# Paraphrase Lattice for Statistical Machine Translation

**Takashi Onishi** and **Masao Utiyama** and **Eiichiro Sumita**
Language Translation Group, MASTAR Project
National Institute of Information and Communications Technology
3-5 Hikaridai, Keihanna Science City, Kyoto, 619-0289, JAPAN
{takashi.onishi,mutiyama,eiichiro.sumita}@nict.go.jp

## Abstract

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. We show that lattice decoding is also useful for handling input variations. 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 the lattice decoder. The decoder selects the best path for decoding. Using these paraphrase lattices as inputs, we obtained significant gains in BLEU scores for IWSLT and Europarl datasets.

## 1 Introduction

Lattice decoding in SMT is useful in speech translation and in the translation of German (Bertoldi et al., 2007; Dyer, 2009). 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 of 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 in *"beauty salon"* and *"beauty parlor"*. Since these variations are frequently found in natural language texts, a mismatch of the expressions in source sentences and the expressions in training corpus leads to a decrease in translation quality. Therefore, we propose a novel method that can handle input variations using paraphrases and lattice decoding. In the proposed method, we regard a given source sentence as one of many variations (1-best). Given an input sentence, we build a paraphrase lattice which represents paraphrases of the input sentence. Then, we give the paraphrase lattice as an input to the Moses decoder (Koehn et al., 2007). Moses selects the best path for decoding. By using paraphrases of source sentences, we can translate expressions which are not found in a training corpus on the condition that paraphrases of them are found in the training corpus. Moreover, by using lattice decoding, we can employ the source-side language model as a decoding feature. Since this feature is affected by the source-side context, the decoder can choose a proper paraphrase and translate correctly.

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

## 2 Related Work

Lattice decoding has been used to handle ambiguities of preprocessing. Bertoldi et al. (2007) employed a confusion network, which is a kind of lattice and represents speech recognition hypotheses in speech translation. Dyer (2009) also employed a segmentation lattice, which represents ambiguities of compound word segmentation in German, Hungarian and Turkish translation. However, to the best of our knowledge, there is no work which employed a lattice representing paraphrases of an input sentence.

On the other hand, paraphrasing has been used to enrich the SMT model. Callison-Burch et

*Proceedings of the ACL 2010 Conference Short Papers*, pages 1–5,
Uppsala, Sweden, 11-16 July 2010.

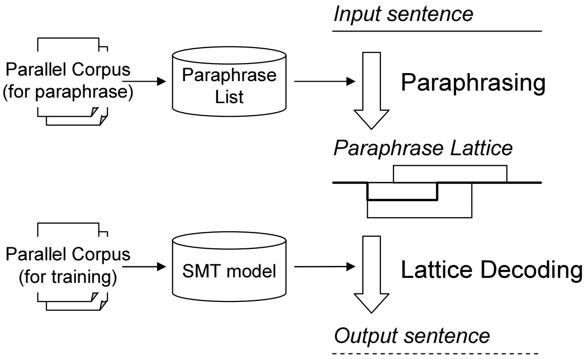

Figure 1: Overview of the proposed method.

al. (2006) and Marton et al. (2009) augmented the translation phrase table with paraphrases to translate unknown phrases. Bond et al. (2008) and Nakov (2008) augmented the training data by paraphrasing. However, there is no work which augments input sentences by paraphrasing and represents them in lattices.

## 3 Paraphrase Lattice for SMT

Overview of the proposed method is shown in Figure 1. In advance, we automatically acquire a paraphrase list from a parallel corpus. In order to acquire paraphrases of unknown phrases, this parallel corpus is different from the parallel corpus for training.

Given an input sentence, we build a lattice which represents paraphrases of the input sentence using the paraphrase list. We call this lattice a paraphrase lattice. Then, we give the paraphrase lattice to the lattice decoder.

### 3.1 Acquiring the paraphrase list

We acquire a paraphrase list using Bannard and Callison-Burch (2005)'s method. Their idea is, if two different phrases $e_1$, $e_2$ in one language are aligned to the same phrase $c$ in another language, they are hypothesized to be paraphrases of each other. Our paraphrase list is acquired in the same way.

