Searching Translation Memories for Paraphrases

Masao Utiyama[1] **Graham Neubig**[1,2] **Takashi Onishi**[1] **Eiichiro Sumita**[1]
[1]National Institute of Information and Communications Technology (NICT)
Keihanna Science City, Kyoto 619-0288, Japan
{mutiyama, eiichiro.sumita}@nict.go.jp
[2]Kyoto University
Yoshida Honmachi, Sakyo-ku, Kyoto 606-8501, Japan
neubig@ar.media.kyoto-u.ac.jp

Abstract

Translation memories (TMs) are very useful tools for translating texts in narrow domains. We propose the use of paraphrases for searching TMs. By using paraphrases, we can retrieve sentences that have the same meaning as the input sentences even if they do not match exactly. The paraphrase pairs used in our system are obtained from parallel corpora and are used to retrieve sentences in a statistical framework.

1 Introduction

Translation memories (TMs[1]) are very useful tools for translating texts in narrow domains, where replications of sentences are abundant. In such a case, a machine translation (MT) system can simply search for a match of the input sentence in the TM, and if a match is found, output its corresponding translation.

TMs may also use soft matching, finding a sentence that is similar, but not identical to the input sentence. In this case, the translations of these similar sentences are modified to produce appropriate output translations. A number of MT systems have used TMs in this way (Sumita, 2001; Vogel et al., 2004; Zhechev and van Genabith, 2010).

In this paper, we propose the use of paraphrases for searching TMs. By using paraphrases, we can retrieve sentences that have the same meaning as the input sentences even if the actual words of the sentences do not match exactly.

Note that previous TM systems retrieve similar sentences based on the number of differing words in the sequence. For example, they would prefer "is there a pen?" over "is there a beauty parlor?", when they are given "is there a salon?" as an input. This is because "is there a pen?" is more similar to "is there a salon?" in terms of the number of different words. In contrast, our system would prefer "is there a beauty parlor?" over "is there a pen?" if the system knows "beauty parlor" and "salon" is a paraphrase pair.[2]

The paraphrase pairs used in our system are obtained from parallel corpora and are used to retrieve sentences in a statistical framework as described in Section 3. Related works are presented in Section 2. Experiments and conclusions are described in Sections 4 and 5.

2 Related Work

A number of MT systems have used TMs to retrieve similar sentences and adjusted the translations of these sentences to produce outputs (Sumita, 2001; Vogel et al., 2004; Zhechev and van Genabith, 2010). For example, let us assume we have a translation pair "I like apples" and "watasi-ha ringo-ga suki-desu" in a TM. In this case, the English sentence "I like oranges" could be translated into a Japanese sentence "watasi-ha orenzi-ga suki-desu" by substituting "apples/ringo" with "oranges/orenzi" in the matched sentence.

In contrast, the paraphrase retrieval proposed in this paper will use the translations of the retrieved sentences without modification. For example, if "is there a beauty parlor?" is retrieved when "is there a salon?" is given as an input, we simply output

[1]We use the term *TM* to refer to a set of parallel sentences.

[2]Previous TM systems could use a thesaurus to detect paraphrases. However, large scale thesauruses do not exist for most languages. In this paper, we propose a method that uses only parallel corpora for getting paraphrases.

the translation of "is there a beauty parlor?" without modification.

Retrieving sentences from TMs for MT has been proposed by (Shimohata et al., 2003). They have proposed a method that retrieves sentences sharing the main meaning with input sentences despite lacking some unimportant information. In contrast, we aim to retrieve sentences with exactly the same meaning as input sentences, with no difference in information content.

Our method uses a TM to perform MT. There are works that use MT for TM (He et al., 2010; Simard and Isabelle, 2009). Our method uses paraphrasing for retrieving sentences from a TM. Paraphrasing has also been used in a number of works on statistical MT (SMT) (Callison-Burch et al., 2006; Onishi et al., 2010).

