Automatic Phrase Reordering Approach for Machine Translation

Myat Thuzar Tun, Ni Lar Thein
University of Computer Studies, Yangon, Myanmar
myathuzar@gmail.com, *nilarthein@mptmail.net.mm*

Abstract

In this paper, we firstly discuss some problems in Machine Translation. We summarize the aims of proposing the phrase reordering model. Next, we present the proposed reordering model that can be incorporated into the Statistical Machine Translation System (SMTS) and working steps of reordering model. Moreover, we discuss how source language reordering model can assist MTS.

Keywords: Machine Translation, Phrase Reordering, Phrase Jumping

1. Introduction

Translation is the process of moving texts from one language (source language) to another (target language). Machine Translation automates the process of fully automatic translation or computer-aided (human) translation [15]. Nowadays, there has been a lot of work on Machine Translation. Some difficulties in MT still remain. We propose the target phrase reordering approach to incorporate Machine Translation System for making translation easier.

Natural Language texts are often written using ambiguous sentences, and human as well as computers may encounter some difficulties in understanding and translating them. Sentences which are not properly disambiguated are likely to be translated incorrectly, leading to a corresponding increase in the amount of post editing required [7]. Although filters provide a model for the description of disambiguation, they are inefficient for many applications because all possible parse trees for a sentence have to be built before the intended ones are selected. The number of possible parse trees grows exponentially with the length of strings [14]. Successful translation can be achieved by stylizing the target language used [13]. Moreover, preservation of the syntactic structure of source texts in the target text is one of a number of dimensions along which one can make a judgment of the quality for MT output [6]. In addition to these facts, restriction to source texts to a particular type of construction reduces the need for post-editing or sometimes be translated without the need for post-editing [13].

We propose an approach to reorder chunked phrases of the source language before full parsing. This can be used to prevent parse steps that lead to parse trees that would be removed by the filter after parsing. This may be the first step to phrase based Machine Translation. Chunking divides text into segments which correspond to a certain syntactic units such as noun phrases, verb phrases etc. The approach will be a rule based one. We use Context Free Grammar (CFG) rules as the base for chunking. Moreover, we use heuristics rules to identify a subject, object and indirect object for each verb which is not an auxiliary. We propose this model with the objectives: to reduce parses disambiguation, to reduce time for target language post editing and to get efficient translation by providing an appropriate sentence structure for translating to the target language.

In this paper, section 1 introduces our approach. Brief explanation of related work with our approach is in Section 2. In section 3 we discuss some problems in MT system. Section 4 presents the proposed reordering model and working steps for reordering. Finally we make a conclusion for our reordering model in section 5.

2. Brief Review of Related Researches

A syntax-based algorithm that automatically builds Finite State Automata from semantically equivalent translation sets was described in [1].[6] described an experimental graphical human/computer interface and showed a new approach to machine translation for monolingual, which supports interactive foreign-language generation and post-editing. In the paper [8], KANT controlled language rewriting architecture, which combines a rewriting engine with interactive dialogue, was presented. C. Munteanu [10] presented an experimental evaluation of automatic disambiguation strategies which could eliminate the need for interactive disambiguation in Machine Translation System. B. Pang et al. [12] presented recent developments of an indexing technique aimed at improving parsing times. The paper [13] showed how to generate parse tree for the English sentence using a C# port of Open NLP.

2. Machine Translation

Translation is the process of moving texts from one language (source language) to another (target language). Machine translation automates the process of fully automatic translation or computer-aided (human) translation. MT is a difficult task because languages are vastly different in:

1. Lexically (the words they use)
2. Syntactically (the construction they allow) and
3. Semantically (the way meaning works) [9].

Types of Ambiguities in Machine Translation

- Lexical ambiguity (a word in a sentence has more than one possible meaning)
- Structural ambiguity (a sentence can produce more than one parse tree)
- Semantic ambiguity (a sentence has different meanings) [9]

Ways to assist Machine Translation

- Restricting input texts to a particular type of construction
- Using a unification grammar to produce legal set of grammatical functional structure for the input [9]
- Providing a syntax structure of the source text which preserves the target text [2]

These ways also can reduce the effect of syntactical difference of languages and structural ambiguity in Machine Translation.

