Reducing Grammar Errors for Translated English Sentences

Nay Yee Lin[1], Khin Mar Soe[2], and Ni Lar Thein[1]

[1] University of Computer Studies, Yangon, Myanmar
[2] Natural Language Processing Laboratory
University of Computer Studies, Yangon, Myanmar
{nayyeelynn,nilarthein,kmsucsy}@gmail.com

Abstract. One challenge of Myanmar-English statistical machine translation system is that the output (translated English sentence) can often be ungrammatical. To address this issue, this paper presents an ongoing grammar checker as a second language by using trigram language model and rule based model. It is able to solve distortion, deficiency and make smooth the translated English sentences. We identify the sentences with chunk types and generate context free grammar (CFG) rules for recognizing grammatical relations of chunks. There are three main tasks to reduce grammar errors: detecting the sentence patterns in chunk level, analyzing the chunk errors and correcting the errors. Such a three level scheme is a useful framework for a chunk based grammar checker. Experimental results show that the proposed grammar checker can improve the correctness of translated English sentences.

Keywords: Statistical machine translation, grammar checker, context free grammar.

1 Introduction

Grammar checking is one of the most widely used tools within natural language processing applications. Grammar checkers check the grammatical structure of sentences based on morphological processing and syntactic processing. These two steps are parts of natural language processing to understand natural languages. Morphological processing is the step where individual words are analyzed into their components and non-word tokens, such as punctuation. Syntactic processing is the analysis where linear sequences of words are transformed into structures that show grammatical relationships between the words in the sentence [10]. The proposed grammar checker determines the syntactical correctness of a sentence.

There are several approaches for Grammar checking such as syntax-based checking, statistics-based checking and rule-based checking [2]. Among them, we build a chunk based grammar checker by using statistical and rule based approach. In this approach, the translated English sentence is used as an input. Firstly, this input sentence is tokenized and tagged POS to each word. Then these tagged words are

D.-S. Huang et al. (Eds.): ICIC 2011, LNBI 6840, pp. 299–306, 2012.

grouped into chunks by parsing the sentence into a form that is a chunk based sentence structure. After making chunks, these chunks relationship for input sentence are detected by using sentence patterns. If the sentence pattern is incorrect, the system analyzes chunk errors and then corrects the grammar errors. The system has currently trained on about 6000 number of sentence patterns for simple, compound and complex sentence types.

This paper is organized as follows. Section 2 presents the overview of Myanmar-English Statistical Machine Translation System. In section 3, the proposed system is described. Section 4 reports the experimental results and finally section 5 concludes the paper and describes future work.

2 Overview of Myanmar-English Statistical Machine Translation System

Myanmar-English statistical machine translation system has developed source language model, alignment model, translation model and target language model to complete translation. Among these models, our proposed system builds target language model to check the grammar errors of translated English sentences.

Input for Myanmar-English machine translation system is Myanmar sentence. After this input sentence has been processed in three models (source language model, alignment model and translation model), translated English sentence is obtained in target language model. However, this sentence might be incomplete in grammar because the syntactic structures of Myanmar and English language are totally different. For example, after translating the Myanmar sentence "စားပွဲပေါ်တွင် စာအုပ်တစ်အုပ် ရှိသည်။", the translated English sentence might be "*is a book on table.*". This sentence has missing words "*There*" and "*the*" for correct English sentence "*There is a book on the table.*". As an another input "သူသည် ရေတစ်ခွက် သောက်နေသည်။", the translated output is "*He is drinking a cup water.*". In this sentence, "of" (preposition) is omitted from "*a cup of water*". These examples are just simple sentence errors. When the sentence types are more complex, reducing grammar errors and correction are more needed. There are many English grammar errors to correct ungrammatical sentences. At present, this grammar checker detects and provides the following errors according to the translated English sentences:

- If the sentence has missing words such as preposition (PPC), conjunction (COC), determiner (DT) and existential (EX) then this system suggests the required words according to the chunk types.
- In Subject-Verb agreement rule, if the subject is plural, verb has to be the plural. We check the verb agreement according to the person and number of the object.
- Sentence can contain inappropriate determiner. Therefore grammatical rules have been identified several kinds of determiner for appropriate noun.
- Translated English sentences can have the incorrect verb form. The system has to memorize all of the commonly used tenses and suggest the possible verb form.

