Automatic Reordering Rule Generation for English-Myanmar Translation

Thinn Thinn Wai, Tin Myat Htwe, Ni Lar Thein
University of Computer Studies, Yangon, Myanmar
thin2wai@gmail.com

Abstract

Reordering is important problem to be considered when translating between language pairs with different orders. Our Language, Myanmar is a verb final language and reordering is needed when we translate our language from other languages with different orders. In this paper, we focus on automatic reordering rule generation for English-Myanmar machine translation. In order to generate reordering rules; English-Myanmar parallel tagged aligned corpus is firstly created. Then reordering rules are generated automatically by using the linguistic information from this parallel tagged aligned corpus. In this paper, function tag and part-of-speech tag reordering rule extraction algorithms are proposed to generate reordering rule automatically. These algorithms can be used for other language pairs which need reordering because these rules generation is only depend on part-of-speech tags and function tags.

Key Words: reordering, English-Myanmar translation, tagged aligned corpus.

1. Introduction

The goal of statistical machine translation is to translate an input word sequence in the source language into a target language word sequence. In order to improve the translation process, it is possible to perform preprocessing steps before training and translation in both source and target language sequence. In machine translation, reordering is one of the major problems, since different languages have different word order requirements. When an English sentence is translated to Myanmar sentence, the verb in the English sentence must be moved to the end of the sentence in order to obtain the correct word order. On a sub sentential level, Myanmar word order diverges from English mostly within the noun phrase and verb phrase. In Myanmar, noun phrase exhibits multitude of word orders. In chunk level, the noun chunk made up of determiner (DT) and noun (NN) is translated as many patterns such as "DT, NN" (original English order) and "NN, DT" (Myanmar order). Moreover, some function tags are missed and some part-of-speech tags in some chunks are combined together as only one tag. For example, formal subject (F-SUBJ) and verb chunk containing verb-to-be and adjective. Without reordering, the correct word order can't be obtained. Therefore, reordering is necessary for translation from English language to Myanmar Language. In this work, corpus creation procedure and reordering rules generation procedures are proposed for English-Myanmar statistical machine translation.

The plan of this paper is as follows. In the next section, related works which use reordering approaches in a preprocessing step are reviewed. In Section 3, novel methods for reordering are described. Section 4 presented the word order differences in English and Myanmar. Section 5 describes analysis steps and corpus creation. In Section 6, proposed reordering rule extraction algorithm and reordering rules are explained in details. In the last two sections, the experiments are reported and then we conclude the experiments and discuss future work respectively.

2. Related Work

Different approaches have been developed to deal with the word order problem. First approaches worked by constraining reordering at decoding time [7]. In [12], the alignment model

introduced the restrictions in word order, which leads also to restrictions at decoding time. A comparison of these two approaches can be found in [2]. They have in common that they do not use any syntactic or lexical information; therefore they rely on a strong language model or on long phrases to get the right word order. Other approaches were introduced that use more linguistic knowledge, for example the use of bitext grammars that allow parsing the source and target language [13]. In [10], syntactic information was used to re rank the output of a translation system with the idea of accounting for different reordering at this stage. In [11], a lexicalized block-oriented reordering model is proposed that decides for a given phrase whether the next phrase should be oriented to its left or right.

The most recent and very promising approaches that have been demonstrated reorder the source sentences based on rules learned from an aligned training corpus with a POS-tagged source side [8, 9, 20]. These rules are then used to reorder the word sequence in the most likely way.

In our approach we follow the idea proposed in [20] of using a parallel training corpus with a tagged source side to extract rules which allow a reordering before the translation task. Moreover, we use the lexical information for some part-of-speech (pos) rules to solve ambiguity problems. By doing this we hope to differentiate between these pos rules.

3. Word Order differences in English Language and Myanmar Language.

When English sentence is translated to Myanmar sentence, many differences of word order can be found. In this section, we describe some word order differences; adjective movement, adverb movement, preposition movement and auxiliary verb movement.

Some adjectives (jj) in noun chunk (nc) of English sentence are necessary to move after its relative noun (nn) according to the translation. For example, when the English phrase “rich man” is translated into the Myanmar phrase “လူချမ်းသာ”, the adjective “rich (jj)” must be moved after the noun “man (nn)”. This can be seen in the Example (1).

Example (1),

nc[rich(jj) man(nn)]

nc[လူ (nn)ချမ်းသာ(jj)]

Myanmar is also modifier and adjunct proceeding language. Therefore, these adjuncts are necessary to move before the relative verb. When the English sentence “He runs quickly” is translated into the Myanmar sentence “သူလျှင်မြန်စွာပြေးသည်”, the adverb “quickly” must be moved before its relative verb “run” in order to fit the correct Myanmar order. Such adverb movement can be seen in the Example (2).

