patterns by hand. In MDS, McKeown et al. (2002) identified paraphrase sentences across documents before generating summarizations. In NLG, Iordanskaja et al. (1991) used paraphrases to generate more varied and fluent texts.

Previous work has examined various resources for acquiring paraphrases, including thesauri, monolingual corpora, bilingual corpora, and the web. Thesauri, such as WordNet, have been widely used for extracting paraphrases. Some researchers extract synonyms as paraphrases (Kauchak and Barzilay, 2006), while some others use looser definitions, such as hypernyms and holonyms (Barzilay and Elhadad, 1997). Besides, the automatically constructed thesauri can also be used. Lin (1998) constructed a thesaurus by automatically clustering words based on context similarity.

Barzilay and McKeown (2001) used monolingual parallel corpora for identifying paraphrases. They exploited a corpus of multiple English translations of the same source text written in a foreign language, from which phrases in aligned sentences that appear in similar contexts were extracted as paraphrases. In addition, Finch et al. (2005) applied MT evaluation methods (BLEU, NIST, WER and PER) to build classifiers for paraphrase identification.

Monolingual parallel corpora are difficult to find, especially in non-literature domains. Alternatively, some researchers utilized monolingual comparable corpora for paraphrase extraction. Different news articles reporting on the same event are commonly used as monolingual comparable corpora, from which both paraphrase patterns and phrasal paraphrases can be derived (Shinyama et al., 2002; Barzilay and Lee, 2003; Quirk et al., 2004).

Lin and Pantel (2001) learned paraphrases 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 monolingual corpus used in their work is not necessarily parallel or comparable. Thus it is easy to obtain. However, since this resource is used to extract paraphrase patterns other than phrasal paraphrases, we do not use it in this paper.

Bannard and Callison-Burch (2005) learned phrasal paraphrases using bilingual parallel corpora. The basic idea is that if two phrases are aligned to the same translation in a foreign language, they may be paraphrases. This method has been demonstrated effective in extracting large volume of phrasal paraphrases. Besides, Wu and Zhou (2003) exploited bilingual corpora and translation information in learning synonymous collocations.

In addition, some researchers extracted paraphrases from the web. For example, Ravichandran and Hovy (2002) retrieved paraphrase patterns from the web using hand-crafted queries. Pasca and Dienes (2005) extracted sentence fragments occurring in identical contexts as paraphrases from one billion web documents. Since web mining is rather time consuming, we do not exploit the web to extract paraphrases in this paper.

So far, two kinds of methods have been proposed for sentence-level paraphrase generation, i.e., the pattern-based and SMT-based methods. Automatically learned patterns have been used in paraphrase generation. For example, Barzilay and Lee (2003) applied multiple-sequence alignment (MSA) to parallel news sentences and induced paraphrasing patterns for generating new sentences. Pang et al. (2003) built finite state automata (FSA) from semantically equivalent translation sets based on syntactic alignment and used the FSAs in paraphrase generation. The pattern-based methods can generate complex paraphrases that usually involve syntactic variation. However, the methods were demonstrated to be of limited generality (Quirk et al., 2004).

Quirk et al. (2004) first recast paraphrase generation as monolingual SMT. They generated paraphrases using a SMT system trained on parallel sentences extracted from clustered news articles. In addition, Madnani et al. (2007) also generated sentence-level paraphrases based on a SMT model. The advantage of the SMT-based method is that it achieves better coverage than the pattern-based method. The main difference between their methods and ours is that they only used bilingual parallel corpora as paraphrase resource, while we exploit and combine multiple resources.