3 Paraphrases for Retrieving Sentences from TMs

3.1 Definition of TM

We first define a TM. A TM, T, is defined as:

$$T = \{\langle f_i, e_{i1}, \ldots, e_{ij}, \ldots, e_{iN_i}\rangle | 1 \leq i \leq N\}$$

where f_i is the i-th source language sentence, e_{ij} is the j-th translation of f_i, N_i is the number of translations of f_i, and N is the number of unique source sentences in T. We use T_F to denote the set of source language sentences in T, i.e., $T_F = \{f_i | 1 \leq i \leq N\}$.

Given an input sentence f, we retrieve the f_i from T_F that receives the highest score according to a scoring function. Then, we use one of e_{ij} as the translation of f.[3]

3.2 Statistical paraphrase retrieval

The statistical paraphrase retrieval model proposed in this paper is defined as follows.

Let f be an input sentence and $Para(f)$ be the set of possible paraphrases of f. We retrieve $\hat{f}$, such that,

$$\begin{aligned} \hat{f} &= \arg \max_{f' \in T_F \cap Para(f)} P(f'|f) \\ &= \arg \max_{f' \in T_F \cap Para(f)} P(f|f')P(f') \quad (1) \end{aligned}$$

Note that we can fail to retrieve $\hat{f}$ if $T_F \cap Para(f)$ is empty, i.e., no paraphrase is found in T_F.

In Equation (1), $P(f')$ is the language model probability of f' and $P(f|f')$ is the paraphrase probability of f given f'. We use an n-gram language model to calculate $P(f')$ and use the model described in the next section to calculate $P(f|f')$.

The statistical paraphrase retrieval model in Equation (1) is almost the same as the statistical paraphrase generation model proposed by Quirk et al. (2004).[4] The important difference is that we retrieve sentences from $T_F \cap Para(f)$, while they search sentences in $Para(f)$. This is because our aim is to retrieve sentences from T_F, not simply find a paraphrase for the input sentence.

There are also two minor differences. One difference is the calculation of $P(f|f')$. $P(f|f')$ can be represented as

$$P(f|f') = \prod_{\langle f_p, f'_p \rangle} P(f_p|f'_p)$$

where f_p and f'_p is one of the set of paraphrase pairs of f and f', respectively, and $P(f_p|f'_p)$ is the paraphrase probability of f_p given f'_p.

Quirk et al. calculated $P(f_p|f'_p)$ from monolingual parallel corpora. In contrast, we calculate $P(f_p|f'_p)$ from bilingual parallel corpora (Bannard and Callison-Burch, 2005).

Another difference lies in the implementation. They used an in-house decoder which was very much like a phrase-based SMT monotone decoder. We use weighted finite state transducers (WFSTs) implemented with open-source software tools.

3.3 Acquiring the paraphrase list

We acquire a paraphrase list using Bannard and Callison-Burch (2005)'s method. Their idea is, if two different phrases f_{p1}, f_{p2} in one language are aligned to the same phrase e_p 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 tech-

[3] How to select one of e_{ij} is not discussed in this paper because we focus on the retrieval part of TMs.

[4] A statistical model for paraphrase detection has also been proposed (Das and Smith, 2009). Their system detects whether two input sentences are paraphrases of one another. However, it does not use paraphrases for searching TMs.

niques. (We used the Moses toolkit (Koehn et al., 2007).)

(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 $P(f_{p2}|f_{p1})$ that f_{p2} is a paraphrase of f_{p1}.

$$P(f_{p2}|f_{p1}) = \sum_{e_p} P(f_{p2}|e_p)P(e_p|f_{p1})$$

where $P(f_{p2}|e_p)$ and $P(e_p|f_{p1})$ are phrase translation probabilities.

(4) **Acquire a paraphrase pair.** Acquire (f_{p1}, f_{p2}) as a paraphrase pair if $P(f_{p2}|f_{p1}) > P(f_{p1}|f_{p1})$. The purpose of this threshold is to keep highly-accurate paraphrase pairs.

