Automatic Reordering Rule Generation Based On Parallel Tagged Aligned Corpus for Myanmar-English Machine Translation

Thinn Thinn Wai, Tin Myat Htwe, Ni Lar Thein

Abstract— Reordering is important problem to be considered when translating between language pairs with different word orders. Myanmar is a verb final language and reordering is needed when it is translated into other languages which are different from Myanmar word order. In this paper, automatic reordering rule generation for Myanmar-English machine machine translation is presented. In order to generate reordering rules; Myanmar-English 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 rules 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.

Index Terms— Constituent Analysis, English-Myanmar Machine translation, parallel tagged aligned corpus, Reordering, Syntactic Analysis,

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 a Myanmar sentence is translated into English sentence, the verb in the Myanmar sentence must be moved after the subject of the English 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. Moreorver, there are many particles that support noun, adjective, and verb in Myanmar Language. They are subject marker particles, object marker particles, adjective support particles and verb support particles. These particles do not exist in English and their missing can make the translation error. So, each particle is needed to move its respective places scuch as beside a noun, verb and so on. To allievate the tag missing, moving these particles to their respective places is essential. Without reordering, the particles can be far from their relative nouns, verbs and adjectives and the correct word order can't be obtained. In addition to this, the meaningful translation can't also be obtained. Therefore, reordering is necessary for translation from Myanmar language to English Language. In this work, corpus creation procedure and reordering rules generation procedures are proposed for Myanmar-English 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, the significant differences of word order in English language and Myanmar language. Section 4 describes analysis steps and corpus creation. In Section 5, 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.

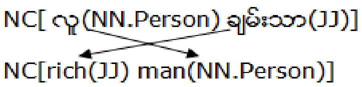
The most recent and very promising approaches that have been demonstrated reorder the source sentences based on rules learned from an aligned training corpus with a POS-tagged source side [8, 9, 20]. These rules are then used to reorder the word sequence in the most likely way.

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.

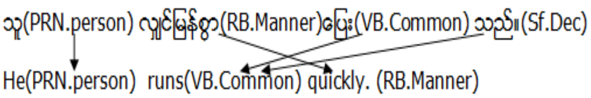
3 Differences of Word Order Between English Language And Myanmar Language

When Myanmar sentence is translated to English sentence, many differences of word order can be found. In this section, significant word order differences; adjective movement, and adverb movement will be described.

Some adjectives (JJ) in noun chunk (NC) of Myanmar sentence are necessary to move before its relative noun (NN.Person) to obtain the correct English order. For example, when the Myanmar phrase “ လူချမ်းသာ ” is translated into the English phrase “rich man”, the adjective “ ချမ်းသာ (JJ)” must be moved before its relative noun “ လူ (NN.Person)” . This can be seen in the Example (1).
Example (1),



Myanmar is also modifier and adjunct proceeding language. Therefore, these adjuncts are needed to move after its relative verb to make the correct word order in English Sentence. When the Myanmar sentence “ သူလျှင်မြန်စွာပြေးသည် ” is translated into the English sentence “He runs quickly.” the adverb of manner “ လျှင်မြန်စွာ ” must be moved behind its relative verb “ပြေး” in order to fit the correct English order. Such adverb movement can be seen in the Example (2).
Example (2),


သူ(PRN.person) လျှင်မြန်စွာ(RB.Manner)ပြေး(VB.Common) သည်။(Sf.Dec)
He(PRN.person) runs(VB.Common) quickly. (RB.Manner)

In this example, the verb particle pos tag (Sf.Dec) is also needed to move beside its relative verb not to miss the Myanmar word meaning. Therefore, the pos tag “VB.Common “and ‘Sf.Dec” in Myanmar phrase are combined to form only one tag “VB.Common” in English phrase.