Example (2),

He(subj) runs(active) quickly.(advl)

သူ(subj)လျှင်မြန်စွာ(advl)ပြေးသည်(active)

Moreover, auxiliary verbs (md) in verb chunk (vc) are necessary to move after the main verb in order to get an appropriate word order. Therefore, auxiliary verb movement also helps in English-Myanmar Translation. This auxiliary verb movement can be seen in the Example (3).

Example (3),

vc[can(md) go(vb)]

vc[သွား(vb)နိုင်သည်။(md)]

All of these above necessities, word reordering is needed for English-Myanmar statistical machine translation.

4. Corpus Creation

Corpus creation steps are described in Figure 1. For corpus creation, plain text corpus is used as a resource. For each sentence in the corpus, analysis process is carried out by using Chunk-based Syntax Analyzer [15]. This Syntax Analyzer consists of two components; Chunker and Grammatical Function Tagger. In this analysis process, there are three main steps.

(1) Morpho-lexical analysis
(2) Constituent analysis and
(3) Syntax analysis

Morpho-lexical analysis and constituent analysis are accomplished by the chunker and

syntax analysis is the role of grammatical function tagger.

Morpho-lexical analysis contains tokenizing and part-of-speech tagging. Tokenizing splits input text into words by using token marker such as space, punctuation marks. Part-of-speech (POS) tagging marks up the words in the text with their corresponding part-of-speech such as noun, verb, and adjective and so on. For this POS tagging, TreeTagger is used.

Constituent analysis consists of chunking and merging some chunks that are necessary to merge. Chunking is done by generating CFG rules based on part-of-speech (POS) tags.

In syntax analysis, Grammatical function tagger searches the functional relation between chunks based on dependency grammar by using Maximum likelihood Estimation and then identifies the function of each chunk.

By aligning the analyzed text resulted from Analyzer, parallel tagged aligned corpus is created.

Our tagged align corpus format can be seen in Figure 2.

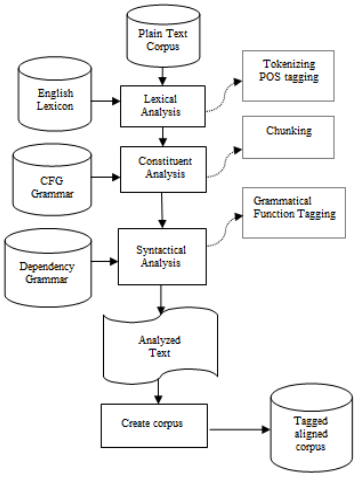


Figure 1: Corpus Creation Steps

SUBJ:NC[I/PP(0/0)#(1/3)ACTIVE:VC[go/VBP(0/0)]#(2/2)ADVL:PPC[to/TO(0/0)]#(3/1)OBJ-P:NC[school/NN(0/0)]

Figure 2: Tagged Aligned Corpus

As we can see in this figure, "SUBJ","ACTIVE" , "ADVL" and "OBJ-P" are function tags of each chunk. "NC", "VC" and "PPC" refer the relevant chunk type and "PP", "VBP", "TO" and "NN" are part of speech of each word. The numbers in the parentheses are alignment position of function tags and part of speech tags. The first number before "/" indicates the position of tags in source language and the number after "/" indicates the position of tags in target language. These are separated by target position with "/". Each chunk is separated by "#".

5. Reordering Rule Extraction

By using the linguistic information from the corpus, two kinds of reordering rules are generated automatically. They are function tags-based reordering rules and part-of-speech tags-based reordering rules. The former is generated for using in chunk-level reordering and the latter is for using word-level reordering. They are extracted from corpus using the following rule extraction algorithms.

function rule extraction algorithm
funSeq=NULL //sequence for function tags
aliSeq=NULL //sequence for alignment position

1. Load the sentences from Tagged Aligned Corpus
2. Store all sentences in S .
3. for each sentence $s_i \in S$ do, where i=1,2,3,…k
4. for each chunk $c_i \in C$ do, where i =1,2,3,…k
5. if (k>1)
6. extract f_i for s_i
7. $funSeq \rightarrow funSeq + f_i$
8. extract alignment position a_i
9. $aliSeq \rightarrow aliSeq + a_i$
10. End if//line 5
11. End for//line 4
12. $rule \rightarrow funSeq + \#+ aliSeq$
13. write *rule*
14. End for//line 3
15. End.

pos rule extraction algorithm

posSeq=NULL *//sequence of pos tags*
aliSeq=NULL *//sequence of alignment position*

