Empirical Dependency-Based Head Finalization for Statistical Chinese-, English-, and French-to-Myanmar (Burmese) Machine Translation

Chenchen Ding[†], *Ye Kyaw Thu*[‡], *Masao Utiyama*[‡], *Andrew Finch*[‡], *Eiichiro Sumita*[‡]

[†]Department of Computer Science, University of Tsukuba
1-1-1 Tennodai, Tsukuba, Ibaraki, 305-8573, Japan
[‡]Multilingual Translation Laboratory,
National Institute of Information and Communications Technology
3-5 Hikaridai, Seikacho, Sorakugun, Kyoto, 619-0289, Japan
[†]tei@mibel.cs.tsukuba.ac.jp
[‡]{yekyawthu, mutiyama, andrew.finch, eiichiro.sumita}@nict.go.jp

Abstract

We conduct dependency-based head finalization for statistical machine translation (SMT) for Myanmar (Burmese). Although Myanmar is an understudied language, linguistically it is a head-final language with similar syntax to Japanese and Korean. So, applying the efficient techniques of Japanese and Korean processing to Myanmar is a natural idea. Our approach is a combination of two approaches. The first is a head-driven phrase structure grammar (HPSG) based head finalization for English-to-Japanese translation, the second is dependency-based pre-ordering originally designed for English-to-Korean translation. We experiment on Chinese-, English-, and French-to-Myanmar translation, using a statistical pre-ordering approach as a comparison method. Experimental results show the dependency-based head finalization was able to consistently improve a baseline SMT system, for different source languages and different segmentation schemes for the Myanmar language.

1. Introduction

The state-of-the-art techniques of statistical machine translation (SMT) [1, 2] demonstrate good performance on translation of languages with relatively similar word orders [3]. However, word reordering is a problematic issue for language pairs with significantly different word orders, such as the translation between a subject-verb-object (SVO) language and a subject-object-verb (SOV) language [4].

To resolve the word reordering problem in SMT, a line of research handles the word reordering as a separate preprocess, which is referred as *pre-ordering*. In pre-ordering, the word order on source-side is arranged into the target-side word order, before a standard SMT system is applied, on both training and decoding phases. The pre-ordering process can be realized in either a rule-based way or a statistical way. Generally, a rule-based approach needs a high-precision parser and effective manually designed rules; and a statistical approach needs data for model training.

An effective rule-based approach, *head finalization* has been proposed for English-to-Japanese translation [4]. The approach takes advantage of the *head final* property of Japanese on the target-side. It designs a head finalization rule to move the head word based on the parsing result by a head-driven phrase structure grammar (HPSG) parser. Generally, the idea can be applied to other SVO-to-Japanese translation tasks, such as its application in Chinese-to-Japanese translation [5]. However, an HPSG parser is not available for many languages, which prevents the HPSG-based head finalization from being applied to more languages. On the other hand, dependency parsers are available for more languages. A typical rule-based pre-ordering using dependency structure was proposed in [6]. Their approach used a rule set to arrange the order of a head word together with its modifiers.

In this paper, we explore dependency-based head finalization for an understudied language, Myanmar[1]. We use the dependency structure to realize the head finalization of [4]. Because the head finalization only moves a head word after all its modifiers, the proposed dependency-based head finalization is a simplified version of [6], which keeps the order of modifiers unchanged. So, our approach is simple and widely applicable for different source languages. On the target-side, there are no standard part-of-speech set and morpheme analysis tools available for Myanmar word segmentation yet, so we employ two word segmentation schemes: syllable-based and dictionary-based maximum matching. Experiments on Chinese-, English-, and French-to-Myanmar translation show that simple head finalization can efficiently and stably improve a baseline SMT system, no matter what the source-side language is or which segmentation scheme is used. We use a statistical pre-ordering approach [7] as a comparison method. We observe it performs well on certain situations, but it is sensitive to the source-side language and segmentation schemes.

[1]The language may be more referred as *Burmese* in English though, in this paper, we refer it consistently as *Myanmar*.

Figure 1: *Example of a Myanmar sentence " သူသည် စာအုပ်ကို ဆရာအား ပေးသည် " (English translation "he gives the book to the teacher"). The first row shows the morphemes in the Myanmar sentence, one-box-one-morpheme. Content morphemes are illustrated in black and functional morphemes are in gray. The second row is the English literal translations of them. In the two lower rows, Japanese and Korean translations of the Myanmar sentence are also shown, morpheme-by-morpheme. Both the Japanese and Korean sentences are grammatically correct, from which the syntactic similarity can be observed. The right-most boxes in Japanese and Korean sentences, which contain the verbs, should be noticed. The corresponding parts of Myanmar present marker in these two languages are inflection endings which cannot be detached from the verb stems (marked by gray, in the case of Korean, more correctly, the " ㄴ다 " part). While Myanmar has a completely detachable marker from the verb stem.*

This paper is organized as follows. In Section 2, we give an introduction to the Myanmar language. In Section 3, we discuss related approaches. In Section 4, we describe the proposed approach. In Section 5, we show the experimental results and present a discussion. Section 6 contains the conclusion and future work.