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

4 Corpus Creation

Corpus creation steps are described in fig 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 [23]. 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 tokenization, word segmentation and part-of-speech tagging. In tokenization, input text is divided into units called tokens where each is either a word or something else which make the word boundary for each syllable.Then the tokenized sentence is processed again by initial segmentation with using Forward Maximum Matching Algorithm [24]. 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, Bigram Part-of-Speech Tagger [22] 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 [23].

By aligning the analyzed text resulted from Analyzer, parallel tagged aligned corpus is created. Our tagged align corpus format can be seen in fig 2.

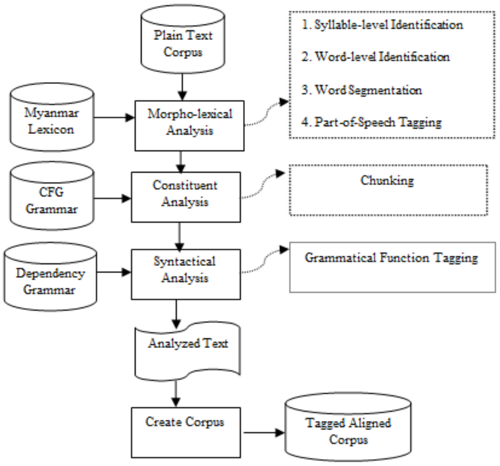
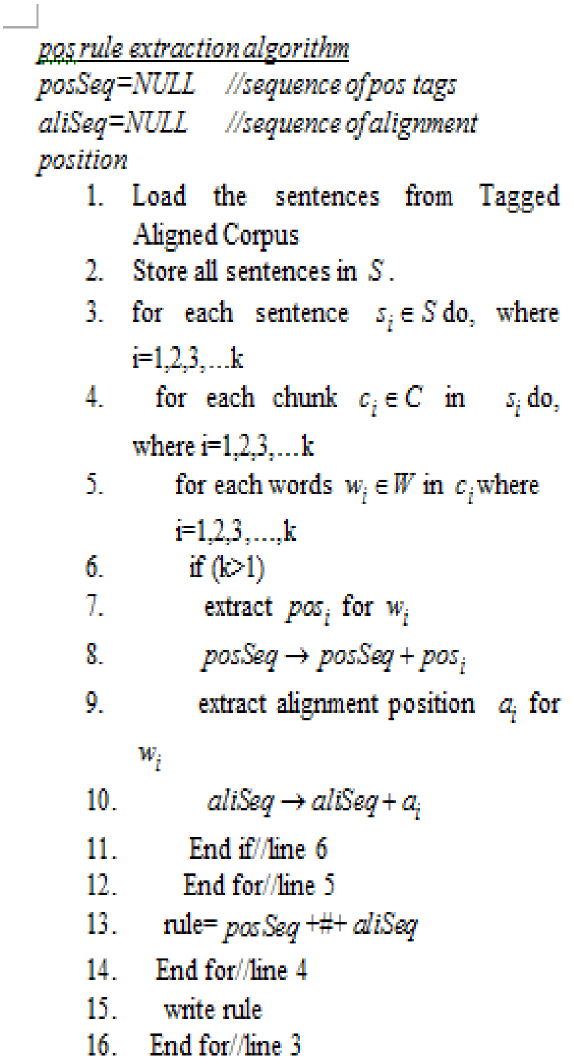


Fig. 1. Corpus Creation Steps

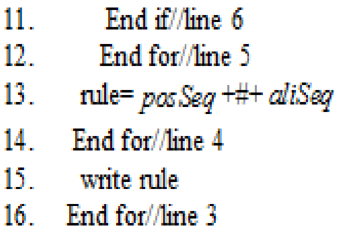
As we can see in fig 2, “Subj”,”Obj” , “Verb” and “Sf” are function tags of each chunk. “NC”,”VC” and “SFC” refer the relevant chunk type and “PRN.Person”, "NN.Objects", and "Part.Type" are part of speech of each word. The numbers in the parentheses are alignment position of function tags and part of 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. Each chunk is separated by "#".

