PAPER

Paraphrase Lattice for Statistical Machine Translation*

Takashi ONISHI[†a)], **Masao UTIYAMA**[†], ***Nonmembers***, *and* **Eiichiro SUMITA**[†], ***Member***

SUMMARY Lattice decoding in statistical machine translation (SMT) is useful in speech translation and in the translation of German because it can handle input ambiguities such as speech recognition ambiguities and German word segmentation ambiguities. In this paper, we show that lattice decoding is also useful for handling input variations. "Input variations" refers to the differences in input texts with the same meaning. Given an input sentence, we build a lattice which represents paraphrases of the input sentence. We call this a paraphrase lattice. Then, we give the paraphrase lattice as an input to a lattice decoder. The lattice decoder searches for the best path of the paraphrase lattice and outputs the best translation. Experimental results using the IWSLT dataset and the Europarl dataset show that our proposed method obtains significant gains in BLEU scores.

key words: *statistical machine translation, lattice decoding, paraphrasing, paraphrase lattice*

1. Introduction

Lattice decoding in SMT is useful in speech translation [1] and in the translation of German [2]. In speech translation, by using lattices that represent not only 1-best result but also other possibilities of speech recognition, we can take into account the ambiguities of speech recognition. Thus, the translation quality for lattice inputs is better than the quality for 1-best inputs.

In this paper, we show that lattice decoding is also useful for handling input variations. "Input variations" refers to the differences in input texts with the same meaning. For example,

"Is there a beauty salon?" and
"Is there a beauty parlor?"

have the same meaning with variations being *"beauty salon"* and *"beauty parlor"*. Since these variations are frequently found in natural language texts, mismatches of the expressions in input sentences and the expressions in training corpus lead to a decrease in translation quality.

Therefore, we propose a novel method that can handle input variations using paraphrases of input sentences and lattice decoding. In the proposed method, we regard a given input sentence as one of many variations (1-best). Given an input sentence, we build a lattice which represents paraphrases of that input sentence. We call this a paraphrase lattice. Then, we give the paraphrase lattice as an input for a lattice decoder. The lattice decoder searches for the best path of the paraphrase lattice and outputs the best translation. By using paraphrases of input sentences, we can translate expressions which are not found in the training corpus on the condition that paraphrases of them are found in the training corpus. Moreover, by using lattice decoding, we can employ a source-side language model as a decoding feature. Since this feature is affected by the source-side context, the lattice decoder can select an appropriate paraphrase and translate correctly.

This paper is organized as follows: Related works on lattice decoding and paraphrasing are presented in Sect. 2. The proposed method is described in Sect. 3. Experimental results on IWSLT dataset and Europarl dataset are shown in Sect. 4. Finally, the paper is concluded with a summary and a few directions for future work in Sect. 5.

2. Related Work

Lattice decoding has been used to handle preprocessing ambiguities. In speech translation, the whole process is divided into two parts, speech recognition and machine translation. An input of machine translation is generated by a speech recognizer. However, there are many ambiguities in speech recognition and 1-best quality of speech recognition is not sufficiently high. The quality of machine translation using only 1-best input is highly affected by errors in the input. Therefore, many approaches which use multiple hypotheses for speech recognition have been proposed. Bertoldi et al. [1] used a confusion network, which is a kind of lattice, as an input of machine translation. They made a confusion network which represents multiple hypotheses generated by a speech recognizer and used confusion network decoding. Bertoldi et al. [3] also proposed a method for handling misspellings using confusion network decoding. In text translation, the preprocessing is word segmentation. However, word segmentation difficulties arise for languages like Chinese, which are not separated by spaces, and languages like German, which have productive compounding. Therefore, Dyer [2] employed a segmentation lattice, which represents word segmentation ambiguities, and used lattice decoding [4]. However, to the best of our knowledge, there is no work employing a lattice representing paraphrases of an input sentence.