1. Load the sentences from Tagged Aligned Corpus
2. Store all sentences in S .
3. for each sentence $s_i \in S$ do, where i=1,2,3,…k
4. for each chunk $c_i \in C$ in s_i do, where i=1,2,3,…k
5. for each words $w_i \in W$ in c_i where i=1,2,3,…,k
6. if (k>1)
7. extract pos_i for w_i
8. $posSeq \rightarrow posSeq + pos_i$
9. extract alignment position a_i for w_i
10. $aliSeq \rightarrow aliSeq + a_i$
11. End if//line 6
12. End for//line 5
13. rule= $posSeq$ +#+ $aliSeq$
14. End for//line 4
15. write rule
16. End for//line 3

alignment extraction algorithm

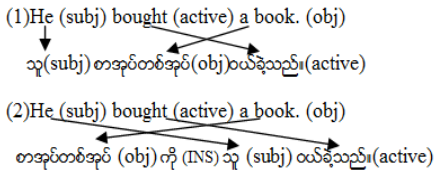
Input: AP //Alignment Position Array
Output: rule // for actual alignment position
A=NULL // Array for final alignment position

1. for each ap_i from Array AP do
2. if ($ap_i = ap_{i-1}$) then
3. a_{i-1} =i-1+i+ ap_i
4. else
5. a_i =i+\+ ap_i
6. end if
7. end for
8. for each $a_i \in$ A do
9. if $a_i \neq NULL$ then
10. rule=rule+ a_i
11. end if
12. end for
13. return rule

5.1. Reordering Rule Generation

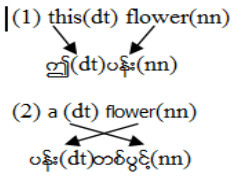
In order to generate reordering rules for English-Myanmar translation, two main cases are needed to consider. In the first case, words can be reorder in several ways if they have different reordering rules. In this case, reordering in several ways does not affect the translation because Myanmar is free chunk order language. In the second case, words are needed to reorder according to the specific translation although they have same pattern with different reordering rules.

According to the first case, reordering can be seen in the following Example (4).
Example (4),



From the above example, the English Sentence “He bought a book.” has two different reordering rules and both of these rules make correct translation. Moreover, we can see that word insertion (INS) is needed when this sentence is translated in second way (2).

The second case suggests that, the use of reordering rules mistakenly makes the translation error if they are not reorder over the specified translation. This can be seen in Example (5),
Example (5),



By studying this example, the POS rules composed of determiner and noun have several reordering patterns as shown in (1) and (2). Although they have same pattern (dt,nn), they must be reordered according to the identified

translation. If these patterns are not reorder according to the specified patterns, there will be error in translation. To solve this condition, lexical information for determiner is needed to consider in generating reordering rules for this pattern. Therefore, reordering rules are generated automatically using part-of-speech tag, function tags and lexical information.

The generated reordering rule consists of two sides: the left-hand-side (lhs), which is a function tags or POS tags pattern, and the right-hand-side (rhs), which corresponds to a possible reordering of that pattern. Different rules can share the lhs: in such cases, the same pattern can be reordered in more than one way. Rules are weighted, according to statistics extracted from training data. There are two kinds of reordering patterns: function tag-based, which define reordering at the clause and phrase level, and pos tag-based, which defines reordering at the word level. Let us consider the following examples:

- **Rules using function tag**

-SUBJ, ACTIVE, OBJ#0/0, 1/2, 2/1:7(10)
-SUBJ, ACTIVE, OBJ#0/1, 1/2, 2/0:3(10)
-SUBJ, ACTIVE, ADVL#0/0, 1/2, 2/1:10(10)
-F-SUBJ, ACTIVE, PCOMPL-S, ADVL, OBJ-P# 0+1/3, 2/2, 3/1, 4/0:10(10)

- **Rules using pos tag**

-the@ DT, NN#0+1/0:10(30)
-this @DT, NN#0/0, 1/1:10(30)
-a @DT, NN#0/1, 1/0:10(30)
-CD, NNS#1/0, 0, 1:10(10)
-DT, JJ, NN#1/0, 2/1, 0/2:10(10)