2. Myanmar Language

Myanmar is an SOV language that demonstrates a consistent head-final typology. Syntactically, Myanmar is quite similar to Japanese and Korean, where functional morphemes succeed content morphemes, and verb phrases succeed noun phrases. We show an example in Fig. 1 to show the features of Myanmar and its similarity to Japanese and Korean.

On the other hand, unlike Japanese and Korean, which are typical agglutinative languages, Myanmar is an analytic language, in which the morphemes are without inflection. This is because Myanmar is a monosyllabic language originally, where morphemes are only composed by non-inflected single syllables. Although Buddhism-related loanwords from the Pali language and modern loanwords from western languages have introduced polysyllabic morphemes into Myanmar, the basic framework of syntax has not been affected.

3. Related Work

As mentioned, Myanmar is an understudied language that has quite similar (or, even simpler) characteristics to Japanese and Korean, both of which are well studied. A natural idea is that we can transfer the Japanese or Korean language processing techniques to Myanmar.

HPSG-based head finalization [4] and dependency-based pre-ordering [6] are two typical rule-based pre-ordering approaches. Originally, the former was designed for Japanese and the latter for Korean. Further differences between the two approaches first lies in the linguistic formulation they used, which leads to differences in their rule sets. Essentially, there is only *one* rule in the HPSG-based head finalization, that is the *head finalization rule* itself. The simplicity of the rule set can be attributed to the sophisticated analysis by an HPSG parser, which shows the phrase structural as well as the syntactic head. On the other hand, the rule set in [6] contains about 20 rules, in order to arrange the position of a head word with its modifiers. It can be observed that a good HPSG parser is required for [4] if we want to expand the approach to more source-side languages, despite the simple rule. While a dependency parser is available for more languages, the rule set in [6] is dependent on the part-of-speech (POS) tag set and dependency arc label set of the dependency parser used. The approach used in our experiments combines the simplicities of the two previous approaches. We use dependency parsers to conduct the head finalization alone without touching the arrangement of various types of modifiers.

There are also statistical pre-ordering approaches. The work of [8, 9] are early syntax-oriented approaches, that they introduce separate reordering modules into SMT systems. Recently, the approach in [10] learns pre-ordering automatically from an aligned corpus. This approach achieves nearly same performance as the rule-based approach of [6] (Table 4 in [10]). In [7], a method to learn a discriminative parser for pre-ordering from an aligned parallel training corpus is proposed. The approach takes the derivation tree as a latent variable and trains a model to maximize reordering measures. The approach is fully unsupervised but needs high quality training data (i.e., a word-aligned parallel corpus). In

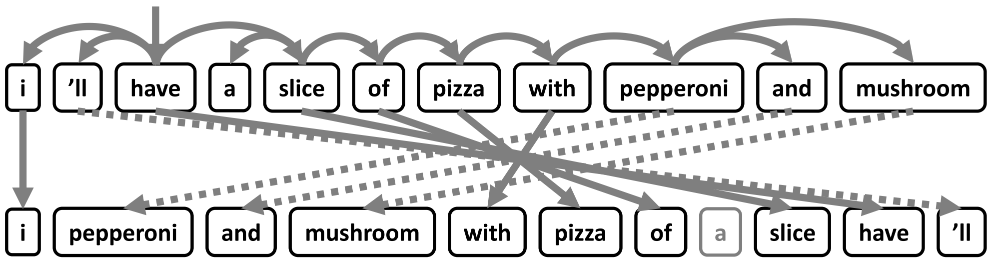

Figure 2: *Pre-ordering example of English sentence "i 'll have a slice of pizza with pepperoni and mushroom".*

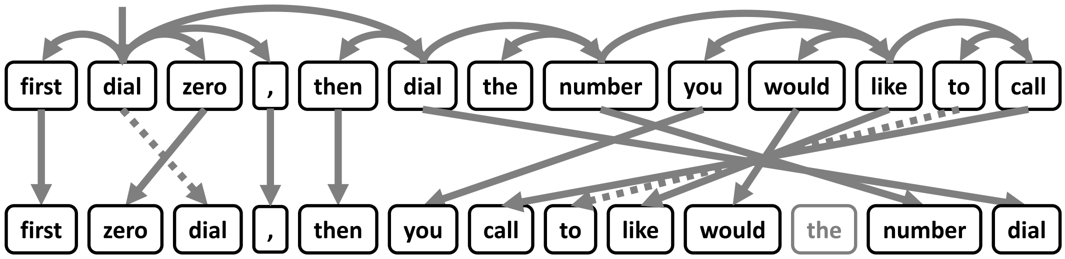

Figure 3: *Pre-ordering example of English sentence "first dial zero , then dial the number you would like to call".*

the experiments reported in [7], they show the model trained by a manually aligned parallel corpus outperforms the model trained by an automatically aligned parallel corpus of more than ten times the size. We take the approach of [7] as a baseline in our experiment, to explore the different characteristics of the rule-based and statistical pre-ordering approaches.