3 SMT-based Paraphrasing Model

The SMT-based paraphrasing model used by Quirk et al. (2004) was the noisy channel model of Brown et al. (1993), which identified the optimal paraphrase T^* of a sentence S by finding:

$$\begin{aligned} T^* &= \arg\max_T \{P(T|S)\} \\ &= \arg\max_T \{P(S|T)P(T)\} \end{aligned} \tag{1}$$

In contrast, we adopt a log-linear model (Och and Ney, 2002) in this work, since multiple paraphrase tables can be easily combined in the log-linear model. Specifically, feature functions are derived from each paraphrase resource and then combined with the language model feature[1]:

$$\begin{aligned} T^* = \arg\max_T \{ & \sum_{i=1}^{N} \lambda_{TM\_i} h_{TM\_i}(T,S) + \\ & \lambda_{LM} h_{LM}(T,S)\} \end{aligned} \tag{2}$$

where N is the number of paraphrase tables. $h_{TM\_i}(T,S)$ is the feature function based on the i-th paraphrase table PT_i. $h_{LM}(T,S)$ is the language model feature. λ_{TM_i} and λ_{LM} are the weights of the feature functions. $h_{TM\_i}(T,S)$ is defined as:

$$h_{TM\_i}(T,S) = \log \prod_{k=1}^{K_i} Score_i(T_k, S_k) \tag{3}$$

where K_i is the number of phrase substitutes from S to T based on PT_i. T_k in T and S_k in S are phrasal paraphrases in PT_i. $Score_i(T_k, S_k)$ is the paraphrase likelihood according to PT_i[2]. A 5-gram language model is used, therefore:

$$h_{LM}(T,S) = \log \prod_{j=1}^{J} p(t_j | t_{j-4}, ..., t_{j-1}) \tag{4}$$

where J is the length of T, t_j is the j-th word of T.

4 Exploiting Multiple Resources

This section describes the extraction of phrasal paraphrases using various resources. Similar to Pharaoh (Koehn, 2004), our decoder[3] uses top 20 paraphrase options for each input phrase in the default setting. Therefore, we keep at most 20 paraphrases for a phrase when extracting phrasal paraphrases using each resource.

1 - Thesaurus: The thesaurus[4] used in this work was automatically constructed by Lin (1998). The similarity of two words e_1 and e_2 was calculated through the surrounding context words that have dependency relations with the investigated words:

$$\begin{aligned} & Sim(e_1, e_2) \\ & = \frac{\sum_{(r,e)\in T_r(e_1)\cap T_r(e_2)} (I(e_1, r, e) + I(e_2, r, e))}{\sum_{(r,e)\in T_r(e_1)} I(e_1, r, e) + \sum_{(r,e)\in T_r(e_2)} I(e_2, r, e)} \end{aligned} \tag{5}$$

where $T_r(e_i)$ denotes the set of words that have dependency relation r with word e_i. $I(e_i, r, e)$ is the mutual information between e_i, r and e.

For each word, we keep 20 most similar words as paraphrases. In this way, we extract 502,305 pairs of paraphrases. The paraphrasing score $Score_1(p_1, p_2)$ used in Equation (3) is defined as the similarity based on Equation (5).

[1] The reordering model is not considered in our model.

[2] If none of the phrase substitutes from S to T is from PT_i (i.e., $K_i = 0$), we cannot compute $h_{TM\_i}(T,S)$ as in Equation (3). In this case, we assign $h_{TM\_i}(T,S)$ a minimum value.

[3] The decoder used here is a re-implementation of Pharaoh.

[4] http://www.cs.ualberta.ca/ lindek/downloads.htm.

2 - Monolingual parallel corpus: Following Barzilay and McKeown (2001), we exploit a corpus of multiple English translations of foreign novels, which contains 25,804 parallel sentence pairs. We find that most paraphrases extracted using the method of Barzilay and McKeown (2001) are quite short. Thus we employ a new approach for paraphrase extraction. Specifically, we parse the sentences with CollinsParser[5] and extract the chunks from the parsing results. Let S_1 and S_2 be a pair of parallel sentences, p_1 and p_2 two chunks from S_1 and S_2, we compute the similarity of p_1 and p_2 as:

$$Sim(p_1, p_2) = \alpha Sim_{content}(p_1, p_2) + (1 - \alpha) Sim_{context}(p_1, p_2) \quad (6)$$

where, $Sim_{content}(p_1, p_2)$ is the content similarity, which is the word overlapping rate of p_1 and p_2. $Sim_{context}(p_1, p_2)$ is the context similarity, which is the word overlapping rate of the contexts of p_1 and p_2[6]. If the similarity of p_1 and p_2 exceeds a threshold Th_1, they are identified as paraphrases. We extract 18,698 pairs of phrasal paraphrases from this resource. The paraphrasing score $Score_2(p_1, p_2)$ is defined as the similarity in Equation (6). For the paraphrases occurring more than once, we use their maximum similarity as the paraphrasing score.