4. Proposed Reordering Method

4.1. Primary Works for the Approach

Specifying Sentence Structure

A sentence structure which is secure for translating target language is specified. A sample sentence structure is presented in table 2. This structure is highly depends on the target language and need efficient linguistic knowledge to decide appropriate structure.

Extracting Rules

Initially an English corpus is taken and it is divided into two or more sets. One of these divided sets will be used as training data. The training data set is taken and manually chunk for phrases. For each sentence in the corpus, the syntactic structure is built by a skeleton tree. Figure 1 shows an example of a sentence. This sentence is ***"we are at the top of a hill".***

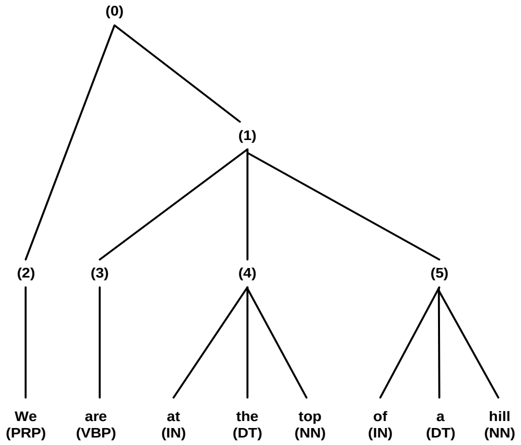

Figure 1.Skeletal Tree for Example Sentence

Each node other than any leaf in a skeletal tree can be transformed into a production rule that has the node itself on the left hand side (LHS) and the immediate children on the right hand side (RHS). Table 1 shows the rules extracted from the skeletal tree shown in figure 1. Each bracketed number in the rules corresponds to an intermediate node in figure 1.

<5>	⟶	of (IN)	a (DT)	hill (NN)
<4>	⟶	at (IN)	the (DT)	top (NN)
<3>	⟶	are (VBP)		
<2>	⟶	we (PRP)		
<1>	⟶	<3>	<4>	<5>
<0>	⟶	<2>	<1>	

Table 1.The Rules Derived from the Skeletal Tree

In order to transform these rules into CFG rules, we need to assign appropriate non-terminal symbols to the bracketed numbers in table 1. When labeling the intermediate nodes, we have to take account of the head of a phrase. We identify the head from amongst the children. For example, we assign <4> "PNP", in other word, "prepositional phrase" because the head of this phrase is at (IN). We have to assign the same symbol to <4> in fifth rule in table 1 at the same time, since it denotes the same node in figure 1.

These rules will serve as the base for chunking. The chunker program will use these rules and will chunk the test data. Precision and recall are calculated for this and the result will be analyzed to check if more rules are needed to improve the coverage of the system. If more rules are needed then additional rules are added and the same process as mentioned above is repeated to check for increase in the precision and recall of the system. The system can then be tested for various other applications.

4.2. Reordering Architecture

There are 6 working steps in phrase reordering as shown in Figure 2. These are

1. Tokenizing
2. Part-of-speech Tagging
3. Chunking
4. Subject/ Object detecting
5. Analyzing and
6. Reordering

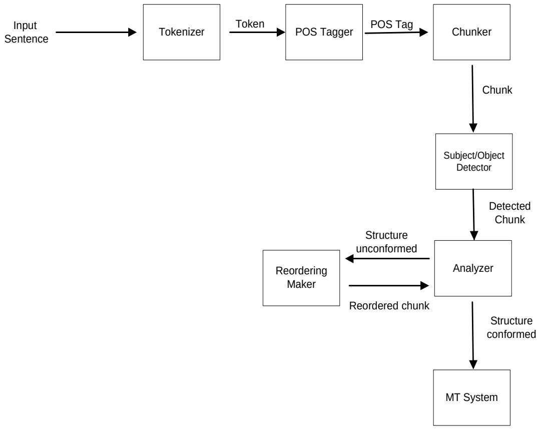

Figure 2.Reordering Model Architecture

We would like to illustrate these working steps with the following sample input text in figure 3 and predefined sentence structure shown in table 2.

