Transliteration Extraction from Classical Chinese Buddhist Literature Using Conditional Random Fields

Yu-Chun Wang
Department of Computer Science
and Information Engineering,
National Taiwan University, Taiwan
Telecommunication Laboratories,
Chunghwa Telecom, Taiwan
d97023@csie.ntu.edu.tw

Richard Tzong-Han Tsai*
Department of Computer Science
and Information Engineering,
National Central University,
Zhongli City, Taiwan
thtsai@csie.ncu.edu.tw

Abstract

Extracting plausible transliterations from historical literature is a key issues in historical linguistics and other resaech fields. In Chinese historical literature, the characters used to transliterate the same loanword may vary because of different translation eras or different Chinese language preferences among translators. To assist historical linguiatics and digial humanity researchers, this paper propose a transliteration extraction method based on the conditional random field method with the features based on the characteristics of the Chinese characters used in transliterations which are suitable to identify transliteration characters. To evaluate our method, we compiled an evaluation set from the two Buddhist texts, the Samyuktagama and the Lotus Sutra. We also construct a baseline approach with suffix array based extraction method and phonetic similarity measurement. Our method outperforms the baseline approach a lot and the recall of our method achieves 0.9561 and the precision is 0.9444. The results show our method is very effective to extract transliterations in classical Chinese texts.

1 Introduction

Cognates and loanwords play important roles in the research of language origins and cultural interchange. Therefore, extracting plausible cognates or loanwords from historical literature is a key issues in historical linguistics. The adoption of loanwords from other languages is usually through transliteration. In Chinese historical literature, the characters used to transliterate the same loanword may vary because of different translation eras or different Chinese language/dialect preferences among translators. For example, in classical Chinese Buddhist scriptures, the translation process of Buddhist scriptures from Sanskrit to classical Chinese occurred mainly from the 1st century to 10th century. In these works, the same Sanskrit words may transliterate into different Chinese loanword forms. For instance, the surname of the Buddha, Gautama, is transliterated into several different forms such as "瞿曇" (*qü-tan*) or "喬答摩" (*qiao-da-mo*), and the name "Culapanthaka" has several different Chinese transliterations such as "朱利槃特" (*zhu-li-pan-te*) and "周利槃陀伽" (*zhou-li-pan-tuo-qie*). In order to assist researchers in historical linguistics and other digital humanity research fields, an approach to extract transliterations in classical Chinese texts is necessary.

Many transliteration extraction methods require a bilingual parallel corpus or text documents containing two languages. For example, (Sherif and Kondrak, 2007) proposed a method for learning the string distance measurement function from a sentence-aligned English-Arabic parallel corpus to extract transliteration pairs. (Kuo et al., 2007) proposed a transliteration pair extraction method using a phonetic similarity model. Their approach is based on the general rule that when a new English term is transliterated into Chinese (in modern Chinese texts, e.g. newswire), the English source term usually appears alongside the transliteration. To exploit this pattern, they identify all the English terms in a Chinese text and measure the phonetic similarity between those English terms and their surrounding Chinese terms, treating the pairs with the highest similarity as the true transliteration pairs. Despite its high accuracy, this approach cannot be applied to transliteration extraction in classical Chinese literature since the prerequisite (of the source terms alongside the transliteration) does not apply.

27th Pacific Asia Conference on Language, Information, and Computation pages 260－266

Some researchers have tried to extract transliterations from a single language corpus. (Oh and Choi, 2003) proposed a Korean transliteration identification method using a Hidden Markov Model (HMM) (Rabiner, 1989). They transformed the transliteration identification problem into a sequential tagging problem in which each Korean syllable block in a Korean sentence is tagged as either belonging to a transliteration or not. They compiled a human-tagged Korean corpus to train a hidden Markov model with predefined phonetic features to extract transliteration terms from sentences by sequential tagging. (Goldberg and Elhadad, 2008) proposed an unsupervised Hebrew transliteration extraction method. They adopted an English-Hebrew phoneme mapping table to convert the English terms in a named entity lexicon into all the possible Hebrew transliteration forms. The Hebrew transliterations are then used to train a Hebrew transliteration identification model. However, Korean and Hebrew are alphabetical writing system, while Chinese is ideographic. These identification methods heavily depend on the phonetic characteristics of the writing system. Since Chinese characters do not necessarily reflect actual pronunciations, these methods are difficult to apply to the transliteration extraction problem in classical Chinese.

This paper proposes an approach to extract transliterations automatically in classical Chinese texts, especially Buddhist scriptures, with supervised learning models based on the probability of the characters used in transliterations and the anguage model features of Chinese characters.

2 Method

To extract the transliterations from the classical Chinese Buddhist scriptures, we adopt a supervised learning method, the conditional random fields (CRF) model. The features we use in the CRF model are described in the following subsections.

2.1 Probability of each Chinese character in transliterations

According to our observation, in the classical Chinese Buddhist texts, the Chinese characters chosen be used in transliteration show some characteristics. Translators tended to choose the characters that do not affect the comprehension of the sentences. The amount of the Chinese characters is huge, but the possible syllables are limited in Chinese. Therefore, one Chinese character may share the same pronunciation with several other characters. Hence, the translators may try to choose the rarely used characters for transliteration.