On the other hand, paraphrasing has been used to enrich SMT models. SMT systems learn translation mod-

Manuscript received August 6, 2010.
Manuscript revised January 18, 2011.

[†]The authors are with the National Institute of Information and Communications Technology, Kyoto-fu, 619–0289 Japan.

[*]This paper was presented at the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010).

a) E-mail: takashi.onishi@nict.go.jp

DOI: 10.1587/transinf.E94.D.1299

els from parallel sentences, so more parallel sentences lead to better translation quality. However, the available parallel sentences are limited. Therefore, many approaches to augment parallel sentences using paraphrasing have been proposed [5], [6]. Moreover, Callison-Burch et al. [7], [8] proposed a method which augments a translation phrase table using paraphrases which are automatically acquired from parallel corpora [9]. However, there is no work which augments input sentences by paraphrasing and representing these paraphrases in lattices.

3. Paraphrase Lattice for SMT

An overview of the proposed method is shown in Fig. 1. In advance, we automatically acquire a paraphrase table from a parallel corpus. Given an input sentence, we build a lattice which represents paraphrases of the input sentence using the paraphrase table which is acquired in advance. We call this lattice a paraphrase lattice. Then, we give the paraphrase lattice to a lattice decoder. The lattice decoder searches for the best path of the paraphrase lattice and outputs the best translation.

3.1 Paraphrase Table

A paraphrase table is a table which contains paraphrase pairs and their paraphrase probabilities. We acquire a paraphrase table from parallel corpus and build paraphrase lattices using the paraphrase table. In order to acquire paraphrases of unseen phrases, a different parallel corpus than the one for training is used to acquire the paraphrase table.

We acquire a paraphrase table from a parallel corpus using Bannard and Callison-Burch's method [9]. Their idea is, if two different phrases f_1, f_2 in one language are aligned to the same phrase c in another language, the two phrases are hypothesized to be paraphrases of each other. We acquired a paraphrase table in the same way.

The procedure is as follows:

1. Build a phrase table.
   Build a phrase table from a parallel corpus using standard phrase-based SMT techniques. We used GIZA++ [10] and grow-diag-final-and heuristic for alignment.
2. Filter the phrase table by the sigtest-filter.
   The phrase table built in 1 contains many inappropriate phrase pairs. Therefore, in order to reduce the computational costs in the next step and to acquire highly-accurate paraphrase pairs, we filter the phrase table by the sigtest-filter [11]. For the sigtest-filter, we used "-l a+e -n 30" setting, which means that phrase pairs which co-occur only once are removed and phrase pairs which are in the top 30 of each source phrase are kept.
3. Calculate the paraphrase probability.
   Calculate the paraphrase probability $p(f_2|f_1)$ if f_2 is hypothesized to be a paraphrase of f_1. The paraphrase probability $p(f_2|f_1)$ is defined at [9].

   $$p(f_2|f_1) = \sum_c P(c|f_1)P(f_2|c)$$

   where $P(c|f_1)$ and $P(f_2|c)$ are phrase translation probabilities which are calculated in 1. The paraphrase probability is also used for lattice decoding.

4. Acquire a paraphrase pair.
   Acquire (f_1, f_2) as a paraphrase pair if $p(f_2|f_1) > p(f_1|f_1)$. The purpose of this threshold is to keep highly accurate paraphrase pairs. In experiments, more than 80% of the paraphrase pairs were eliminated by this threshold.

