

# Cross-language Projection of Dependency Trees with Constrained Partial Parsing for Tree-to-Tree Machine Translation

Yu Shen<sup>1</sup>, Chenhui Chu<sup>2\*</sup>, Fabien Cromieres<sup>2</sup> and Sadao Kurohashi<sup>1</sup>

<sup>1</sup>Graduate School of Informatics, Kyoto University

<sup>2</sup>Japan Science and Technology Agency

{shen-yu, kuro}@nlp.ist.i.kyoto-u.ac.jp {chu, fabien}@pa.jst.jp

## Abstract

Tree-to-tree machine translation (MT) that utilizes syntactic parse trees on both source and target sides suffers from the non-isomorphism of the parse trees due to parsing errors and the difference of annotation criterion between the two languages. In this paper, we present a method that projects dependency parse trees from the language side that has a high quality parser, to the side that has a low quality parser, to improve the isomorphism of the parse trees. We first project a part of the dependencies with high confidence to make a partial parse tree, and then complement the remaining dependencies with partial parsing constrained by the already projected dependencies. MT experiments verify the effectiveness of our proposed method.

## 1 Introduction

According to how syntactic parse trees are used in machine translation (MT), there are 4 types of MT approaches: string-to-string that does not use parse trees (Chiang, 2005; Koehn et al., 2007), string-to-tree that uses parse trees on the target side (Galley et al., 2006; Shen et al., 2008), tree-to-string that uses parse trees on the source side (Quirk et al., 2005; Liu et al., 2006; Mi and Huang, 2008), and tree-to-tree that uses parse trees on both sides (Zhang et al., 2008; Richardson et al., 2015). Intuitively, the tree-to-tree approach seems to be the most appropriate. The reason is that it could preserve the structure information on both sides, which leads to fluent and accurate translations.

In practice, however, good quality parsers on both the source and target sides are difficult to ac-

quire. In many cases, the parsing quality of one side is much higher than that of the other side, because the higher quality side has a well annotated treebank or is linguistically easier to parse. For example, in the case of Japanese-Chinese MT that we study in this paper, the head-final characteristic of Japanese (Isozaki et al., 2010) makes the dependency parsing for Japanese much easier than that of Chinese. Currently, the dependency parsing accuracy of Japanese is over 90% (Kawahara and Kurohashi, 2006), while the Chinese parsing accuracy is less than 80% (Shen et al., 2012). Another problem is the annotation criterion difference of the treebanks in different languages, which are used for training the parsers. For example, the dependency annotations of noun phrases and coordination could be different among different languages. For example, in Japanese, noun phrases and coordination are annotated as modifier-head dependencies (Kawahara and Kurohashi, 2006), while in Chinese they are annotated as sibling dependencies (Shen et al., 2012). These two problems lead to the parse difference between the source and target parse trees, which affects the translation rule extraction in tree-to-tree MT that requires the isomorphism of the parse trees. This extremely limits the translation quality of tree-to-tree MT.

In this paper, we present an approach that projects dependency trees from a high quality (HQ) parser to a low quality (LQ) parser using alignment information. The projection could reduce the parsing errors on the LQ side, and address the annotation criterion difference problem. This can make the LQ trees isomorphic to the HQ trees, which can benefit the translation rule extraction in tree-to-tree MT, and thus improve the MT performance. The idea of cross-language projection of parse trees has been proposed previously, e.g., (Ganchev et al., 2009; Jiang et al., 2010; Goto

\*Corresponding author.

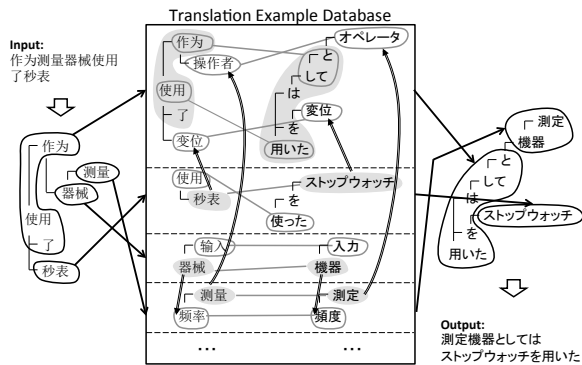


Fig. 1: An example of the KyotoEBMT system on Chinese-to-Japanese translation.

et al., 2015). However, few studies have been conducted in the context of dependency based tree-to-tree MT, which is the setting of this paper.

In addition, we propose a novel constrained partial parsing method to address the word alignment problems such as unaligned words and alignment errors in projection. In detail, we first apply a partial projection step to project a part of the dependencies with high confidence judged by the alignment information and a projectivity criterion. We thus obtain a projected “partial tree.” We then find the missing dependencies from this partial tree by applying a “partial parsing” method: we apply a parser to find the missing dependencies subject to respecting the projected dependencies, so that we obtain a full dependency tree. Initially, the LQ parser is used for the partial parsing process. Once the entire projection process has been finished, we select a part of the projected trees based on the dependency projection ratio of the partial projection step, and re-train a parser for the LQ side. This re-trained parser tends to be more isomorphic to the HQ parser, and thus we again apply it for the partial parsing process.

We conduct experiments with an open source dependency based tree-to-tree MT system KyotoEBMT<sup>1</sup> (Richardson et al., 2015) on the Japanese-Chinese language pair. Because of the improvement of the isomorphism of the source and target parse trees by our proposed method, we achieve significant MT performance improvements on both Japanese-to-Chinese and Chinese-to-Japanese translation directions.

