Effects of Parsing Errors on Pre-reordering Performance for Chinese-to-Japanese SMT

Dan Han\textsuperscript{1,2}, Pascual Martínez-Gómez\textsuperscript{2,3}, Yusuke Miyao\textsuperscript{1,2}, Katsuhito Sudoh\textsuperscript{4}, Masaaki Nagata\textsuperscript{4}

\textsuperscript{1}The Graduate University For Advanced Studies
\textsuperscript{2}National Institute of Informatics, \textsuperscript{3}The University of Tokyo
\textsuperscript{4}NTT Communication Science Laboratories, NTT Corporation

\{handan, pascual, yusuke\}@nii.ac.jp
\{sudoh.katsuhito, nagata.masaaki\}@lab.ntt.co.jp

Abstract

Linguistically motivated reordering methods have been developed to improve word alignment especially for Statistical Machine Translation (SMT) on long distance language pairs. However, since they highly rely on the parsing accuracy, it is useful to explore the relationship between parsing and reordering. For Chinese-to-Japanese SMT, we carry out a three-stage incremental comparative analysis to observe the effects of different parsing errors on reordering performance by combining empirical and descriptive approaches. For the empirical approach, we quantify the distribution of general parsing errors along with reordering qualities whereas for the descriptive approach, we extract seven influential error patterns and examine their correlation with reordering errors.

1 Introduction

Statistical machine translation is a challenging and well established task in the community of computational linguistics. One of the key components of statistical machine translation systems are word alignment techniques, where the words from sentences in a source language are mapped to words from sentences in a target language. When estimating the most appropriate word alignments, it is unfeasible to explore every possible word correspondence due to the combinatorial complexity. Considering local permutations of words might be effective to translate languages with a similar sentence structure, but these methods have a limited performance when translating sentences from languages with different syntactical structures.

An effective technique to translate sentences between distant language pairs is pre-reordering, where words in sentences from the source language are re-arranged with the objective to resemble the word order of the target language. Re-arranging rules are automatically extracted (Xia and McCord, 2004; Genzel, 2010), or linguistically motivated (Xu et al., 2009; Isozaki et al., 2010; Han et al., 2012; Han et al., 2013). We work following the latter strategy, where the source sentence is parsed to find its syntactical structure, and linguistically-motivated rules are used in combination with the structure of the sentence to guide the word reordering. The language pair under consideration is Chinese-to-Japanese, which despite their common roots, it is a well known language pair for their different sentence structure.

However, syntax-based pre-reordering techniques are sensitive to parsing errors, but insight into their relationship has been elusive. The contribution of this work is two fold. First, we provide an empirical analysis where we quantify the aggregated impact of parsing errors on pre-reordering performance. Second, we define seven patterns of the most common and influential parsing errors and we carry out a descriptive analysis to examine their relationship with reordering errors. We combine an empirical and descriptive approach to present a three-stage incremental comparative analysis to observe the effect of different parsing errors on reordering performance.

In Section 2, after a brief description on the pre-reordering method that we use for experiments, we will introduce some related works on parsing error analysis and analysis on the relation between parsing and machine translation. From a general perspective, we describe our analysis methods for this work in Section 3. Then, we carry out the analysis and exhibit the results in Section 4 and Section 5. The last two sections are dedicated to discussion, future directions and summarize our findings.
2 Background

2.1 Reordering Model

Since local reordering models which are integrated in phrase-based SMT systems do not perform well for distant language pairs due to their different syntactic structures, pre-reordering methods have been proposed to supply the need for improving the word alignment. Han et al. (2013) described one of the latest pre-reordering methods (DPC) which was based on dependency parsing. The authors were using an unlabeled dependency parser to extract the syntactic information of Chinese sentences, and then by combining with part-of-speech (POS) tags\(^1\), they defined a set of heuristic reordering rules to guide the reordering. The essential idea of DPC is to move so-called verbal block (Vb)\(^2\) to the right-hand side of its right-most dependent (RM-D) for a Subject-Verb-Object (SVO) language to resemble a Subject-Object-Verb (SOV) language’s word order. Table 1 shows the POS tags that are used to identify words as Vb-H, RM-D, or BEI (a Vb-H involves in a bei-construction) in a sentence from Han et al. (2013).