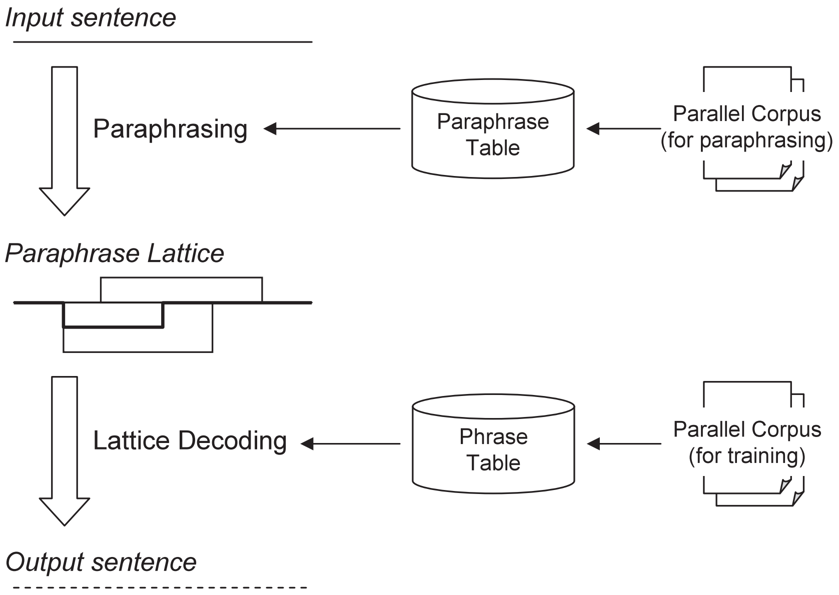

Fig. 1 Overview of the proposed method.

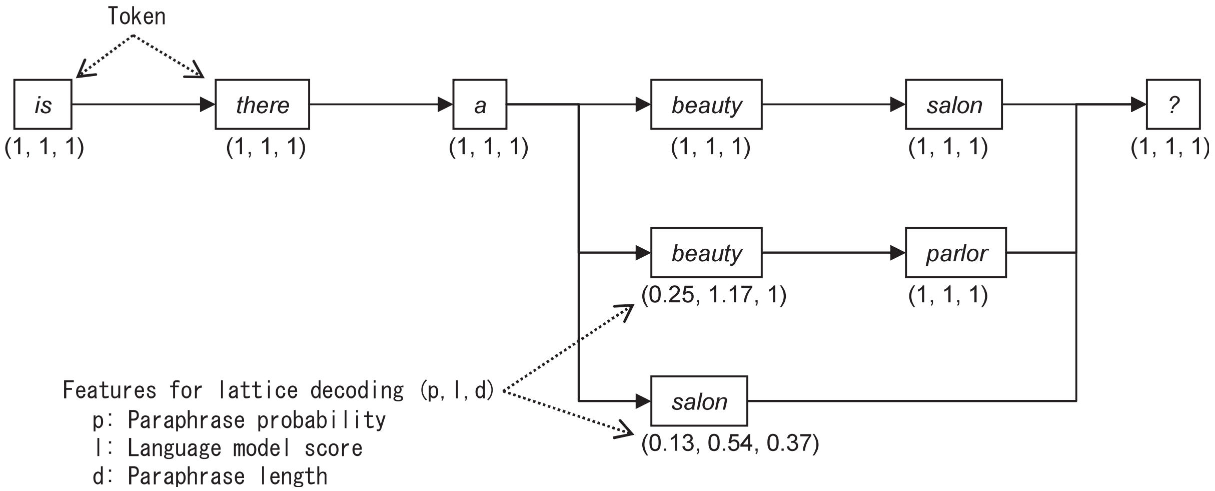

Fig. 2 An example of a paraphrase lattice. The input sentence is *"is there a beauty salon ?"*. Each node represents a token. Three values under each node are features for lattice decoding.

3.2 Paraphrase Lattice

An input sentence is paraphrased using the paraphrase table which is described in previous section and is transformed into a paraphrase lattice. A paraphrase lattice is a lattice which represents paraphrases of the input sentence. An example of a paraphrase lattice is shown in Fig. 2. In this example, an input sentence is

"is there a beauty salon ?".

This paraphrase lattice contains two paraphrase pairs

"beauty salon" → *"beauty parlor"*
"beauty salon" → *"salon"*

and represents following three sentences.

"is there a beauty salon ?"
"is there a beauty parlor ?"
"is there a salon ?"

