Tree as a Pivot: Syntactic Matching Methods in Pivot Translation

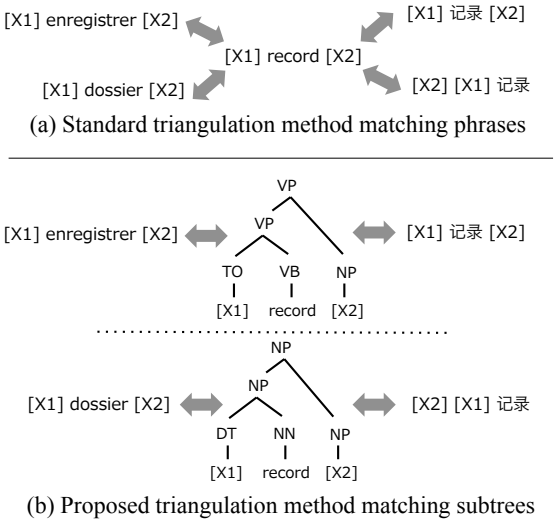
Akiva Miura[†], **Graham Neubig**[‡,†], **Katsuhito Sudoh**[†], **Satoshi Nakamura**[†]
[†] Nara Institute of Science and Technology, Japan
[‡] Carnegie Mellon University, USA
miura.akiba.lr9@is.naist.jp gneubig@cs.cmu.edu
sudoh@is.naist.jp s-nakamura@is.naist.jp

Abstract

Pivot translation is a useful method for translating between languages with little or no parallel data by utilizing parallel data in an intermediate language such as English. A popular approach for pivot translation used in phrase-based or tree-based translation models combines source-pivot and pivot-target translation models into a source-target model, as known as *triangulation*. However, this combination is based on the constituent words' surface forms and often produces incorrect source-target phrase pairs due to semantic ambiguity in the pivot language, and interlingual differences. This degrades translation accuracy. In this paper, we propose a approach for the triangulation using syntactic subtrees in the pivot language to distinguish pivot language words by their syntactic roles to avoid incorrect phrase combinations. Experimental results on the United Nations Parallel Corpus show the proposed method gains in all tested combinations of language, up to 2.3 BLEU points.[1]

1 Introduction

In statistical machine translation (SMT) (Brown et al., 1993), it is known that translation with models trained on larger parallel corpora can achieve greater accuracy (Dyer et al., 2008). Unfortunately, large bilingual corpora are not readily available for many language pairs, particularly those that do not include English. One effective solution to overcome the scarceness of bilingual data is to introduce a pivot language for which parallel data with the source and target languages exists (de Gispert and Mariño, 2006).



(a) Standard triangulation method matching phrases

(b) Proposed triangulation method matching subtrees

Figure 1: Example of disambiguation by parse subtree matching (Fr-En-Zh), [X1] and [X2] are non-terminals for sub-phrases.

Among various methods using pivot languages, one popular and effective method is the triangulation method (Utiyama and Isahara, 2007; Cohn and Lapata, 2007), which first combines source-pivot and pivot-target translation models (TMs) into a source-target model, then translates using this combined model. The procedure of triangulating two TMs into one has been examined for different frameworks of SMT and its effectiveness has been confirmed both in Phrase-Based SMT (PBMT) (Koehn et al., 2003; Utiyama and Isahara, 2007) and in Hierarchical Phrase-Based SMT (Hiero) (Chiang, 2007; Miura et al., 2015). However, word sense ambiguity and interlingual differences of word usage cause difficulty in accurately learning correspondences between source and target phrases, and thus the accuracy obtained by triangulated models lags behind that of models

[1] Code to replicate the experiments can be found at https://github.com/akivajp/wmt2017

trained on direct parallel corpora.

In the triangulation method, source-pivot and pivot-target phrase pairs are connected as a source-target phrase pair when a common pivot-side phrase exists. In Figure 1 (a), we show an example of standard triangulation on Hiero TMs that combines hierarchical rules of phrase pairs by matching pivot phrases with equivalent surface forms. This example also demonstrates problems of ambiguity: the English word "record" can correspond to several different parts-of-speech according to the context. More broadly, phrases including this word also have different possible grammatical structures, but it is impossible to uniquely identify this structure unless information about the surrounding context is given.

This varying syntactic structure will affect translation. For example, the French verb "enregistrer" corresponds to the English verb "record", but the French noun "dossier" also corresponds to "record" — as a noun. As a more extreme example, Chinese is a languages that does not have inflections according to the part-of-speech of the word. As a result, even in the contexts where "record" is used with different parts-of-speech, the Chinese word "记录" will be used, although the word order will change. These facts might result in an incorrect connection of "[X1] enregistrer [X2]" and "[X2] [X1] 记录" even though proper correspondence of "[X1] enregistrer [X2]" and "[X1] dossier [X2]" would be "[X1] 记录 [X2]" and "[X2] [X1] 记录". Hence a superficial phrase matching method based solely on the surface form of the pivot will often combine incorrect phrase pairs, causing translation errors if their translation scores are estimated to be higher than the proper correspondences.