From our observation, the probability of each Chinese character used to be transliterated is an important feature to identify transliteration from the classical Buddhist texts. In order to measure the probability of every character used in transliterations, we collect the frequency of all the Chinese characters in the Chinese Buddhist Canon. Furthermore, we apply the suffix array method (Manzini and Ferragina, 2004) to extract the terms with their counts from all the texts of the Chinese Buddhist Canon. Next, the extracted terms are filtered out by the a list of selected transliteration terms from the Buddhist Translation Lexicon and Ding Fubao's Dictionary of Buddhist Studies. The extracted terms in the list are retained and the frequency of each Chinese character can be calculated. Thus, the probability of a given Chinese character c in transliteration can be defined as:

$$Prob(c) = \log \frac{freq_{trans}(c)}{freq_{all}(c)}$$

where $freq_{trans}(c)$ is c's frequency used in transliterations, and $freq_{all}(c)$ is c's frequency appearing in the whole Chinese Buddhist Canon. The logarithm in the formula is designed for CRF discrete feature values.

2.2 Language model of the transliteration

Transliterations may appear many times in one Buddhist sutra. The preceding character and the following character of the transliteration may be different. For example, for the phrase "於憍薩羅國" (*yu-jiao-sa-luo-guo*, in Kosala state), if we want to identify the actual transliteration, "憍薩羅" (*jiao-sa-luo*, Kosala), from the extra characters "於" (*yu*, in) and "國" (*guo*, state), we must first use an effective feature to identify the boundaries of the transliteration.

In order to identify the boundaries of transliterations, we propose a language-model-based feature. A language model assigns a probability to a sequence of m words $P(w_1, w_2, \ldots, w_m)$ by means of a probability distribution. The probability of a sequence of m words can be transformed

into a conditional probability:

$$\begin{aligned} P(w_1, w_2, \cdots, w_m) &= P(w_1)P(w_2|w_1) \\ &\quad P(w_3|w_1, w_2) \cdots \\ &\quad P(w_m|w_1, w_2, \cdots, \\ &\quad w_{m-1}) \\ &= \prod_{i=1}^{m} P(w_i|w_1, w_2, \cdots, \\ &\quad w_{i-1}) \end{aligned}$$

In practice, we can assume the probability of a word only depends on its previous word (bi-gram assumption). Therefore, the probability of a sequence can be approximated as:

$$\begin{aligned} P(w_1, w_2, \cdots, w_m) &= \prod_{i=1}^{m} P(w_i|w_1, w_2, \\ &\quad \cdots, w_{i-1}) \\ &\approx \prod_{i=1}^{m} P(w_i|w_{i-1}) \end{aligned}$$

We collect person and location names from the Buddhist Authority Database[1] and the known Buddhist transliteration terms from The Buddhist Translation Lexicon (翻譯名義集)[2] to create a dataset with 4,301 transliterations for our bi-gram language model.

After building the bi-gram language model, we apply it as a feature for the supervised model. Following the previous example, "於憍薩羅國" (*yu-jiao-sa-luo-guo*, in Kosala state), for each character in the sentence, we first compute the probability of the current character and its previous character. For the first character "於" , since there is no previous word, the probability is $P(於)$. For the second character "憍", the probability of the two characters is $P(於憍) = P(於)P(憍|於)$. We then compute the probability of the second and third characters: $P(憍薩) = P(憍)P(薩|憍)$, and so on. If the probability changes sharply from that of the previous bi-gram, the previous bi-gram may be the boundary of the transliteration. Because the character "於" rarely appears in transliterations, $P(於憍)$ is much lower than $P(憍薩)$. We may conclude that the left boundary is between the first two characters "於憍".

2.3 Functional Words

We take the classical Chinese functional words into consideration. These characters have special grammatical functions in classical Chinese; thus, they are seldom used to transliterate foreign names. This is a binary feature which records the character is a functional word or not. The functional words are listed as follows: 之 (*zhi*), 乎 (*hu*), 且 (*qie*), 矣 (*yi*), 邪 (*ye*), 於 (*yu*), 哉 (*zai*), 相 (*xiang*), 遂 (*sui*), 嗟 (*jie*), 與 (*yu*), and 噫 (*yi*).

2.4 Appellation and Quantifier Words

After observing the transliterations appearing in classical Chinese literature, we note that there are some specific patterns of the characters follows the transliteration terms. Most of the characters following the transliteration are appellation or quantifier words, such as 山 (*san*, mountain), 海 (*hai*, sea), 國 (*guo*, state), 洲 (*zhou*, continent). For example, there are some cases like 耆闍崛山 (*qi-du-jui-san*, Vulture mountain), 拘薩羅國 (*jü-sa-luo-guo*, Kosala state), and 贍部洲 (*zhan-bu-zhou*, Jambu continent). Therefore, we collect the Chinese characters that are usually used as appellation or quantifiers following transliterations and then design this feature. This is also a binary feature that records the character is used as an appellation or quantifier word or not.

2.5 CRF Model Training

We adopt the supervised learning models, conditional random field (CRF) (Lafferty et al., 2011), to extract the transliterations in classical Buddhist texts. For CRF model, we formulate the transliteration extraction problem as a sequential tagging problem.