If *"beauty salon"* is not found in a phrase table, we can't translate *"is there a beauty salon ?"* properly even if *"beauty parlor"* is found in the phrase table. However, using the paraphrase lattice as an input of a lattice decoder, we can translate it properly through the path for *"is there a beauty parlor ?"*.

3.3 Lattice Decoding

We use Moses [12] to decode paraphrase lattices. Moses is an open source SMT system which supports lattice decoding. In lattice decoding, Moses searches the best path and the best translation according to not only normal SMT features but also additional features associated with each node of the paraphrase lattice. Weights of these features are optimized using Minimum Error Rate Training (MERT) [13].

We use the following four features as additional features:

- Paraphrase probability (p)
  A paraphrase probability is the probability that a source-side phrase f_1 can be paraphrased into f_2. The paraphrase probabilities are calculated when acquiring the paraphrase table.
  $$h_p = p(f_2|f_1) = \sum_c P(c|f_1)P(f_2|c)$$
- Language model score (l)
  A language model score is the ratio between the source-side language model probability of the paraphrased sentence ($para$) and that of the original input sentence ($orig$). We use a 5-gram language model trained by the SRI Language Modeling Toolkit [14].
  $$h_l = \frac{lm(para)}{lm(orig)}$$
- Normalized language model score (L)
  A normalized language model score is the same as the language model score, except the language model probability is normalized by the sentence length. The sentence length is calculated as the number of tokens.
  $$h_L = \frac{LM(para)}{LM(orig)}$$
  where
  $$LM(sent) = lm(sent)^{\frac{1}{length(sent)}}$$
- Paraphrase length (d)
  A paraphrase length is the difference between the original input sentence length and the paraphrased sentence length.
  $$h_d = \exp(length(para) - length(orig))$$

Table 1 Training corpus statistics of IWSLT dataset.

	English ↔ Japanese		English ↔ Chinese	
Sentences	39,953		39,953	
Tokens	365,229	429,365	365,229	351,060
Distinct	9,814	11,413	9,814	11,178

Table 2 Training corpus statistics of Europarl dataset. We filtered out long sentences (more than 40 tokens).

	German ↔ English		German ↔ Spanish		Spanish ↔ English	
Sentences	997,575		948,385		964,133	
Tokens	20,341,901	21,432,529	18,973,290	20,579,857	20,823,726	20,210,162
Distinct	226,385	74,579	215,005	112,321	112,771	72,504

Values of these features are calculated only if the node is the first node of a paraphrased path. In other nodes, including original nodes, the values of these features are 1. For example, in Fig. 2, *"beauty"* and *"parlor"* node are added to the paraphrase lattice because *"beauty parlor"* is a paraphrase of *"beauty salon"*. Then, the feature values of *"beauty"* are calculated as follows, and the feature values of *"parlor"* are all 1.

$$h_p = p(\text{“beauty parlor”} \mid \text{“beauty salon”}) = 0.25$$

$$h_l = \frac{lm(\text{“is there a beauty parlor ?”})}{lm(\text{“is there a beauty salon ?”})} = 1.17$$

$$h_d = \exp(6 - 6) = 1$$

The features related to the source-side language model, such as (l) and (L), are affected by the source-side context even if the same paraphrase pair is applied. As these features can penalize paraphrases which are not appropriate to the source-side context, likely appropriate paraphrases are selected and appropriate translations are output in lattice decoding. The features related to the sentence length, such as (L) and (d), are added to penalize the source-side language model score in case the paraphrased sentence length is shorter than the original input sentence length and the source-side language model score is unreasonably low.

4. Experiments

In order to evaluate our proposed method, we conducted experiments with various translation directions and various training corpus sizes.

4.1 Datasets

We used the IWSLT dataset [15] and the Europarl dataset [16].