3.4 Implementation using WFSTs

We use WFSTs to retrieve sentences in a TM. Given an input sentence f, the best sentence $\hat{f}$ in Equation (1) is represented in Equation (2) using finite-state transducer operations.[5]

$$\textit{BestPath}(\textit{InputFST} \circ \textit{ParaFST} \circ \textit{LMFST} \circ \textit{TMFST}) \quad (2)$$

where *BestPath* is the function that finds the minimum cost path in the WFST, "$\circ$" represents the composite operation of two WFSTs, and *InputFST*, *ParaFST*, *LMFST*, and *TMFST* are the WFSTs for the input sentence, paraphrase list obtained in Section 3.3, n-gram language model, and TM. We describe each WFST below.

***InputFST*:** The *InputFST* of an input sentence accepts and outputs the input sentence without modification and cost.

***ParaFST*:** The *ParaFST* of the paraphrase list acquired in Section 3.3 accepts the input sentence and outputs its paraphrases. *ParaFST* consists of all paraphrase pairs in the paraphrase list and all words in the source language vocabulary. The cost of paraphrasing a phrase f_{p1} into f_{p2} is $-\log_{10} P(f_{p1}|f_{p2})$. The cost of paraphrasing a word into the same word is 0.

***LMFST*:** The *LMFST* of the language model, which is made from T_F, is made by using the Kyoto Language Modeling toolkit (Kylm).[6] We build a 5-gram language model with modified Kneser-Ney smoothing. The cost of each n-gram is the negative base 10 logarithm of its probability.

***TMFST*:** The *TMFST* of the TM is used to retrieve the index of the input sentence, i.e., when f_i is the input, *TMFST* outputs i with cost 0. Using this i, we retrieve f_i. Note that when an input sentence is not found in the TM, the search fails.

Examples of WFSTs are shown in Figure 1. In the Figure, the *ParaFST* replaces "fine" with "very good" at the cost of 2. It also outputs "it" and "is" without modification and cost. "⟨eps⟩" indicates the empty string. The *TMFST* consists of "it is good" and "it is bad" whose ids are 1 and 2, respectively. "⟨/s⟩" is the end of sentence marker.

InputFST, *ParaFST*, *LMFST*, and *TMFST* are compiled, determinized and minimized using the OpenFST library.[7] Then, Equation 2 is solved by using the Kyoto FST Decoder (Kyfd), a general purpose beam-search decoder for WFSTs.[8]

4 Experiments

We used texts in the E-commerce domain to examine the performance of our method.

4.1 Basic statistics

The texts were on Japanese products in a health-care domain. From that domain, we obtained 462,460 Japanese sentences. These sentences contain descriptions of products as well as headings such as "Product Name", "Usage", or "Country Of Origin" (in Japanese). The number of unique Japanese sentences in these 462,460 sentences was 132,210, which was 28.6% of the total number. These unique sentences consist of 41,712 sentences that occurred more than once (average length = 14.4 words[9]) and 90,498 sentences that occurred only once (average length = 17.6 words). Hereafter, we use $S2$ and $S1$

[5] Note that *BestPath*(*InputFST* $\circ$ *ParaFST* $\circ$ *LMFST*) can be used to obtain the maximum probability paraphrased sentence in the statistical paraphrase generation model.

[6] http://www.phontron.com/kylm/

[7] http://www.openfst.org/

[8] http://www.phontron.com/kyfd/

[9] We used ChaSen to segment Japanese texts in words.