3 - Monolingual comparable corpus: Similar to the methods in (Shinyama et al., 2002; Barzilay and Lee, 2003), we construct a corpus of comparable documents from a large corpus D of news articles. The corpus D contains 612,549 news articles. Given articles d_1 and d_2 from D, if their publication date interval is less than 2 days and their similarity[7] exceeds a threshold Th_2, they are recognized as comparable documents. In this way, a corpus containing 5,672,864 pairs of comparable documents is constructed. From the comparable corpus, parallel sentences are extracted. Let s_1 and s_2 be two sentences from comparable documents d_1 and d_2, if their similarity based on word overlapping rate is above a threshold Th_3, s_1 and s_2 are identified as parallel sentences. In this way, 872,330 parallel sentence pairs are extracted.

We run Giza++ (Och and Ney, 2000) on the parallel sentences and then extract aligned phrases as described in (Koehn, 2004). The generated paraphrase table is pruned by keeping the top 20 paraphrases for each phrase. After pruning, 100,621 pairs of paraphrases are extracted. Given phrase p_1 and its paraphrase p_2, we compute $Score_3(p_1, p_2)$ by relative frequency (Koehn et al., 2003):

$$Score_3(p_1, p_2) = p(p_2|p_1) = \frac{count(p_2, p_1)}{\sum_{p'} count(p', p_1)} \quad (7)$$

People may wonder why we do not use the same method on the monolingual parallel and comparable corpora. This is mainly because the volumes of the two corpora differ a lot. In detail, the monolingual parallel corpus is fairly small, thus automatical word alignment tool like Giza++ may not work well on it. In contrast, the monolingual comparable corpus is quite large, hence we cannot conduct the time-consuming syntactic parsing on it as we do on the monolingual parallel corpus.

4 - Bilingual phrase table: We first construct a bilingual phrase table that contains 15,352,469 phrase pairs from an English-Chinese parallel corpus. We extract paraphrases from the bilingual phrase table and compute the paraphrasing score of phrases p_1 and p_2 as in (Bannard and Callison-Burch, 2005):

$$Score_4(p_1, p_2) = \sum_f p(f|p_1) p(p_2|f) \quad (8)$$

where f denotes a Chinese translation of both p_1 and p_2. $p(f|p_1)$ and $p(p_2|f)$ are the translation probabilities provided by the bilingual phrase table. For each phrase, the top 20 paraphrases are kept according to the score in Equation (8). As a result, 3,177,600 pairs of phrasal paraphrases are extracted.

5 - Encarta dictionary definitions: Words and their definitions can be regarded as paraphrases. Here are some examples from Encarta dictionary: "*hurricane: severe storm*", "*clever: intelligent*", "*travel: go on journey*". In this work, we extract words' definitions from Encarta dictionary web pages[8]. If a word has more than one definition, all of them are extracted. Note that the words and definitions in the

[5] http://people.csail.mit.edu/mcollins/code.html

[6] The context of a chunk is made up of 6 words around the chunk, 3 to the left and 3 to the right.

[7] The similarity of two documents is computed using the vector space model and the word weights are based on tf·idf.

[8] http://encarta.msn.com/encnet/features/dictionary/dictionaryhome.aspx

dictionary are lemmatized, but words in sentences are usually inflected. Hence, we expand the word - definition pairs by providing the inflected forms. Here we use an inflection list and some rules for inflection. After expanding, 159,456 pairs of phrasal paraphrases are extracted. Let $< p_1, p_2 >$ be a word - definition pair, the paraphrasing score is defined according to the rank of p_2 in all of p_1's definitions:

$$Score_5(p_1, p_2) = \gamma^{i-1} \tag{9}$$

where γ is a constant (we empirically set $\gamma = 0.9$) and i is the rank of p_2 in p_1's definitions.