4.1.1 IWSLT

We conducted English-to-Japanese (en→ja) and English-to-Chinese (en→zh) translation experiments using the IWSLT dataset. This dataset contains training sets of nearly 40 K sentences (detailed statistics are shown in Table 1) and about 500-sentence sets (dev1, dev2 and dev3) for development and evaluation. The domain for the IWSLT dataset is travel, and the average sentence length is short (about 8 tokens). We used the dev1 set for parameter tuning (MERT), the dev2 set for selecting the best setting, which is described below, for the proposed system and the dev3 set for evaluation.

4.1.2 Europarl

We conducted German (de), English (en) and Spanish (es) translation experiments using the Europarl dataset provided for the WMT08 shared task[†]. This dataset contains training sets of close to 1 M sentences (detailed statistics are shown in Table 2), a nearly 1.4 M-sentence set for building a language model and 2000-sentence sets (dev2006, devtest2006, test2006 and test2007 and test2008) for development and evaluation. The Europarl corpus is extracted from European Parliament proceedings, and the average sentence length is long (about 30 tokens). We used the dev2006 set for parameter tuning, the devtest2006 set for selecting the best setting of the proposed system and the other sets for evaluation.

4.2 Baseline Systems

We used a default Moses system (Moses) and a system proposed by Callison-Burch et al. [7] (CCB) as baseline systems.

4.2.1 Moses

We used Moses with the default settings and no paraphrasing. In Moses, we used phrase translation model (5 features), 5-gram target-side language model (1 feature), distance-based reordering model (1 feature), lexicalized reordering model (6 features) and word penalty (1 feature). These features were also used in CCB system and the proposed system.

4.2.2 CCB

Callison-Burch et al. proposed a method that augments a translation phrase table using paraphrases in order to translate unseen phrases. We implemented their method in the

[†] http://www.statmt.org/wmt08/shared-task.html

Table 3 Translation results for IWSLT dataset (%BLEU). ∗∗, ∗ and ⋄ indicate significance level of $p < 0.01$, $p < 0.05$ and $p < 0.1$ respectively.

Direction	Moses	CCB	Proposed	vs Moses	vs CCB	Paraphrase table size
en→ja	38.98	39.24	**40.34**	+1.36 ∗	+1.10 ⋄	53 K (from en-zh)
en→zh	25.11	26.14	**27.06**	+1.95 ∗∗	+0.92 ∗	47 K (from en-ja)

Table 4 Translation results with various translation directions. The training corpus size of each direction is 10 K sentences. Paraphrase table size is calculated as the number of paraphrase pairs. ∗∗ and ⋄ indicate significance level of $p < 0.01$ and $p < 0.1$ respectively.

Direction	Moses	CCB	Proposed	vs Moses	vs CCB	Optimal setting	Paraphrase table size
de→en	20.61	21.63	**21.69**	+1.07 ∗∗	+0.05 ⋄	(p), (p, l, d)	5.3 M (from de-es)
de→es	19.66	20.45	**20.85**	+1.19 ∗∗	+0.40 ∗∗	(p), (p, l, d)	6.2 M (from de-en)
en→de	15.82	15.89	**16.08**	+0.26 ∗∗	+0.19 ∗∗	(p), (p, l, d)	2.7 M (from en-es)
en→es	27.23	27.50	**27.65**	+0.42 ∗∗	+0.15 ∗∗	(L), (p)	3.7 M (from en-de)
es→de	15.10	15.37	**15.54**	+0.44 ∗∗	+0.17 ∗∗	(p), (p, l, d)	4.2 M (from es-en)
es→en	26.66	27.19	**27.38**	+0.71 ∗∗	+0.18 ∗∗	(L), (p, l)	4.2 M (from es-de)

Moses decoder. Using the paraphrase table described in Sect. 3.1, we augmented the phrase table with paraphrased phrases not found in the original phrase table. As mentioned in [7], we also used an additional feature. If the entry is generated by paraphrasing, the value of this feature is the paraphrase probability (p). If otherwise, the value is 1. Weights of this feature and other Moses features described above were optimized using MERT. In experiments using the Europarl dataset, we used 1-best paraphrase pair per phrase to avoid combinatorial explosion of the phrase table.