We are at the top of a hill. From here we can see the roof of our school.

Figure 3.Sample Input Text

4.2.1. Tokenizing

1. Input text is cut into sentences using sentence boundary '.' as follow:

We are at the top of a hill.

From here we can see the roof of our school.

2 .The sentences are split into tokens.
We/are/at/the/top/of/a/hill/.

From/here/we/can/see/the/roof/of/our/school/.

4.2.2. Part-of-Speech Tagging

Each word in a sentence is labelled with its appropriate part of speech using lexical information.

We/PRP are/VBP at/IN the/DT top/NN of/IN a/DT hill/NN ./.

from/IN here/ADV we/PRP can/AUX see/VB the/DT roof/NN of/IN our/PRP$ school/NN ./.

4.2.3. Chunking

Chunking can also serve as a possible first step for full parsing only need to look at part-of-speech tags (ignore lexical content. In this step input text is divided into segments which correspond to certain syntactic unit. Each chunk corresponds to a syntactic unit such as a noun phrase or adverb phrase.

We get the chunk sequences shown in figure 4 for each of two input sentences.

[*NP we/PRP*] [*VP are/VBP*] [*PNP at/IN the/DT top/NN] [PNP of/IN a /DT hill/NN*] *./.*

[*ADVP from/IN here/ADV*] [N*P we/PRP*] [*VP can/AUX see/VB*] [N*P the/DT roof/NN*] [PNP *of/IN our/PRP$ school/NN*] *./.*

Figure 4.Sequence of Chunks for Input Sentences

4.2.4 Subject/Object Detecting

Depending on our purpose of phrase reordering approach, we need to identify noun phrases whether subject or object using the following heuristics rules in addition to dividing syntactical unit.

1. Case: Pronouns have different forms according to case (eg. he & him, we & us).
2. Agreement: A finite verb agrees with its subject while it doesn't agree with its object.
3. Position: Subjects generally precede predicates and complements follow.

Finally, we get the labeled chunk sequence as shown in figure 5.

[*SP_1 we/PRP*] [*VP_1 are/VBP*] [*PNP at/IN the/DT top/NN] [PNP of/IN a /DT hill/NN*] *./.*

[*ADVP from/IN here/ADV*] [*SP_1 we/PRP*] [*VP_1 can/AUX see/VB*] [O*P_1 the/DT roof/NN*] [PNP *of/IN our/PRP$ school/NN*] *./.*

Figure 5. Subject/ Object Detector Output for Sequence of Chunks in Figure 4

Subject and object phrases are marked by [*SP_i...]* and *[OP_i…]* respectively. ' i ' is an integer number that indicated that the noun phrase is the subject/object of the verb phrase with the same index [V*P_i...]*.

4.2.5. Analyzing

The sentence structure we predefined and which is suitable for translating to target language is as in Table 2.

Subject Phrase	*Subject Modifi-ed Phrase*	*Verb Phrase*	*Object phrase*	*Object Modifi-ed Phrase*	*Adverb phrase*

Table 2.A Sample Predefined Sentence Structure

The predefined sentence structure is a sequence of phrases with a modification structure which can be described by a diagram as in figure 6.

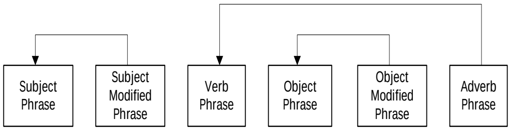

Figure 6.Predefined Modification Structure

For a sequence of phrases $x_1, x_2, ..., x_n$ to conform to predefine structure, it must have a structure satisfying the following constraints:

Constraint 1: For any i (x_i is subject phrase), j (x_j is verb phrase) , k (x_k is object phrase) , $(1 \le i < j < k \le n)$, such that the phrase order is Subject Verb Object (SVO).