4. Head Finalization for Myanmar

4.1. Basic Principle

The dependency-based head finalization used in our experiment is according the following principle.

- To move the head word after all its modifiers, but
  1. do not break a coordination structure;
  2. do not cross a punctuation mark;
  3. auxiliary verbs come after their head verb.

We show two examples of the English sentence in Fig. 2 and Fig. 3. The dependency structures are marked by arcs over the words. In Fig. 2, "*pepperoni and mushroom*" is a coordination structure with the first word "*pepperoni*" as a head word. We do not apply head finalization in this kind of structure in order to keep the original order of coordinating components. In Fig. 3, the root word of the sentence, i.e. the first *dial*, does not cross the comma after it. We disable head finalization in this situation to avoid excess reordering between clauses.

As to the auxiliary verbs (3), many widely-used dependency parsers handle this kind of functional word as the modifier of a verb, just as an article becoming the modifier of a noun. While we consider auxiliary verb should be the head of a verb, and actually, in typical head-final languages the auxiliary verbs are always placed after the verb. So we arrange auxiliary verbs after their head verb. E.g., in Fig. 2, we keep the "*'ll*" after the verb "*have*"; and in Fig. 3, "*to* " after "*call*".

We describe detailed source-language dependent features in the appendices.

4.2. Myanmar Oriented Process

In the original head finalization approach, a morpheme generation process is used also to generate certain target-side grammatical markers which are absent in the source-side language. Specifically, in [4], they insert three types of tag for *topic marker*, *nominative marker*, and *accusative marker* in the source-side English. However, this issue is not so serious for Myanmar because it has a strong tendency to omit these grammatic markers as long as no ambiguity arises[2]. So we do not apply this generation process in [4] in our approach.

On the other hand, *negation* in Myanmar, unlike in Japanese or in Korean, where it is realized by a negation auxiliary word as a suffix of the verb, is realized by a prefix "မ" before the verb[3]. Further, as a collocation of the negation prefix, a negation suffix "ဘူး" must succeed the verb. Finally, the prefix and suffix surround a verb to form a negation. The phenomenon is rather like the "*ne ... pas*" in French. However, the "*pas*" is not fixed and can be replace with "*plus*"

[2] Actually, the example of a Myanmar sentence given in Fig. 1 is a quite formal expression which is rare in daily communication. We show it mainly to illustrate the syntactical similarity to Japanese and Korean.

[3] In Korean, there are also alternative prefixes used instead of negation suffixes. While in Myanmar, the negation prefix is used consistently.

or "*jamais*" and so forth according the meaning in French, while the prefix and suffix are fixed in Myanmar. We use a *neg* tag for the negation suffix generation. Specifically, the negation word of a verb is placed immediately before the verb and the *neg* tag is inserted immediately after the verb.

We use the same strategy as [4] to delete the articles in the source-side language (if any). As shown in Fig. 2 and Fig. 3, the "*a*" and "*the*" are deleted (marked in gray).

5. Experiments

5.1. Corpus and Settings

We use *Basic Travel Expression Corpus* (BTEC) [11] in the experiments. The source languages are Chinese (zh), English (en), French (fr) and the target language is Myanmar (my). The corpus statistics are is shown in Tables 1, 2 and 3. Specifically, the training, development, and test data for zh-, en-, and fr-my translations contain identical Myanmar sentences. We use two segmentation schemes for the morpheme process of Myanmar sentences. One is syllable-based (syl) [12] and the other is maximum marching (mmx) based on a dictionary with more than $20,000$ Myanmar lexicon entries. The token numbers of the two schemes are listed in the my rows in the tables (syl / mmx). Due to multi-syllable tokens, the syl has larger token numbers than mmx. We show a simple segmentation example in Fig. 4. [4]

Table 1: *Training Corpus.*

Lang.	Sentences	Tokens (syl / mmx for my)
my	$155,121$	$1,835,687$ / $1,508,234$
zh	$155,121$	$1,062,809$
en	$155,121$	$1,161,283$
fr	$155,121$	$1,248,764$