3 Proposed System

There are very few spelling errors in the translation output, because all words are come from the corpus of the SMT system. Therefore, this system proposes a target-dominant grammar checking for Myanmar-English machine translation system as shown in Fig. 1.

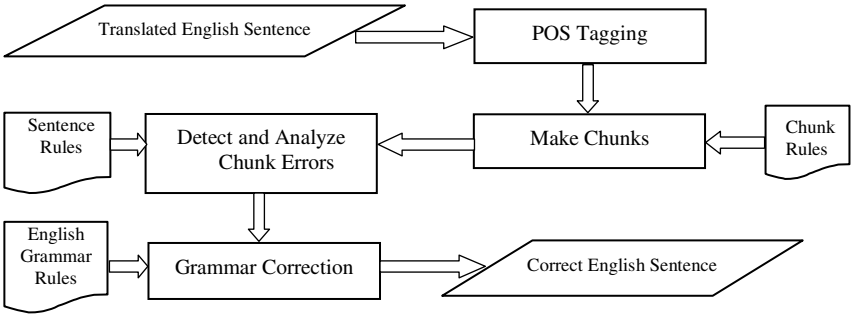

Fig. 1. Overview of Proposed System

3.1 Part-of-Speech (POS) Tagging

POS tagging is the process of assigning a part-of-speech tag such as noun, verb, pronoun, preposition, adverb, adjective or other tags to each word in a sentence. Nouns can be further divided into singular and plural nouns, verbs can be divided into past tense verbs and present tense verbs and so on [5].

POS tagging is the main process of making up the chunks in a sentence. There are many approaches to automate part of speech tagging. This system tags each word by using Tree Tagger which is a Java based open source tagger. However, it often fails to tag correctly some words when one word has more than one POS tags. In this case, refinement of POS tags for these words is made by using the rules according to POS tags of the neighbor words. The example for refinement tags is shown in Table 1.

Table 1. Example of Refinement Tags

Example	Incorrect Tag	POS tags of neighbor words	Refine Tag
He ***bit*** a rope.	***bit***=NN	Previous tag is PP	***bit***=VBD
He is a ***tailor***.	***tailor***=VB	Previous tag is DT	***tailor***=NN

3.2 Making Chunks

A chunk is a textual unit of adjacent POS tags which display the relations between their internal words. Making chunks is a process to parse the sentence into a form that is a chunk based sentence structure. Input English sentence is made in chunk structure by using hand written rules. It represents how these chunks fit together to form the constituents of the sentence.

Context Free Grammar (CFG): Context Free Grammars constitute an important class of grammars, with a broad range of applications including programming languages, natural language processing, bio informatics and so on. CFG's rules present a single symbol on the left-hand-side, are a sufficiently powerful formalism to describe most of the structure in natural language.

A context-free grammar G = (V, T, S, P) is given by

- A finite set V of variables or non terminal symbols.
- A finite set T of symbols or terminal symbols. We assume that the sets V and T are disjoint.
- A start symbol $S \in V$.
- A finite set $P \subseteq V \times (V \cup T)^*$ of productions.

A production (A, α), where $A \in V$ and $\alpha \in (V \cup T)^*$ is a sequence of terminals and variables, is written as $A \rightarrow \alpha$. Context Free Grammars are powerful enough to express sophisticated relations among the words in a sentence. It is also tractable enough to be computed using parsing algorithms [9]. NLP applications like Grammar Checker need a parser with an optional parsing model. Parsing is the process of analyzing the text automatically by assigning syntactic structure according to the grammar of language. Parser is used to understand the syntax and semantics of a natural language sentences confined to the grammar. There are two methods for parsing such as Top-down parsing and Bottom-up parsing [3]. In this system, Bottom-up parsing is used to parse the sentences.