Table 1: Lists of POS tags for identifying words as Vb-H, RM-D, and BEI. (Han et al., 2013)

| POS  | Tag   |
|------|-------|
| Vb-H | VV V E VC VA P |
| BEI  | LB SB |
| RM-D | NN NR NT PN OD CD M FW CC |
| ETC  | LC DEV DT JJ SP IJ ON |

Figure 1 shows an example of unlabeled dependency parse tree of a Chinese sentence (SVO) with word aligned to its Japanese counterpart (SOV). Arrows are pointing from heads to dependents.

is important to observe the effects of parsing errors on reordering performance.

In this analysis, we borrow this state-of-the-art pre-reordering model for our experiments since it is a rule-based pre-reordering method for a distant language pair based on dependency parsing as well as its extensibility to other language pairs.

2.2 Related Work

Although there are studies on analyzing parsing errors and reordering errors, as far as we know, there is not any work on observing the relationship between these two types of errors.

One most relevant work to ours is observing the impact of parsing accuracy on a SMT system introduced in Quirk and Corston-Oliver (2006). They showed the general idea that syntax-based SMT models are sensitive to syntactic analysis. However, they did not further analyze concrete parsing error types that affect task accuracy.

Green (2011) explored the effects of noun phrase bracketing in dependency parsing in English, and further on English to Czech machine translation. But the work focused on using noun phrase structure to improve a machine translation framework. In the work of Katz-Brown et al. (2011), they proposed a training method to improve a parser’s performance by using reordering quality to examine the parse quality. But they did not study the relationship between reordering quality and parse quality.

There are more works on parsing error analysis. For instance, Hara et al. (2009) defined several types of parsing error patterns on predicate argument relation and tested them with a Head-driven phrase structure grammar (HPSG) (Pollard and Sag, 1994) parser (Miyao and Tsujii, 2008). McDonald and Nivre (2007) explored parsing errors for data-driven dependency parsing by

\(^1\)In this work, POS tag definitions follow the POS tag guidelines of the Penn Chinese Treebank v3.0.

\(^2\)According to (Han et al., 2013), a Vb includes the head of the Vb (Vb-H) and an optional component (Vb-D).
comparing a graph-based parser with a transition-based parser, which are representing two dominant parsing models. At the same time, Dredze et al. (2007) provided a comparison analysis on differences in annotation guidelines among treebanks which were suspected to be responsible for dependency parsing errors in domain adaptation tasks. Unlike analyzing parsing errors, authors in Yu et al. (2011) focused on the difficulties in Chinese deep parsing by comparing the linguistic properties between Chinese and English.

There are also works on reordering error analysis like Han et al. (2012) which examined an existing reordering method and refined it after a detailed linguistic analysis on reordering issues. Although they discovered that parsing errors affect the reordering quality, they did not observe the concrete relationship. On the other hand, Giménez and Márquez (2008) proposed an automatic error analysis method of machine translation output, by compiling a set of metric variants. However, they did not provide insight on what SMT component caused low translation performance.

3 Analysis Method

We combine an empirical approach with a descriptive approach to observe the effects of parsing errors on pre-reordering performance in three stages: preliminary experiment stage, POS tag level stage, and dependency type level stage. First, we provide a general idea of the sensitiveness of parsing errors on reordering method. Then, we use POS tags to identify parsing errors and quantify the aggregate impact on reordering performance. Finally, we define several concrete error patterns and examine their effects on reordering qualities.

In order to test for an upper bound of the reordering performance and examine the specific parsing errors that affect reordering, one way is to contrast the reordering based on error-free parse trees with the reordering based on auto-parse trees. Error-free parse trees are considered as gold trees.