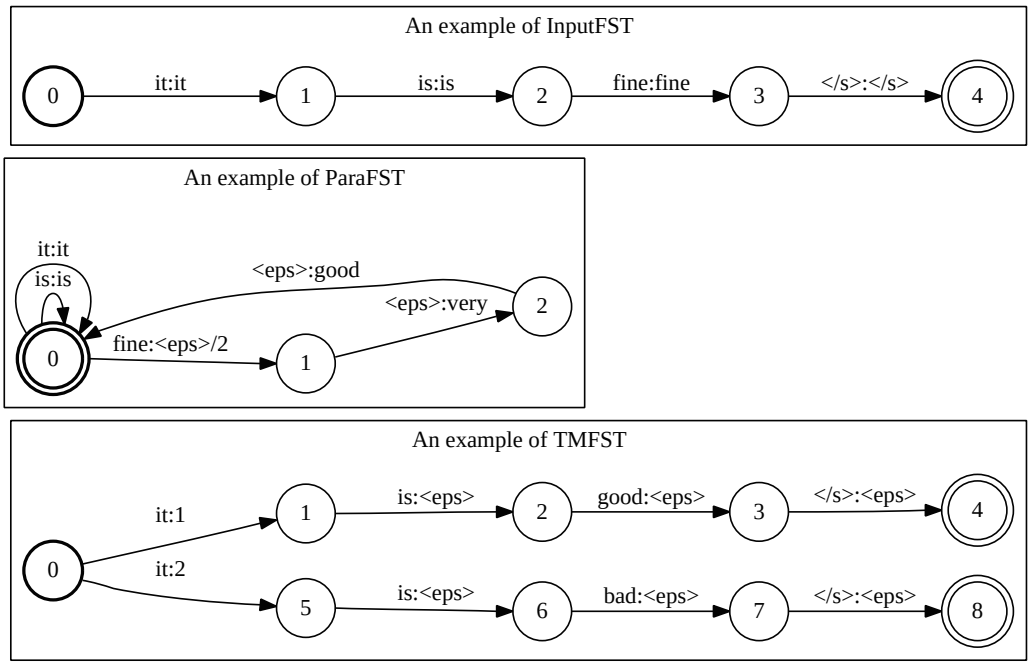

Figure 1: Examples of WFSTs.

to refer to these 41,712 and 90,498 sentences, respectively. These figures show that there are significant replications of sentences in this domain.

We obtained translations of part of $S2$[10] and examined how the sentences in $S1$ could be translated by using our paraphrase retrieval method. This situation represents a common real world case where we translate frequently occurring sentences manually then translate rarely occurring sentences by machine (and post-edit them manually if necessary).

4.2 Distribution of word accuracy

We first examined how the sentences in $S1$ resembled the sentences in $S2$. We used the word accuracy between two sentences to measure the similarity between them. The word accuracy of a sentence s w.r.t a reference sentence r, *WAcc*$(s|r)$ is defined by

$$\textit{WAcc}(s|r) = 1 - \textit{WER}(s|r) = 1 - \frac{I + D + S}{|r|}$$

where $|r|$ is the number of words in r, I is the number of insertions, D is the number of deletions and S is the number of substitutions required in terms of words to transform s to r. *WER*$(s|r)$ is the word error rate, which is often used in measuring the performance of MT.

For each sentence in $S1$, we obtained the highest *WAcc* sentence in $S2$. Thus, we obtained a set of sentence pairs, $\{\langle f, f'\rangle | f \in S1, f' \in S2, f' = \arg\max_{f'' \in S2} \textit{WAcc}(f''|f)\}$. We discarded pairs which shared no words while obtaining these sentence pairs. As a result, we obtained 89,792 sentence pairs. Then, we randomly divided these sentence pairs into two parts, DEV and TEST. DEV and TEST had 44,817 and 44,975 sentences, respectively.

The distribution of *WAcc* in DEV was shown in Table 1. "Freq.", "Percent", and "Cum. Percent" are the numbers of sentence pairs, their percent, and their cumulative percent, respectively. This table shows that there were indeed highly similar sentence pairs (*WAcc* ≥ 0.9) in DEV.

4.3 Comparing paraphrasing with word accuracy

We used the translations of part of $S2$, as described in Section 4.1, to acquire a paraphrase list. The para-

[10] We obtained 41,611 translation pairs. The unique Japanese sentences in these pairs was 30,898, because some Japanese sentences were translated more than once.