## 2 The Difficulties of Tree-to-Tree MT

### 2.1 Overview of the KyotoEBMT System

This study is conducted on the KyotoEBMT system (Richardson et al., 2015), which is a represen-

<sup>1</sup> <http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?KyotoEBMT>

tative dependency based tree-to-tree MT system. Figure 1 shows an overview of the KyotoEBMT system on Chinese-to-Japanese translation. The translation example database is automatically constructed from a parallel training corpus by means of a discriminative alignment model (Riesa et al., 2011). It contains “examples” that form the hypotheses to be combined during decoding. Note that both source and target sides of all the examples are stored in dependency trees. An input sentence is also parsed and transformed into a dependency tree. For all the subtrees in the input dependency tree, matching hypotheses are searched in the example database. This step is the most time consuming part, and a fast subtree retrieval method (Cromieres and Kurohashi, 2011) is used. There are many available hypotheses for one subtree, and also, there are many possible hypothesis combinations. The best combination is detected by a lattice-based decoder, which optimizes a log-linear model (Cromieres and Kurohashi, 2014). In the example in Figure 1, four hypotheses are used. They are combined and produce an output dependency tree, which is the final translation. For more details of the system, please refer to (Richardson et al., 2015).

### 2.2 The Translation Example Extraction Problem

One advantage of the KyotoEBMT system is that it can handle examples that are discontinuous as a word sequence but continuous structurally, because of the usage of both source and target parse trees. In Figure 2, for example, the translation example of “26-31:類似することを示唆する/4:表明 14:类似 (show the similarity)” and “0-2:このことは 30-35:示唆するものと思われる/0-4:认为这一现象表明 (I think that this phenomenon shows)” can be extracted by the KyotoEBMT system, because they are continuous in the parse trees. However, in phrase based MT (Koehn et al., 2007), both of these two translation examples could not be extracted. The reason for this is that “4:表明 (show)” and “14:类似 (similarity)” are discontinuous in the Chinese sentence; similarly, “0-2:このことは (this phenomenon)” and “30-35:示唆するものと思われる (I think that shows)” are discontinuous in the Japanese sentence.

On the other hand, it also adds the constraint that a translation example has to share the same structure on the parse trees to guarantee the quality

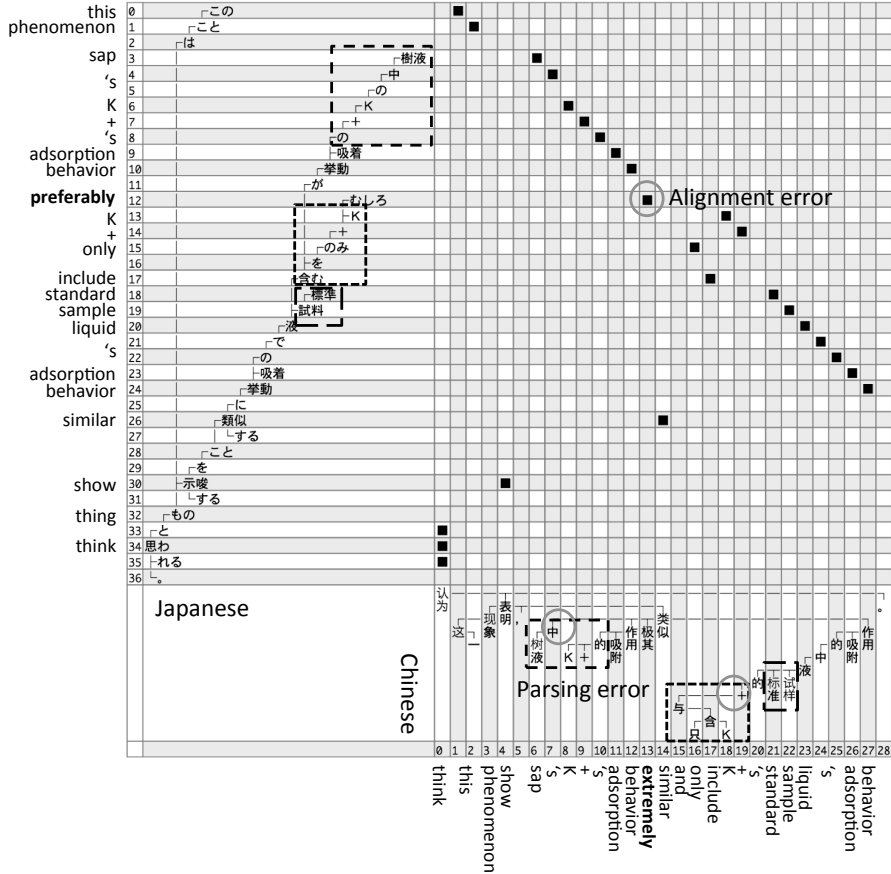


Fig. 2: A motivated example that shows a word aligned Japanese-Chinese parse tree pair, where the solid black boxes show the word alignments.

of the extracted examples. This could be a problem because of two reasons. The main reason is parsing error. In Figure 2, for example, because of the parsing errors in the Chinese parse tree, the translation examples of “3-8:樹液中のK+の/6-10:樹液中K+的 (sap’s K+),” and “13-17:K+のみを含む/16-19:只含K+ (only include K+)” could not be extracted. The other reason is the annotation criterion difference. In Figure 2, for example, the translation example of “18:標準 19:試料/21:标准 22:试样(standard sample)” could not be extracted, though both of the parses are correct. In Japanese this kind of noun phrase structure is annotated as the modifier-head, while in Chinese it is annotated as siblings depending on the last word.