The procedure is as follows:

1. Build a phrase table.

   Build a phrase table from parallel corpus using standard SMT techniques.

2. Filter the phrase table by the sigtest-filter.

   The phrase table built in 1 has many inappropriate phrase pairs. Therefore, we filter the phrase table and keep only appropriate phrase pairs using the sigtest-filter (Johnson et al., 2007).

3. Calculate the paraphrase probability.

   Calculate the paraphrase probability $p(e_2|e_1)$ if $e_2$ is hypothesized to be a paraphrase of $e_1$.

   $$p(e_2|e_1) = \sum_c P(c|e_1)P(e_2|c)$$

   where $P(\cdot|\cdot)$ is phrase translation probability.

4. Acquire a paraphrase pair.

   Acquire ($e_1$, $e_2$) as a paraphrase pair if $p(e_2|e_1) > p(e_1|e_1)$. The purpose of this threshold is to keep highly-accurate paraphrase pairs. In experiments, more than 80% of paraphrase pairs were eliminated by this threshold.

### 3.2 Building paraphrase lattice

An input sentence is paraphrased using the paraphrase list and transformed into a paraphrase lattice. The paraphrase lattice is a lattice which represents paraphrases of the input sentence. An example of a paraphrase lattice is shown in Figure 2. In this example, an input sentence is *"is there a beauty salon ?"*. This paraphrase lattice contains two paraphrase pairs *"beauty salon"* = *"beauty parlor"* and *"beauty salon"* = *"salon"*, and represents following three sentences.

- *is there a beauty salon ?*
- *is there a beauty parlor ?*
- *is there a salon ?*

In the paraphrase lattice, each node consists of a token, the distance to the next node and features for lattice decoding. We use following four features for lattice decoding.

- Paraphrase probability (p)

  A paraphrase probability $p(e_2|e_1)$ calculated when acquiring the paraphrase.

  $h_p = p(e_2|e_1)$

- Language model score (l)

  A ratio between the language model probability of the paraphrased sentence (para) and that of the original sentence (orig).

  $h_l = \frac{lm(para)}{lm(orig)}$

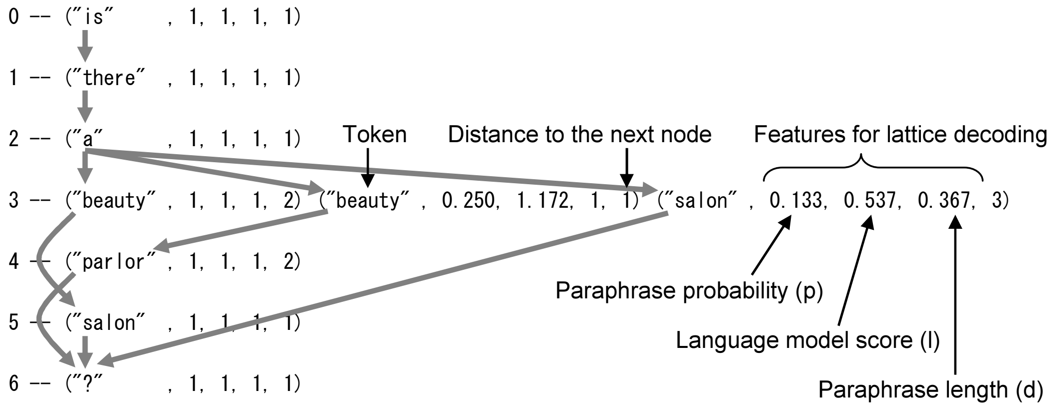
0 -- ("is" , 1, 1, 1, 1)
1 -- ("there" , 1, 1, 1, 1)
2 -- ("a" , 1, 1, 1, 1)
3 -- ("beauty" , 1, 1, 1, 2) ("beauty" , 0.250, 1.172, 1, 1) ("salon" , 0.133, 0.537, 0.367, 3)
4 -- ("parlor" , 1, 1, 1, 2)
5 -- ("salon" , 1, 1, 1, 1)
6 -- ("?" , 1, 1, 1, 1)
Token
Distance to the next node
Features for lattice decoding
Paraphrase probability (p)
Language model score (l)
Paraphrase length (d)

Figure 2: An example of a paraphrase lattice, which contains three features of (p, l, d).