Table 2: *Development Data.*

Lang.	Sentences	Tokens (syl / mmx for my)
my	$5,000$	$59,058$ / $48,546$
zh	$5,000$	$34,103$
en	$5,000$	$37,496$
fr	$5,000$	$40,256$

Table 3: *Test Data.*

Lang.	Sentences	Tokens (syl / mmx for my)
my	$2,000$	$23,661$ / $19,425$
zh	$2,000$	$13,799$
en	$2,000$	$15,146$
fr	$2,000$	$16,173$

[4] As we have mentioned, original Myanmar morphemes are monosyllabic and there are polysyllabic morphemes of loanwords. Actually, "*word*" is not a clear (and natural) unit in Myanmar sentence. In mmx scheme, we have polysyllabic words not only derived from polysyllabic morphemes, but also derived from fixed patterns of monosyllabic morphemes, as Fig. 4 shows.

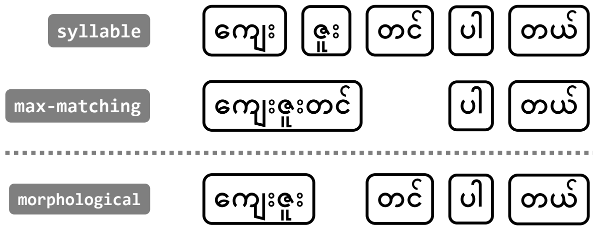

Figure 4: *Segmentation example of a Myanmar expression, meaning "thank you". The two upper rows are the syllable-based segmentation, where each box contains a syllable, and dictionary-based maximum matching, where the first three syllables are merged. The lower row illustrates a morphologically oriented analysis, where the first two syllables should be merged. The meanings of four boxes in the lower row are approximately: "gratitude", "put", polite marker, and sentence-ending marker.*

For the source-side language parsing, we use the Stanford dependency parser[5] for Chinese and English parsing [13, 14]. We use the Stanford tagger[6] [15] for French tagging (*CC* tag set [16]) and Malt parser[7] [17] for French parsing. LADER[8] is used to realize the unsupervised approach in [7] as a comparison approach. For the model training in LADER, we randomly sample $1,000$ automatically aligned sentence pairs from training set because we do not have manually-aligned data. Table 4 of [7] shows that increasing the training data for LADER from 600 to $10,000$ automatically aligned sentence pairs only brought a gain of 0.1 – 0.2 BLEU, therefore we considered a training set size of $1,000$ to be sufficient[9].

We use the phrase-based (PB) SMT system in Moses[10] [2] as a baseline system. GIZA++[11] [18] is used to align word and alignment is symmetrized by *grow-diag-final-and* heuristics [1]. The lexicalized reordering model is trained with the *msd-bidirectional-fe* option [19]. The maximum phrase length is 7. We use SRILM[12] [20] to training 5-gram language model with interpolated modified Kneser-Ney discounting [21] on Myanmar training data.

In decoding, we adopt the default settings of the Moses decoder except the *distortion-limit* (DL). That is, *ttable-limit* is 20 and *stack* is 200. We use DL of 0, 6, 12, and ∞ in the experiments to analyze the reordering abilities of the pre-ordering and the SMT reordering. We tuned the parameter weights on the development sets by MERT [22] and evaluated the translation on test sets by using two automatic measures: BLEU [23] and RIBES [24]. Identical decoding settings were applied on both development sets and test sets.

[5] http://nlp.stanford.edu/software/lex-parser.shtml

[6] http://nlp.stanford.edu/software/tagger.shtml

[7] http://www.maltparser.org/index.html

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

[9] The training of LADER usually takes long time. Under the default settings of LADER, 500 iterations on $1,000$ sentences with 32 threads took more than 10 hours for each translation task in our experiment.

[10] http://www.statmt.org/moses/

[11] https://code.google.com/p/giza-pp/

[12] http://www.speech.sri.com/projects/srilm/

Table 4: *Test set BLEU / RIBES of* `zh-my.syl`.

DL	Baseline	LADER	Head Final.
0	35.5 / .817	36.2 / .816	38.5 / .835
6	37.9 / .831	37.9 / .830	38.7 / .832
12	38.5 / .832	37.9 / .830	38.8 / .832
∞	38.4 / .834	38.3 / .831	38.6 / .832

Table 5: *Test set BLEU / RIBES of* `en-my.syl`.

DL	Baseline	LADER	Head Final.
0	40.4 / .789	47.8 / .861	47.8 / .870
6	45.7 / .842	49.2 / .874	49.9 / .885
12	48.8 / .873	49.6 / .878	**50.3** / .886
∞	49.3 / .875	49.6 / .877	50.2 / .882

Table 6: *Test set BLEU / RIBES of* `fr-my.syl`.