One possible solution to address the above problem is to loosen the constraint for translation example extraction. For example, to extract the “18:標準 19:試料/21:标准 22:试样(standard sample)” example caused by the annotation criterion difference, we might allow the extraction of examples that are modifier-head and sibling subtrees on the source and target sides, respectively. However,

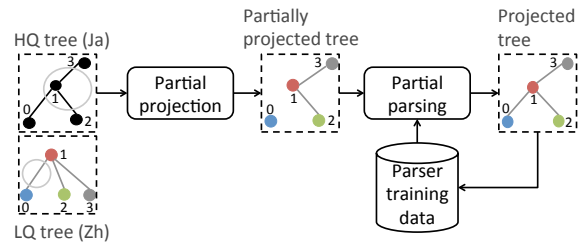


Fig. 3: An overview of our constrained partial parsing based projection method.

firstly, even the loosening in this degree could also lead to other noisy translation examples; secondly, what kind of loosening is required for the parse error case is unclear, because the types of parse errors are diverse. Therefore, instead of loosening the constraint, we choose the cross-lingual projection approach to address the problem.

### 3 Projection of Dependency Trees with Constrained Partial Parsing

Figure 3 is an overview of our proposed constrained partial parsing method. Firstly, we apply a partial projection process to project a part of the

dependencies from the HQ tree using the HQ tree, word alignment information and a projectivity criterion. Note that the word alignment information is omitted in Figure 3 for simplification. In Figure 3, the circled part in the HQ tree is projected. Next, we apply partial parsing to complement the other dependencies in the partially projected tree using the LQ parser. In Figure 3, as the LQ parser could parse the circled part in the original LQ tree correctly, it also complements the dependencies for the partially projected tree correctly. Once we obtained the projected trees, we select a part of the highly confident projected trees as training data to re-train the LQ parser. Finally, we apply the re-trained LQ parser for the partial parsing process, which further improves the quality of projection.

In the remaining of this section, we describe the details of partial projection, partial parsing, and re-training of the LQ parser in Section 3.1, 3.2, and 3.3 respectively.

### 3.1 Partial Projection

#### 3.1.1 Direct Mapping for Dependency Tree Projection

We first present a direct mapping method for dependency tree projection using word alignment, which can be formalized as below.

Given a parallel sentence pair  $(S, T)$ , where  $S = s_1 \dots s_i \dots s_n$ , and  $T = t_1 \dots t_j \dots t_m$  are sentences of the HQ and LQ sides, respectively;  $s_i$  and  $t_j$  denote the word index (which also denotes the node index in the dependency tree) in the corresponding sentences. We have a dependency tree for  $S$  denoted as  $Tree_S = \{(s_i, s_k) \dots\}$  that is composed of a set of dependencies, where  $(s_i, s_k)$  means that the word  $s_i$  is dependent on the word  $s_k$ . We also have an alignment set  $A = \{a(s_i, t_j) \dots\}$  from  $S$  to  $T$ , where  $a(s_i, t_j)$  means that the HQ word  $s_i$  is aligned to the LQ word  $t_j$ . The new LQ parse tree  $Tree_T^{new}$  is projected from  $Tree_S$ . We first perform the following preprocessing for the unaligned HQ words.

- *unaligned words (HQ side)*: If  $s_i$  is an unaligned word, link the dependencies around  $s_i$ . More specifically, if  $s_i$  is unaligned, and  $(s_h, s_i) \in Tree_S$ ,  $(s_i, s_k) \in Tree_S$ , we add  $(s_h, s_k)$  to  $Tree_S$ , and discard  $(s_h, s_i)$  and  $(s_i, s_k)$  from  $Tree_S$ . This preprocess can make two distinct words separated by unaligned words be a modifier-head pair. For example, in Figure 2, because “32:もの

(thing)” is an unaligned word, we add (30:示唆 (show), 33:と (and)) to  $Tree_S$ .

We then process each source node  $s_i$  in  $Tree_S$  in a top-down manner (from the root node to the leaf node) by applying the following rules divided by the alignment types.

- *one to one alignment*: If  $s_i$  aligns to a unique  $t_j$ ,  $s_k$  aligns to a unique  $t_l$ , and  $(s_i, s_k) \in Tree_S$ , add  $(t_j, t_l)$  to  $Tree_T^{new}$ . For example, in Figure 2, the Japanese dependency (0:この (this), 1:こと (phenomenon)) is projected to the Chinese side as (1:这2:一 (this)) by applying this rule.
- *many to one alignment*: If  $(s_i, s_k, \dots)$  aligns to  $t_j$ , we take the head  $s_r$  (e.g.,  $s_k$ ) from  $(s_i, s_k, \dots)$  as the representative, and then perform the same process as in the one to one alignment case. For example, in Figure 2, a(33:と 34:思わ 35:れる (think), 0:认为(think)) is a many to one alignment, and we select the head “34:思わ” as the representative.
- *one to many alignment*: If  $s_i$  aligns to several words  $(t_j, t_l, \dots)$ , similar to the many to one alignment case, we take the head  $t_r$  (e.g.,  $t_j$ ) from  $(t_j, t_l, \dots)$  based on the original LQ tree as the representative, and then perform the same process as in the one to one alignment case for  $s_i$  and  $t_r$ .
- *many to many alignment*: Reduce this to one-to-many and many-to-one cases, i.e., select the representatives for both sides, and then perform the same process as in the one to one alignment case.