In the preliminary experiment stage, we set up two benchmarks in two scenarios. For scenario 1, the benchmark is manually reordered Chinese sentence on the basis of Japanese reference. By measuring the word order similarities between the benchmark and the gold-tree based reordered sentence as well as between the benchmark and the auto-parse tree based reordered sentence separately, we quantify the extent of parsing errors that influence reordering. Meanwhile, the former measurement shows additionally the general figure of the upper bound of the reordering method. However, since it is not only time-consuming but also labor-intensive to set up the benchmark in scenario 1, we use the Japanese reference as the benchmark in scenario 2 and follow the same strategies as in scenario 1 to calculate the word order similarities. More detailed description on the preliminary experiment is given in Section 4.

In POS tag level stage, we compare the gold-tree with auto-parse tree along with reordering quality to explore the relationship between general parsing errors and reordering from two aspects: the percentages of top three most frequent dependent’s POS tags that point to wrong heads and the percentages of top two most frequent head’s POS tags that are recognized wrongly. The percentages of other POS tags are not provided because they are negligible. Our objective is to profile general parsing errors’ distribution. However, this does not imply that those errors are the cause of the reordering errors. Section 5.1 includes more concrete analysis results.

In dependency type level stage, we classify the most influential parsing errors on reordering into three superclasses and seven subclasses according to the methodology of the reordering method. We then plot the distribution of these parsing errors for various reordering qualities. In Section 5.2, we illustrate these parsing errors with examples.

4 Preliminary Experiment

4.1 Gold Data

In order to build up gold parse tree sets for comparison, we used the annotated sentences from Chinese Penn Treebank ver. 7.0 (CTB-7) which is a well known corpus that consists of parsed text in five genres. They are Chinese newswire (NS), magazine news (NM), broadcast news (BN), broadcast conversation programs (BC), and web newsgroups, weblogs (NW).

We first randomly selected 517 unique sentences (hereinafter set-1) from all five genres in development set of CTB-7 which is split according to (Wang et al., 2011). However, we found that sentences in BC and NW are mainly from spoken language, which tend to have faults like repetitions, incomplete sentences, corrections, or incorrect sentence segmentation. Therefore, we randomly selected another 2,126 unique sentences.
Table 2: Statistics of selected sentences in five genres of CTB-7. AL stands for the average length of sentences, while Voc. for vocabulary.

|        | BN | BC | NM | NS | NW | Total |
|--------|----|----|----|----|----|-------|
| set-1  | 100| 100| 100| 117| 100| 517   |
| set-2  | 797| -  | 578| 751|-  | 2,126 |
| Total  | 897| 100| 678| 808| 100| 2,643 |

Table 3: The average value of Kendall’s tau ($\tau$) of 517 Chinese sentences by comparing manually reordered sentences, unordered sentences, and automatically reordered sentences. M-reordered is short for manually reordered.

|        | Baseline | Gold-DPC | Auto-DPC |
|--------|----------|----------|----------|
| M-reordered | 0.82     | 0.90     | 0.88     |
| Gold-DPC  | -        | -        | 0.95     |

Table 3 shows the average $\tau$ value.

For baseline system, the average $\tau$ value shows how similar these 517 Chinese sentences between manually reordered ones and non-reordered ones are. Comparing with manually reordered Chinese, both Auto-DPC and Gold-DPC achieved higher average $\tau$ value than baseline, which imply that the reordering method DPC positively reordered the Chinese sentences and improved the word alignment. Nevertheless, a slightly lower average $\tau$ value of Auto-DPC shows that DPC is sensitive on parsing errors. This assumption is also confirmed by the average $\tau$ value between Auto-DPC and Gold-DPC. However, the difference of $\tau$ values are limited. We hence increase the test data by adding set-2 for further experiments in scenario 2.

Scenario 1 Preliminary observation about the effects of parsing errors on reordering performance is to compare word order similarities between manually reordered Chinese sentences and automatically reordered Chinese sentences from set-1. Table 3 shows the average $\tau$ value.