5 REORDERING RULE EXTRACTION

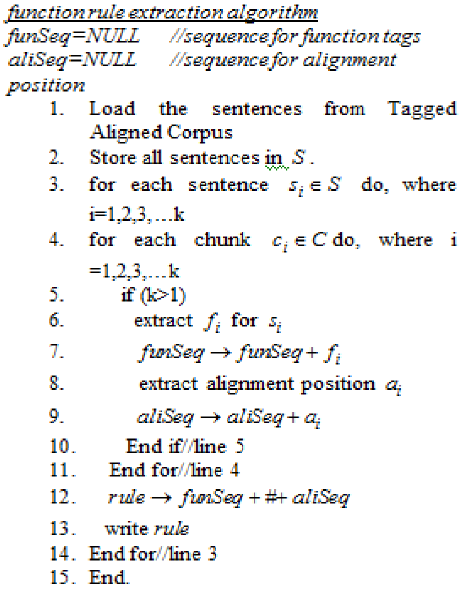
By using the linguistic information from the corpus, two kinds of reordering rules are generated automatically. They are function tags-based reordering rules and part-of-speech tags-based reordering rules. The former is generated for using in chunk-level reordering and the latter is for using word-level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



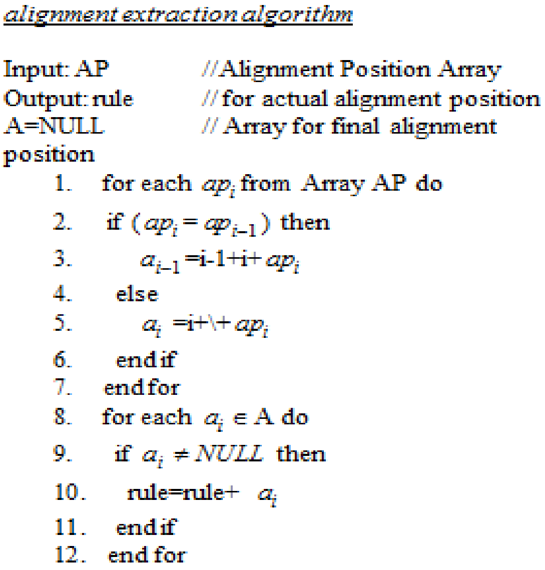
```
pos rule extraction algorithm
posSeq=NULL    //sequence of pos tags
aliSeq=NULL    //sequence of alignment
position
1.  Load the sentences from Tagged
    Aligned Corpus
2.  Store all sentences in $S$.
3.  for each sentence $s_i \in S$ do, where
    i=1,2,3,…k
4.    for each chunk $c_i \in C$ in $s_i$ 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
```



```
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
```



```
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.
```



```
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
```

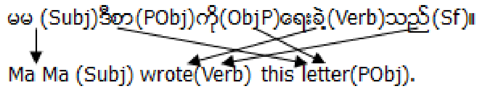
5.1 Reordering Rule Generation

In order to generate reordering rules for English-Myanmar translation, there are two main cases that cause the same pattern with different reorderings. The first case is caused by active form translation and passive form translation. The second case is caused by different translations.

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

Example (4),

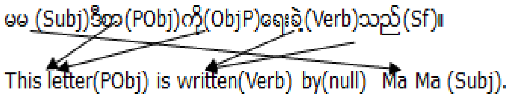
(1)



မမ (Subj)ဒီစာ(PObj)ကို(ObjP)ရေးခဲ့(Verb)သည်(Sf)။
Ma Ma (Subj) wrote(Verb) this letter(PObj).

- *Thinn Thinn Wai, University of Computer Studies, Yangon, Myanmar, PH-095-09-5059987. E-mail: thin2wai@gmail.com*
- *Ni Lar Thein , University of Computer Studies, Yangon, Myanmar. E-mail: nilarthein@gmail.com*

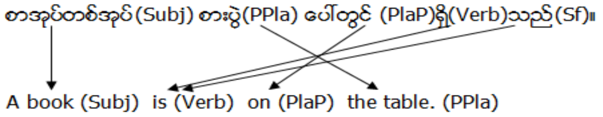
(2)


မမ (Subj)ဒီစာ(PObj)ကို(ObjP)ရေးခဲ့(Verb)သည်(Sf)။
This letter(PObj) is written(Verb) by(null) Ma Ma (Subj).