2.5.1 Conditional Random Fields

Conditional random fields (CRFs) are undirected graphical models trained to maximize a conditional probability (Lafferty et al., 2011). A linear-chain CRF with parameters $\Lambda = \lambda_1, \lambda_2, \ldots$ defines a conditional probability for a state sequence $\mathbf{y} = \mathbf{y}_1 \ldots \mathbf{y}_T$, given that an input sequence $\mathbf{x} = \mathbf{x}_1 \ldots \mathbf{x}_T$ is

$$P_\Lambda(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_\mathbf{x}} \exp\left(\sum_{t=1}^{T}\sum_{k} \lambda_k f_k(\mathbf{y}_{t-1}, \mathbf{y}_t, \mathbf{x}, t)\right)$$

where $Z_\mathbf{x}$ is the normalization factor that makes the probability of all state sequences sum to one; $f_k(\mathbf{y}_{t-1}, \mathbf{y}_t, \mathbf{x}, t)$ is often a binary-valued feature function and λ_k is its weight. The feature functions can measure any aspect of a state transition, $\mathbf{y}_{t-1} \rightarrow \mathbf{y}_t$, and the entire observation sequence,

[1] http://authority.ddbc.edu.tw/

[2] http://www.cbeta.org/result/T54/T54n2131.htm

$\mathbf{x}$, centered at the current time step, t. For example, one feature function might have the value 1 when $\mathbf{y}_{t-1}$ is the state *B*, $\mathbf{y}_t$ is the state *I*, and $\mathbf{x}_t$ is the character "國" (*guo*). Large positive values for λ_k indicate a preference for such an event; large negative values make the event unlikely.

The most probable label sequence for $\mathbf{x}$,

$$\mathbf{y}^* = \arg\max_y P_\Lambda(\mathbf{y}|\mathbf{x})$$

can be efficiently determined using the Viterbi algorithm.

2.5.2 Sequential Tagging and Feature Template

The classical Buddhist texts are separated into sentences by the Chinese punctuation. Then, each character in the sentences is taken as a data row for CRF model. We adopt the tagging approach motivated by the Chinese segmentation (Tsai et al., 2006) which treat Chinese segmentation as a tagging problem. The characters in a sentence are tagged in **B** class if it is the first character of a transliteration word or in **I** class if it is in a transliteration word but not the first character. The characters that do not belong to a transliteration words are tagged in **O** class. We adopt the CRF++ open-source toolkit[3]. We train our CRF models with the unigram and bigram features over the input Chinese character sequences. The features are shown as follows.

- Unigram: s_{-2}, s_{-1}, s_0, s_1, s_2
- Bigram: $s_{-1}s_0$, s_0s_1

where current substring is s_0 and s_i is other characters relative to the position of the current character.

3 Evaluation

3.1 Data set

We choose one Buddhist scripture as our data set for evaluation from the Chinese Buddhist Canon maintained by Chinese Buddhist Electronic Text Association (CBETA). The scripture we choose to compile the training and test sets is the Samyuktagama (雜阿含經). The Samyuktagama is one of the most important scriptures in Early Buddhism and contains a lot of transliterations because it detailedly records the speech and the lives of the Buddha and many of his diciples.

[3] http://crfpp.googlecode.com

The Samyuktagama is an early Buddhist scripture collected shortly after the Buddha's death. The term agama in Buddhism refers to a collection of discourses, and the name Samyuktagama means "connected discourses." It is among the most important sutras in Early Buddhism. The authorship of the Samyuktagama is traditionally regarded as the most early sutra collected by the Mahakssyapa, the Buddha's disciple, and five hundred Arhats three months after the Buddha's death. An Indian monk, Gunabhadra, translated this sutra into classical Chinese in Liu Song dynasty around 443 C.E. The classical Chinese Samyuktagama has 50 volumes containing about 660,000 characters. Because the amount of Samyuktagama is too tremendous, we take the first 20 volumes as the training set, and the last 10 volumes as the test set.

In addition, we want to evaluate if the supervised learning model trained by one Buddhist scripture can be applied to another Buddhist scrupture translated in different era. Therefore, we choose another scripture, the Lotus Sutra (妙法蓮華經), to create another test set. The Lotus sutra is a famous Mahayana Buddhist scripture probably written down between 100 BC and 100 C.E. The earliest known Sanskrit title for the sutra is the Saddharma Pundarika Sutra, which translates to "the Good Dharma Lotus Flower Sutra." In English, the shortened form Lotus Sutra is common. The Lotus Sutra has also been highly regarded in a number of Asian countries where Mahayana Buddhism has been traditionally practiced, such as China, Japan, and Korea. The Lotus Sutra has several classical Chinese translation versions. The most widely used version is translated by Kumarajiva ("鳩摩羅什" in Chinese) in 406 C.E. It has eight volumes and 28 chapters containing more then 25,000 characters. We select the first 5 chapters as a different test set to evaluate our method.

3.2 Baseline Method

There are a few reseaches focusing on transliteration extraction from classical Chinese literature. However, in order to compare and show the benefits of our method, we construct a baseline system with widely used information extraction methods. Because many previous researches on transliteration extraction are based on phonetic similarity or phoneme mapping approaches, we also use these methods to construct the baseline system. First,

Table 1: Evaluation Results of Tranliteration Extraction

		Precision	Recall	F_1-score
Our Approach	The Samyuktagama test set	0.8810	0.9561	0.9170
	The Lotus Sutra test set	0.9444	0.9474	0.9459
Baseline	The Samyuktagama test set	0.0399	0.7771	0.0759
	The Lotus Sutra test set	0.0146	0.5789	0.2848

the baseline system use the suffix array method to extract all the possible terms for the classical Chinese Buddhsit scriptures. Then, the extracted terms are converted into Pinyin sequences by a modern Chinese pronunication dictionary. We also adopt the collected transliteation list used in section 2.1 and also convert the transliterations into Pyinyin sequences. Next, for each extracted terms, the baseline system measures the Levenshtein distance between the Pinyin sequences of the extracted terms and all the transliterations as the phonetic similairy. If the extracted term has a Levenshtein distance less than threshold (distance $\leq$ 3 in our baseline) from one of the transliterations we collect, the extracted term will be regarded as a transliteration; otherwise, the term will be dropped.