#### 3.1.2 Partial Projection with Direct Mapping

There are several cases that the direct mapping method could not deal with:

1. *the other nodes in the one to many alignment case*: For the nodes (e.g.,  $t_l$ ) (in  $(t_j, t_l, \dots)$  that align to one word  $s_i$ ) other than the representative  $t_r$ , there are no clues to determine their dependencies during the projection.
2. *unaligned words (LQ side)*: If  $t_j$  is an unaligned word, there are also no clues for the projection. For example, in Figure 2, because the word Chinese “3:现象 (phenomenon)”

“15:与 (and)” and “20:的 (’s)” are unaligned words, we cannot determine their dependencies by projection.

3. *alignment errors*: Because the direct mapping method highly depends on word alignments, erroneous word alignments would lead to wrong projected dependency results. For example, in Figure 2, the Japanese word “12:むしろ (preferably)” is incorrectly aligned to the Chinese word “13:及其 (extremely)”; this erroneous alignment would project the Japanese dependency (12:むしろ (preferably), 14:+) to the Chinese side, leading to a projected dependency of (13:及其 (extremely), 19:+), which is obviously incorrect. Alignment errors could happen due to many factors, one of which is translation shift. The erroneous alignment in Figure 2 is caused by this.

Because of the existence of the above cases, we only apply the direct mapping method for partial projection. For the (1) and (2) cases, we leave the dependencies for these words as null. For the (3) case, we propose a projectivity criterion to detect the alignment error, and again leave the dependencies as null. Note that all of these three cases are processed during the top-down projection process.

### 3.1.3 Adding a Projectivity Criterion to the Projection Process

Projectivity is a property of dependency parsing, which informally means that there should not be crossing arcs in a dependency tree (Kubler et al., 2009). For example,  $Treew_T^{new} = \{(0,2)(1,3)(2,3)(3,-1)\}$  (-1 denotes the root) is not projective, because the arc of modifier-head pair (0,2) and that of modifier-head pair (1,3) is crossed. We use the projectivity property to detect alignment errors during the top-down projection process. Suppose that by processing the HQ tree from the root, we already have a partially projected LQ subtree. Next, we want to project a new dependency in the HQ tree to the LQ side. If adding this newly projected dependency to the partially projected subtree leads to non-projectivity,<sup>2</sup> we give up this projection and leave the dependency as null.

Many alignment errors can be detected by the property of projectivity. For example, in Figure

<sup>2</sup> Note that not all non-projectivities are caused by alignment errors; a few of them are also due to translation shift.

2, if we use the erroneous alignment a(12:むしろ (preferably), 13:及其 (extremely)) to project the Japanese dependency (12:むしろ (preferably), 14:+) to the Chinese side, we obtain the dependency of (13:及其 (extremely), 19:+). Before the projection for the node “12:むしろ (preferably)”, because the node “24:挙動 (behavior)” is an ancestor of this node in the Japanese tree, it has been projected. The dependency (24:挙動 (behavior), 26:類似 (similar)) has been projected to the Chinese side, leading the dependency of (27:作用 (behavior), 14:类似 (similar)). (13:及其 (extremely), 19:+) and (27:作用 (behavior), 14:类似 (similar)) lead to non-projectivity. Therefore, we leave the dependency for “13:及其 (extremely)” as null.

## 3.2 Partial Parsing

After the partial projection step, we obtain partial projected trees, with null dependencies discussed in Section 3.1.2. We then perform partial parsing to complement these null dependencies. Before the description of the partial parsing method, we first review the formalism of dependency parsing used in many previous studies such as (Kubler et al., 2009; Shen et al., 2012):

$$Y^* = \operatorname{argmax}_{Y \in \Phi(X)} \operatorname{score}(Y, X) \quad (1)$$

where  $X = x_1 \dots x_i \dots x_n$  is the input sentence,  $Y$  is a candidate tree,  $\Phi(X)$  is a set of all possible dependency trees over  $X$ .  $Y$  can be denoted as  $Y = \{(m, h) : 0 \leq m \leq n, 0 \leq h \leq n\}$ , where  $(m, h)$  is a dependency from the modifier  $x_m$  to the head  $x_h$ . The problem of dependency parsing is to search the best tree from  $\Phi(X)$  that maximizes the score function  $\operatorname{score}(Y, X)$ . The score function can be factorized as the summation of the scores of its factors (subtrees):

$$\operatorname{score}(Y, X) = \sum_{F \in Y} \operatorname{score}(F, X) \quad (2)$$

The score function for each factor is denoted as the inner product of a feature and a weight vector:

$$\operatorname{score}(F, X) = w \cdot f(F, X) \quad (3)$$

The weight vector can be learnt by e.g., the averaged structured perceptron algorithm (Collins, 2002) on an annotated treebank. During parsing, the parser would utilize the learnt weight vector to determine the best parse tree.