Given this background, we hypothesize that disambiguation of these cases would be easier if the necessary syntactic information such as phrase structures are considered during pivoting. To incorporate this intuition into our models, we propose a method that considers syntactic information of the pivot phrase, as shown in Figure 1 (b). In this way, the model will distinguish translation rules extracted in contexts in which the English symbol string "[X1] record [X2]" behaves as a verbal phrase, from contexts in which the same string acts as nominal phrase.

Specifically, we propose a method based on Synchronous Context-Free Grammars (SCFGs) (Aho and Ullman, 1969; Chiang, 2007), which are widely used in tree-based machine translation frameworks (§2). After describing the baseline triangulation method (§3), which uses only the surface forms for performing triangulation, we propose two methods for triangulation based on syntactic matching (§4). The first places a hard restriction on exact matching of parse trees (§4.1) included in translation rules, while the second places a softer restriction allowing partial matches (§4.2). To investigate the effect of our proposed method on pivot translation quality, we perform experiments of pivot translation on the United Nations Parallel Corpus (Ziemski et al., 2016), which shows that our method indeed provide significant gains in accuracy (of up to 2.3 BLEU points), in almost all combinations of 5 languages with English as a pivot language (§5). In addition, as an auxiliary result, we compare pivot translation using the proposed method with zero-shot neural machine translation, and find that triangulation of symbolic translation models still significantly outperforms neural MT in the zero-resource scenario.

2 Translation Framework

2.1 Synchronous Context-Free Grammars

In this section, first we cover SCFGs, which are widely used in machine translation, particularly hierarchical phrase-based translation (Hiero) (Chiang, 2007). In SCFGs, the elementary structures used in translation are synchronous rewrite rules with aligned pairs of source and target symbols on the right-hand side:

$$X \rightarrow \langle \bar{s},\ \bar{t} \rangle \tag{1}$$

where X is the head symbol of the rewrite rule, and $\bar{s}$ and $\bar{t}$ are both strings of terminals and non-terminals on the source and target side respectively. Each string in the right side pair has the same number of indexed non-terminals, and identically indexed non-terminals correspond to each-other. For example, a synchronous rule could take the form of:

$$X \rightarrow \langle X_0 \text{ of } X_1,\ X_1 \text{ 的 } X_0 \rangle . \tag{2}$$

Synchronous rules can be extracted based on parallel sentences and automatically obtained word alignments. Each extracted rule is scored with phrase translation probabilities in both directions $\phi(\bar{s}|\bar{t})$ and $\phi(\bar{t}|\bar{s})$, lexical translation probabilities in both directions $\phi_{lex}(\bar{s}|\bar{t})$ and $\phi_{lex}(\bar{t}|\bar{s})$,

a word penalty counting the terminals in $\bar{t}$, and a constant phrase penalty of 1.

At translation time, the decoder searches for the target sentence that maximizes the derivation probability, which is defined as the sum of the scores of the rules used in the derivation, and the log of the language model (LM) probability over the target strings. When not considering an LM, it is possible to efficiently find the best translation for an input sentence using the CKY+ algorithm (Chappelier et al., 1998). When using an LM, the expanded search space is further reduced based on a limit on expanded edges, or total states per span, through a procedure such as cube pruning (Chiang, 2007).

2.2 Hierarchical Rules

In this section, we specifically cover the rules used in Hiero. Hierarchical rules are composed of initial head symbol S, and synchronous rules containing terminals and single kind of non-terminals X.[2] Hierarchical rules are extracted using the same phrase extraction procedure used in phrase-based translation (Koehn et al., 2003) based on word alignments, followed by a step that performs recursive extraction of hierarchical phrases (Chiang, 2007).

For example, hierarchical rules could take the form of:

$$X \rightarrow \langle \text{Officers, 主席团 成員} \rangle \quad (3)$$
$$X \rightarrow \langle \text{the Committee, 委员会} \rangle \quad (4)$$
$$X \rightarrow \langle X_0 \text{ of } X_1,\ X_1 \text{ 的 } X_0 \rangle . \quad (5)$$

From these rules, we can translate the input sentence by derivation:

$$\begin{aligned} S &\rightarrow \langle X_0,\ X_0 \rangle \\ &\Rightarrow \langle X_1 \text{ of } X_2,\ X_2 \text{ 的 } X_1 \rangle \\ &\Rightarrow \langle \text{Officers of } X_2,\ X_2 \text{ 主席团 成員} \rangle \\ &\Rightarrow \langle \text{Officers of the Committee,} \\ &\qquad \text{委员会 的 主席团 成員} \rangle \end{aligned}$$

The advantage of Hiero is that it is able to achieve relatively high word re-ordering accuracy (compared to other symbolic SMT alternatives such as standard phrase-based MT) without language-dependent processing. On the other hand, since it does not use syntactic information and tries to extract all possible combinations of rules, it has the tendency to extract very large translation rule tables and also tends to be less syntactically faithful in its derivations.