- Normalized language model score (L)

  A language model score where 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)

  The difference between the original sentence length and the paraphrased sentence length.

  $h_d = \exp(length(para) - length(orig))$

The values of these features are calculated only if the node is the first node of the paraphrase, for example the second *"beauty"* and *"salon"* in line 3 of Figure 2. In other nodes, for example *"parlor"* in line 4 and original nodes, we use 1 as the values of features.

The features related to the language model, such as (l) and (L), are affected by the context of source sentences even if the same paraphrase pair is applied. As these features can penalize paraphrases which are not appropriate to the context, appropriate paraphrases are chosen 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 language model score in case the paraphrased sentence length is shorter than the original sentence length and the language model score is unreasonably low.

In experiments, we use four combinations of these features, (p), (p, l), (p, L) and (p, l, d).

### 3.3 Lattice decoding

We use Moses (Koehn et al., 2007) as a decoder for lattice decoding. Moses is an open source SMT system which allows lattice decoding. In lattice decoding, Moses selects the best path and the best translation according to features added in each node and other SMT features. These weights are optimized using Minimum Error Rate Training (MERT) (Och, 2003).

## 4 Experiments

In order to evaluate the proposed method, we conducted English-to-Japanese and English-to-Chinese translation experiments using IWSLT 2007 (Fordyce, 2007) dataset. This dataset contains EJ and EC parallel corpus for the travel domain and consists of 40k sentences for training and about 500 sentences sets (dev1, dev2 and dev3) for development and testing. We used the dev1 set for parameter tuning, the dev2 set for choosing the setting of the proposed method, which is described below, and the dev3 set for testing.

The English-English paraphrase list was acquired from the EC corpus for EJ translation and 53K pairs were acquired. Similarly, 47K pairs were acquired from the EJ corpus for EC translation.

### 4.1 Baseline

As baselines, we used Moses and Callison-Burch et al. (2006)'s method (hereafter CCB). In Moses, we used default settings without paraphrases. In CCB, we paraphrased the phrase table using the automatically acquired paraphrase list. Then, we augmented the phrase table with paraphrased phrases which were not found in the original phrase table. Moreover, we used an additional feature whose value was the paraphrase probability (p) if the entry was generated by paraphrasing and

|  | Moses (w/o Paraphrases) | CCB | Proposed Method |
|---|---|---|---|
| EJ | 38.98 | 39.24 (+0.26) | **40.34** (+1.36) |
| EC | 25.11 | 26.14 (+1.03) | **27.06** (+1.95) |

Table 1: Experimental results for IWSLT (%BLEU).

1 if otherwise. Weights of the feature and other features in SMT were optimized using MERT.

### 4.2 Proposed method

In the proposed method, we conducted experiments with various settings for paraphrasing and lattice decoding. Then, we chose the best setting according to the result of the dev2 set.

#### 4.2.1 Limitation of paraphrasing

As the paraphrase list was automatically acquired, there were many erroneous paraphrase pairs. Building paraphrase lattices with all erroneous paraphrase pairs and decoding these paraphrase lattices caused high computational complexity. Therefore, we limited the number of paraphrasing per phrase and per sentence. The number of paraphrasing per phrase was limited to three and the number of paraphrasing per sentence was limited to twice the size of the sentence length.

As a criterion for limiting the number of paraphrasing, we use three features (p), (l) and (L), which are same as the features described in Subsection 3.2. When building paraphrase lattices, we apply paraphrases in descending order of the value of the criterion.

#### 4.2.2 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 with the dev1 set for each setting and used as best the setting which got the highest BLEU score for the dev2 set.

### 4.3 Results

The experimental results are shown in Table 1. We used the case-insensitive BLEU metric for evaluation. In EJ translation, the proposed method obtained the highest score of 40.34%, which achieved an absolute improvement of 1.36 BLEU points over Moses and 1.10 BLEU points over CCB. In EC translation, the proposed method also obtained the highest score of 27.06% and achieved 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 Method, paraphrasing is useful for SMT and using paraphrase lattices and lattice decoding is especially more useful than augmenting the phrase table. In Proposed Method, the criterion for building paraphrase lattices and the combination of features for lattice decoding were (p) and (p, L) in EJ translation and (L) and (p, l) in EC translation. Since features related to the source-side language model were chosen in each direction, using the source-side language model is useful for decoding paraphrase lattices.