In our partial parsing method, we aim to keep the dependencies in partial projected trees, while

complement the null dependencies to construct a projective tree. To realize this, we set extremely high scores to the projected dependencies to maximize the  $score(F, X)$  for these dependencies, while for the null dependencies we set relatively small scores. Doing so, the parser would search the best tree that respects the partial projected dependencies. In our experiments, we used the projective second order graph based dependency parser (Shen et al., 2012). We set the initial dependency scores for the projected dependencies to  $1e12$ , and 0 to the null dependencies.

### 3.3 Re-train a New Low Quality Side Parser

Re-training a new LQ parser on the projected trees is necessary for two reasons. Initially, we use the original LQ parser for the partial parsing process, because we do not have a better choice; due to the low accuracy and the annotation criterion difference problem of the LQ parser, we have the risk that it will produce unsatisfying parsing results, especially for the trees with a low ratio of dependencies being projected. Secondly, if we perform the LQ-to-HQ direction MT, we should make the parsed trees of the input sentences isomorphic to the projected trees. Re-training a new LQ parser on the projected trees could address both of these two problems. As the re-trained parser tend to be more isomorphic to the HQ parser, it could be more effective for the partial parsing process, and could be applied for parsing the input sentences for the LQ-to-HQ direction MT task.

Therefore, after the entire projection process, we select a part of the projected trees, and re-train a parser for the LQ side. How to select the projected trees for training the new LQ parser is an open question. The main question is how to take the balance of the quality and quantity of the projected trees. Currently, the selection criterion is empirical based on the ratio of dependencies projected by the partial projection process in a tree, defined by

$$ratio = \frac{\#projected\_dependencies}{\#all\_dependencies} \quad (4)$$

The motivation behind this is that the more dependencies projected by the partial projection in a tree, the more isomorphic would the projected tree be as the HQ tree, and the less affect would be introduced by the original LQ parser during the partial parsing process. We set a threshold, and

use the trees with the ratio higher than the threshold for training the parser. We tried several thresholds in our preliminary experiments, and selected the best threshold of 0.78 (170k trees) based on the MT performance.<sup>3</sup>

## 4 Experiments

We conducted Japanese-Chinese MT experiments to verify the effectiveness of our constrained partial parsing based projection method.

### 4.1 Settings

We conducted experiments on the scientific domain MT task on the Japanese-Chinese paper excerpt corpus (ASPEC-JC),<sup>4</sup> which is one sub-task of the workshop on Asian translation (WAT)<sup>5</sup> (Nakazawa et al., 2015). The ASPEC-JC task uses 672,315, 2,090, and 2,107 sentences for training, development, and testing, respectively. We used the tree-to-tree MT system KyotoEBMT<sup>6</sup> (Richardson et al., 2015) for all of our MT experiments. For Chinese, we used the Chinese analyzing tool KyotoMorph<sup>7</sup> proposed by Shen et al. (2014) for segmentation and part-of-speech (POS) tagging, and the SKP parser<sup>8</sup> (Shen et al., 2012) for parsing. As the baseline Chinese parser, we trained SKP with the Penn Chinese treebank version 5 (CTB5)<sup>9</sup> containing 18k sentences in news domain, and an in-house scientific domain treebank of 10k sentences. For Japanese, we used JUMAN<sup>10</sup> (Kurohashi et al., 1994) for morphological analyzing, and the KNP parser for parsing<sup>11</sup> (Kawahara and Kurohashi, 2006). We trained two 5-gram language models for Chinese and Japanese, respectively, on the training data of the ASPEC corpus using the KenLM toolkit<sup>12</sup> with interpolated Kneser-Ney discounting, and used them for all the experiments. In all of our experiments, we used the discriminative alignment model Nile<sup>13</sup> (Riesa et al., 2011) for word alignment; tuning was performed by the k-best batch

<sup>3</sup> The average partial projection ratio was 0.70.

<sup>4</sup> <http://lotus.kuee.kyoto-u.ac.jp/ASPEC/>

<sup>5</sup> <http://orchid.kuee.kyoto-u.ac.jp/WAT/>

<sup>6</sup> <http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?KyotoEBMT>

<sup>7</sup> <https://bitbucket.org/msmoshen/kyotomorph-beta>

<sup>8</sup> <https://bitbucket.org/msmoshen/skp-beta>

<sup>9</sup> <https://catalog.ldc.upenn.edu/LDC2005T01>

<sup>10</sup> <http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?JUMAN>

<sup>11</sup> <http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?KNP>

<sup>12</sup> <https://github.com/kpu/kenlm/>

<sup>13</sup> <https://github.com/neubig/nile>

MIRA (Cherry and Foster, 2012) with 10 iterations, and it was re-run for every experiment.

Note that, in our task, Japanese is the HQ parser side, and Chinese is the LQ parser side, because of the parsing accuracy difference (90% v.s. 80%). Therefore, in our experiments, we projected the Japanese parse trees to Chinese. We compared the MT performance of our proposed projection method with the baseline Chinese parser. For Japanese-to-Chinese MT experiments, we compared the MT results of the Chinese training data parsed by the baseline parser, to those of the projected trees. For Chinese-to-Japanese MT, we also re-parsed the development and test Chinese sentences using the SKP model trained on the projected Chinese trees, for the comparison.