3.3 Evaluation Metrics

We use two evaluation metrics, recall and precision, to estimate the performance of our system. Recall and precision are widely used measurements in many research fields, sucn as information retrieval and information extraction. (Manning et al., 2008) In the digital humanities research field, a key issue is the coverage of the extraction method. To maximize usefulness to researchers, a method should be able to extract as many potential transliterations from literature as possible. Therefore, in our evaluation, we use recall, defined as follows:

$$Recall = \frac{|\text{Correctly extracted transliterations}|}{|\text{Transliterations in the data set}|}$$

In addition, the correctness of the extracted transliterations are also important. To avoid wasting time on the useless information, a method should be able to extract correct transliterations from literature as possible. Thus, we also use precision, defined as follows:

$$Precision = \frac{|\text{Correctly extracted transliterations}|}{|\text{All extracted transliterations}|}$$

With precision and recall, the F-score measurement is also adopted as a weighted average of the precision and recall. The F_1-score is defineds as follows:

$$F_1\text{-score} = \frac{2 \times recision \times recall}{precision + recall}$$

3.4 Evaluation Results

Table 1 shows the results of our method and the baseline system on different test sets. The gold standards of these two test sets are compiled by human experts who examine all the sentences in the test sets and regconize each transliterations for evaluation. The results show that our method can extract 95.61% transliterations on the Sumyuktagama and 94.74% on the Lotus Sutra. On the precision measurement, our method also achieves pretty good results, which show that most of the terms our method extract are actual transliterations. Our method outperforms the baseline system and the precision of the baseline system is very poor. The baseline system cannot extract most transliterations due to the limit of the suffix array method since the suffix array method only extracts the terms that appear twice or more in the context. Besides, the phonetic similarity is not effective to filter the transliteartions; the problem causes the low precision. These results demonstrate that our method can save a lot of labor-intensive work to examine the transliteration for the historical and humanity researchers.

4 Discussion

4.1 Effectiveness of transliteration extraction

Our method can extract many transliterations from the Samyuktagama such as "迦毘羅衛" (*jia-pi-luo-wei*, *Kapilavastu*, the name of an ancient kingdom where the Buddha was born and grew up), "尼拘律" (*ni-jü-lü*, *Nyagro*, the forest name in Kapilavastu kingdom), and "摩伽陀" (*muo-qie-tuo*, *Magadha*, the name of an ancient Indian kingdom). These transliteration do not appear in the training set, but our method can still identify them. In addition, our method also finds out many transliterations in the Lotus Sutra which

are unseen in the Samyuktagama, such as "娑伽羅" (*suo-qie-luo*, *Sagara*, the name of the king of the sea world in ancient Indian mythology), "鳩槃荼/鳩槃茶" (*jiu-pan-cha*/*jiu-pan-tu*, *Kumbhanda*, one of a group of dwarfish, misshapen spirits among the lesser deities of Buddhist mythology), and "阿鞞跋致" (*a-pi-ba-zhi*, *Avaivart*, "not turn back" in Sanskrit). Since the characteristics of the Lotus Sutra are different from the Samyuktagama in many aspects, it shows that the supervised learning model trained by one Buddhist scripture may apply to other Buddhist scriptures translated in different eras and translators.

We also discovered that transliterations may vary even in the same scripture. In the Samyuktagama, the Sanskrit term "Chandala" (someone who deals with disposal of corpses, and is a Hindu lower caste, formerly considered untouchables) has two different transliterations: "旃陀羅" (*zhan-tuo-luo*) and "栴陀羅" (*zhan-tuo-luo*). The Sanskrit term "*Magadha*" (the name of an ancient Indian kingdom) has three different transliterations: "摩竭陀" (*muo-jie-tuo*), "摩竭提" (*muo-jie-ti*), and "摩伽陀" (*muo-qie-tuo*). The variations of the transliterations of the same word give the clues of translators and translation progress. These variations may help the study of historical Chinese phonology and philology.

4.2 Error cases

Although our method can extract and identify most transliteration pairs, some transliteration pairs cannot be identified. The error cases can be divided into several categories. The first one is that a few terms cannot be extracted, such as "闍維" (*she-wei*, *Jhapita*, cremation, a monk's funeral pyre). This transliteration is less used and only appears three times in the final part of the Samyuktagama. The widely used transliteration of the term "*Jhapita*" is "荼毘" (*tu-pi*). It may cause the difficulty for the supervised learning model to identify these terms.