WAcc	Freq.	Percent	Cum. Percent
0.9≤	2530	5.65	5.65
0.8≤	3157	7.04	12.69
0.7≤	3258	7.27	19.96
0.6≤	5430	12.12	32.08
0.5≤	8921	19.91	51.98

Table 1: Distribution of word accuracy

phrase list consisted of 266,519 pairs.

We used the sentences of $S1$ in TEST as the inputs to our paraphrase retrieval method. For each input sentence, it retrieved the paraphrased sentence from $S2$ with the highest score. As a result, we succeeded in retrieving paraphrases for 2189 input sentences. This was 4.87% of TEST. Thus the coverage of our method was not very high. We describe our attempts to increase the coverage in the next section.

We then evaluated the precision of the 2189 retrieved sentences. First, we sampled 100 pairs consisting of input and retrieved sentences. Next, we asked a Japanese-English translator to judge if each pair of sentences could be translated into the same English sentence. She judged that 91 of 100 pairs could be translated into the same ones. Thus, the precision was 0.91.

Next, we compared our method with *WAcc*. We obtained the highest *WAcc* sentence for each sentence of $S1$ in TEST. Then, we sorted the input and retrieved sentence pairs in descending order of *WAcc*. Finally, we extracted the top-2189 pairs for comparison. We asked her to judge in the same way. She judged that 76 of 100 pairs could be translated into the same English sentences. The precision was 0.76.

This precision difference was statistically significant at the 1% level, according to the two-sided proportional test. The summary of the comparison is shown in Table 2.

Next, we estimated the recall and F-measure by sampling. The precision of our method was estimated to be 0.91 by sampling as described above. It means that there were about 1992 ($= 2189 \times 0.91$) pairs that had the same meanings in the 2189 retrieved sentences. Similarly, we estimated by sampling that the number of the same meaning pairs in TEST was 6997. From this, we estimated that the recall of our method was 0.285 ($\frac{2189 \times 0.91}{6997}$). Thus, the F-measure was estimated to be 0.434.

On the other hand, the recall and F-measure of *WAcc* were estimated to be 0.238 ($\frac{2189 \times 0.76}{6997}$) and 0.362, respectively, when using the top-2189 pairs. We could also estimate the recall of *WAcc* to be 1 if we regarded all pairs in TEST were retrieved by *WAcc*. In this case, the precision of *WAcc* was 0.156 ($= \frac{6997}{44975}$) and the F-measure was estimated to be 0.270. Thus, the precision, recall and F-measure of our method were superior to those of *WAcc* in both cases.

	Our method	*WAcc*
Precision	0.91	0.76
Recall	0.285	0.238
F-measure	0.434	0.362

Table 2: Comparison of precision/recall/F-measure

We also examined the pairs that were judged incorrect. We found that the differences of meanings were not critical when using our method. For example, some unimportant words were omitted in retrieved sentences. On the other hand, when using *WAcc*, the differences of meanings were often critical. For example, content words were frequently substituted with other content words.

The nine paraphrase pairs that were judged incorrect in our method were listed below. The differences are indicated in *italic*. Note that these paraphrases were literal translations of the Japanese paraphrases. As shown in these paraphrases, the differences of meanings were often not critical.

- Please take around *2-3* capsules per day with cold or hot water. // Please take around *3* capsules per day with cold or hot water.
- People with food allergies, please check the ingredients *before using*. // People with food allergies, please check the ingredients.
- Please close the cap tightly *after opening*. // Please close the cap tightly.
- Please avoid *storing in places with* direct sunlight *or* high temperature and high humidity. // Please avoid direct sunlight, high temperature and high humidity.

- Please stop taking it if *you believe* it is unsuitable for your constitution. // Please stop taking it if it is unsuitable for your constitution.
- Please *do not eat this* in such cases. // Please *abstain* in such cases.
- *moisture* // *water*
- *Fluid* type. // *Liquid* type.
- Please use *immediately* after opening. // Please use *quickly* after opening.