Constraint 2: For any i $(x_i$ is subject (or) object phrase) and $(x_j$ modifies x_i), the phrase sequence is in the form of i, j, such that the modifier immediately follows the phrase intended to modify.

Constraint 3: If x_j is an adverb phrase, n=j such that x_j is the last phrase in phrase sequence.

According to the above constraints and chunk sequence in figure 5, 1st sentence has conformed sentence structure but 2nd has not. And so Analyzer determines to reorder the chunk sequence of 2nd sentence.

4.2.6. Reordering

Reordering model permutes input phrases x_1, x_2,…,x_K into phrase sequence y_1,y_2,…,y_K and then sends back to analyzer to check if the sentence structure is conformed. This model can be built using first order Markov process with a single parameter that controls the degree of movement. The phrase alignment sequence m_1^k specifies a reordering of phrases into predefined phrase order; the words within the phrases remain in the source language order.

$$P(m_1^K \backslash x_1^K, K, g_1^K) = P(m_1^K \backslash x_1^K)$$

$$= \prod_{k=1}^{K} P(m_k \backslash x_k, \Phi_{k-1})$$

Where g_1,g_2,….,g_K is the predefined sentence structure, Φ_{k-1} is the state arrived by m_1^{k-1} and m_k is movement parameter for phrase x_k. And so y_k is determined by x_k and m_k. Displacement of kth phrase x_k is

$$x_k \to y_k + m_k, k \in \{1,2,..., K\}$$

The jump sequence m_1^K is constructed such that y_1^K is the permutation of x_1^K.

The simple ways to swap phrases are

(i) Adjacent phrases (jump 1 phrase)
(ii) Within three phrases (jump 2 phrase)

Figure 7 and 8 show the values of movement parameter when we make a swapping.

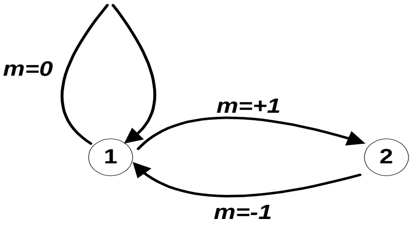

Figure 7.Jumping 1 Phrase (adjacent)

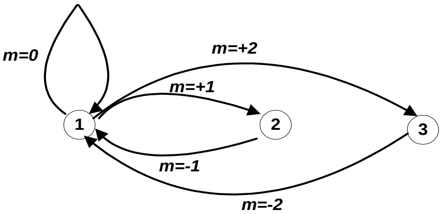

Figure 8.Jumping 2 Phrases (three phrases window)

Now we use jumping 1 phrase method for reordering. Figure 9 illustrates reordering steps to make input chunk sequence conformed to predefined sentence structure. For using 1 phrase jumping method, the value allowed for movement parameter m$_k$ is $m_k \in \{-1,0,1\}$. And which has two equivalence states for any history m_1^{k-1}; $\phi(m_1^{k-1}) \in \{1,2\}$. A jump of +1 has to be followed by a jump of -1 and 1 is the start and end state and so $\sum_{k=1}^{K} m_k = 0$