In the above rules, “SUBJ”,”ACTIVE” and OBJ are function tags and “DT”,,”NN”,”CD”,”NNS” are POS’s tags. Therefore, “SUBJ, ACTIVE, OBJ” is function rule pattern and “DT, JJ, NN” is POS rule pattern. The string of numbers after “#” is position of source and target words and source word position is divided by “/” target word position. For example, in the rhs of the third pos rule pattern“1/0, 2/1, 0/ 2”, the 1/0 means that the pos tag at the position 1,”JJ” is move to the position 0. In this model, we used array structure to store the position and so the starting index is 0. Moreover, in the function tag rule, the formal subject(F-SUBJ) “*there*” in English is not in the Myanmar Function tag and then it is translated as “ရှိသည်” (ACTIVE) in Myanmar language by combining it into the Function tag(ACTIVE) containing the words “*am ,is are, was, were*”. This means that the two function tags F-SUBJ and ACTIVE are become only ACTIVE and F-SUBJ is dismissed in Myanmar. Therefore, in the third function tag rule described above, the string after #, “0+1/3” means that the words at position “0”and “1” are move together into the position “3”.

The sequences “SUBJ, ACTIVE, OBJ” and “DT, JJ, NN” are function and pos rule patterns (p_1^n).The strings of numbers in between the symbols “#” and “:” represent suggested reordering (r_1^n): each integer after “/”, r_i represents the new position of (the translation of) p_i. The two numbers after the colon (:) are collected from training data and are respectively the number of times the rhs (reordering suggestion) of the rule has been observed and (inside brackets) the number of occurrences of the rule pattern $count(p_1^n)$. The probability of each reordering suggestion is computed as follows:

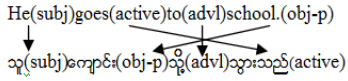
$$P(r_1^n / p_1^n) = \frac{count(r_1^n)}{count(p_1^n)} \qquad (1)$$

Moreover, some pos rules composed of determiner (a, an, the, this… etc) have same pattern with different reordering rules according to the kind of determiner used. To solve the ambiguity problem of these pos rules, lexical information concerned with determiner is added before the rule pattern divided by “@” to obtain the correct order.

5.2. Reordering rule Examples

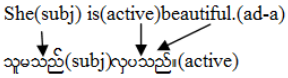
In this section, we describe some of the reordering rule examples for function tags-based and part-of-speech based reordering rules. And then we also describe some reordering rules which have more than one reordering rule suggestions.

Example (6)



In this above example, function tags are aligned according to one-to-one correspondence. In this example, there is only one word in each phrase and so there is no part-of-speech reordering rule.

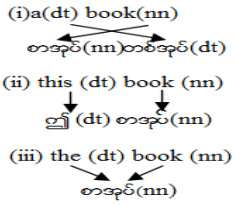
Example (7)



In Example (7), many-to-one correspondence alignment rule can be seen. As can be seen in this example, the function tags (active) and (ad-a) in English sentence are combined to only one function tag (active) in Myanmar sentence. If the information of function tag (ad-a) is missed, the meaning of this word is also missed and so translation of this sentence to Myanmar sentence is meaningless. Therefore many-to-one correspondence is also needed.

As the function tag rule described above, part-of-speech rules have many-to-one correspondence and many reordering pattern. In the part-of-speech rule composed of determiner and noun, this is translated in three ways such as: 1)these are combined into only one pos tag(noun), 2)they are translated according to original order, and 3) they are translated reverse order. This reordering that contain same pattern with different reordering can be seen in Example (8).

Example (8),



6. Experiments

These generated reordering rules are tested on the English –Myanmar machine translation system. Our Experiment shows that the use of reordering rules provide translation effectively. Moreover, these reordering rules can be used as a rule base for English-Myanmar machine translation. Besides, they can be combined with mathematical Models to create reordering model for English-Myanmar translation. By using these rules as an embedded component, English-Myanmar translation system can perform translation effectively and efficiently.

6.1. Accuracy of the reordering rules

The purpose of this experiment was to see how many reordering rules are accurate when they were applied to the test set. The test set was obtained randomly from High School English books. In the test set, Lengths of the sentences are between 3 and 20 words. Average length is 5.96 words. The test set was split into three subsets:

- 500 simple sentences
- 500 compound sentences
- 500 complex sentences

After reordering the test set by the reordering rules, the accuracy values of the reordering rules were collected for each subset on the test set. The accuracy values were given in percentage form. Human evaluation was used for evaluating how accurately the reordering rules are applied to the test set.

Table 3 shows the accuracies of the reordering rules for each subset of English sentences on the test set. The experiment showed that the most common causes of errors of the reordering rules are incorrect part-of-speech tagging and function tagging.