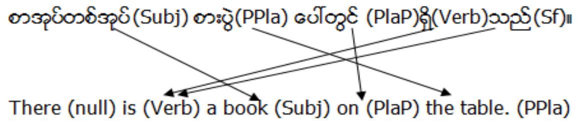
In this example, the Myanmar Sentence" မမဒီစာကိုရေးခဲ့သည်။ "can be translated in two ways; active form and passive form. According to active form translation, the translated English sentence is "Ma Ma wrote this letter.". In passive translation, this sentence is translated into "This letter is written by Ma Ma". According to active and passive translation, this Myanmar sentence has two reordering rules.

In the second case, some Myanmar sentences such as " စာအုပ်တစ်အုပ်စားပွဲပေါ်တွင်ရှိသည်။ "have two different translations such as "A book is on the table" and "There is a book on the table." According to these two different traslation, two different reordering rules are obtained. This can be seen in Example (5),
Example (5),

(1)


စာအုပ်တစ်အုပ်(Subj) စားပွဲ(PPla) ပေါ်တွင် (PlaP)ရှိ(Verb)သည်(Sf)။
A book (Subj) is (Verb) on (PlaP) the table. (PPla)

(2)


စာအုပ်တစ်အုပ်(Subj) စားပွဲ(PPla) ပေါ်တွင် (PlaP)ရှိ(Verb)သည်(Sf)။
There (null) is (Verb) a book (Subj) on (PlaP) the table. (PPla)

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, PObj, ObjP, Verb, Sf#0/0, 1+2/2, 3+4/1:7(10)
-Subj, PObj, ObjP, Verb, Sf#0/2, 1+2/0, 3+4/1:3(10)
-Subj, PPla, PlaP, Verb, Sf#0/0, 1/3, 2/2, 3+4/1:4(10)
-Subj, PPla, PlaP, Verb, Sf#0/1, 1/3, 2/2, 3+4/0:6(10)

- Rules using pos tag

-NN.Person, JJ#0/1, 1/0:10(10)
-NN.Objects,CRD,Part.Type#0/1, 1+2/0:10(10)
-VB.Common, Part.Support#0+1/0:10(30)
-VB.Common, Part.Support#0/1, 1/0:20(30)

In the above rules, Subj, PObj, ObjP, Verb, Sf, PPla and PlaP are function tags and NN.Person, JJ, NN.Objects, CRD, Part.Type, VB.Common, and Part.Support are POS's tags. Therefore, "Subj, PObj, ObjP, Verb, Sf" is function rule pattern and "NN.Person, JJ" is POS rule pattern. The string of numbers between "#" and ":"is the position of source and target words and source word position is divided by "/" with target word position. For example, in the rhs of the first pos rule pattern"0/1, 1/ 0", 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 part-of-speech tag rule, the verb particle part-of-speech tag (Part.Support) is not in English and so there is no alignment. This tag missing can cause the error in translation and so this tag is needed to add beside its relative verb to allievate the translation error caused by tag missing. In this reordering rule generation, this problem is solved by combining soure tag positions which have same target alignment postions. This can be seen in the third part-of-speech tag rule described above. In this rule, the string after #, "0+1/0" means that the words at position "0"and "1" are move together into the position "1" because they have the same target position .

The sequences "Subj, PObj, ObjP, Verb, Sf" and "VB.Common, Part.Support" 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 observed and (inside brackets) the number of occurrences of the rule pattern. The probability of each reordering suggestion is computed as in (1).

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

6 Experiments

These generated reordering rules are tested on the Myanmar-English 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 Myanmar-English machine translation. Besides, they can be combined with mathematical models to create reordering model for Myanmar-English translation. By using these rules as an embedded component, Myanmar-English translation system can perform translation effectively and efficiently.