For baseline system, the average $\tau$ value shows how similar these 517 Chinese sentences between manually reordered ones and non-reordered ones are. Comparing with manually reordered Chinese, both Auto-DPC and Gold-DPC achieved higher average $\tau$ value than baseline, which imply that the reordering method DPC positively reordered the Chinese sentences and improved the word alignment. Nevertheless, a slightly lower average $\tau$ value of Auto-DPC shows that DPC is sensitive on parsing errors. This assumption is also confirmed by the average $\tau$ value between Auto-DPC and Gold-DPC. However, the difference of $\tau$ values are limited. We hence increase the test data by adding set-2 for further experiments in scenario 2.

Scenario 2 Since we do not have manually reordered Chinese sentences as benchmark for set-2, we calculate the Kendall’s tau between Chinese sentences and their Japanese counterparts for both data sets by using the MGIZA++ alignment

(hereinafter set-2) within a limit to three genres: NS, NM, and BN. Table 2 shows the statistics of all selected sentences in five genres respectively.

For converting CTB-7 parsed text to dependency parse trees, we used an open utility Penn2Malt\(^3\) which converts Penn Treebank into MaltTab format containing dependency information. Since the head rules that Penn2Malt recommended for converting on its website do not contain three new annotation types in CTB-7, we added three new ones for them as follows: FLR (Fillers) and DFL (Disfluency) head on right-hand branch; INC (Incomplete sentences) follows the same head rule as FRAG (Fragment).

Meanwhile, professional human translators translated all Chinese sentences in both set-1 and set-2 into Japanese. Thereafter, according to the Japanese references, Chinese sentences in set-1 have been manually reordered as the same word orders as their Japanese counterparts by a bilingual speaker of Chinese and Japanese for the experiments in scenario 1. For example, the Chinese sentence in Figure 1 is following the word order of “He bookstore went (to) a book buy (-ed).” in the handcrafted reordered set since it resembles the Japanese word order.

4\textsuperscript{http://triplet.cc/software/corbit

5\textsuperscript{Note that Corbit was tuned with the development set of CTB-7.

4\textsuperscript{http://triplet.cc/software/corbit

5\textsuperscript{Note that Corbit was tuned with the development set of CTB-7.
trees. Since the syntactic information that guides reordering in DPC is limited to dependency structure and POS tags, for analysis on the causes of reordering errors, we examine parsing errors from these two linguistic categories. In this section, the value of Kendall’s tau measures the word order similarity between Gold-DPC and Auto-DPC.

5.1 Part-of-Speech Tag Error

There are two types of parsing errors to a token in a dependency parse tree. One is that the token points to a wrong head, namely dependent-error, and another one is that the token is recognized wrongly as a head of other tokens, namely head-error. For example, Figure 3 presents a possible wrong parse tree of the example shown in Figure 1. By comparing with the gold-tree in Figure 1, tokens (POS tag) of “he (PN)”, “went (VV)”, “bookstore (NN)”, “buy (VV)”, “a (CD)”, and “.” (PU) in the dependency tree in Figure 3 all point to different wrong heads, which are dependent-errors. Concurrently, tokens (POS tag) of “went (VV)”, “buy (VV)”, and “book (NN)” are wrongly recognized as heads of other tokens (e.g., “he”, “bookstore”, “a”), which are head-errors. According to the definition, every head-error has at least one corresponding dependent-error. However, in the case that a token is not the root in a gold-tree but is root in the wrong tree, this token is a dependent-error corresponding with no head-error. An example is the dependent-error “went (VV)” in Figure 3.

We count the number of POS tag mis-recognitions separately for dependent- and head-errors. In the example of Figure 3, dependent-error counts are for VV, 2 errors, and PN, NN, CD, PU each 1 error. The number of POS tag mis-recognitions for head-errors are VV with 2 errors, and NN with 1 error. In our analysis, we will compute these counts for all POS tags at every sentence in our data set. However, our reordering method performed differently at each sentence in our data set, and the reordering quality varied from

![Figure 2: The distribution of Kendall's tau values for 2,236 bilingual sentences (Chinese-Japanese) in which the Chinese is from three systems of baseline, Auto-DPC, and Gold-DPC.](image)

![Figure 3: A possible wrong dependency parse tree of the example in Figure 1.](image)
sentence to sentence. With the objective of observing the correlation between reordering quality and each type of error, we will first group sentences according to their Kendall’s τ values. Then, we will compute proportions of POS tag errors at each τ value, for every type of POS tag error.