Chunking or shallow parsing segments a sentence into a sequence of syntactic constituents or chunks, i.e. sequences of adjacent words grouped on the basis of linguistic properties [7]. The system has used ten general chunk types for parsing. Some chunk types are subdivided into more detail chunk levels such as common chunk types for sentence patterns and specific chunk levels as shown in Table 2.

We make the chunk based sentence structure by assembling POS tags using CFG based chunk rules. For a simple sentence "A young man is reading a book in the library." is chunked as follows:

POS Tagging:

A [DT] young [JJ] man [NN] is [VBZ] reading [VBG] a [DT] book [NN] in [IN] the [DT] library [NN] . [SENT]

Making Chunks:

[DT][JJ][NN] [VBZ] [VBG] [DT] [NN] [IN] [DT] [NN] [SENT]

NCB1_ [VBZ][VBG] [DT] [NN] [IN] [DT] [NN] [SENT]

NCB1 _ PRV1 _ [DT] [NN] [IN] [DT] [NN] [SENT]

NCB1 _ PRV1 _ NCB1 _ [IN] [DT] [NN] [SENT]

NCB1 _ PRV1 _ NCB1 _ PPC_ [DT] [NN] [SENT]

NCB1 _ PRV1 _ NCB1 _ PPC_ NCB1 _ [SENT]

NCB1 _ PRV1 _ NCB1 _ PPC_ NCB1 _ END

Chunk Based Sentence Pattern:

S = NCB1_PRV1_NCB1_PPC_NCB1_END

Table 2. Proposed Chunk Types

General Chunk Types	Common Chunk Types	Description	Specific Chunk Types	Example Words
NC	NCS1	Singular Noun Chunk for Subject only	PN1	He, She
	NCS2	Plural Noun Chunk for Subject only	PN2, PNC1	They, We all
	NCS	Singular and Plural Noun Chunk for Subject only	NEC	There
	NCB1	Singular Noun Chunk for both Subject and Object	ANN1,DNN1	A boy, This car
	NCB2	Plural Noun Chunk for both Subject and Object	ANN2,DNN2	The boys, These girls
	NCB	Singular and Plural Noun Chunk for both Subject and Object	NDC	This, These, Those
	NCO	Singular and Plural Noun Chunk for Object only	PN3	him, them, us
VC	PAVC	Past Tense Verb Chunk for both singular and plural noun	DV	wrote, gave
	PAV1	Past Tense Verb Chunk for singular noun	VDZ	was
	PAV2	Past Tense Verb Chunk for plural noun	VDP	were
	PRV1	Present Tense Verb Chunk for singular noun	ZV	is, writes, gives
	PRV2	Present Tense Verb Chunk for plural noun	PV	are, write, give
	FUVC	Future Tense Verb Chunk for singular and plural noun	MVB	will go, will be
TC	TC1	Time Chunk for Present	ADV1	Now, Today
	TC2	Time Chunk for Past	ADV2	Yesterday, last
	TC3	Time Chunk for Future	ADV3	Tomorrow, next
COC	XC	Subordinated Conjunction Chunk	NPR,NDT	Which, who, that
	CC	Coordinated Conjunction Chunk	COC	And, but, or
INFC	INC	Infinitive Chunk with Noun Chunk	IN1	to market, to school
	IVC	Infinitive Chunk with Verb Chunk	IBV	to go, to give
AC	AC	Adjective Chunk	R2A1,A1	more beautiful, old
RC	RC	Adverb Chunk	R1	usually, quickly
PTC	PTC	Particle Chunk	PTC	up, down
PPC	PPC	Prepositional Chunk	PPC	at, on, in
QC	QC	Question Chunk	QDT, QRB	Which, Where

3.3 Detecting Sentence Patterns and Analyzing Chunk Errors

After making chunks, these chunks relationship for input sentence are detected and analyzed chunk errors using trigram language model and rule based model.