[2] It is also standard to include a glue rule $S \rightarrow \langle X_0,\ X_0 \rangle$, $S \rightarrow \langle S_0\ X_1,\ S_0\ X_1 \rangle$, $S \rightarrow \langle S_0\ X_1,\ X_1\ S_0 \rangle$ to fall back on when standard rules cannot result in a proper derivation.

2.3 Explicitly Syntactic Rules

An alternative to Hiero rules is the use of synchronous context-free grammar or synchronous tree-substitution grammar (Graehl and Knight, 2004) rules that explicitly take into account the syntax of the source side (tree-to-string rules), target side (string-to-tree rules), or both (tree-to-tree rules). Taking the example of tree-to-string (T2S) rules, these use parse trees on the source language side, and the head symbols of the synchronous rules are not limited to S or X, but instead use non-terminal symbols corresponding to the phrase structure tags of a given parse tree. For example, T2S rules could take the form of:

$$X_{\text{NP}} \rightarrow \langle \text{(NP (NNS Officers))},\ \text{主席团 成員} \rangle \quad (6)$$
$$X_{\text{NP}} \rightarrow \langle \text{(NP (DT the) (NNP Committee))},\ \text{委员会} \rangle \quad (7)$$
$$X_{\text{PP}} \rightarrow \left\langle \text{(PP (IN of) } X_{\text{NP},0}),\ X_0 \text{ 的} \right\rangle \quad (8)$$
$$X_{\text{NP}} \rightarrow \left\langle \text{(NP } X_{\text{NP},0}\ X_{\text{PP},1}),\ X_1\ X_0 \right\rangle \quad (9)$$

Here, parse subtrees of the source language rules are given in the form of S-expressions.

From these rules, we can translate from the parse tree of the input sentence by derivation:

$$\begin{aligned} X_{\text{ROOT}} &\rightarrow \left\langle X_{\text{NP},0},\ X_0 \right\rangle \\ &\Rightarrow \left\langle \text{(NP } X_{\text{NP},1}\ X_{\text{PP},2}),\ X_2\ X_1 \right\rangle \\ &\Rightarrow \left\langle \text{(NP (NP (NNS Officers) } X_{\text{PP},2})),\ X_2 \text{ 主席团 成員} \right\rangle \\ &\overset{*}{\Rightarrow} \left\langle \begin{array}{l} \text{(NP} \\ \quad \text{(NP (NNS Officers))} \\ \quad \text{(PP (IN of)} \\ \quad\quad \text{(NP (DT the)} \\ \quad\quad\quad \text{(NNP Committee))))} \end{array} \text{ , 委员会 的 主席团 成員} \right\rangle \end{aligned}$$

In this way, it is possible in T2S translation to obtain a result conforming to the source language's grammar. This method also has the advantage the number of less-useful synchronous rules extracted by syntax-agnostic methods such as Hiero are reduced, making it possible to learn more compact rule tables and allowing for faster translation.

3 Standard Triangulation Method

In the triangulation method by Cohn and Lapata (2007), we first train source-pivot and pivot-target rule tables as T_{SP} and T_{PT} respectively. Then we search T_{SP} and T_{PT} for source-pivot and pivot-target rules having a common pivot phrase, and

synthesize them into source-target rules to create rule table T_{ST}:

$$\begin{aligned} &X \rightarrow \langle \overline{s},\ \overline{t} \rangle \in T_{ST} \\ &s.t.\ X \rightarrow \langle \overline{s}, \overline{p} \rangle \in T_{SP} \ \wedge\ X \rightarrow \langle \overline{p}, \overline{t} \rangle \in T_{PT}. \end{aligned} \tag{10}$$

For all the combined source-target rules, phrase translation probability $\phi(\cdot)$ and lexical translation probability $\phi_{lex}(\cdot)$ are estimated according to the following equations:

$$\phi\left(\overline{t}|\overline{s}\right) = \sum_{\overline{p} \in T_{SP} \cap T_{PT}} \phi\left(\overline{t}|\overline{p}\right) \phi\left(\overline{p}|\overline{s}\right), \tag{11}$$

$$\phi\left(\overline{s}|\overline{t}\right) = \sum_{\overline{p} \in T_{SP} \cap T_{PT}} \phi\left(\overline{s}|\overline{p}\right) \phi\left(\overline{p}|\overline{t}\right), \tag{12}$$

$$\phi_{lex}\left(\overline{t}|\overline{s}\right) = \sum_{\overline{p} \in T_{SP} \cap T_{PT}} \phi_{lex}\left(\overline{t}|\overline{p}\right) \phi_{lex}\left(\overline{p}|\overline{s}\right), \tag{13}$$

$$\phi_{lex}\left(\overline{s}|\overline{t}\right) = \sum_{\overline{p} \in T_{SP} \cap T_{PT}} \phi_{lex}\left(\overline{s}|\overline{p}\right) \phi_{lex}\left(\overline{p}|\overline{t}\right). \tag{14}$$

