Using a Grammar Checker for Detection Translated English Sentence

Nay Yee Lin, Khin Mar Soe, Ni Lar Thein
University of Computer Studies, Yangon
nayyeelynn@gmail.com, nilarthein@gmail.com

Abstract

Machine Translation Systems expect target language output to be grammatically correct within the frame of proper grammatical category. In Myanmar-English statistical machine translation system, the proposed system concerns with the target language model to smooth the translated English sentences. Most of the typical grammar checkers can detect ungrammatical sentences and seek for what error it is. However, they often fail to detect grammar errors for translated English sentence such as missing words. Therefore, we intend to develop an ongoing grammar checker for English as a second language (ESL). There are two main tasks in this approach: detecting the sentence pattern in chunk structure and analyzing the chunk errors. This system identifies the chunk types and generates context free grammar (CFG) rules for recognizing grammatical relations of chunks. It also presents a hybrid approach of trigram-based markov model with rule-based model. Experimental results show that the proposed grammar checker can improve the effectiveness of translation quality.

Keywords: Statistical Machine Translation, English Grammar Checker, Context Free Grammar

1. Introduction

Grammar checking is one of the most widely used tools within natural language processing applications. The syntactic structure of a sentence is necessary to determine its grammar correctness. The syntactic detection is verification of the structure of a sentence and of relations between words (e.g. agreement).

Grammar checkers are most often implemented as a feature of a larger program, such as a word processor. Although all major Open Source word processors offer spell checking and grammar checker feature. Such a feature is not available as a separate free program either for machine translation. Therefore, our approach is a free program which can be used both as a stand-alone grammar checker.

Three methods are widely used for grammar checking in a language; syntax-based checking, statistics-based checking and rule-based checking. In *syntax based grammar checking*, each sentence is completely parsed to check the grammatical correctness of it. The text is considered incorrect if the syntactic parsing fails. In *statistics-based approach*, POS tag sequences are built from an annotated corpus, and the frequency, and thus the probability, of these sequences are noted. The text is considered incorrect if the POS-tagged text contains POS sequences with frequencies lower than some threshold. The statistics based approach essentially learns the rules from the tagged training corpus. In *rule-based approach*, the approach is very similar to the statistics based one, except that the rules must be handcrafted [5].

There are a variety of techniques for Grammar checking. Among them, this paper presents a chunk based grammar checker by using hybrid approach to decide whether the sentence rule is correct or not and to analyze the errors.

This paper is organized as follows. Section 2 describes the related work. Section 3 presents the overview of Myanmar-English Statistical

Machine Translation System. In section 4, chunk based grammar detection system is described. Section 5 reports the experimental results and finally section 6 concludes the paper.

2. Related Work

Several researchers worked the grammar checking in natural language processing for various languages.

An approach based on n-gram statistical grammar checker for both Bangla and English is proposed in [10]. It considers the n-gram based analysis of words and POS tags to decide whether the sentence is grammatically correct or not.

In [1], a model is applied for reducing errors in translation using Pre-editor for Indian English Sentences. They have used a major corpus in tourism and health domains. They formed structures of English practiced mostly in India have been identified to design the predictor. This was incorporated in the AnglaBharti Engine and gave significant improvement in the Machine Translation output.

An alternative approach is proposed in [13], where they check the Swedish grammar for evaluation tool and post processing tool of Statistical Machine Translation. They have performed experiments for English-Swedish translation using a factored phrase-based statistical machine translation (PBSMT) system based on Moses (Koehn et al., 2007) and the mainly rule-based Swedish grammar checker Granska (Domeij et al., 2000; Knutsson, 2001).

In [6], a user model which can be tailored to different types of users to identify and correct English language errors. It is presented in the context of a written English tutoring system for deaf people. The model consists of a static model of the expected language and a dynamic model that represents how a language might be acquired over time.

In [7], the ongoing developments in the LRE-2 project SECC (A Simplified English Grammar and Style Checker/Corrector) check if the documents comply with the syntactic and lexical rules; if not, error messages are given, and automatic correction is attempted wherever possible to reduce the amount of human correction needed.

An approach based on hybrid approach [2] that presents an implemented hybrid approach for grammar and style checking, combining an industrial pattern based grammar and style checker with bidirectional, large-scale HPSG grammars for German and English.

Another kind of approach [3] developed a way of producing context free grammar for solving Noun and Verb agreement in Kannada Sentences. They showed the implementation of this agreement using Context Free Grammar.

In [8], the authors presented an analysis of the most frequently encountered style and text structure errors produced by a variety of types of authors when producing texts. They showed an argumentation system can be used so that the user can get arguments for or against a certain correction.

3. Myanmar-English Statistical Machine Translation System

In Myanmar-English statistical machine translation system, source language model, alignment model, translation model and target language model are required to complete translation. Among these models, our proposed system develops a target language model to check the grammar of translated English sentences.

Input for Myanmar-English machine translation system is Myanmar sentence. After this input sentence has been passed three models, translated English sentence is obtained in target language model. However, this sentence might be incomplete in grammar because the syntactic structure of Myanmar and English language are totally different. For example, after translating the Myanmar sentence “မြန်မာပြည်တွင် ရာသီဥတု သုံးမျိုး ရှိပါသည်။”, the translated English sentence might be “*are three seasons in Myanmar*.”. This sentence has missing words “*There*” for correct English sentence “*There are three seasons in Myanmar.*”. As an another input “ကျွန်တော် မနေ့က သံတစ်ထုပ် ဝယ်ခဲ့သည်။”, the translated output is “*I bought a packet nails yesterday.*”. In this

sentence, "of" (preposition) is omitted for "*a packet of nails*". These examples are just simple sentence errors. When the sentence types are more complex, grammar errors detection is more needed.