4.3 Proposed System

In the proposed system, we conducted experiments with various settings for paraphrasing and lattice decoding. Then, we selected the best setting according to the BLEU score of the dev2 set on IWSLT and the devtest2006 set on Europarl.

4.3.1 Paraphrase Limiting

Since the paraphrase table is automatically acquired, there are many erroneous paraphrase pairs. Building paraphrase lattices using all the erroneous paraphrase pairs and decoding these paraphrase lattices causes degradation in translation quality and a high computational complexity. Therefore, we limited the number of paraphrasing per phrase and per sentence. The number of paraphrasings per phrase was limited to 3 in IWSLT and 1 in Europarl. The number of paraphrasings per sentence was limited to 2 × (sentence length) in IWSLT and 0.5 × (sentence length) in Europarl.

As a criterion for limiting the number of paraphrasings, we use three features (p), (l) and (L), which are the same as the features described in Sect. 3.3. When building paraphrase lattices, we apply paraphrase pairs in descending order of the value of the criterion. Using the source-side language model score as a criterion, paraphrase pairs which are suitable for the context are preferred.

4.3.2 Features for Lattice Decoding

In experiments, we use four combinations of features, (p), (p, l), (p, L) and (p, l, d). Weights of these features and other Moses features are optimized using MERT.

4.3.3 Finding Optimal Settings

As previously mentioned, we have three choices for the criterion for building paraphrase lattices and four combinations of features for lattice decoding. Thus, there are $3 \times 4 = 12$ combinations of these settings. We conducted parameter tuning (MERT) with the dev1 set and the dev2006 set for each setting and selected as best the setting which received the highest BLEU score for the dev2 set and the devtest2006 set.

4.4 Results

We conducted experiments with various translation directions and various training corpus sizes using the IWSLT dataset and the Europarl dataset. The experimental results are shown in Table 3 to Table 5. We used the case-insensitive BLEU score [17] for evaluation. We used the BLEU score for the dev3 set of the IWSLT dataset and the average of the BLEU scores for the test2006, test2007 and test2008 sets of the Europarl dataset. Statistical significance of the difference from the baseline systems was measured by using paired bootstrap resampling [18].

4.4.1 IWSLT

Table 3 shows the experimental results for en→ja and en→zh translations. For en→ja translation, 53 K pairs of English paraphrases were acquired from the en-zh parallel corpus. Similarly, for en→zh translation, 47 K pairs of English paraphrases were acquired from the en-ja parallel corpus. In en→ja translation, the proposed system obtained the highest score with 40.34 and an absolute improvement of 1.36 BLEU points over Moses and 1.10 BLEU

Table 5 Translation results (de→en) with various training corpus sizes. The paraphrase rate is a percentage of sentences which were translated through paraphrased path. ** and ⋄ indicate significance level of $p < 0.01$ and $p < 0.1$ respectively.

Corpus size	Moses	CCB	Proposed	vs Moses	vs CCB	Optimal setting	Paraphrase rate
10 K	20.61	21.63	**21.69**	+1.07 **	+0.05 ⋄	(p), (p, l, d)	82.1%
20 K	22.64	22.83	**23.54**	+0.90 **	+0.70 **	(p), (p, l, d)	75.5%
40 K	24.04	**24.71**	24.70	+0.66 **	−0.01	(L), (p, L)	65.4%
80 K	25.25	25.42	**25.83**	+0.58 **	+0.41 **	(p), (p, l, d)	57.6%
160 K	26.35	26.23	**26.44**	+0.10 ⋄	+0.21 **	(p), (p, l)	19.5%
320 K	27.19	27.23	**27.32**	+0.13 ⋄	+0.09	(p), (p, l, d)	24.6%
640 K	27.69	27.47	**27.75**	+0.06	+0.27 **	(L), (p, l, d)	5.8%
All (1.0 M)	27.90	27.92	**28.05**	+0.16 **	+0.13 **	(p), (p, l)	16.5%

Table 6 Translation examples on IWSLT.