The equations (11)-(14) are based on the memoryless channel model, which assumes:

$$\phi\left(\overline{t}|\overline{p},\overline{s}\right) = \phi\left(\overline{t}|\overline{p}\right), \tag{15}$$

$$\phi\left(\overline{s}|\overline{p},\overline{t}\right) = \phi\left(\overline{s}|\overline{p}\right) \tag{16}$$

For example, in equation (15), it is assumed that the translation probability of target phrase given pivot and source phrases is never affected by the source phrase. However, it is easy to come up with examples where this assumption does not hold. Specifically, if there are multiple interpretations of the pivot phrase as shown in the example of Figure 1, source and target phrases that do not correspond to each other semantically might be connected, and over-estimation by summing products of the translation probabilities is likely to cause failed translations.

4 Triangulation with Syntactic Matching

In the previous section, we explained about the standard triangulation method and mentioned that the pivot-side ambiguity causes incorrect estimation of translation probability and the translation accuracy might decrease. To address this problem, it is desirable to be able to distinguish pivot-side phrases that have different syntactic roles or meanings, even if the symbol strings are exactly equivalent. In the following two sections, we describe two methods to distinguish pivot phrases that have syntactically different roles, one based on exact matching of parse trees, and one based on soft matching.

4.1 Exact Matching of Parse Subtrees

In the exact matching method, we first train pivot-source and pivot-target T2S TMs by parsing the pivot side of parallel corpora, and store them into rule tables as T_{PS} and T_{PT} respectively. Synchronous rules of T_{PS} and T_{PT} take the form of $X \rightarrow \langle \hat{p}, \overline{s} \rangle$ and $X \rightarrow \langle \hat{p}, \overline{t} \rangle$ respectively, where $\hat{p}$ is a symbol string that expresses pivot-side parse subtree (S-expression), $\overline{s}$ and $\overline{t}$ express source and target symbol strings. The procedure of synthesizing source-target synchronous rules essentially follows equations (11)-(14), except using T_{PS} instead of T_{SP} (direction of probability features is reversed) and pivot subtree $\hat{p}$ instead of pivot phrase $\overline{p}$. Here $\overline{s}$ and $\overline{t}$ do not have syntactic information, therefore the synthesized synchronous rules should be hierarchical rules explained in §2.2.

The matching condition of this method has harder constraints than matching of superficial symbols in standard triangulation, and has the potential to reduce incorrect connections of phrase pairs, resulting in a more reliable triangulated TM. On the other hand, the number of connected rules decreases as well in this restricted triangulation, and the coverage of the triangulated model might be reduced. Therefore it is important to create TMs that are both reliabile and have high coverage.

4.2 Partial Matching of Parse Subtrees

To prevent the problem of the reduction of coverage in the exact matching method, we also propose a partial matching method that keeps coverage just like standard triangulation by allowing connection of incompletely equivalent pivot subtrees. To estimate translation probabilities in partial matching, we first define *weighted triangulation* generalizing the equations (11)-(14) of standard triangulation with weight function $\psi(\cdot)$:

$$\phi\left(\overline{t}|\overline{s}\right) = \sum_{\hat{p_T}} \sum_{\hat{p_S}} \phi\left(\overline{t}|\hat{p_T}\right) \psi\left(\hat{p_T}|\hat{p_S}\right) \phi\left(\hat{p_S}|\overline{s}\right), \tag{17}$$

$$\phi\left(\overline{s}|\overline{t}\right) = \sum_{\hat{p_S}} \sum_{\hat{p_T}} \phi\left(\overline{s}|\hat{p_S}\right) \psi\left(\hat{p_S}|\hat{p_T}\right) \phi\left(\hat{p_T}|\overline{t}\right), \tag{18}$$

$$\phi_{lex}\left(\overline{t}|\overline{s}\right) = \sum_{\hat{p_T}} \sum_{\hat{p_S}} \phi_{lex}\left(\overline{t}|\hat{p_T}\right) \psi\left(\hat{p_T}|\hat{p_S}\right) \phi_{lex}\left(\hat{p_S}|\overline{s}\right), \tag{19}$$