Figure 4 shows the distribution of top three dependent-error POS tags, which means that they are the three most frequent POS tags that point to a wrong head in auto-parse trees. VV represents all verbs except predicative adjective (VA), copula (VC), and you3 as the main verb (VE). PU represents punctuation and NN represents all nouns except proper noun (NR), temporal noun (NT), and the ones for locations which cannot modify verb phrases with or without de0. The dependent-error on VV accounts for a larger proportion in low reordering accuracy sentences whereas more NN dependent-error occurred in high reordering accuracy sentences. On the other hand, the proportion of PU dependent-error is more consistent.

Figure 5 shows the distribution of top two head-error POS tags, which means that they are the two most frequent POS tags that are recognized wrongly as heads in auto-parse trees. Comparing to Figure 4, the tendency of both VV and NN is the same but distincter.

The analysis results on the proportion distributions of dependent-error POS tags and head-error POS tags in different reordering quality sentence groups exhibit that there are more parsing errors on verbs than nouns in low reordering accuracy sentences and thus the parsing errors on verbs influence more on the reordering performance. However, it is still difficult to reveal the effects of more concrete parsing errors on reordering considering that not all verb parsing errors influence the reordering. As an illustration, in Figure 3, if the head of “bookstore” were “went”, the VV head-error of “went” would not cause any reordering error since it would be reordered consistently to the right-hand side of its RM-D “bookstore”. Consequently, we use a descriptive approach to analyze dependency types to explore the effects from more concrete parsing errors in the next section.

5.2 Dependency Type Error

As introduced in Section 2.1, DPC first identifies Vb, RM-D, and then reorderers necessary words. Thus, DPC reorderers not only Vb-H, but also Vb-D in a Vb, which means that the failure on identifying Vbs may also cause unexpected reordering on particles, such as aspect markers. However, in this work, we only focus on reordering issues of Vb-H candidates. To discover the effects of more concrete parsing errors on reordering, we distinguish three categories of dependency types, i.e., ROOT, RM-D, and BEI. Among them, ROOT denotes whether the Vb-H candidate is the root of the sentence or not. RM-D is the right-most object dependent of the Vb-H candidate if it has one, and BEI denotes whether the Vb-H candidate is involved in a bei-construction.

According to the methodology of the reordering method DPC, we define seven patterns of parsing error phenomena and classify them into three types by comparing the gold-tree (GT) with auto-parse tree (Corbit-tree, CT). Table 4 lists all parsing error patterns in three error types, ROOT error, RM-D error and BEI error by considering three dependency types ROOT, RM-D and BEI. Symbols of “√”, “×”, “?” represent the status of a cer-

---

*A Chinese character expresses possession and existence.

*A Chinese character is specially used to connect the verb phrase and its modifier.

---

9We use “Vb-H candidate” in this work for the reason that if the Vb-H is involved into a bei-construction, then it can not be Vb-H according to (Han et al., 2013).
Table 4: Seven error patterns (Root-C, Root-G, RM_D-C, RM_D-G, RM_D-D, BEI-C, BEI-G) that cause three types of reordering issues (ROOT error, RM-D error, and BEI error). GT stands for gold-tree, and CT stands for Corbit-tree. Symbols “✓”, “✗”, “?” represent the status of True, False, and Unknown, respectively. “diff.” means that the RM-Ds exist in both GT and CT but are different.

The sentence according to the CT whereas it will not be reordered according to GT, which is already in the same position as its Japanese counterpart.

**Root-G** is the opposite case of Root-C where a Vb-H candidate is the root of the sentence but was not parsed as the root in CT. This affects the reordering under the two same constraints as Root-C. Figure 7b shows an example of Root-G. In Figure 7a, the word alignment shows that the Vb-H “agree” should be reordered to the end of the sentence. However, it will not be reordered for the wrong parse tree shown in Figure 7b.