6 - Similar user queries: Clusters of similar user queries have been used for query expansion and suggestion (Gao et al., 2007). Since most queries are at the phrase level, we exploit similar user queries as phrasal paraphrases. In our experiment, we use the corpus of clustered similar MSN queries constructed by Gao et al. (2007). The similarity of two queries p_1 and p_2 is computed as:

$$\begin{aligned} Sim(p_1, p_2) = \beta Sim_{content}(p_1, p_2) + \\ (1 - \beta) Sim_{click-through}(p_1, p_2) \end{aligned} \tag{10}$$

where $Sim_{content}(p_1, p_2)$ is the content similarity, which is computed as the word overlapping rate of p_1 and p_2. $Sim_{click-through}(p_1, p_2)$ is the click through similarity, which is the overlapping rate of the user clicked documents for p_1 and p_2. For each query q, we keep the top 20 similar queries, whose similarity with q exceeds a threshold Th_4. As a result, 395,284 pairs of paraphrases are extracted. The score $Score_6(p_1, p_2)$ is defined as the similarity in Equation (10).

7 - Self-paraphrase: In addition to the six resources introduced above, a special paraphrase table is used, which is made up of pairs of identical words. The reason why this paraphrase table is necessary is that a word should be allowed to keep unchanged in paraphrasing. This is a difference between paraphrasing and MT, since all words should be translated in MT. In our experiments, all the words that occur in the six paraphrase table extracted above are gathered to form the self-paraphrase table, which contains 110,403 word pairs. The score $Score_7(p_1, p_2)$ is set 1 for each identical word pair.

5 Parameter Estimation

The weights of the feature functions, namely λ_{TM_i} $(i = 1, 2, ..., 7)$ and λ_{LM}, need estimation[9]. In MT, the max-BLEU algorithm is widely used to estimate parameters. However, it may not work in our case, since it is more difficult to create a reference set of paraphrases.

We propose a new technique to estimate parameters in paraphrasing. The assumption is that, since a SMT-based paraphrase is generated through phrase substitution, we can measure the quality of a generated paraphrase by measuring its phrase substitutes. Generally, the paraphrases containing more correct phrase substitutes are judged as better paraphrases[10]. We therefore present the phrase substitution error rate (PSER) to score a generated paraphrase T:

$$PSER(T) = \|PS_0(T)\| / \|PS(T)\| \tag{11}$$

where $PS(T)$ is the set of phrase substitutes in T and $PS_0(T)$ is the set of incorrect substitutes.

In practice, we keep top n paraphrases for each sentence S. Thus we calculate the PSER for each source sentence S as:

$$PSER(S) = \| \bigcup_{i=1}^{n} PS_0(T_i) \| / \| \bigcup_{i=1}^{n} PS(T_i) \| \tag{12}$$

where T_i is the i-th generated paraphrase of S.

Suppose there are N sentences in the development set, the overall PSER is computed as:

$$PSER = \sum_{j=1}^{N} PSER(S_j) \tag{13}$$

where S_j is the j-th sentence in the development set.

Our development set contains 75 sentences (described in detail in Section 6). For each sentence, all possible phrase substitutes are extracted from the six paraphrase tables above. The extracted phrase substitutes are then manually labeled as "correct" or "incorrect". A phrase substitute is considered as correct only if the two phrases have the same meaning in the given sentence and the sentence generated by

[9] Note that, we also use some other parameters when extracting phrasal paraphrases from different resources, such as the thresholds Th_1, Th_2, Th_3, Th_4, as well as α and β in Equation (6) and (10). These parameters are estimated using different development sets from the investigated resources. We do not describe the estimation of them due to space limitation.

[10] Paraphrasing a word to itself (based on the 7-th paraphrase table above) is not regarded as a substitute.

substituting the source phrase with the target phrase remains grammatical. In decoding, the phrase substitutes are printed out and then the PSER is computed based on the labeled data.