The other case is incorrect boundary of the transliterations. Sometimes our method may extract shorter terms, such as "韋提" (*wei-ti*, correct transliteration is "韋提希", *wei-ti-xi*, *Vaidehi*, a female person name), "波羅" (*po-luo*, correct transliteration is "波羅柰", *po-luo-nai*, *Varanasi*, a location name in northen India), "瞿利摩羅" (*qü-li-muo-luo*, correct transliteration is "央瞿利摩羅", *yang-qü-li-muo-luo*, *Angulimala*, one of the Buddha's disciples). This problem is due to the probability generated by the language model. For example, the probability of the first two charactgers of the transliteration "央瞿利摩羅", $P(\text{央瞿})$, is very low. It causes the CRF model predicts the first character "央" (*yang*) does not belong to the transliteration. If more transliterations can be collected to build a better language model, this problem can be overcome.

In some cases, our method extracts much longer terms, like "阿那律陀夜" (*a-na-lü-tuo-ye*, correct transliteration is "阿那律陀", *a-na-lü-tuo*, *Aniruddha*, one of the Buddha's closest disciples), and "兒富那婆藪" (*er-fu-na-po-sou*, correct transliteration is "富那婆藪", *fu-na-po-sou*, *Punabbasu*, a kind of ghost in Buddhist mythology). In these cases, the previous or following characters are often used in transliterations. Therefore, it is very difficult to distinguish the boundary of the actual transliteration. In addition, there are some cases that a transliteration followed by another transliteration immediately. For example, our method extracts out the term "闡陀舍利" (*chan-tuo-she-li*), which comprises two transliteration terms such as "闡陀" (*chan-tuo*, *Chanda*, one of the Buddhist's disciples) and "舍利" (*she-li*, *Sarira*, Buddhist relics). It is also difficult to separate them without any additional semantic clues. Although our method sometimes might extract incomplete transliterations with incorrect boundary, checking the boundary of a transliteration is not difficult to a human expert. Therefore, the extracted incorrect transliteartions also have the benefits to help humanity researchers quickly find and check plausible transliterations.

5 Conclusion

The transliteration extrction of foreign loanwords is an important task in research fields such as historical linguistics and digital humanities. We propose an approach which can extract transliteration automatically from classical Chinese Buddhist scriptures. Our approach comprises the conditional random fields method with designed features which are suitable to identify transliteration characters. The first feature is the probability of each Chinese character used in transliterations. The second feature is probability of the sequential bigram characters measured by the language model method. In addiition, the functional words, appellation and quantifier words also be regarded

as binary features. Next, the transliteration extration problem is formulated as a sequential tagging problem and the CRF method is used to train a model to extract the transliterations from the input classical Chinese sentences. To evaluate our method, we constructed an evaluation set from the two Buddhist texts, the Samyuktagama and the Lotus Sutra, which were translated into Chinese in different eras. We also construct a baseline system with proach with suffix array based extraction method and phonetic similarity measurementfor comparison. The recall of our method achieves 0.9561 and the precision is 0.9444. The results show our method outperforms the baseline system a lot and is effective to extract transliterations from classical Chinese texts. Our method can find the transliterations among the immense classical literatures to help many research fields such as historical linguistics and philology.

References

Y. Goldberg and M. Elhadad. 2008. Identification of transliterated foreign words in hebrew script. *Computational Linguistics and Intelligent Text Processing*.

J-S. Kuo, H. Li, and Y-K. Yang. 2007. A phonetic similarity model for automatic extraction of transliteration pairs. *ACM Trans. Asian Language Information Processing*, 6(2).

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2011. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proc. of ICML*, pages 282–289.

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*, volume 1. Cambridge University Press Cambridge.

G. Manzini and P. Ferragina. 2004. Engineering a lightweight suffix array construction algorithm. *Algorithmica*, 40(1):33–50.

J. Oh and K. Choi. 2003. A statistical model for automatic extraction of korean transliterated foreign words. *International Journal of Computer Processing of Oriental Languages*, 16(1):41–62.

L. Rabiner. 1989. tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77.

T. Sherif and G. Kondrak. 2007. Bootstrapping a stochastic transducer for arabic-english transliteration extraction. *Proceedings of Annual Meeting- Association for Computational Linguistics*.

Richard Tzong-Han Tsai, Hsieh-Chuan Hung, Cheng-Lung Sung, Hong-Jie Dai, and Wen-Lian Hsu. 2006. On closed task of chinese word segmentation: An improved crf model coupled with character clustering and automatically generated template matching. *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 134–137.

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

Dan Han[1,2] **Pascual Martínez-Gómez**[2,3] **Yusuke Miyao**[1,2]
Katsuhito Sudoh[4] **Masaaki Nagata**[4]
[1]The Graduate University For Advanced Studies
[2]National Institute of Informatics, [3]The University of Tokyo
[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. Rearranging 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.

27th Pacific Asia Conference on Language, Information, and Computation pages 267－276

Vb-H	VV VE 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

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

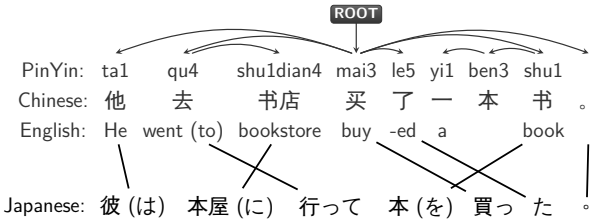


Figure 1: Example of unlabeled dependency parse tree of a Chinese sentence (SVO) with word aligned to its Japanese counterpart (SOV). Arrows are pointing from heads to dependents.

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).