We also tried a combination of Proposed Method and CCB, which is a method of decoding paraphrase lattices with an augmented phrase table. However, the result showed no significant improvements. This is because the proposed method includes the effect of augmenting the phrase table.

Moreover, we conducted German-English translation using the Europarl corpus (Koehn, 2005). We used the WMT08 dataset[1], which consists of 1M sentences for training and 2K sentences for development and testing. We acquired 5.3M pairs of German-German paraphrases from a 1M German-Spanish parallel corpus. We conducted experiments with various sizes of training corpus, using 10K, 20K, 40K, 80K, 160K and 1M. Figure 3 shows the proposed method consistently get higher score than Moses and CCB.

## 5 Conclusion

This paper has proposed a novel method for transforming a source sentence into a paraphrase lattice and applying lattice decoding. Since our method can employ source-side language models as a decoding feature, the decoder can choose proper paraphrases and translate properly. The experimental results showed significant gains for the IWSLT and Europarl dataset. In IWSLT dataset, we obtained 1.36 BLEU points over Moses in EJ translation and 1.95 BLEU points over Moses in

[1] http://www.statmt.org/wmt08/

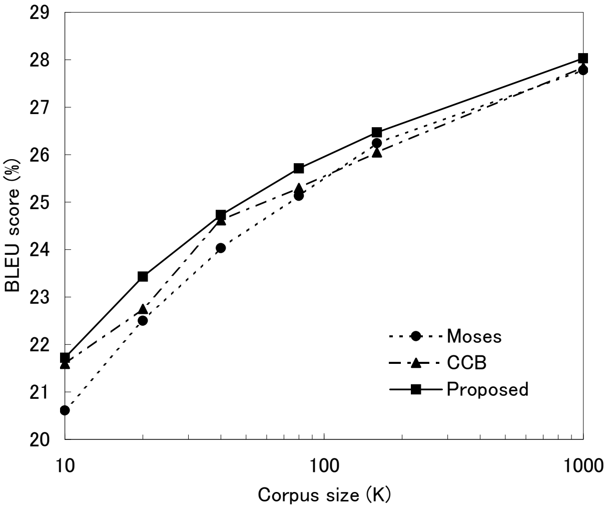

Figure 3: Effect of training corpus size.

EC translation. In Europarl dataset, the proposed method consistently get higher score than baselines.

In future work, 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

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with Bilingual Parallel Corpora. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 597–604.

Nicola Bertoldi, Richard Zens, and Marcello Federico. 2007. Speech translation by confusion network decoding. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1297–1300.

Francis Bond, Eric Nichols, Darren Scott Appling, and Michael Paul. 2008. Improving Statistical Machine Translation by Paraphrasing the Training Data. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, pages 150–157.

Chris Callison-Burch, Philipp Koehn, and Miles Osborne. 2006. Improved Statistical Machine Translation Using Paraphrases. In *Proceedings of the Human Language Technology conference - North American chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 17–24.

Chris Dyer. 2009. Using a maximum entropy model to build segmentation lattices for MT. In *Proceedings of the Human Language Technology conference - North American chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 406–414.

Cameron S. Fordyce. 2007. Overview of the IWSLT 2007 Evaluation Campaign. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, pages 1–12.

J Howard Johnson, Joel Martin, George Foster, and Roland Kuhn. 2007. Improving Translation Quality by Discarding Most of the Phrasetable. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 967–975.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 177–180.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the 10th Machine Translation Summit (MT Summit)*, pages 79–86.

Yuval Marton, Chris Callison-Burch, and Philip Resnik. 2009. Improved Statistical Machine Translation Using Monolingually-Derived Paraphrases. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 381–390.

Preslav Nakov. 2008. Improved Statistical Machine Translation Using Monolingual Paraphrases. In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, pages 338–342.

Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 160–167.