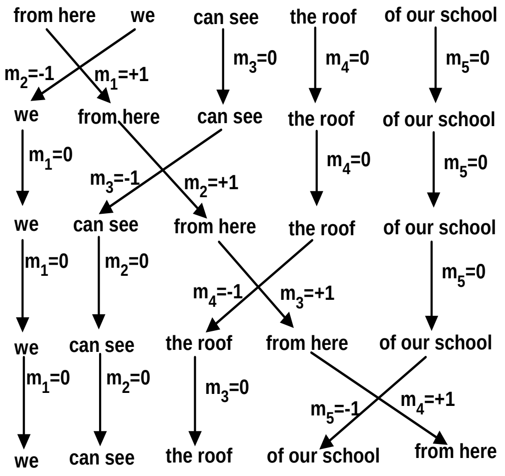

Figure 9.Phrase Reordering and Jumping Sequences

5. Conclusion

In this paper, we have presented an approach for automatic phrase reordering which can be incorporated to Machine Translation System. We use array structure representations for tokens, POS tags and chunks. We use no tree representation and it is sure that output phrases sequences are valid permutation of input phrase sequences. Our approach can reduce time for target language post-editing by providing unified source sentence structure. We can also predefine secure source sentence structure to encourage clear writing which can improve the quality of the source text and the translation output. Our approach doesn't concern with compound sentences. Future work will build more powerful models which can transform compound sentences into simple sentences conformed to the predefined sentence structure.

References

[1]. P. Abbeel and Andrew Y.Ng, “Learning first-order Markov models for control”, *Computer Science Department Stanford University.*

[2]. D.Freudenthal and J.Pine, "Resloving Amgiguities in the Extraction of Syntactic Categories through Chunking", *Proceedings of the sixth International Conference on Cognitive Modelling, 94-99.*

[3]. K.Hacioglu, "A Lightweight Semantic Chunking Model Based On Tagging", *Center for Spoken Language Research.*

[4]. V. Kodaganallur, "Incorporating Language Processing into Java Applications: A JavaCC Tutorial", *Published by the IEEE Computer Society.*

[5]. C. D. Manning and Hinrich Schutze "Foundations of Statistical Natural Language Processing", *The MIT Press, Cambridge, Massachusetts, London, England.*

[6]. T. Mitamura and E. Nyberg, “Automatic Rewriting for Controlled Language Translation”, *Proceeding of the NLPRS 2001.*

[7]. T. Mitamura, E. Nyberg, , E. Torrejon and R. Lgo, " Mutliple Strategies for Automatic Disambiguatin in Technical Translation", *Proceeding of the 8th International Conference on Theoretical and Methodological Issues in Machine Translation, Chester, UK.*

[8]. W.G. Mitchener, “Mathematical Models of Word Order Change in Middle English”, *August 2004.http://www.math.duke. edu*

[9]. A. Molina and F. Pla, "Shallow Parsing using Specialized HMMS", *Journal of Machine Learning Research 2(2002) 595-613.*

[10]. C. Munteanu," Indexing Methods for Efficient Parsing", *Proceedings of HLT-NAACL 2003.*

[11]. M. Osgorne, "Shallow Parsing as Part-of-Speech Tagging", *Proceedings of CoNLL-2000 and LLL-2000, PP 145-147, Lisbon, Portugal, 2000*

[12]. B. Pang, K. Knight and D. Marcu," Syntax-based Alignment of Multiple Translations: Extraction Paraphrases and Generating New Sentences", *Department of Computer Science Cornell University.*

[13]. Russell and Norvig, "Practical Natural Language Processing", *Artificial Intelligence: A Modern Approach, 1995, chapter.*

[14]. E. Visser, "A Case Study in Optimizing Parsing Schemata by Disambiguation Filters", *In International Workshop on Parsing Technology (IWPT'97), pages 210--224, Boston, USA, September 1997. Massachusetts Institute of Technology*

[15]. Y.Zhaing and Q. Zhoug,"Chinese Base-Phrases Chunking", *State key Laboratory of Intellignet Technology and Systems Department and Computer Science and Technology, Tsinghua University , China.*

Appendix

ADV=Adverb

AUX=Auxiliary Verb

DT=Determiner

IN=Preposition

NN=Noun

PRP$= Possessive Pronoun

PRP= Personal Pronoun

singular present

VB=Verb, base form

VBP= Verb, non 3rd –person

ADVP=Adverb Phrase

NP=Noun Phrase

OP= Object Phrase

PNP= Prepositional Phrase

SP=Subject Phrase

VP=Verb Phrase