Using each set of parameters, we generate paraphrases for the sentences in the development set based on Equation (2). PSER is then computed as in Equation (13). We use the gradient descent algorithm (Press et al., 1992) to minimize PSER on the development set and get the optimal parameters.

6 Experiments

To evaluate the performance of the method on different types of test data, we used three kinds of sentences for testing, which were randomly extracted from Google news, free online novels, and forums, respectively. For each type, 50 sentences were extracted as test data and another 25 were extracted as development data. For each test sentence, top 10 of the generated paraphrases were kept for evaluation.

6.1 Phrase-level Evaluation

The phrase-level evaluation was carried out to investigate the contributions of the paraphrase tables. For each test sentence, all possible phrase substitutes were first extracted from the paraphrase tables and manually labeled as "correct" or "incorrect". Here, the criterion for identifying paraphrases is the same as that described in Section 5. Then, in the stage of decoding, the phrase substitutes were printed out and evaluated using the labeled data.

Two metrics were used here. The first is the number of distinct correct substitutes (#DCS). Obviously, the more distinct correct phrase substitutes a paraphrase table can provide, the more valuable it is. The second is the accuracy of the phrase substitutes, which is computed as:

$$Accuracy = \frac{\#correct\ phrase\ substitutes}{\#all\ phrase\ substitutes} \quad (14)$$

To evaluate the PTs learned from different resources, we first used each PT (from 1 to 6) along with PT-7 in decoding. The results are shown in Table 1. It can be seen that PT-4 is the most useful, as it provides the most correct substitutes and the accuracy is the highest. We believe that it is because PT-4 is much larger than the other PTs. Compared with PT-4, the accuracies of the other PTs are fairly low. This is because those PTs are smaller, thus they can provide fewer correct phrase substitutes. As a result, plenty of incorrect substitutes were included in the top 10 generated paraphrases.

PT combination	#DCS	Accuracy
1+7	178	14.61%
2+7	94	25.06%
3+7	202	18.35%
4+7	553	56.93%
5+7	231	20.48%
6+7	21	14.42%

Table 1: Contributions of the paraphrase tables. PT-1: from the thesaurus; PT-2: from the monolingual parallel corpus; PT-3: from the monolingual comparable corpus; PT-4: from the bilingual parallel corpus; PT-5: from the Encarta dictionary definitions; PT-6: from the similar MSN user queries; PT-7: self-paraphrases.

PT-6 provides the least correct phrase substitutes and the accuracy is the lowest. There are several reasons. First, many phrases in PT-6 are not real phrases but only sets of keywords (e.g., "*lottery results ny*"), which may not appear in sentences. Second, many words in this table have spelling mistakes (e.g., "*widows vista*"). Third, some phrase pairs in PT-6 are not paraphrases but only "related queries" (e.g., "*back tattoo*" vs. "*butterfly tattoo*"). Fourth, many phrases of PT-6 contain proper names or out-of-vocabulary words, which are difficult to be matched. The accuracy based on PT-1 is also quite low. We found that it is mainly because the phrase pairs in PT-1 are automatically clustered, many of which are merely "similar" words rather than synonyms (e.g., "*borrow*" vs. "*buy*").

Next, we try to find out whether it is necessary to combine all PTs. Thus we conducted several runs, each of which added the most useful PT from the left ones. The results are shown in Table 2. We can see that all the PTs are useful, as each PT provides some new correct phrase substitutes and the accuracy increases when adding each PT except PT-1.

Since the PTs are extracted from different resources, they have different contributions. Here we only discuss the contributions of PT-5 and PT-6, which are first used in paraphrasing in this paper. PT-5 is useful for paraphrasing uncommon concepts since it can "explain" concepts with their definitions.

PT combination	#DCS	Accuracy
4+7	553	56.93%
4+5+7	581	58.97%
4+5+3+7	638	59.42%
4+5+3+2+7	649	60.15%
4+5+3+2+1+7	699	60.14%
4+5+3+2+1+6+7	711	60.16%

Table 2: Performances of different combinations of paraphrase tables.