## 4.2 MT Results

Table 1 shows the results, where KyotoEBMT is the baseline system that used the Chinese parser trained on CTB5; Baseline partial parsing denotes the projection systems that used the Chinese parser trained on CTB5 for the partial parsing process; Re-trained partial parsing denotes the systems that used the Chinese parser re-trained on the projected trees for the partial parsing process. For reference, we also show the MT performance of the phrase based, string-to-tree, and tree-to-string systems, which are based on the open-source GIZA++/Moses pipeline (Koehn et al., 2007). Note that in all of the Moses, string-to-tree, and tree-to-string settings, Japanese is always in the string format, and Chinese is parsed by the Berkeley parser<sup>14</sup> (Petrov and Klein, 2007).<sup>15</sup> The significance tests were performed using the bootstrap resampling method (Koehn, 2004).

We can see that, the Baseline KyotoEBMT system outperforms the Moses, string-to-tree, and tree-to-string systems, which verifies the effectiveness of the tree-to-tree approach. The performance difference of KyotoEBMT against the other three MT approaches on the Ja-to-Zh direction is much larger than those of the Zh-to-Ja direction. The reason for this is that KyotoEBMT is much more sensitive to the parsing accuracy on the source side, because the source tree is utilized in the ordering of the final translation. Therefore using Chinese as the source side limits the effectiveness

<sup>14</sup> <https://github.com/slavpetrov/berkeleyparser>

<sup>15</sup> We show the MT performance of Moses that only parsed the Chinese data, because these were the baseline systems of WAT.

System	Ja-to-Zh	Zh-to-Ja
Moses phrase based	27.25	33.94
Moses string-to-tree	26.20	N/A
Moses tree-to-string	N/A	33.49
Baseline KyotoEBMT	29.33	34.73
Baseline partial parsing	30.12 <sup>†</sup>	35.84 <sup>†</sup>
Re-trained partial parsing	<b>30.28<sup>†</sup></b>	<b>36.18<sup>†‡</sup></b>

Tab. 1: BLEU scores for ASPEC Ja-to-Zh and Zh-to-Ja (“†,” and “‡” indicate that the result is significantly better than “Baseline KyotoEBMT” and “Baseline partial parsing” at  $p < 0.01$ , respectively).

System	Ja-to-Zh	Zh-to-Ja
Baseline KyotoEBMT	13.13M	8.43M
Baseline partial parsing	15.69M	9.88M
Re-trained partial parsing	15.69M	9.90M

Tab. 2: Number of hypotheses for the test sentences.

of the KyotoEBMT system. Baseline partial parsing performs significantly better than the Baseline KyotoEBMT, and Re-trained partial parsing further improves the performance significantly. We also observe slightly more improvement on the Zh-to-Ja direction than the Ja-to-Zh direction. The reason is similar to the one above that in Zh-to-Ja task, we not only improve the translation example extraction, but also the quality of the input trees.

To further understand the reason for the MT improvement, we investigated the number of hypotheses for the test sentences. The number of hypotheses for a test sentence is the number all the matching hypotheses in the example database for all the subtrees in the input dependency structure of the test sentence (refer to Section 2.1). The entire number of hypotheses for all the test sentences of different systems are shown in table 2. We can see that the number of hypotheses for the partial parsing systems is greatly larger than the baseline KyotoEBMT system. The reason for this is that our projection method significantly increased the isomorphism of the source and target trees in the training corpus, making more translation examples being extractable. More hypotheses are potentially to improve the final MT performance.

In addition, we investigated the translation results of the Baseline KyotoEBMT and Re-trained partial parsing systems. We found that there are

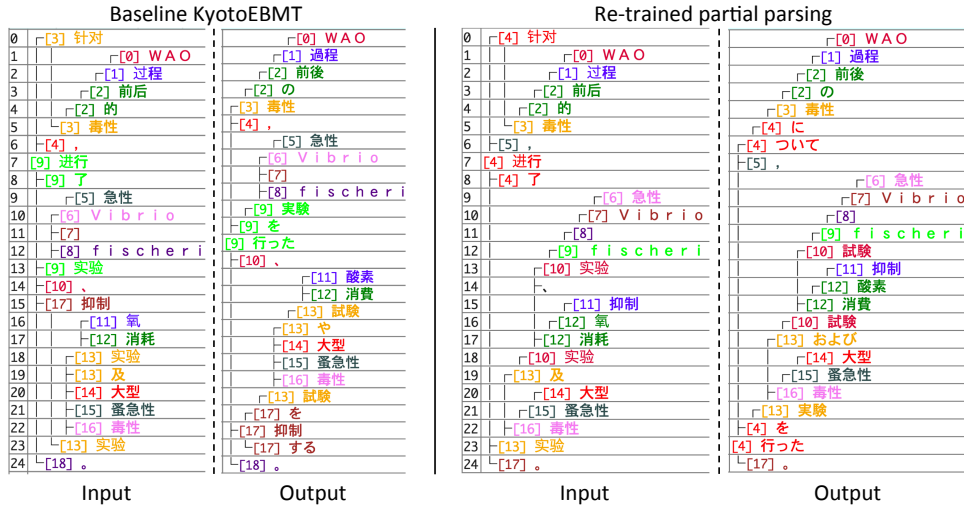


Fig. 4: An improved example of Zh-to-Ja translation (The subtrees in corresponding IDs/colors in the input and output dependency trees show the translation examples being used during translation).