$$\phi_{lex}\left(\overline{s}|\overline{t}\right) = \sum_{\hat{p_S}} \sum_{\hat{p_T}} \phi_{lex}\left(\overline{s}|\hat{p_S}\right) \psi\left(\hat{p_S}|\hat{p_T}\right) \phi_{lex}\left(\hat{p_T}|\overline{t}\right) \tag{20}$$

where $\hat{p_S} \in T_{SP}$ and $\hat{p_T} \in P_{PT}$ are pivot parse subtrees of source-pivot and pivot-target synchronous rules respectively. By adjusting $\psi(\cdot)$, we can control the magnitude of the penalty for the case of incompletely matched connections. If we

define $\psi(\hat{p_T}|\hat{p_S}) = 1$ when $\hat{p_T}$ is equal to $\hat{p_S}$ and $\psi(\hat{p_T}|\hat{p_S}) = 0$ otherwise, equations (17)-(20) are equivalent with equations (11)-(14).

Better estimating $\psi(\cdot)$ is not trivial, and co-occurrence counts of $\hat{p_S}$ and $\hat{p_T}$ are not available. Therefore we introduce a heuristic estimation method as follows:

$$\psi(\hat{p_T}|\hat{p_S}) = \frac{w(\hat{p_S}, \hat{p_T})}{\sum_{\hat{p} \in T_{PT}} w(\hat{p_S}, \hat{p})} \cdot \max_{\hat{p} \in T_{PT}} w(\hat{p_S}, \hat{p}) \quad (21)$$

$$\psi(\hat{p_S}|\hat{p_T}) = \frac{w(\hat{p_S}, \hat{p_T})}{\sum_{\hat{p} \in T_{SP}} w(\hat{p}, \hat{p_T})} \cdot \max_{\hat{p} \in T_{SP}} w(\hat{p}, \hat{p_T}) \quad (22)$$

$$w(\hat{p_S}, \hat{p_T}) = \begin{cases} 0 & (flat(\hat{p_S}) \neq flat(\hat{p_T})) \\ \exp\left(-d\left(\hat{p_S}, \hat{p_T}\right)\right) & (otherwise) \end{cases} \quad (23)$$

$$d(\hat{p_S}, \hat{p_T}) = TreeEditDistance(\hat{p_S}, \hat{p_T}) \quad (24)$$

where $flat(\hat{p})$ returns the symbol string of $\hat{p}$ keeping non-terminals, and $TreeEditDistance(\hat{p_S}, \hat{p_T})$ is minimum cost of a sequence of operations (contract an edge, uncontract an edge, modify the label of an edge) needed to transform $\hat{p_S}$ into $\hat{p_T}$ (Klein, 1998).

According to equations (21)-(24), we can assure that incomplete match of pivot subtrees leads $d(\cdot) \geq 1$ and penalizes such that $\psi(\cdot) \leq 1/e^d \leq 1/e$, while exact match of subtrees leads to a value of $\psi(\cdot)$ at least $e \approx 2.718$ times larger than when using partially matched subtrees.

5 Experiments

5.1 Experimental Set-Up

To investigate the effect of our proposed approach, we evaluate the translation accuracy through pivot translation experiments on the United Nations Parallel Corpus (UN6Way) (Ziemski et al., 2016). UN6Way is a line-aligned multilingual parallel corpus that includes data in English (En), Arabic (Ar), Spanish (Es), French (Fr), Russian (Ru) and Chinese (Zh), covering different families of languages. It contains more than 11M sentences for each language pair, and is therefore suitable for multilingual translation tasks such as pivot translation. In these experiments, we fixed English as the pivot language considering that it is the language most frequently used as a pivot language. This has the positive side-effect that accurate phrase structure parsers are available in the pivot language, which is good for our proposed method. We perform pivot translation on all the combinations of the other 5 languages, and compared the accuracy of each method. For tokenization, we adopt SentencePiece,[3] an unsupervised text tokenizer and detokenizer, that is although designed mainly for neural MT, we confirmed that it also helps to reduce training time and even improves translation accuracy in our Hiero model as well. We first trained a single shared tokenization model by feeding a total of 10M sentences from the data of all the 6 languages, set the maximum shared vocabulary size to be 16k, and tokenized all available text with the trained model. We used English raw text without tokenization for phrase structure analysis and for training Hiero and T2S TMs on the pivot side. To generate parse trees, we used the Ckylark PCFG-LA parser (Oda et al., 2015), and filtered out lines of length over 60 tokens from all the parallel data to ensure accuracy of parsing and alignment. About 7.6M lines remained. Since Hiero requires a large amount of computational resources for training and decoding, so we decided not to use all available training data but first 1M lines for training each TM. As a decoder, we use Travatar (Neubig, 2013), and train Hiero and T2S TMs with its rule extraction code. We train 5-gram LMs over the target side of the same parallel data used for training TMs using KenLM (Heafield, 2011). For testing and parameter tuning, we used the first 1,000 lines of the 4,000 lines test and dev sets respectively. For the evaluation of translation results, we first detokenize with the SentencePiece model and re-tokenized with the tokenizer of the Moses toolkit (Koehn et al., 2007) for Arabic, Spanish, French and Russian and re-tokenized Chinese text with Kytea tokenizer (Neubig et al., 2011), then evaluated using case-sensitive BLEU-4 (Papineni et al., 2002).