**RM_D-C** is the case where the RM-D of a Vb-H candidate exists in a CT but not in GT. In other words, a RM-D candidate was parsed wrongly on its head. There are four varieties of combination with the status of ROOT, BEI of the Vb-H candidate that lead to incorrect reorderings. The Vb-H “agree” in Figure 7c matches the last combination of RM_D-C, which will be reordered right after “journalist” instead of at the end of the sentence.

**RM_D-G** is the opposite case of RM_D-C where the RM-D of a Vb-H candidate was missed in a CT. There are also four cases of reordering errors according to the status of BEI, ROOT and RM-D. Vb-H “went” in Figure 3 matches the second combination of RM_D-G so that it will not be able to reorder after “bookstore”.

**RM_D-D** is the case where a bei-construction-free Vb-H candidate obtains two different RM-D candidates in CT and GT, which causes the reordering issue. In Figure 6, Vb-H “join” received different RM-Ds in two trees. According to the
word alignment, it should be reordered next to “navy” instead of “combat power”.

**BEI-C** is the case where a Vb-H candidate received a wrong BEI dependent in CT. This will prevent reordering independently on whether the Vb-H candidate has RM-D or is the root.

**BEI-G** is the opposite case of BEI-C, where Vb-H in GT will not be reordered but in CT it will.

After defining seven patterns of parsing errors and classifying them into three types, we calculate the average frequency proportions of each type in different $\tau$ value groups of sentences.

Figure 8 shows the distribution of the three types of parsing errors and their tendencies. In low $\tau$ value sentences, there are higher proportions of ROOT errors, and relatively lower proportions in high $\tau$ value sentences. RM-D errors follow the opposite tendency. This implies that the effects of ROOT errors on reordering are stronger than the effects from RM-D errors. The reason could be that ROOT errors cause long distance reordering failure while RM-D errors lead to more local reordering errors. Since there are very few BEI errors, it was difficult to capture their trends.

Figure 9 and Figure 10 provide the correlations between parsing error patterns and reordering accuracy. In ROOT errors types, Root-C had a larger percentage than Root-G in low reordering accuracy sentences which shows that the Vb-H candidate that does not have any object dependent tends to be recognized as root by parser. This is consistent with the distribution results that are shown in Figure 10. The error pattern of RM_D-G had larger percentage than the other two patterns, which also implies that a Vb-H candidate in a CT tends to have less or none object dependents.

### 5.3 Further Analysis Possibilities

Due to the time limitation, we only focused on analyzing parsing errors that cause reordering issues on Vb-H candidates while defining the error patterns. However, it is not only that Vb-H candidates are reordered in DPC, but also other words like Vb-D candidates and particles will be reordered. It is also meaningful to explore the parsing error patterns which cause unexpected reordering on these words and the correlation between them as well.

The current study on exploring influential parsing errors is not exhaustive, and another analysis possibility would be to explore what types of parsing errors do not affect reordering so that parsers can sacrifice their performance on those types of issues in order to improve on influential types.
For automatically learned reordering rules, those able POS tags or unreliable dependency relations could be designed to be more robust against unreli- tivated pre-reordering methods, reordering rules parsing accuracies. In case of linguistically mo- benefit of preliminary analysis of POS tagging and of syntax-based pre-reordering methods would high accuracy to recognize the sentence root. similar strategy should rely on parsers that have a our analysis demonstrated that systems that follow speech constructions. For that purpose, it is cru- rivalsity to correctly reorder sentences with reported errors by using the linguistic feature of depen- mation to the cause of the effects. The analysis results assist us to achieve a better the syntactic structures and the reordering model. dependency types based on a deep linguistic study of errors by using POS tag information. The distributions of two types of parsing errors on reordering quality of the reordering model. objective: (i) quantify effects of parsing errors on reordering, (ii) estimate upper bounds in perfor- mance of the reordering method, (iii) profile gen- eral parsing errors, and (iv) examine effects of spec- ific parsing errors on reordering.

In the first stage, we set up benchmarks in two scenarios for reordered Chinese sentences. By calculating the word order similarity between the benchmarks and the dependency parse tree based auto-reordered Chinese sentences, we quantified the correlation between parsing errors and reorder- ing accuracies as well as explored the upper bound in reordering quality of the reordering model.