4.4 Coverage

The previous section showed that the coverage of our method was not very high. A simple method for increasing the coverage is using a larger amount of paraphrase pairs. Thus, we used all paraphrase pairs acquired by using the method described in 3.3 without discarding the paraphrase candidates even when they did not satisfy $P(f_{p2}|f_{p1}) > P(f_{p1}|f_{p1})$.

We applied this paraphrase list to DEV. As a result, 5500 input sentences succeeded in retrieving paraphrases. This was 12.27% of DEV. We also applied the original our method to DEV. As a result, 2094 input sentences succeeded in retrieving paraphrases. This was 4.67% of DEV. Thus, the coverage increased from 4.67% to 12.27% when we use all paraphrase candidates.

Next, we estimated the precisions of these methods using the same procedure as in the previous section. We found that the precision of the method that used all paraphrases was 63% and that of our original method was 89%. Thus, the precision decreased from 89% to 63%.

Consequently, we decided to use the paraphrase candidates that satisfied $P(f_{p2}|f_{p1}) > P(f_{p1}|f_{p1})$.

Our future work is extending the coverage without decreasing the precision. It would be interesting to use monolingually-derived paraphrases to improve the coverage (Marton et al., 2009). We may also be able to use a thesaurus to increase the coverage. However, large scale thesauruses do not exist for most languages. Thus, using a thesaurus for increasing the coverage is only applicable to a few languages.

5 Conclusion

We have proposed the use of paraphrases for searching TMs based on a statistical framework. Although the coverage of our method has room for improvement, the precision is high and the meaning differences caused by the method are not critical. Consequently, the method are useful for translating texts on narrow domains, where similar sentences are abundant.

References

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *ACL*, pages 597–604.

Chris Callison-Burch, Philipp Koehn, and Miles Osborne. 2006. Improved statistical machine translation using paraphrases. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 17–24.

Dipanjan Das and Noah A. Smith. 2009. Paraphrase identification as probabilistic quasi-synchronous recognition. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 468–476.

Yifan He, Yanjun Ma, Andy Way, and Josef van Genabith. 2010. Integrating n-best SMT outputs into a tm system. In *Coling 2010: Posters*, pages 374–382.

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 Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180.

Yuval Marton, Chris Callison-Burch, and Philip Resnik. 2009. Improved statistical machine translation using monolingually-derived paraphrases. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Takashi Onishi, Masao Utiyama, and Eiichiro Sumita. 2010. Paraphrase lattice for statistical machine trans-

lation. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 1–5.

Chris Quirk, Chris Brockett, and William Dolan. 2004. Monolingual machine translation for paraphrase generation. In *Proceedings of EMNLP 2004*, pages 142–149.

Mitsuo Shimohata, Eiichiro Sumita, and Yuji Matsumoto. 2003. Retrieving meaning-equivalent sentences for example-based rough translation. In *Proceedings of the HLT-NAACL 2003 Workshop on Building and using parallel texts: data driven machine translation and beyond*.

Michel Simard and Pierre Isabelle. 2009. Phrase-based machine translation in a computer-assisted translation environment. In *MT Summit XII*.

Eiichiro Sumita. 2001. Example-based machine translation using DP-matching between work sequences. In *Proceedings of the ACL 2001 Workshop on Data-Driven Methods in Machine Translation*.

Stephan Vogel, Sanjika Hewavitharana, Muntsin Kolss, and Alex Waibel. 2004. The ISL statistical translation system for spoken language translation. In *International Workshop on Spoken Language Translation (IWSLT)*.

Ventsislav Zhechev and Josef van Genabith. 2010. Seeding statistical machine translation with translation memory output through tree-based structural alignment. In *Proceedings of the 4th Workshop on Syntax and Structure in Statistical Translation*, pages 43–51.