We evaluate 6 translation methods:

Direct:
Translating with a Hiero TM directly trained on the source-target parallel corpus without using pivot language (as an oracle).

Tri. Hiero:
Triangulating source-pivot and pivot-target Hiero TMs into a source-target Hiero TM using the traditional method (baseline, §3).

Tri. TreeExact
Triangulating pivot-source and pivot-target T2S TMs into a source-target Hiero TM using

[3] https://github.com/google/sentencepiece

Source	Target	BLEU Score [%]			
		Direct	Tri. Hiero (baseline)	Tri. TreeExact (proposed 1)	Tri. TreePartial (proposed 2)
Ar	Es	38.49	34.20	‡ 34.97	‡ **35.94**
	Fr	33.34	29.93	‡ 30.68	‡ **30.83**
	Ru	24.63	22.94	‡ 23.94	‡ **24.15**
	Zh	27.27	22.78	‡ **25.17**	‡ 25.07
Es	Ar	27.18	22.97	‡ 24.09	‡ **24.45**
	Fr	43.24	38.74	‡ 39.62	‡ **40.12**
	Ru	28.83	26.35	‡ 27.25	‡ **27.41**
	Zh	27.08	24.54	25.00	† **25.16**
Fr	Ar	25.10	21.65	21.40	† **22.13**
	Es	45.20	40.16	‡ 41.03	‡ **41.99**
	Ru	27.42	24.71	† 25.24	‡ **25.64**
	Zh	25.84	23.16	**23.56**	23.53
Ru	Ar	22.53	19.82	19.86	**20.35**
	Es	37.60	34.56	34.96	‡ **35.62**
	Fr	34.05	30.75	† 31.43	‡ **31.67**
	Zh	28.03	24.88	25.07	**25.12**
Zh	Ar	20.09	16.66	17.01	‡ **17.73**
	Es	30.66	27.84	27.99	**28.05**
	Fr	25.97	23.82	24.34	† **24.35**
	Ru	21.16	18.63	‡ 19.58	‡ **19.59**

Table 1: Comparison of each triangulation methods. Bold face indicates the highest BLEU score in pivot translation, and daggers indicate statistically significant gains over Tri. Hiero († : $p < 0.05$, ‡ : $p < 0.01$).

the proposed exact matching of pivot subtrees (proposed 1, §4.1).

Tri. TreePartial
Triangulating pivot-source and pivot-target T2S TMs into a source-target Hiero TM using the proposed partial matching of pivot subtrees (proposed 2, §4.2).

5.2 Experimental Results

The result of experiments using all combinations of pivot translation tasks for 5 languages via English is shown in Table 1. From the results, we can see that the proposed partial matching method of pivot subtrees in triangulation outperforms the standard triangulation method for all language pairs and achieves higher or almost equal scores than proposed exact matching method. The exact matching method also outperforms the standard triangulation method in the majority of the language pairs, but has a lesser improvement than partial matching method. In Table 2 we show the comparison of coverage of each proposed triangulated method. From this table, we can see that the exact matching method reduces several percent in number of unique phrases while the partial matching method keeps the same coverage with surface-form matching. We can consider that it is one of the reasons of the difference in improvement stability between the partial and exact matching methods.

We show an example of a translated sentences for which pivot-side ambiguity is resolved in the the syntactic matching methods:

Source Sentence in French:
La Suisse encourage **tous les États parties** à soutenir le travail conceptuel que fait actuellement le Secrétariat .

Corresponding Sentence in English:
Switzerland encourages all parties to support the current conceptual work of the secretariat.

Reference in Spanish:
Suiza alienta a **todos los Estados partes** a que apoyen la actual labor *conceptual* de la Secretaría .