DL	Baseline	LADER	Head Final.
0	36.8 / .786	43.9 / .852	43.7 / .850
6	40.9 / .825	45.2 / .859	45.6 / .860
12	45.1 / .861	45.5 / .859	**46.5** / .866
∞	45.7 / .862	45.7 / .857	**46.5** / .860

Table 7: *Test set BLEU / RIBES of* `zh-my.mmx`.

DL	Baseline	LADER	Head Final.
0	32.9 / .799	34.6 / .810	35.4 / .811
6	34.9 / .816	35.1 / .816	**36.5** / .821
12	35.5 / .817	35.7 / .817	**36.5** / .819
∞	35.2 / .816	35.6 / .814	**36.5** / .820

Table 8: *Test set BLEU / RIBES of* `en-my.mmx`.

DL	Baseline	LADER	Head Final.
0	40.4 / .802	48.0 / .867	47.8 / .871
6	44.7 / .835	48.9 / .871	49.0 / .881
12	48.6 / .873	49.5 / .877	49.8 / .880
∞	49.0 / .876	49.5 / .875	49.7 / .878

Table 9: *Test set BLEU / RIBES of* `fr-my.mmx`.

DL	Baseline	LADER	Head Final.
0	36.9 / .791	43.6 / .844	43.6 / .847
6	39.7 / .818	44.7 / .852	44.9 / .855
12	44.3 / .855	45.1 / .852	45.4 / .855
∞	44.7 / .856	45.4 / .853	45.3 / .853

5.2. Results

We list the experimental results of three source languages (`zh`, `en`, `fr`) with two target Myanmar segmentation schemes (`my.syl`, `my.mmx`) in Tables 4 – 12. In each table, two evaluation measures (BLEU / RIBES) are given with different distortion limits (DLs). The best BLEU scores among the different DLs are underlined and bold BLEU scores are significantly different ($p < 0.05$) to the best baseline BLEU score. As the log-linear model weights were tuned to optimize the BLEU rather than the RIBES score on the development sets with MERT, the RIBES scores shown in the tables are only a complementary evaluation of translation performance on word order.

Table 10: *Test set BLEU / RIBES on* `syl` *of* `zh-my.mmx`.

DL	Baseline	LADER	Head Final.
0	36.8 / .818	38.0 / .829	38.5 / .829
6	38.4 / .836	39.0 / .835	39.7 / .838
12	39.0 / .837	39.4 / .835	**39.9** / .838
∞	38.6 / .833	39.2 / .832	39.6 / .838

Table 11: *Test set BLEU / RIBES on* `syl` *of* `en-my.mmx`.

DL	Baseline	LADER	Head Final.
0	45.0 / .814	51.2 / .879	51.5 / .882
6	48.6 / .847	52.1 / .883	52.7 / .891
12	52.0 / .882	52.8 / .887	**53.4** / .890
∞	52.5 / .885	52.8 / .887	53.2 / .889

Table 12: *Test set BLEU / RIBES on* `syl` *of* `fr-my.mmx`.

DL	Baseline	LADER	Head Final.
0	40.5 / .803	47.0 / .857	46.6 / .860
6	43.1 / .831	48.0 / .865	47.9 / .869
12	47.5 / .867	48.0 / .864	48.4 / .870
∞	47.8 / .867	48.4 / .865	48.3 / .865

In Tables 4 – 6, the evaluation is on `syl` and in Tables 7 – 9, on `mmx`. So the results in the corresponding tables of these two groups are not comparable. In Tables 10 – 12, we show the results on `syl` for `mmx` outputs. So, the corresponding results in Tables 4 – 6 and Tables 10 – 12 are comparable.

5.3. Discussion

In Tables 1 – 3, it can be observed that the average sentence length of the corpus used is quite small (all less than 10 except for `my.syl`). This is because the corpus mainly contains colloquial, rather than literary sentences. This bias suggests two problems. First, the state-of-the-art Moses system can handle the reordering well for short sentences, where a pre-ordering approach may not show its power. Second, there may be more errors in parsing colloquial sentences than literary ones, which may reduce the performance of rule-based head finalization.

Using the same analysis as in [4], first we calculate the average *Kendall's* τ on the training sets (Table 13) to investigate the reordering performance. We observed the following phenomena:

Table 13: *Average Kendall's τ on training sets.*

Language Pair	Baseline	LADER	Head Final.
zh-my.syl	.69	.79	.83
en-my.syl	.53	.79	.79
fr-my.syl	.53	.76	.75
zh-my.mmx	.69	.80	.83
en-my.mmx	.53	.79	.79
fr-my.mmx	.54	.76	.76

- The two different segmentation schemes of Myanmar lead to very similar average Kendell's τ.
- LADER can produce an average Kendall's τ of around .75 – .80 irrespective of the value of average Kendall's τ in its input corpus.
- Dependency-based head finalization shows identical performance to LADER in en-my and fr-my, but better performance on zh-my, where the corpus before pre-ordering already has a relatively high average Kendall's τ.