Figure 1 shows an example of unlabeled dependency parse tree of a Chinese sentence aligned with its Japanese translation. According to the reordering method, "went" will be reordered behind of "bookstore" while "buy -ed" will be reordered to the right-hand side of "book", and thus the sentence will follow a SOV word order as Japanese. However, if "book" was wrongly recognized as the dependent of "went" in the dependency structure, "went" will be wrongly reordered to the right-hand side of "book". Therefore, syntactic structure based reordering methods highly rely on the parsing accuracy. In order to further improve word alignments or refine existing reordering models, it 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 Màrquez (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

	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	868	100	2,643
AL	29.8	20.0	33.5	28.4	25.9	29.8
Voc.	5.5K	690	5K	5.1K	972	9.5K

Table 2: Statistics of selected sentences in five genres of CTB-7. AL stands for the average length of sentences, while Voc. for vocabulary.

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

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

(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.

4.2 Evaluation

We use Kendall's tau (τ) rank correlation coefficient (Isozaki et al., 2010) to measure word order similarities between sentences in two different scenarios. In the first scenario, we use the set of manually reordered Chinese sentences from set-1 as benchmark and compare it with the set of automatically reordered Chinese sentences. In the second scenario, we combine set-1 and set-2 to obtain a larger data set. The set of Japanese references plays the role of benchmark and is compared with the set of automatically reordered Chinese sentences. Word alignments are produced by MGIZA++ (Gao and Vogel, 2008).

In both scenarios, we carry out the reordering method DPC (See Section 2.1). Auto-parse trees are generated by an unlabeled Chinese dependency parser, Corbit[4] (Hatori et al., 2011). Gold trees[5] are converted from CTB-7 parsed text which are created by human annotators. More specifically, we refer to auto-parse tree based reordering system as Auto-DPC and to gold-tree based reordering system as Gold-DPC. Baseline system uses unreordered Chinese sentences.

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 τ value.

For baseline system, the average τ 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 τ 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 τ value of Auto-DPC shows that DPC is sensitive on parsing errors. This assumption is also confirmed by the average τ value between Auto-DPC and Gold-DPC. However, the difference of τ 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

[3]http://stp.lingfil.uu.se/ nivre/research/Penn2Malt.html

[4]http://triplet.cc/software/corbit

[5]Note that Corbit was tuned with the development set of CTB-7.

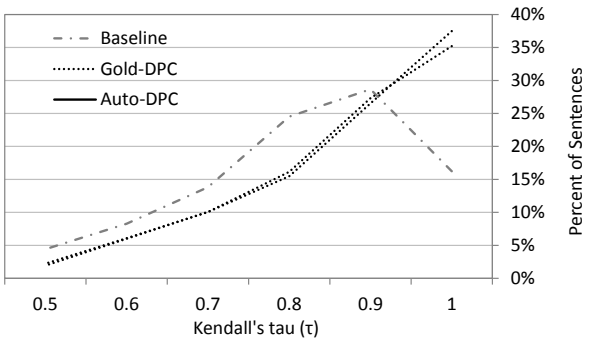


Figure 2: The distribution of Kendall's tau values for $2,236$ bilingual sentences (Chinese-Japanese) in which the Chinese is from three systems of baseline, Auto-DPC, and Gold-DPC.

file, `ch-ja.A3.final`. The comparison implies how monotonically the Chinese sentences have been reordered to align with Japanese. We use MeCab[6] (Kudo and Matsumoto, 2000) to segment Japanese sentences and also filter out sentences with more than 64 tokens. There are $2,236$ valid Chinese-Japanese bilingual sentences in total. Figure 2 shows the distribution of Kendall's tau from three systems in which the baseline is built up by using ordinary Chinese.

In Figure 2, baseline system contains a large numbers of non-monotonic aligned sentences, whereas both Auto-DPC and Gold-DPC increased the amount of sentences that achieved high τ values. Reordering based on gold-tree reduced more percentage of low τ sentences than reordering based on automatically parsed trees. Especially, the amount of sentence difference in $0.9 < \tau <= 1$ between Gold-DPC and Auto-DPC shows that reordering method DPC has a high sensitivity on parsing errors, which enhances the conclusions from the preliminary observation in scenario 1. Furthermore, the performance of reordering system Gold-DPC sketches the figure of upper bound of the reordering method.

5 Analysis on Causes of Reordering Errors

Preliminary experiments in Section 4 provide a general idea of the effects of parsing errors on reordering. In order to achieve more explicit relationship between specific parsing errors and reordering issues, we first identify concrete parsing errors by comparing gold-trees with auto-parse 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.

[6] http://mecab.googlecode.com

Figure 3: A possible wrong dependency parse tree of the example in Figure 1.

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

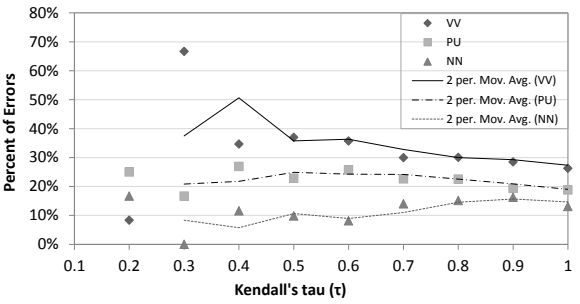


Figure 4: The distribution of top three dependent-error POS tags and their tendency lines.