In the second stage, we examined the effects of two types of parsing errors on reordering quality by using POS tag information. The distributions of parsing errors’ POS tags provide a general view of the influential parsing error types and an approxi- mation to the cause of the effects.

In the last stage, we defined several patterns of parsing errors that assuredly cause reordering errors by using the linguistic feature of depen- dency types based on a deep linguistic study of the syntactic structures and the reordering model. The analysis results assist us to achieve a better and more explicit understanding on the relation- ship between parsing errors and reordering performance. Furthermore, we captured the effects of more concrete parsing errors on reordering.

7 Conclusion

In this work, we carried out linguistically moti- vated analysis methods by combining empirical and descriptive approaches in three analysis stages to examine the effects of different parsing errors on pre-reordering performance. We achieved four objectives: (i) quantify effects of parsing errors on reordering, (ii) estimate upper bounds in performance of the reordering method, (iii) profile general parsing errors, and (iv) examine effects of spe- cific parsing errors on reordering.

In the first stage, we set up benchmarks in two scenarios for reordered Chinese sentences. By calculating the word order similarity between the benchmarks and the dependency parse tree based auto-reordered Chinese sentences, we quantified the correlation between parsing errors and reorder- ing accuracies as well as explored the upper bound in reordering quality of the reordering model.

In the second stage, we examined the effects of two types of parsing errors on reordering quality by using POS tag information. The distributions of parsing errors’ POS tags provide a general view of the influential parsing error types and an approxi- mation to the cause of the effects.

In the last stage, we defined several patterns of parsing errors that assuredly cause reordering errors by using the linguistic feature of depen- dency types based on a deep linguistic study of the syntactic structures and the reordering model. The analysis results assist us to achieve a better and more explicit understanding on the relation- ship between parsing errors and reordering performance. Furthermore, we captured the effects of more concrete parsing errors on reordering.
References

Mark Dredze, John Blitzer, Partha Pratim Talukdar, Kuzman Ganchev, Joao Graca, and Fernando Pereira. 2007. Frustratingly hard domain adaptation for dependency parsing. In Proc. of the CoNLL Shared Task Session of EMNLP-CoNLL, pages 1051–1055.

Qin Gao and Stephan Vogel. 2008. Parallel implementations of word alignment tool. In Software Engineering, Testing, and Quality Assurance for Natural Language Processing, pages 49–57.

Dmitriy Genzel. 2010. Automatically learning source-side reordering rules for large scale machine translation. In Proc. of COLING, pages 376–384.

Jesús Giménez and Lluís Marquez. 2008. Towards heterogeneous automatic MT error analysis. In Proc. of the 6th International Conference on Language Resources and Evaluation, pages 1894–1901.

Nathan Green. 2011. Effects of noun phrase bracketing in dependency parsing and machine translation. In Proc. of ACL-HLT, Student Session, pages 69–74.

Dan Han, Katsuhito Sudoh, Xianchao Wu, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. 2012. Head finalization reordering for Chinese-to-Japanese machine translation. In Proc. of the ACL 6th Workshop on SSST, pages 57–66.

Dan Han, Pascual Martínez-Gómez, Yusuke Miyao, Katsuhito Sudoh, and Masaaki Nagata. 2013. Using unlabeled dependency parsing for pre-reordering for Chinese-to-Japanese statistical machine translation. In Proc. of the ACL Second Workshop on Hybrid Approaches to Translation, pages 25–33.

Tadayoshi Hara, Yusuke Miyao, and Jun’ichi Tsujii. 2009. Descriptive and empirical approaches to capturing underlying dependencies among parsing errors. In Proc. of EMNLP, pages 1162–1171.

Kun Yu, Yusuke Miyao, Takuya Matsuzaki, Xiangli Wang, and Junichi Tsujii. 2011. Analysis of the difficulties in Chinese deep parsing. In Proc. of the 12th International Conference on Parsing Technologies, pages 48–57.