Source	Target	Number of source-side unique phrases/words	
		Tri. TreeExact	Tri. TreePartial
Ar	Es	2.580M / 5,072	2.646M / 5,077
	Fr	2.589M / 5,067	2.658M / 5,071
	Ru	2.347M / 5,085	2.406M / 5,088
	Zh	2.324M / 5,034	2.386M / 5,040
Es	Ar	1.942M / 5,182	2.013M / 5,188
	Fr	2.062M / 5,205	2.129M / 5,210
	Ru	1,978M / 5,191	2.037M / 5,197
	Zh	1,920M / 5,175	1.986M / 5,180
Fr	Ar	2.176M / 5,310	2.233M / 5,316
	Es	2.302M / 5,337	2.366M / 5,342
	Ru	2.203M / 5.311	2.266M / 5,318
	Zh	2.162M / 5.313	2.215M / 5,321
Ru	Ar	2.437M / 5,637	2.505M / 5,644
	Es	2.478M / 5.677	2.536M / 5,682
	Fr	2.479M / 5,661	2.531M / 5,665
	Zh	2.466M / 5,682	2.515M / 5,688
Zh	Ar	1.480M / 9,428	1.556M / 9,474
	Es	1.504M / 9,523	1.570M / 9,555
	Fr	1.499M / 9,490	1,568M / 9,520
	Ru	1.518M / 9,457	1.593M / 9,487

Table 2: Comparison of rule table coverage in proposed triangulation methods.

Direct:

Suiza alienta a todos los Estados partes a que apoyen el trabajo conceptual que se examinan en la Secretaría . (BLEU+1: 55.99)

Tri. Hiero:

Suiza *conceptuales* para apoyar la labor que en estos momentos la Secretaría alienta a todos los Estados Partes . (BLEU+1: 29.74)

Tri. TreeExact:

Suiza alienta a **todos los Estados Partes** a apoyar la labor conceptual que actualmente la Secretaría . (BLEU+1: 43.08)

Tri. TreePartial:

Suiza alienta a **todos los Estados Partes** a apoyar la labor conceptual que actualmente la Secretaría . (BLEU+1: 43.08)

The results of Tri.TreeExact and Tri.TreePartial are same in this example. We find that the derivation in Tri.Hiero uses rule $X \rightarrow \langle X_0$ _parties X_1, X_1 X_0 _Partes$\rangle$[4] causing incorrect re-ordering of phrases followed by steps of incorrect word selection.[5] On the other hand, derivation in Tri.TreeExact and Tri.TreePartial uses rule $X \rightarrow \langle$_tous _les X_0 _parties, _todos X_0 _Partes$\rangle$[6] synthesized from T2S rules with common pivot subtree (NP (DT all) (NP' X_{NNP} (NNS parties)). We can confirm that the derivation improves word-selection and word-reordering by using this rule.

5.3 Comparison with Neural MT:

Recent results (Firat et al., 2016; Johnson et al., 2016) have found that neural machine translation systems can gain the ability to perform translation with zero parallel resources by training on multiple sets of bilingual data. However, previous work has not examined the competitiveness of these methods with pivot-based symbolic SMT frameworks such as PBMT or Hiero. In this section, we compare a zero-shot NMT model (detailed parameters in Table 3) with our pivot-based Hiero models.

[4]The words emphasized with underline and wavy-underline in the example correspond to X_0 and X_1 respectively.

[5]For example, the word "conceptuales" with italic face in Tri.Hiero takes the wrong form and position.

[6]The words emphasized in bold face in the example correspond to the rule.

vocabulary size:	16k (shared)
source embedding size:	512
target embedding size:	512
output embedding size:	512
encoder hidden size:	512
decoder hidden size:	512
LSTM layers:	1
attention type:	MLP
attention hidden size:	512
optimizer type:	Adam
loss integration type:	mean
batch size:	2048
max iteration:	200k
dropout rate:	0.3
decoder type:	Luong+ 2015

Table 3: Main parameters of NMT training

Direct NMT is trained with the same data of Direct Hiero, Cascade NMT translates by bridging source-pivot and pivot-target NMT models, and Zero-Shot NMT is trained on single shared model with pvt ↔ {src,target} parallel data according to Johnson et al. (2016). To train and evaluate NMT models, we adopt NMTKit.[7] From the results we see the tendency of NMT that directly trained model achieves high translation accuracy even for translation between languages of different families, on the other hand, the accuracy is drastically reduced in the situation when there is no source-target parallel corpora for training. Cascade is one immediate method connecting two TMs, and NMT cascade translation shows the medium performance in this experiment. In our setting, while bilingually trained NMT systems were competitive or outperformed Hiero-based models, zero-shot translation is uniformly weaker. This may be because we used only 1 LSTM layer for encoder/decoder, or because the amount of parallel corpora or language pairs were not sufficient. Thus, we can posit that while zero-shot translation has demonstrated reasonable results in some settings, successful zero-shot translation systems are far from trivial to build, and pivot-based symbolic MT systems such as PBMT or Hiero may still be a competitive alternative.

[7] https://github.com/odashi/nmtkit

6 Conclusion

In this paper, we have proposed a method of pivot translation using triangulation with exact or partial matching method of pivot-side parse subtrees. In experiments, we found that these triangulated models are effective in particular when allowing partial matching. To estimate translation probabilities, we introduced heuristic that has no guarantee to be optimal. Therefore in the future, we plan to explore more refined estimation methods that utilize machine learning.