three reasons that lead to the improvement. We explain these reasons through an improved example of Zh-to-Ja translation shown in Figure 4. The first reason is the improvement of the input parse tree. There is a crucial parsing error in the input tree of the Baseline KyotoEBMT system. The KyotoMorph incorrectly assigned a wrong POS tag “VV (verb)” for the word “15:抑制 (inhibition)”, which should be “NN (noun)” in fact. This leads to this word be the head of the whole following noun phrase. Using this erroneous input parse tree, this word is also translated into the head of the entire noun phrase. Our Re-trained partial parsing correctly parsed the word “15:抑制 (inhibition)” as a part of the noun phrase “15-18:抑制氧消耗实验(inhibition of oxygen consumption test)”, leading to the correct translation. Although the Re-trained partial parsing could not correct the wrong POS tag of the word, because we also used this kind of data to train the parser, it successfully parsed this sentence. The second reason is the increase of translation hypotheses. The number of hypotheses for the Baseline KyotoEBMT system is 2,447, while the number of hypotheses of the Re-trained partial parsing system is 3,311. The number of hypotheses for “0:针对...7:进行 8:了 (about...performed)” increased from 52 to 176 by the Re-trained partial parsing system, which improved the translation. The third reason is the isomorphism of the input and output target dependency trees. Note that the noun phrases “15-18:抑制氧消耗实验(inhibition of oxygen consumption test)” and “20-23:大型蚤急性毒性实验(large-

scale flea acute toxicity test)” are parsed as siblings in the Baseline KyotoEBMT system, while in our Re-trained partial parsing model they are parsed as modifier-head dependencies, which are isomorphic to the Japanese parse tree. One unsatisfying point is that “21:蚤急性 (flea acute)” is an unknown word, which is a difficult technical term that could not be translated by both of the two systems.

## 5 Related Work

There are many previous studies that propose many methods to address the difficulties in projecting the parse trees from a resource rich language (e.g., English) to a low resource language, to improve the parsing accuracy of the low resource language. The difficulties in projection can be mainly divided into two categories: word alignment errors and annotation criterion difference (Ganchev et al., 2009).

To address the word alignment error problem, several studies have proposed to train a target parser on high confidence partially projected trees. Ganchev et al. (2009) presented a partial projection method with constraints such as language-specific annotation rules. They then trained a target parser using the partially projected trees. Spreyer and Kuhn (2009) proposed a similar method that trains both graph-based and transition-based dependency parsers on the partially projected trees. Rasooli and Collins (2015) proposed a method to train a target parser on



“dense” projected trees. The “dense” projected trees might only contain a part of dependencies over a threshold. Our proposed method differs from the previous studies in several aspects: we propose the use of the projectivity criterion for partial projection; we utilize the original target parser and propose a constrained partial parsing algorithm; we re-train a target parser on the full trees generated by the partial parsing.

To address the annotation criterion difference problem in projection, Hwa et al. (2005) firstly projected the dependency parse trees, and then applied post projection transformations based on manually created rules. Jiang et al. (2011) presented a method that tolerates the syntactic non-isomorphism between languages. This allows the projected parse trees do not have to follow the annotation criterion of the source parse trees. Our proposed method does not adjust the annotation criterion difference between the source and the projected trees, because in our tree-to-tree MT task, we prefer isomorphic trees.

Only a few studies have been conducted to improve MT performance via projection. For string-to-string MT (Koehn et al., 2007), Goto et al. (2015) proposed a pre-ordering method that projects target side constituency trees to the source side, and then generates pre-ordering rules based on the projected trees. For tree-to-string MT, Jiang et al. (2010) combined projection and supervised constituency parsing by guiding the parsing procedure of the supervised parser with the projected parser. They showed that the guided parser achieved comparable MT results on a tree-to-string system (Liu et al., 2006), compared to a normal supervised parser trained on thousands of CTB trees. For tree-to-tree MT (Richardson et al., 2015), Shen et al. (2015) proposed a naive projection method. They complemented the remaining dependencies for a partially projected tree with a backtracking method. Namely, they reused the dependencies in the original target tree for the complement without considering the partially projected dependencies. In contrast, in this paper we propose partial parsing for the complement, in which we search for the best parse tree by taking account of the partially projected dependencies.

## 6 Conclusion

In this paper, we proposed a constrained partial parsing method for projection to address the non-

isomorphic parse tree problem in a dependency based tree-to-tree MT system. Experiments verified the effectiveness of our proposed method. As future work, firstly, we plan to design a better way for selecting the projected trees for re-training the LQ parser. Secondly, we plan to perform the partial parsing in several iterations. Finally, we plan to conduct experiments on more language pairs to show the language-dependence of our proposed method.

## Acknowledgments

We especially thank to Dr. Mo Shen for the insightful discussion of the constrained partial parsing method with us.