From Table 13, it is noticeable that en-my and fr-my have nearly identical characteristics while zh-my is different from them. This phenomenon is reflected in the evaluation results on the test sets.

In zh-my translation, we find LADER hardly improves performance over the baseline SMT system in both syl and mmx, while the head finalization approach improves performance over the baseline in both cases and more substantially for mmx. LADER has higher performance on en-my and fr-my, and the proposed head finalization technique has identical or better performance. Since the difference in word order is not as severe for zh-my as for en-my and fr-my (as indicated by the Kendall's τ statistics), we consider rule-based head finalization to be a better complementary approach for the SMT system for zh-my. For language pairs with considerably different word orders as en-my and fr-my, LADER and rule-based head finalization, despite their essentially different mechanisms, attain similar levels of performance.

In Tables 4 – 12, it can also be noticed that the differences are quite large between DL = 0 (i.e. monotone translation) and the corresponding best BLEU in each baseline result, but the differences are reduced by both pre-ordering approaches. So, the performance gains over the baseline by using pre-ordering diminish as the DL is increased. As to the RIBES score, the differences actually are not substantial between the baseline, LADER, and the head finalization approach. We consider these to be reasonable phenomena caused by the short length of the sentences in the corpus.

A major factor affecting the performance of the rule-based head finalization approach is the precision of the parser used, and perhaps the most important factor affecting the performance of a statistical approach, such as LADER, is the quality of the training data. In the survey conducted in [25], they reported "*we observed relatively small effects on reordering quality in response of parsing errors*". We visually inspected a sample of the parsing results used in our experiments and found parsing errors did not have a large effect on the performance of our head finalization approach. We consider a major benefit of our approach is that we almost always use the "*head*" information from a dependency parse, which leads to robustness. The performance of LADER is greatly affected by the quality rather than the amount. So it is sensitive to the nature of languages involved, and also to their word segmentation schemes because they affect the quality of word alignment used to train LADER.

Among the various segmentation schemes for Myanmar, we believe the syl strategy has a tendency to over-split sentences and mmx may lead to some long expressions without necessary splits as illustrated in Figure 4. It can be seen that the data segmented using mmx has fewer tokens and relatively longer words. It was expected that the the evaluation scores in Table 7 – 9 would be lower than those in Table 4 – 6. Conversely, if the translation is done on mmx and evaluated by syl, as shown in Table 10 – 12, we find the results are better than those in Table 4 – 6. The experimental results show that the mmx strategy is a better segmentation strategy than syl. Although mmx introduces long expressions, it can offer more meaningful units in word alignment and translation, which lead to a better performance. However, a more useful standard morpheme analysis system should hopefully be built for Myanmar in the future.

We show translation examples of zh-my, en-my and fr-my. The examples are selected from the best results of mmx and illustrated using syl segmentation. It can be seen that the head finalization has a rigidity with respect to the syntactic structure. For example, the objects of verbs are strictly arranged in front positions in head finalization (actually, untouched), such as the Chinese "我" in Fig. 5, the English "*i*" in Fig. 6, and the French "*j'* " in Fig. 7. While in the pre-ordering from LADER, those words are scattered. For example, in the first example of Fig. 6 and in Fig. 7, the "*i*" and "*j'* " are moved to the end of the sentences. This is because LADER does not have information on the syntactic structure of a sentence. In this example, LADER moves the phrases "*i want*" and "*j' ai*" as whole units to the sentence ends, and makes further local swapping within the phrases. The second example of Fig. 6 shows the simplicity of our head finalization approach; in this example, only the verb "*bring*" is moved to the end of the sentence.

6. Conclusion and Future Work

In this paper, we conducted pre-ordering experiments on Chinese-, English-, French-to-Myanmar translation. We found that a simple dependency-based head finalization pre-ordering strategy can consistently and efficiently improve a baseline SMT system. The proposed head finalization approach does not require parallel training data, and only de-

Baseline Input	对不起， 请 告诉 我 这个 怎么 用 ？ *(sorry , please tell me this how use ?)*
LADER Input	这个 对不起， 请 告诉 我 怎么 用 ？ *(this sorry , please tell me how use ?)*
Head Final. Input	对不起， 我 这个 怎么 用 告诉 请 ？ *(sorry , me this how use tell please ?)*
Baseline Output	တ ဆိတ် လောက် ၊ ကျေး ဇူး ပြု ပြီး ဒီ ဟာ ကို ဘယ် လို သုံး ရ မ လဲ ။
LADER Output	ကျေး ဇူး ပြု ပြီး တ ဆိတ် လောက် ။ ကျေး ဇူး ပြု ပြီး ဒီ ဟာ ကို ဘယ် လို သုံး ရ မ လဲ ။
Head Final. Output	တ ဆိတ် လောက် ၊ ဒါ ကို ဘယ် လို သုံး ရ မ လဲ ဆို တာ ပြော ပြ ပေး နိုင် မ လား ။
Reference	တ ဆိတ် လောက် ၊ ဒါ ကို ဘယ် လို သုံး ရ တယ် ဆို တာ ပြ ပေး ပါ လား ။