For instance, in the following test sentence S_1, the word "*amnesia*" is a relatively uncommon word, especially for the people using English as the second language. Based on PT-5, S_1 can be paraphrased into T_1, which is much easier to understand.

> S_1: *I was suffering from* ***amnesia****.*
> T_1: *I was suffering from* ***memory loss****.*

The disadvantage of PT-5 is that substituting words with the definitions sometimes leads to grammatical errors. For instance, substituting "*heat shield*" in the sentence S_2 with "*protective barrier against heat*" keeps the meaning unchanged. However, the paraphrased sentence T_2 is ungrammatical.

> S_2: *The U.S. space agency has been cautious about* ***heat shield*** *damage.*
> T_2: *The U.S. space administration has been cautious about* ***protective barrier against heat*** *damage.*

As previously mentioned, PT-6 is less effective compared with the other PTs. However, it is useful for paraphrasing some special phrases, such as digital products, computer software, etc, since these phrases often appear in user queries. For example, S_3 below can be paraphrased into T_3 using PT-6.

> S_3: *I have a* ***canon powershot*** *S230 that uses CF memory cards.*
> T_3: *I have a* ***canon digital camera*** *S230 that uses CF memory cards.*

The phrase "*canon powershot*" can hardly be paraphrased using the other PTs. It suggests that PT-6 is useful for paraphrasing new emerging concepts and expressions.

Test sentences	Top-1	Top-5	Top-10
All 150	55.33%	45.20%	39.28%
50 from news	70.00%	62.00%	57.03%
50 from novel	56.00%	46.00%	37.42%
50 from forum	40.00%	27.60%	23.34%

Table 3: Top-n accuracy on different test sentences.

6.2 Sentence-level Evaluation

In this section, we evaluated the sentence-level quality of the generated paraphrases[11]. In detail, each generated paraphrase was manually labeled as "acceptable" or "unacceptable". Here, the criterion for counting a sentence *T* as an acceptable paraphrase of sentence *S* is that *T* is understandable and its meaning is not evidently changed compared with *S*. For example, for the sentence S_4, T_4 is an acceptable paraphrase generated using our method.

> S_4: *The* ***strain on*** *US* ***forces*** *of fighting in Iraq and Afghanistan* ***was exposed*** *yesterday when the Pentagon* ***published a report*** *showing that the* ***number*** *of suicides among US* ***troops*** *is at its* ***highest level*** *since the 1991 Gulf* ***war****.*
> T_4: *The* ***pressure on*** *US* ***troops*** *of fighting in Iraq and Afghanistan* ***was revealed*** *yesterday when the Pentagon* ***released a report*** *showing that the* ***amount*** *of suicides among US* ***forces*** *is at its* ***top*** *since the 1991 Gulf* ***conflict****.*

We carried out sentence-level evaluation using the top-1, top-5, and top-10 results of each test sentence. The accuracy of the top-n results was computed as:

$$Accuracy_{top-n} = \frac{\sum_{i=1}^{N} n_i}{N \times n} \tag{15}$$

where N is the number of test sentences. n_i is the number of acceptable paraphrases in the top-n paraphrases of the i-th test sentence.

We computed the accuracy on the whole test set (150 sentences) as well as on the three subsets, i.e., the 50 news sentences, 50 novel sentences, and 50 forum sentences. The results are shown in table 3.

It can be seen that the accuracy varies greatly on different test sets. The accuracy on the news sentences is the highest, while that on the forum sentences is the lowest. There are several reasons. First,

[11] The evaluation was based on the paraphrasing results using the combination of all seven PTs.

the largest PT used in the experiments is extracted using the bilingual parallel data, which are mostly from news documents. Thus, the test set of news sentences is more similar to the training data.

Second, the news sentences are formal while the novel and forum sentences are less formal. Especially, some of the forum sentences contain spelling mistakes and grammar mistakes.

Third, we find in the results that, most phrases paraphrased in the novel and forum sentences are commonly used phrases or words, such as "*food*", "*good*", "*find*", etc. These phrases are more difficult to paraphrase than the less common phrases, since they usually have much more paraphrases in the PTs. Therefore, it is more difficult to choose the right paraphrase from all the candidates when conducting sentence-level paraphrase generation.