6.1 Accuracy of Reordering Rules

The purpose of this experiment was to see how many reordering rules are accurate when they are applied to the test set. The test set was obtained randomly from High School Myanmar Grammar book. The test set was split into two subsets:

- 1000 simple sentences

- 1000 compound sentences

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

Table 1 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. Moreover, descriptions of some function tags and pos tags are described in Table 2 and 3.

TABLE 1
ACCURACY OF REORDERING RULES

English test subsets	**Accuracy**
Simple sentences	98.9%
Complex sentences	97.2%

7 Conclusion And Future Work

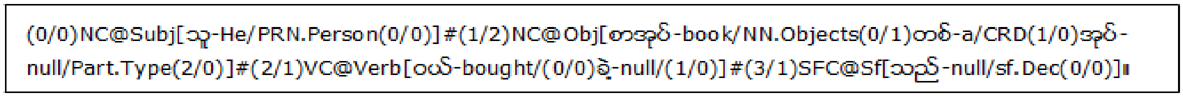
This paper proposes automatic reordering rules generation algorithms for Myanmar-English machine translation.These algorithms use the parallel tagged aligned corpus as a resource. These rules are extracted based on the part-of-speech tags and function tags extracted from Chunk based Analyzer. These rule 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 simple sentences and complex sentences and my future work is to generate reordering rules for more complex Myanmar sentences and the to implement the novel Myanmar-English reodering model effectively.

TABLE 2
SOME PART-OF-SPEECH TAG DESCRIPTIONS

Part-of-speech Tag	**Description**
JJ	Adjective
NN.Person	Noun indicates person
PRN.Person	Personal pronoun
RB.Manner	Adverb of Manner
Sf.Dec	Declarative Sentence Final
VB.Common	Common Verb
NN.Objects	Noun indicates objects
CRD	Cardinal Number
Part.Type	Particle of Cardinal Number
Part.Support	Support Particle

TABLE 3
SOME FUNCTION TAGS DESCRIPTIONS

Function Tag	**Description**
Subj	Subject
PObj	Direct object of preposition
ObjP	Preposition of direct object.
Sf.Dec	Declarative Sentence Final
PPla	Place of Preposition
PlaP	Preposition of Place
Sf	Sentence Final



(0/0)NC@Subj[သူ-He/PRN.Person(0/0)]#(1/2)NC@Obj[စာအုပ်-book/NN.Objects(0/1)တစ်-a/CRD(1/0)အုပ်-null/Part.Type(2/0)]#(2/1)VC@Verb[ဝယ်-bought/(0/0)ခဲ့-null/(1/0)]#(3/1)SFC@Sf[သည်-null/sf.Dec(0/0)]။

Fig. 2. Parallel Tagged Aligned Corpus

References

[1] C. Tillmann and H. Ney. 2002, "Word reordering and DP beam search for statistical machine translation to appear in Computational Linguistics.," *Neurocomputing – Algorithms, Architectures and Applications,* F. Fogelman-Soulie and J. Herault, eds., NATO ASI Series F68, Berlin: Springer-Verlag, pp. 227-236, 1989. (Book style with paper title and editor)

[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 As-soc. 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.

[22] Phyu Hnin Myint, Tin Myat Htwe and Ni Lar Thein. "Bigram Part-of-Speech Tagger for Myanmar Language", Proceedings of International Conference on Information Communication and Management (ICICM 2011), October 14-16, 2011, Singapore.

[23] Win Win Thant,Tin Myat Htwe and Ni Lar Thein ." Syntactic Analysis of Myanmar Language", Proceedings of International Conference on Computer Applications (ICCA 2011), Yangon, Myanmar, May 5-6, 2011.

[24] Win Pa Pa and Ni Lar Thein. "Myanmar Word Segmentation using Hybrid Approach." In Proc. 7th International Conference for Computer Application. Yangon, Myanmar, May 5-6, 2009..