Source:	i'd like to get my **trousers** pressed by ten tomorrow morning .
Reference:	この ズボン を 明朝 十 時 迄 に プレス し て 下さい 。
Moses Output:	明日 の 朝 の 十 時 迄 に **trousers** に アイロン を 掛け て 欲しい の です が 。
Paraphrase:	trousers ⇒ pants
Proposed Output:	明日 の 朝 十 時 迄 に 私 の ズボン に アイロン を 掛け て 頂き たい の です が 。
Source:	give me some **anodyne** , please .
Reference:	鎮痛 剤 を 下さい 。
Moses Output:	**anodyne** を 見せ て 下さい 。
Paraphrase:	anodyne ⇒ sedative
Proposed Output:	鎮痛 剤 を 御 願い し ます 。
Source:	i'd like the **smallest** one you have .
Reference:	一 番 小さい の が 良い です 。
Moses Output:	一番 小さい 物 が 欲しい の です が 。
Paraphrase:	smallest ⇒ small
Proposed Output:	小さい の が 欲しい の です が 。

points over CCB. In en→zh translation, the proposed system also obtained the highest score with 27.06 and an absolute improvement of 1.95 BLEU points over Moses and 0.92 BLEU points over CCB. As the relation of three systems is Moses < CCB < Proposed, paraphrasing is useful for SMT and using paraphrase lattices and lattice decoding especially is more useful than augmenting the phrase table.

In the Proposed system, the optimal criteria for building paraphrase lattices and the combination of features for lattice decoding were (p) and (p, L) in en→ja translation and (L) and (p, l) in en→zh translation. In each case, since the features related to the source-side language model were selected, using a source-side language model is useful for decoding paraphrase lattices.

Table 6 shows translation examples. The first and second examples show that paraphrasing improves the translation quality. On the other hand, the third example shows a degraded example where the meaning of the sentence is changed by paraphrasing.

4.4.2 Europarl

We conducted translation experiments with various directions. Table 4 shows the experimental results. The training corpus size in each direction is 10 K sentences. In all directions, the proposed system received a higher BLEU score than the baseline systems. We acquired paraphrase tables from different parallel corpora than those for training. For example, in de→en translation, we acquired a paraphrase table from 1 M de-es parallel corpus. The sizes of the acquired paraphrase tables vary from 2.7 M pairs to 6.2 M pairs. As German had many distinct tokens, about three times as many as English, the size of German paraphrase table was larger than English one. As a result, improvements against Moses are large in de→en and de→es translations but small in en→de and en→es translations.

Table 5 shows the experimental results of de→en translation with various sizes of training corpora. We used 10 K, 20 K, 40 K, 80 K, 160 K, 320 K, 640 K and all (about 1.0 M) sentences for training and 5.3 M pairs of paraphrases. The proposed system consistently received a higher score than the baseline systems except for 40 K, where the proposed system was slightly inferior to CCB. However, as the size of the training corpus increases, gains over the baseline decrease and the paraphrase rate drops. This shows that the number of useful paraphrases decreases as a result of the broadness of the training corpus coverage.

The optimal criteria for building paraphrase lattices and the combination of features for lattice decoding on each experiment are shown in Table 4 and Table 5. The setting of (p) and (p, l, d) was selected in many cases and the features related to the source-side language model were selected in every case.

5. Conclusion

This paper proposed a novel method for transforming an input sentence into a paraphrase lattice, which represents paraphrases of the input sentence, and applying lattice decoding. Since our method can employ source-side language models as a decoding feature, a lattice decoder can select a proper paraphrased path and translate it properly. The experimental results showed that the proposed method consistently outperformed baseline systems in various translation directions and various training corpus sizes.

In the future, we plan to apply this method with paraphrases derived from a massive corpus such as the Web corpus and apply this method to a hierarchical phrase-based SMT.

References

[1] N. Bertoldi, R. Zens, and M. Federico, "Speech translation by confusion network decoding," Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp.1297–1300, 2007.