Fourth, the forum sentences contain plenty of words such as "*board* (means *computer board*)", "*site* (means *web site*)", "*mouse* (means *computer mouse*)", etc. These words are polysemous and have particular meanings in the domains of computer science and internet. Our method performs poor when paraphrasing these words since the domain of a context sentence is hard to identify.

After observing the results, we find that there are three types of errors: (1) syntactic errors: the generated sentences are ungrammatical. About 32% of the unacceptable results are due to syntactic errors. (2) semantic errors: the generated sentences are incomprehensible. Nearly 60% of the unacceptable paraphrases have semantic errors. (3) non-paraphrase: the generated sentences are well formed and comprehensible but are not paraphrases of the input sentences. 8% of the unacceptable results are of this type. We believe that many of the errors above can be avoided by applying syntactic constraints and by making better use of context information in decoding, which is left as our future work.

7 Conclusion

This paper proposes a method that improves the SMT-based sentence-level paraphrase generation using phrasal paraphrases automatically extracted from different resources. Our contribution is that we combine multiple resources in the framework of SMT for paraphrase generation, in which the dictionary definitions and similar user queries are first used as phrasal paraphrases. In addition, we analyze and compare the contributions of different resources.

Experimental results indicate that although the contributions of the exploited resources differ a lot, they are all useful to sentence-level paraphrase generation. Especially, the dictionary definitions and similar user queries are effective for paraphrasing some certain types of phrases.

In the future work, we will try to use syntactic and context constraints in paraphrase generation to enhance the acceptability of the paraphrases. In addition, we will extract paraphrase patterns that contain more structural variation and try to combine the SMT-based and pattern-based systems for sentence-level paraphrase generation.

Acknowledgments

We would like to thank Mu Li for providing us with the SMT decoder. We are also grateful to Dongdong Zhang for his help in the experiments.

References

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

Regina Barzilay and Michael Elhadad. 1997. Using Lexical Chains for Text Summarization. In *Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, pages 10-17.

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

Regina Barzilay and Kathleen R. McKeown. 2001. Extracting Paraphrases from a Parallel Corpus. In *Proceedings of ACL*, pages 50-57.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. In *Computational Linguistics* 19(2): 263-311.

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

Andrew Finch, Young-Sook Hwang, and Eiichiro Sumita. 2005. Using Machine Translation Evaluation Techniques to Determine Sentence-level Semantic Equivalence. In *Proceedings of IWP*, pages 17-24.

Wei Gao, Cheng Niu, Jian-Yun Nie, Ming Zhou, Jian Hu, Kam-Fai Wong, and Hsiao-Wuen Hon. 2007. Cross-Lingual Query Suggestion Using Query Logs of Different Languages. In *Proceedings of SIGIR*, pages 463-470.

Lidija Iordanskaja, Richard Kittredge, and Alain Polguère. 1991. Lexical Selection and Paraphrase in a Meaning-Text Generation Model. In 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. 2004. Pharaoh: a Beam Search Decoder for Phrase-Based Statistical Machine Translation Models: User Manual and Description for Version 1.2.

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

De-Kang Lin. 1998. Automatic Retrieval and Clustering of Similar Words. In *Proceedings of COLING/ACL*, pages 768-774.

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

Nitin Madnani, Necip Fazil Ayan, Philip Resnik, and Bonnie J. Dorr. 2007. Using Paraphrases for Parameter Tuning in Statistical Machine Translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 120-127.

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.

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

Franz Josef Och and Hermann Ney. 2002. Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. In *Proceedings of ACL*, pages 295-302.

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.

Marius Pasca and Péter Dienes. 2005. Aligning Needles in a Haystack: Paraphrase Acquisition Across the Web. In *Proceedings of IJCNLP*, pages 119-130.

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.

Hua Wu and Ming Zhou. 2003. Synonymous Collocation Extraction Using Translation Information. In *Proceedings of ACL*, pages 120-127.