Figure 5: *Chinese-to-Myanmar translation example. (For the input rows, word-by-word English literal translations are annotated in gray. An unconstrained translation of the original Chinese sentence is "Excuse me, could you tell me how this works?")*

Baseline Input	**i want to send this parcel to japan .**	**please bring me some ice .**
LADER Input	**this parcel to japan send to want i .**	**ice some me please bring .**
Head Final. Input	**i this parcel japan to send to want .**	**please me some ice bring .**
Baseline Output	ဒီ အ ထုတ် ကို ဂျ ပန် ကို ပို့ ချင် လို့ ပါ ။	ရေ ခဲ ကို နည်း နည်း လောက် ယူ လာ ပေး ပါ ။
LADER Output	ဒီ ပါ ဆယ် ဂျ ပန် ကို ပို့ ချင် ပါ တယ် ။	ရေ ခဲ နည်း နည်း ယူ လာ ပေး ပါ နော် ။
Head Final. Output	ဒီ ပါ ဆယ် ဂျ ပန် ကို ပို့ ချင် လို့ ပါ ။	ရေ ခဲ ယူ လာ ပေး ပါ ။
Reference	ဒီ ပါ ဆယ် ဂျ ပန် ကို ပို့ ချင် လို့ ပါ ။	ရေ ခဲ ကို ယူ လာ ပေး ပါ ။

Figure 6: *English-to-Myanmar translation examples.*

Baseline Input	**j' ai oublié mon billet d' avion .** *(I have forgotten my ticket of aeroplane .)*
LADER Input	**avion d' billet mon oublié ai j' .** *(aeroplane of ticket my forgetten have I .)*
Head Final. Input	**j' mon avion d' billet oublié ai .** *(I my aeroplane of ticket forgetten have .)*
Baseline Output	ကျွန် တော့် ရဲ့ လေ ယာဉ် လက် မှတ် မေ့ ကျန် ခဲ့ တယ် ။
LADER Output	လေ ယာဉ် လက် မှတ် မေ့ ကျန် ခဲ့ တယ် ။
Head Final. Output	လေ ယာဉ် လက် မှတ် မေ့ ကျန် ခဲ့ ပါ တယ် ။
Reference	လေ ယာဉ် လက် မှတ် မေ့ ကျန် ခဲ့ ပါ တယ် ။

Figure 7: *French-to-Myanmar translation example. (For the input rows, word-by-word English literal translations are annotated in gray. An unconstrained translation of the original French sentence is "I forgot my airline ticket.")*

pends on a source-side dependency parser, which allowed it to attain higher performance than an unsupervised baseline in our experiment. The simplicity and efficiency of the proposed head finalization approach should allow it to find practical application on large scale data sets.

In further work, we plan to expand the parallel data and conduct experiments on larger corpora. We are also developing a morpheme analyzer and parsers for Myanmar to facilitate the transference of more techniques of Japanese and Korean language processing to Myanmar language processing.

7. References

[1] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation." in *Proc. of HTL-NAACL*, 2003, pp. 48–54.

[2] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation." in *Proc. of ACL*, 2007, pp. 177–180.

[3] P. Koehn, "Europarl: A parallel corpus for statistical machine translation." in *Proc. of MT summit*, 2005, pp. 79–86.

[4] H. Isozaki, K. Sudoh, H. Tsukada, and K. Duh, "HPSG-based preprocessing for English-to-Japanese translation." *ACM Transactions on Asian Language Information Processing*, vol. 11, no. 3, 2012, article 8.

[5] H. Dan, K. Sudoh, X. Wu, K. Duh, H. Tsukada, and M. Nagata, "Head finalization reordering for Chinese-to-Japanese machine translation." in *Proc. of SSST*, 2012, pp. 57–66.

[6] P. Xu, J. Kang, M. Ringgaard, and F. Och, "Using a de-

pendency parser to improve SMT for subject-object-verb languages." in *Proc. of HLT-NAACL*, 2009, pp. 245–253.

[7] G. Neubig, T. Watanabe, and S. Mori, "Inducing a discriminative parser to optimize machine translation reordering." in *Proc. of EMNLP-CoNLL*, 2012, pp. 843–853.