Acknowledgments

Part of this work was supported by JSPS KAKENHI Grant Numbers JP16H05873 and JP17H06101.

References

Alfred V. Aho and Jeffrey D. Ullman. 1969. Syntax directed translations and the pushdown assembler. *Journal of Computer and System Sciences* 3(1):37–56.

Peter F. Brown, Vincent J.Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics* 19:263–312.

Jean-Cédric Chappelier, Martin Rajman, et al. 1998. A Generalized CYK Algorithm for Parsing Stochastic CFG. In *Proc. TAPD*. Citeseer, volume 98, pages 133–137.

David Chiang. 2007. Hierarchical Phrase-Based Translation. *Computational Linguistics* 33(2):201–228.

Trevor Cohn and Mirella Lapata. 2007. Machine Translation by Triangulation: Making Effective Use of Multi-Parallel Corpora. In *Proc. ACL*. pages 728–735.

Adrià de Gispert and José B. Mariño. 2006. Catalan-English Statistical Machine Translation without Parallel Corpus: Bridging through Spanish. In *Proc. of LREC 5th Workshop on Strategies for developing machine translation for minority languages*. pages 65–68.

Chris Dyer, Aaron Cordova, Alex Mont, and Jimmy Lin. 2008. Fast, Easy, and Cheap: Construction of Statistical Machine Translation Models with MapReduce. In *Proc. WMT*. pages 199–207.

Orhan Firat, Baskaran Sankaran, Yaser Al-Onaizan, Fatos T. Yarman Vural, and Kyunghyun Cho. 2016. Zero-Resource Translation with Multi-Lingual Neural Machine Translation. In *Proc. EMNLP*. pages 268–277.

Source	Target	BLEU Score [%]				
		Direct Hiero	Direct NMT	Tri. TreePartial	Cascade NMT	Zero-Shot NMT
Ar	Es	38.49	38.25	35.94	31.62	8.18
	Fr	33.34	33.16	30.83	26.91	8.57
	Ru	24.63	27.00	24.15	21.67	5.79
	Zh	27.27	30.04	25.07	23.70	5.04
Es	Ar	27.18	26.02	24.45	21.21	5.22
	Fr	43.24	41.83	40.12	31.84	15.04
	Ru	28.83	30.65	27.41	23.60	7.57
	Zh	27.08	32.36	25.16	26.03	8.62
Fr	Ar	25.10	23.28	22.13	18.66	8.08
	Es	45.20	44.49	41.99	32.93	14.37
	Ru	27.42	28.29	25.64	20.87	8.77
	Zh	25.84	29.10	23.53	23.14	11.95
Ru	Ar	22.53	23.19	20.35	19.71	3.18
	Es	37.60	38.67	35.62	31.25	10.42
	Fr	34.05	33.26	31.67	27.34	9.76
	Zh	28.03	31.39	25.12	24.25	9.46
Zh	Ar	20.09	20.17	17.73	16.89	10.38
	Es	30.66	32.69	28.05	26.01	6.13
	Fr	25.97	27.68	24.35	23.35	7.12
	Ru	21.16	23.17	19.59	18.40	3.21

Table 4: Comparison of SMT and NMT in multilingual translation tasks.

Jonathan Graehl and Kevin Knight. 2004. Training Tree Transducers. In *Proc. NAACL*. pages 105–112.

Kenneth Heafield. 2011. KenLM: Faster and Smaller Language Model Queries. In *Proc, WMT*. pages 187–197.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *CoRR* abs/1611.04558. http://arxiv.org/abs/1611.04558.

Philip N. Klein. 1998. Computing the Edit-Distance Between Unrooted Ordered Trees. In *Proc. of European Symposium on Algorithms*. pages 91–102.

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 *Proc. ACL*. pages 177–180.

Phillip Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. In *Proc. NAACL*. pages 48–54.

Akiva Miura, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. 2015. Improving Pivot Translation by Remembering the Pivot. In *Proc. ACL*. pages 573–577.

Graham Neubig. 2013. Travatar: A Forest-to-String Machine Translation Engine based on Tree Transducers. In *Proc. ACL Demo Track*. pages 91–96.

Graham Neubig, Yosuke Nakata, and Shinsuke Mori. 2011. Pointwise Prediction for Robust, Adaptable Japanese Morphological Analysis. In *Proc. ACL*. pages 529–533.

Yusuke Oda, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. 2015. Ckylark: A More Robust PCFG-LA Parser. In *Proc. NAACL*. pages 41–45.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proc. ACL*. pages 311–318.

Masao Utiyama and Hitoshi Isahara. 2007. A Comparison of Pivot Methods for Phrase-Based Statistical Machine Translation. In *Proc. NAACL*. pages 484–491.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The United Nations Parallel Corpus v1.0. In *Proc. LREC*. pages 3530–3534.