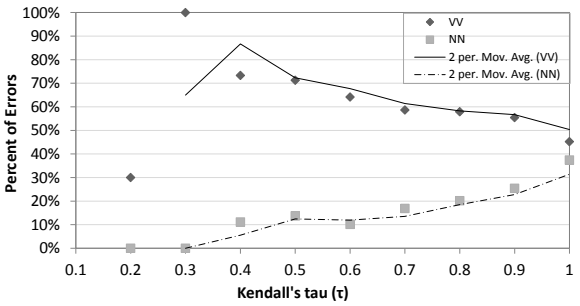


Figure 5: The distribution of top two head-error POS tags and their tendency lines.

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[7] 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[8]. 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 reorders necessary words. Thus, DPC reorders 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[9]. 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 "$\surd$", "$\times$", "?" represent the status of a cer-

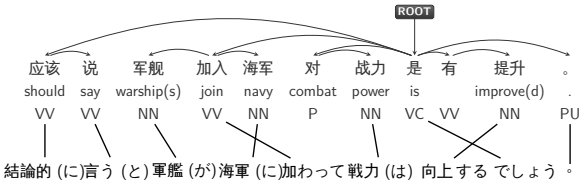
[7] A Chinese character expresses possession and existence.

[8] A Chinese character is specially used to connect the verb phrase and its modifier.

[9] We 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).

	BEI		ROOT		RM-D	
	GT	CT	GT	CT	GT	CT
ROOT Error						
Root-C	×	×	×	√	×	×
Root-G	×	×	√	×	×	×
RM-D Error						
RM_D-C	×	×	×	×	×	√
	×	×	×	√	×	√
	×	×	√	×	×	√
	×	×	√	√	×	√
RM_D-G	×	×	×	×	√	×
	×	×	×	√	√	×
	×	×	√	×	√	×
	×	×	√	√	√	×
RM_D-D	×	×	×	×	√	diff.
	×	×	×	√	√	diff.
	×	×	√	×	√	diff.
	×	×	√	√	√	diff.
BEI Error						
BEI-C	×	√	√	?	×	?
	×	√	×	?	√	?
	×	√	√	?	√	?
BEI-G	√	×	?	√	?	×
	√	×	?	×	?	√
	√	×	?	√	?	√

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.



(a) Gold tree

(b) A possible wrong parse tree.

Figure 6: An example for parsing error patterns of Root-C and RM_D-D. English translation: One should say that, the additions of warships will help to improve the navy's combat power.

tain dependency type in gold-tree or Corbit-tree. For every Vb-H candidate, the 6 status are conditions to match the error pattern. For example, to match a Root-C error pattern, the Vb-H candidate needs to satisfy the following conditions: in gold-tree, it is not the root, and does not have any RM-D or bei dependent; in Corbit tree, it does not have any RM-D or bei dependent, but it is the root.

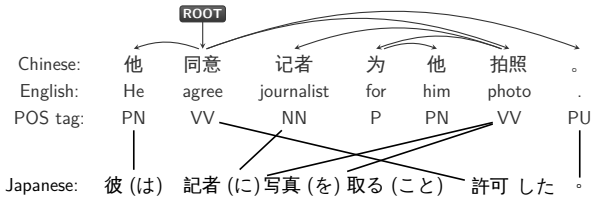
Root-C is the case where a Vb-H candidate has been wrongly parsed as the root of the sentence. However, it only affects the reordering with two constrains, namely that RM-D of the Vb-H candidate does not exist and Vb-H is not involved in a bei-construction. For instance, the Vb-H "should" in the example of Figure 6 was recognized as root in auto-parse tree in Figure 6b. However, the actual root is the Vb-H "is" in gold tree of Figure 6a. Therefore, since "should" does not have any dependent as either BEI or RM-D in both GT and CT, it will be reordered incorrectly to the end of 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



(a) Gold-tree.

(b) A possible erroneous parse tree.

(c) Another possible erroneous parse tree.

Figure 7: An example for parsing error patterns of Root-G and RM_D-C. English translation: He agreed to the journalist to take a picture of him.

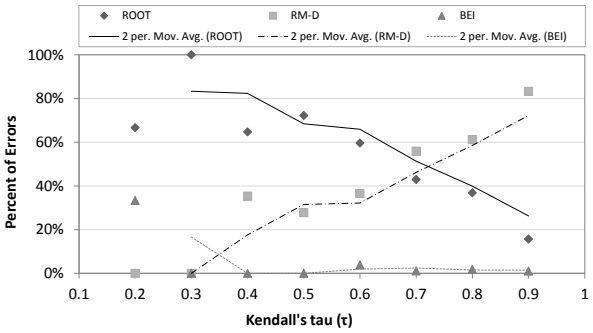


Figure 8: Distribution of three types of parsing errors in different τ groups and their trend curves.

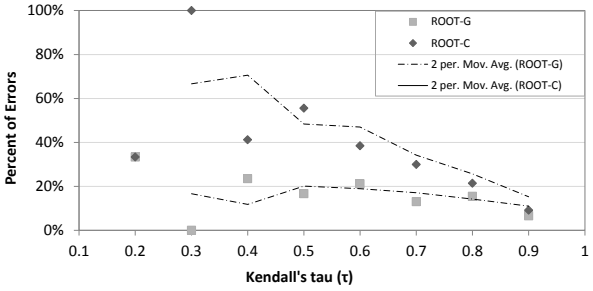


Figure 9: Distribution of patterns of ROOT error in different τ groups and their trend curves.