[8] F. Xia and M. McCord, "Improving a statistical mt system with automatically learned rewrite patterns," in *Proc. of COLING*, no. 508, 2004.

[9] M. Collins, P. Koehn, and I. Kučerová, "Clause restructuring for statistical machine translation," in *Proc. of ACL*, 2005, pp. 531–540.

[10] D. Genzel, "Automatically learning source-side reordering rules for large scale machine translation." in *Proc. of COLING*, 2010, pp. 376–384.

[11] G. Kikui, E. Sumita, T. Takezawa, and S. Yamamoto, "Creating corpora for speech-to-speech translation." in *Proc. of EUROSPEECH*, 2003, pp. 381–384.

[12] Y. K. Thu, A. Finch, Y. Sagisaka, and E. Sumita, "A study of Myanmar word segmentation schemes for statistical machine translation," in *Proc. of ICCA*, 2013, pp. 167–179.

[13] R. Levy and C. Manning, "Is it harder to parse Chinese, or the Chinese treebank?" in *Proc. of ACL*, 2003, pp. 439–446.

[14] R. Socher, J. Bauer, C. D. Manning, and A. Y. Ng, "Parsing with compositional vector grammars." in *Proc. of ACL*, 2013, pp. 455–465.

[15] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network." in *Proc. of HLT-NAACL*, 2003, pp. 173–180.

[16] B. Crabbé and M. Candito, "Expériences d'analyse syntaxique statistique du français." in *Proc. of TALN*, 2008.

[17] J. Nivre, J. Hall, J. Nilsson, A. Chanev, G. Eryigit, S. Kübler, S. Marinov, and E. Marsi, "MaltParser: A language-independent system for data-driven dependency parsing." *Natural Language Engineering*, vol. 13, no. 02, pp. 95–135, 2007.

[18] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models." *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[19] C. Tillmann, "A unigram orientation model for statistical machine translation." in *Proc. of HTL-NAACL*, 2004, pp. 101–104.

[20] A. Stolcke, "SRILM–an extensible language modeling toolkit." in *Proc. of ICSLP 2002*, 2002, pp. 901–904.

[21] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling." TR-10-98, Computer Science Group, Harvard Univ., Tech. Rep., 1998.

[22] F. J. Och, "Minimum error rate training in statistical machine translation." in *Proc. of ACL*, 2003, pp. 160–167.

[23] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation." in *Proc. of ACL*, 2002, pp. 311–318.

[24] H. Isozaki, T. Hirao, K. Duh, K. Sudoh, and H. Tsukada, "Automatic evaluation of translation quality for distant language pairs." in *Proc. of EMNLP*, 2010, pp. 944–952.

[25] D. Han, P. Martınez-Gómez, Y. Miyao, K. Sudoh, and M. Nagata, "Effects of parsing errors on pre-reordering performance for Chinese-to-Japanese SMT," in *Proc. of PACLIC*, 2013, pp. 267–276.

A. Head Finalization for Chinese

We use the Standford Chinese dependency parser.

- The `conj` arc is used to identify coordination.
- The `punct` arc is used to identify punctuation marks.
- The `asp`, `assm`, `ba`, `cop`, `cpm`, `dvpm`, `mmod` arcs are taken as auxiliary verbs or post-positioned particles. They are always arranged after their heads.
- The `neg` arc is used to identify the negation.
- We clean up parsing errors around several common Chinese function words, to insure:
  - sentence final particles "啊", "吧", "的", "了", "吗", "呢", "呀" are always after their head words;
  - determiners "这", "那", "哪" are always before their head words.
- The article deletion process is not applied in Chinese.

B. Head Finalization for English

We use the Standford English dependency parser.

- The `conj`, `cc` arcs are used to identify coordination.
- The `punct` arc is used to identify punctuation marks.
- The `aux`, `auxpass`, `cop` arcs are taken as auxiliary verbs. They are always arranged after their heads.
- The `mark` arc and "*when*", "*where*" with `advmod` arc are always arranged after their heads.
- The `neg` arc is used to identify the negation.
- The "*there be*" of an existential clause is kept together.
- For the process of article deletion, we delete "*a*", "*an*", "*the*".

C. Head Finalization for French

We use Malt French parser with the CC tag set.

- The `*coord*` arcs are used to identify coordination.
- The `ponct` arc is used to identify punctuation marks.
- The `*aux*` arcs are taken as auxiliary verbs. They are always arranged after their heads.
- The "*ne*", "*n'* " with `mod` arc is used to identify the negation.
- The "*il y a*" and "*y a-t-il*" of an existential clause is kept together.
- For the process of article deletion, we delete "*le*", "*la*", "*l'* ", "*les*", "*un*", "*une*".