Trigram Language Model. The simplest models of natural language are n- gram Markov models. The Markov models for any n-gram are called Markov Chains. A Markov Chain is at most one path through the model for any given input [4]. N-grams are traditionally presented as an approximation to a distribution of strings of fixed length.

According to the n-gram language model, a sentence has a fixed set of chunks, $\{C_0, C_1, C_2, \cdots C_n\}$. This is a set of chunks in our training sentences, e.g., {NCB1, PAVC, AC,…, END}. In N-gram language model, each chunk depends probabilistically on the n-1 preceding words. This is expressed as shown in equation (1).

$$p(C_{o,n}) = \prod_{i=0}^{n-1} p(C_i | C_{i-n+1}, \ldots, C_{i-1}) \tag{1}$$

Where (C_0) is the current chunk of the input sentence and it depends on the previous chunks. In trigram language model, each chunk (C_i) depends probabilistically on previous two chunks (C_{i-1} , C_{i-2}) and is shown in equation (2) [6].

$$p(C_{o,n}) = \prod_{i=0}^{n-1} p(C_i | C_{i-1}, C_{i-2}) \tag{2}$$

Trigram language model is most suitable due to the capacity, coverage and computational power [1]. The trigram model is used in a greater level of some advanced and optimizing techniques such as smoothing, caching, skipping, clustering, sentence mixing, structuring and text normalization. This model makes use of the history events in assigning the current event some probability value and therefore, it suits for our approach.

Rule-Based Model. Rule-based model has successfully used to develop natural language processing tools and applications. English grammatical rules are developed to define precisely how and where to assign the various words in a sentence. Rule-based system is more transparent and errors are easier to diagnose and debug.

Rule-based model relies on hand-constructed rules that are to be acquired from language specialists, requires only small amount of training data and development could be very time consuming. It can be used with both well-formed and ill-formed input. It is extensible and maintainable. Rules play major role in various stages of translation: syntactic processing, semantic interpretation, and contextual processing of language [8]. Therefore, the accuracy of translation system can be increased by the product of the rule based correcting ungrammatical sentences.

3.4 Correcting Grammar Errors

The final step of our proposed system is controlled by English grammar rules. These rules can determine syntactic structure and ensure the agreement relations between various chunks in the sentence. Common chunk types for each general chunk are used to correct grammar errors. When the sentence patterns increased, the grammar rules will be improved. Some correction rules for subject verb agreement are NCS2_PRV2, NCS2_PAVC, NCS2_PAV2, NCS2_FUVC, NCS1_PAVC, NCS1_PAV1, NCS1_FUVC, NCS1_PRV1, NCB1_PAV1, NCB1_PAVC, NCS_FUVC and NCB_PAVC.

4 Experimental Results

For each input sentence, the system has classified the kinds of sentence such as simple, compound and complex. It also describes whether the sentence type is interrogative or declarative. The proposed system is tested on about 1800 number of sentences. The grammar errors mainly found in the tested sentences are subject verb agreement, missing chunks and incorrect verb form. The performance of this approach is measured with precision and recall. Precision is the ratio of the number of correctly reduced errors to the number of reduced errors in equation (3). Recall is the ratio of the number of correctly reduced errors to the number of errors in equation (4). The resulting precision and recall of reducing grammar errors on different sentence types are shown in Table 3.

$$\text{PRECISION} = \frac{Number of Correctly\ Reduced Errors}{Number of\ Reduced Errors} \times 100\% \quad (3)$$

$$\text{RECALL} = \frac{Number of Correctly\ Reduced Errors}{Number of Errors} \times 100\% \quad (4)$$

Table 3. Experimental Results of Reducing Grammar Errors

Sentence Type	Actual	Reduce	Correct	Precision	Recall
Simple	650	570	512	89.83 %	78.77 %
Compound	560	530	440	83.02 %	78.57 %
Complex	530	480	402	83.75 %	75.85 %

5 Conclusion and Future Work

This paper has presented a grammar checker for reducing errors of translated English sentences which makes use of a context free grammar based bottom up parsing, trigram language model and rule based model. It is expected that this ongoing research will yield benefits for Myanmar-English machine translation system. We use our own trained sentence patterns (dataset). Sample sentence patterns are presented as shown in Table 4. The proposed system currently detects the syntactic structure of the sentence and limits the detection of semantic errors.