## References

- Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436, Montréal, Canada, June. Association for Computational Linguistics.
- David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL’05)*, pages 263–270, Ann Arbor, Michigan, June. Association for Computational Linguistics.
- Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 1–8. Association for Computational Linguistics, July.
- Fabien Cromieres and Sadao Kurohashi. 2011. Efficient retrieval of tree translation examples for syntax-based machine translation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 508–518, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.
- Fabien Cromieres and Sadao Kurohashi. 2014. Translation rules with right-hand side lattices. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 577–588, Doha, Qatar, October. Association for Computational Linguistics.
- Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proceedings of the 21st International Conference on*

- Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 961–968, Sydney, Australia, July. Association for Computational Linguistics.
- Kuzman Ganchev, Jennifer Gillenwater, and Ben Taskar. 2009. Dependency grammar induction via bitext projection constraints. 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 369–377, Suntec, Singapore, August. Association for Computational Linguistics.
- Isao Goto, Masao Utiyama, Eiichiro Sumita, and Sadao Kurohashi. 2015. Preordering using a target-language parser via cross-language syntactic projection for statistical machine translation. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 14(3):13:1–13:23, June.
- Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Natural language engineering*, 11(03):311–325.
- Hideki Isozaki, Katsuhito Sudoh, Hajime Tsukada, and Kevin Duh. 2010. Head finalization: A simple reordering rule for sov languages. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 244–251, Uppsala, Sweden, July. Association for Computational Linguistics.
- Wenbin Jiang, Yajuan Lv, Yang Liu, and Qun Liu. 2010. Effective constituent projection across languages. In *Coling 2010: Posters*, pages 516–524, Beijing, China, August. Coling 2010 Organizing Committee.
- Wenbin Jiang, Qun Liu, and Yajuan Lv. 2011. Relaxed cross-lingual projection of constituent syntax. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1201, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.
- Daisuke Kawahara and Sadao Kurohashi. 2006. A fully-lexicalized probabilistic model for japanese syntactic and case structure analysis. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 176–183, New York City, USA, June. Association for Computational Linguistics.
- 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, Prague, Czech Republic, June. Association for Computational Linguistics.
- Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain, July. Association for Computational Linguistics.
- Sandra Kubler, Ryan McDonald, Joakim Nivre, and Graeme Hirst. 2009. *Dependency Parsing*. Morgan and Claypool Publishers.
- Sadao Kurohashi, Toshihisa Nakamura, Yuji Matsumoto, and Makoto Nagao. 1994. Improvements of Japanese morphological analyzer JUMAN. In *Proceedings of the International Workshop on Sharable Natural Language*, pages 22–28.
- Yang (1) Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-string alignment template for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 609–616, Sydney, Australia, July. Association for Computational Linguistics.
- Haitao Mi and Liang Huang. 2008. Forest-based translation rule extraction. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 206–214, Honolulu, Hawaii, October. Association for Computational Linguistics.
- Toshiaki Nakazawa, Hideya Mino, Isao Goto, Graham Neubig, Sadao Kurohashi, and Eiichiro Sumita. 2015. Overview of the 2nd Workshop on Asian Translation. In *Proceedings of the 2nd Workshop on Asian Translation (WAT2015)*, pages 1–28, Kyoto, Japan, October.
- Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 404–411, Rochester, New York, April. Association for Computational Linguistics.
- Chris Quirk, Arul Menezes, and Colin Cherry. 2005. Dependency treelet translation: Syntactically informed phrasal SMT. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL’05)*, pages 271–279, Ann Arbor, Michigan, June. Association for Computational Linguistics.
- Mohammad Sadegh Rasooli and Michael Collins. 2015. Density-driven cross-lingual transfer of dependency parsers. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 328–338, Lisbon, Portugal, September. Association for Computational Linguistics.
- John Richardson, Raj Dabre, Chenhui Chu, Fabien Cromières, Toshiaki Nakazawa, and Sadao Kurohashi. 2015. KyotoEBMT System Description for

- the 2nd Workshop on Asian Translation. In *Proceedings of the 2nd Workshop on Asian Translation (WAT2015)*, pages 54–60, Kyoto, Japan, October.
- Jason Riesa, Ann Irvine, and Daniel Marcu. 2011. Feature-rich language-independent syntax-based alignment for statistical machine translation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 497–507, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.
- Libin Shen, Jinxi Xu, and Ralph Weischedel. 2008. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proceedings of ACL-08: HLT*, pages 577–585, Columbus, Ohio, June. Association for Computational Linguistics.
- Mo Shen, Daisuke Kawahara, and Sadao Kurohashi. 2012. A reranking approach for dependency parsing with variable-sized subtree features. In *Proceedings of the 26th Pacific Asia Conference on Language, Information, and Computation*, pages 308–317, Bali, Indonesia, November. Faculty of Computer Science, Universitas Indonesia.
- Mo Shen, Hongxiao Liu, Daisuke Kawahara, and Sadao Kurohashi. 2014. Chinese morphological analysis with character-level pos tagging. In *Proceedings of ACL*, pages 253–258.
- Yu Shen, Chenhui Chu, Fabien Cromieres, and Sadao Kurohashi. 2015. Cross-language projection of dependency trees for tree-to-tree machine translation. In *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computing (PACLIC2015)*, pages 80–88, Shanghai, China, 10.
- Kathrin Spreyer and Jonas Kuhn. 2009. Data-driven dependency parsing of new languages using incomplete and noisy training data. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 12–20, Boulder, Colorado, June. Association for Computational Linguistics.
- Min Zhang, Hongfei Jiang, Aiti Aw, Haizhou Li, Chew Lim Tan, and Sheng Li. 2008. A tree sequence alignment-based tree-to-tree translation model. In *Proceedings of ACL-08: HLT*, pages 559–567, Columbus, Ohio, June. Association for Computational Linguistics.