[2] C. Dyer, "Using a maximum entropy model to build segmentation lattices for MT," Proc. Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL), pp.406–414, 2009.

[3] N. Bertoldi, M. Cettolo, and M. Federico, "Statistical machine translation of texts with misspelled words," Proc. Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT), pp.412–419, 2010.

[4] C. Dyer, S. Muresan, and P. Resnik, "Generalizing word lattice translation," Proc. 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT), pp.1012–1020, 2008.

[5] F. Bond, E. Nichols, D.S. Appling, and M. Paul, "Improving statistical machine translation by paraphrasing the training data," Proc. International Workshop on Spoken Language Translation (IWSLT), pp.150–157, 2008.

[6] P. Nakov, "Improved statistical machine translation using monolingual paraphrases," Proc. European Conference on Artificial Intelligence (ECAI), pp.338–342, 2008.

[7] C. Callison-Burch, P. Koehn, and M. Osborne, "Improved statistical machine translation using paraphrases," Proc. Human Language Technology conference — North American chapter of the Association for Computational Linguistics (HLT-NAACL), pp.17–24, 2006.

[8] Y. Marton, C. Callison-Burch, and P. Resnik, "Improved statistical machine translation using monolingually-derived paraphrases," Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP), pp.381–390, 2009.

[9] C. Bannard and C. Callison-Burch, "Paraphrasing with bilingual parallel corpora," Proc. 43rd Annual Meeting of the Association for Computational Linguistics (ACL), pp.597–604, 2005.

[10] F.J. Och and H. Ney, "A systematic comparison of various statistical alignment models," Computational Linguistics, vol.29, no.1, pp.19–51, 2003.

[11] J.H. Johnson, J. Martin, G. Foster, and R. Kuhn, "Improving translation quality by discarding most of the phrasetable," Proc. 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp.967–975, 2007.

[12] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," Proc. 45th Annual Meeting of the Association for Computational Linguistics (ACL), pp.177–180, 2007.

[13] F.J. Och, "Minimum error rate training in statistical machine translation," Proc. 41st Annual Meeting of the Association for Computational Linguistics (ACL), pp.160–167, 2003.

[14] A. Stolcke, "SRILM — An extensible language modeling toolkit," Proc. International Conference on Spoken Language Processing (ICSLP), pp.901–904, 2002.

[15] C.S. Fordyce, "Overview of the IWSLT 2007 evaluation campaign," Proc. International Workshop on Spoken Language Translation (IWSLT), pp.1–12, 2007.

[16] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," Proc. 10th Machine Translation Summit (MT Summit), pp.79–86, 2005.

[17] K. Papineni, S. Roukos, T. Ward, and W.J. Zhu, "BLEU: A method for automatic evaluation of machine translation," Proc. 40th Annual Meeting of the Association for Computational Linguistics (ACL), pp.311–318, 2002.

[18] P. Koehn, "Statistical significance tests for machine translation evaluation," Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP), pp.388–395, 2004.

Takashi Onishi received the M.S. degree in Information and Communication Engineering from the University of Tokyo in 2006. He is currently an expert researcher of National Institute of Information and Communications Technology. His research interests include machine translation.

Masao Utiyama received the B.S. degree in computer science from University of Tsukuba, Japan in 1992, the M.S. degree in computer science from University of Tsukuba in 1994, the Ph.D. degree in engineering from University of Tsukuba in 1997. From 1997 to 1999, he was a research associate at the Shinshu University, Japan. He has been a member of National Institute of Information and Communications Technology (NICT), Japan since 1999. He is a senior researcher at NICT.

Eiichiro Sumita received the M.S. degree in computer science from the University of Electro-Communications in 1982 and the Ph.D. degree in engineering from Kyoto University in 1999. Dr. Sumita is the group leader of NICT/MASTAR Project/Language Translation Group, and the visiting professor of Kobe University. His research interests include machine translation and e-Learning.