In the future, we plan to check the semantic grammar errors for translated English sentences. We will expand more trained sentence rules to access all sentence types. If we get more sentence patterns, we can reduce more errors. We also plan to apply more English grammar rules for fully correction. Moreover, we plan to improve the accuracies of detection, analyzing and correction grammar errors.

Table 4. Sample Sentence Patterns

NC_VC_NC (Declarative)	QC_VC_NC_VC (Interrogative)
NCB_PRV1_NCB_END=S	QC_PAV2_NCB2_VC_IEND=S
NCB_PRV1_NCB1_END=S	QC_PAV1_NCB1_VC_IEND=S
NCB_PRV1_NCB2_END=S	QC_PAV1_NCB_VC_IEND=S
NCB_PRV1_NCO_END=S	QC_PAV2_NCB_VC_IEND=S
NCB1_PRV1_NCB_END=S	QC_PAV2_NCS2_VC_IEND=S
NCB1_PRV1_NCB1_END=S	QC_PAV1_NCS1_VC_IEND=S
NCB1_PRV1_NCB2_END=S	QC_PAV1_NCS_VC_IEND=S
NCB1_PAV1_NCB_END=S	QC_PAV2_NCS_VC_IEND=S
NCB1_PAV1_NCB1_END=S	QC_PRV2_NCB2_VC_IEND=S
NCS2_PRV2_NCB2_END=S	QC_PRV1_NCB1_VC_IEND=S
NCS2_PRV2_NCO_END=S	QC_PRV2_NCB_VC_IEND=S
:	:

References

1. Brian, R., Eugene, C.: Measuring Efficiency in High-Accuracy, Broad-Coverage Statistical Parsing. In: Proceedings of the COLING 2000 Workshop on Efficiency in Large-Scale Parsing Systems, pp. 29–36 (2000)
2. Daniel, N.: A Rule-Based Style and Grammar Checker (2003)
3. Keith, D.C., Ken, K., Linda, T.: Bottom-up Parsing (2003)
4. Lawrence, S., Fernando, P.: Aggregate and mixed order Markov models for statistical language processing. In: Proceedings of the Second Conference on Empirical Methods in Natural Language Processing, pp. 81–89. ACM Press, New York (1997)
5. Myat, T.Z.T.: An English Syntax Analyzer for English-to-Myanmar Machine Translation. University of Computer Studies, Yangon (2007)
6. Selvam, M., Natarajan, A.M., Thangarajan, R.: Structural Parsing of Natural Language Text in Tamil Using Phrase Structure Hybrid Language Model. International Journal of Computer, Information and Systems Science, and Engineering, 2–4 (2008)
7. Steven, A.: Tagging and Partial Parsing. In: Ken, C., Steve, Y., Gerrit, B. (eds.) Corpus-Based Methods in Language and Speech. Kluwer Academic Publishers, Dordrecht (1996)
8. Paisarn, C., Virach, S., Thatsanee, C.: Improving Translation Quality of Rule-based Machine Translation. In: 19th International Conference on Computational Linguistics (Coling 2002): Workshop on Machine Translation in Asia, Taipei, Taiwan (2002)
9. Ramki, T.: Context Free Grammars (2005), `http://web.cs.du.edu/~ramki/courses/3351/2009Fall/notes/cfl.pdf`
10. Elaine, R., Kevin, K.: Artificial Intelligent, 2nd edn. McGraw Hill, Inc., New York (1991)