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 τ value groups of sentences.

Figure 8 shows the distribution of the three types of parsing errors and their tendencies. In low τ value sentences, there are higher proportions of ROOT errors, and relatively lower proportions in high τ 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.

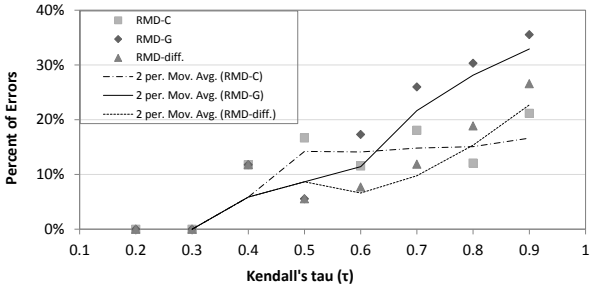


Figure 10: The distribution of different patterns of RM-D error in different τ groups.

6 Discussion and Future Work

Two important research directions concentrate on either improving parsers or developing linguistically motivated pre-reordering methods. We believe that analyzing the link between those directions can help us to refine future developments.

We observed relatively small effects on reordering quality in response of parsing errors. However, reordering quality affect word alignments, which in turn affect the quality of bilingual phrases that are extracted. It would be interesting to extend this work to quantify the propagation of parsing and reordering errors in SMT pipelines, to observe the factored effect on the overall MT quality.

We found that not all POS tagging and parsing errors correlate equally with reordering quality. In the case of DPC reordering method, mis-recognitions of VV words correlate with low reordering performance, whereas mis-recognitions of NN words had a smaller impact. Indeed, DPC heavily relies on detecting verbal blocks that are candidates for reordering, and systems that use the same strategy should choose POS taggers that display high accuracy of VV recognition.

One of the key characteristics of DPC is its ability to correctly reorder sentences with reported speech constructions. For that purpose, it is crucial for parsers to recognize the sentence root, and our analysis demonstrated that systems that follow similar strategy should rely on parsers that have a high accuracy to recognize the sentence root.

In general, we believe that future developments of syntax-based pre-reordering methods would benefit of preliminary analysis of POS tagging and parsing accuracies. In case of linguistically motivated pre-reordering methods, reordering rules could be designed to be more robust against unreliable POS tags or unreliable dependency relations. For automatically learned reordering rules, those systems could be designed to make use of N-best lists of certain POS tags or dependencies that are critical but that parsers cannot reliably provide.

There are other popular syntax-based pre-reordering methods that may use different types of parsing grammars (i.e. Head-driven phrase structure grammar), and similar analysis would also be interesting in those contexts, possibly with a larger set of gold parsed and reordered sentences. Additionally, researchers interested in developing POS taggers and parsers with the objective to aid pre-reordering could attempt to maximize the accuracy of POS tags or dependencies that are relevant to the reordering task, maybe at the expense of lower accuracies on other elements.

7 Conclusion

In this work, we carried out linguistically motivated 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 specific 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 reordering 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 approximation 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 dependency 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 relationship 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 Lluis Màrquez. 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.

Jun Hatori, Takuya Matsuzaki, Yusuke Miyao, and Junichi Tsujii. 2011. Incremental joint POS tagging and dependency parsing in Chinese. In *Proc. of 5th IJCNLP*, pages 1216–1224.

Hideki Isozaki, Katsuhito Sudoh, Hajime Tsukada, and Kevin Duh. 2010. Head finalization: A simple reordering rule for SOV languages. In *Proc. of WMTMetricsMATR*, pages 244–251.

Jason Katz-Brown, Slav Petrov, Ryan McDonald, Franz Och, David Talbot, Hiroshi Ichikawa, and Masakazu Seno. 2011. Training a parser for machine translation reordering. In *Proc. of the 2011 Conference on EMNLP*, pages 183–192.

Taku Kudo and Yuji Matsumoto. 2000. Japanese dependency structure analysis based on support vector machines. In *Proc. of the EMNLP/VLC-2000*, pages 18–25.

Ryan McDonald and Joakim Nivre. 2007. Characterizing the errors of data-driven dependency parsing models. In *Proc. of the 2007 Joint Conference on EMNLP-CoNLL*, pages 122–131.

Yusuke Miyao and Jun'ichi Tsujii. 2008. Feature forest models for probabilistic HPSG parsing. *Computational Linguistics*, 34:35–80.

Carl Jesse Pollard and Ivan A. Sag. 1994. *Head-driven phrase structure grammar*. The University of Chicago Press and CSLI Publications.

Chris Quirk and Simon Corston-Oliver. 2006. The impact of parse quality on syntactically-informed statistical machine translation. In *Proc. of EMNLP*, pages 62–69.

Yiou Wang, Junichi Kazama, Yoshimasa Tsuruoka, Wenliang Chen, Yujie Zhang, and Kentaro Torisawa. 2011. Improving Chinese word segmentation and POS tagging with semi-supervised methods using large auto-analyzed data. In *Proc. of 5th IJCNLP*, pages 309–317.

Fei Xia and Michael McCord. 2004. Improving a statistical MT system with automatically learned rewrite patterns. In *Proc. of COLING*.

Peng Xu, Jaeho Kang, Michael Ringgaard, and Franz Och. 2009. Using a dependency parser to improve SMT for subject-object-verb languages. In *Proc. of NAACL*, pages 245–253.

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.