Table 3: Accuracy of Reordering Rules

English test subsets	Accuracy
Simple sentences	98.9%
Complex sentences	86.5%
Compound sentences	83.8%

7. Conclusions and future works

This paper proposes automatic reordering rules generation algorithms for English-Myanmar translation. These rules are extracted based on the part-of-speech tags and function tags extracted from Chunk based Analyzer. These rule extraction algorithms can be used to reorder other language pairs those have their own analyzer because the input of these algorithm only depend on the result of Language Analyzer. In this work, rules are extracted for English-Myanmar translation and my future work is that these rules extraction algorithm is used in Myanmar-English translation because Myanmar Analyzer is ongoing research. These reordering rules can also improve English-Myanmar translation qualities of simple, complex and compound sentences.

8. References

[1] C. Tillmann and H. Ney. 2002. Word reordering and DP beam search for statistical machine translation to appear in Computational Linguistics.

[2] R. Zens and H. Ney. 2003. A comparative study on reordering constraints in statistical machine trans lation. In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, vol ume 1, pages 144–151, Sapporo, Japan.

[3] S. Vogel, F.J. Och, C. Tillmann, S. Nießen, H. Sawaf, and H. Ney. 2000. Statistical methods for machine translation. InW.Wahlster, editor, Verbmobil: Foundations of Speech-to-Speech Translation, pages 377–393. Springer Verlag: Berlin, Heidelberg, New York.

[4] Y.Y. Wang and A. Waibel. 1997. Decoding algorithm in statistical translation. In Proc. 35th Annual Meeting of the Assoc. for Computational Linguistics, pages 366–372, Madrid, Spain, July.

[5] Ei Ei Han and Ni Lar Thein, "Morphological Synthesis For Myanmar Language", Proceeding of International Conference on Internet Information Retrieval, Korea, 2007.

[6] Yaser Al-Onaizan and Kishore Papineno. 2006. Distortion models for statistical machine translation. In Proceedings of the 21st International Conference on Computational Linguistics and the 4th annual meeting of the ACL, pages 529–536, Sydney, Australia.

[7] A. L. Berger, S. A. Della Pietra, and V. J. Della Pietra,1996. A maximum entropy approach to natural language processing. Computational Linguistics, 22(1):39.

[8] B. Chen, M. Cettolo, and M. Federico. 2006. Reordering rules for phrase-based statistical machine translation. In Int. Workshop on Spoken Language Translation Evaluation Campaign on Spoken Language Translation, pages 1–15.

[9] M. Popovic and H. Ney. 2006. POS-based word reorderings for statistical machine translation. In Proc. of the 5th Int. Conf. on Language Resources and Evaluation (LREC), page 1278, Genoa, Italy.

[10] L. Shen, A. Sarkar, and F. J. Och. 2004. Discriminative reranking for machine translation. In HLTNAACL 2004: Main Proc., page 177.

[11] C. Tillmann and T. Zhang. 2005. A localized prediction model for statistical machine translation. In Proceedings of the 43rd Annual Meeting of the Assoc. for Computational Linguistics (ACL), pages 557–564, Ann Arbor, MI.

[12] D. Wu. 1996. A polynomial-time algorithm for statistical machine translation. Proc. 34th Annual Meeting of the Assoc. for Computational Linguistics, page 152.

[13] D. Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. Computational Linguistics, 23(3):377.

[14] Y. Zhang, R. Zens, and H. Ney. 2007. Chunk-Level Reordering of Source Language Sentences with Automatically Learned Rules for Statistical Machine Translation. In Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL): Proceedings of the Workshop on Syntax and Structure in Statistical Translation (SSST), pages 1–8, Rochester, NY.

[15] Myat Thuzar Tun and Ni Lar Thein, " English Syntax Analyzer for English-to-Myanmar Machine Translation", In proceedings of the Fifth International Conference on Computer Application, Myanmar, February, 8-9,2007.

[16] Myat Thuzar Tun, Tin Myat Htwe and Ni Lar Thein, "EMTM: An Effective Language Translation Model", In proceedings of International Conference on Internet Information Retrieval, Korea, November 30, 2005.

[17] Shankar Kumar “Local Phrase Reordering Models for Statistical Machine Translation”, Center for Language and Speech Processing, Johns Hopkins University, 3400 North Charles Street, Baltimore, MD 21218, U.S.A.

[18] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer, “The Mathematics of Statistical Machine Translation: Parameter Estimation,”

Computational Linguistics, vol. 19(2), pp. 263–312, 1993.
[19] Kenji Yamada and Kevin Knight. 2000. A Syntax based Statistical Translation Model. ACL 2000.
[20] Josep M. Crego and Jose B. Marino. 2006. Reordering Experiments for N-Gram-based SMT. In Spoken Language Technology Workshop, pages 242-245, Palm Beach, Aruba.
[21] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, "BLEU: a Method for Automatic Evaluation of Machine Translation", Association for Computational Linguistics, 2002, pp. 311-318.