There are various English grammar rules to correct the ungrammatical sentences. The proposed grammar checker detects and provides subject verb agreement, incorrect verb form, missing markers (.,?) and missing words such as preposition (IN), conjunction (CC), determiner (DT), existential (EX) based on the translated English sentences.

4. Proposed Chunk Based Grammar Detection

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

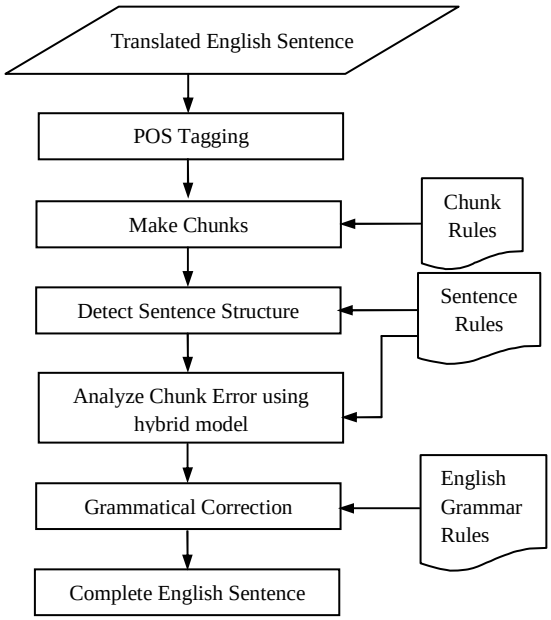

Figure 1. Overview of the proposed system

In our proposed system, the translated English sentence is used as an input. Firstly, this sentence is tokenized and tagged POS to each word. Secondly, these tagged words are grouped into chunks by parsing the sentence into a form which is a chunk based sentence structure. After making chunks, these chunks relationship for input sentence are detected by using sentence rules. If the sentence rule is incorrect, we analyze the chunk errors using trigram language model and rule based model.

4.1. 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. POS-tagging is one of the main processes of making up the chunks in a sentence as corresponding to a particular part of speech. There are many approaches to automated part of speech tagging.

This system tags each word by using Tree Tagger which is a Java based open source tagger. It can work on any sentences. However, it often fails to tag correctly some words when one word has more than one tag. For example, POS tags for the word "sweet" are noun [NN] and also adjective [JJ]. In this case, we need to refine the POS tags for these words based on the rules according to the position of the neighbour words' POS tags. The example of refinement tags is shown in Table 1.

Table 1. Example of refinement tags

Example Sentence	Incorrect POS Tag	Position of Neighbors	Refine POS Tag
He ***bit*** a rope.	bit -NN	Previous tag is PP	VBD
He is a ***tailor***.	tailor-VB	Previous tag is DT	NN

4.2. 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 which builds a chunk based sentence structure associated with a given sentence. The system makes English sentence in chunk structure by using Context Free Grammar (CFG). It represents how these chunks fit together to form the constituents of that sentence.

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 (Abney, 1996).

The syntactic chunk structure of a sentence is necessary to determine its grammar correctness. There are ten general chunk types are used in the proposed system as shown in Table 2.

Table 2. Chunk types

NC	Noun Chunk
VC	Verb Chunk
AC	Adjective Chunk
RC	Adverb Chunk
PTC	Particle Chunk
PPC	Prepositional Chunk
COC	Conjunction Chunk
QC	Question Chunk
INFC	Infinitive Chunk
TC	Time Chunk

4.2.1. Context Free Grammar (CFG)

Context Free Grammars constitute an important class of grammars, with a broad range of applications such as programming languages, natural language processing, bio-informatics and others.

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

- A finite set V of variables or non terminal symbols.
- A finite set Σ of symbols or terminal symbols. We assume that the sets V and Σ 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 $A \rightarrow \alpha$.

CFGs are powerful enough to express sophisticated relations among the words in a sentence. It is also tractable enough to be computed using parsing algorithms [12]. 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.

There are two methods for parsing such as Top-down parsing and Bottom-up parsing. In Top-down parsing, begin with the start symbol and attempt to derive the input sentence by substituting the right hand side of productions for non terminals. In Bottom-up (shift–reduce) parsing, begin with the input sentence and attempt to work back to the start symbol. Bottom-up parsers handle a large class of grammars. In this system, Bottom-up parsing is used to parse sentence structure.

4.2.2. Parsing Chunks by using Context Free Grammar

In our proposed system, sample Context Free Grammar G = (V, Σ, S, P) is shown as follows:

S= S

V={S, NC, VC, PPC, TC, DT, JJ, NN, VBZ…}

∑= {a, young, man, is, reading, in, writes, wrote…}

P=

S =>	NC_VC_NC_PPC_NC_TC_END
NC =>	DT_JJ_NN
NC =>	DT_NN
VC =>	VBZ_VBG
VC =>	VBZ
PPC=>	IN
DT =>	A,The ,This
JJ =>	young, tall, clever
NN =>	man, apple, book
VBZ=>	is, writes, reads

The proposed grammar checker identifies the chunks using CFG based bottom-up parsing for assembling POS tags into higher level chunks, until a complete sentence (S) has been found.

Example:

"*A young girl is reading a book in the library*."

POS Tagging:

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

Making Chunks:

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

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

NC _ VC _ [DT][NN] [IN] [DT] [NN] [SENT]

NC _ VC _ NC _ [IN] [DT] [NN] [SENT]

NC _ VC _ NC _ PPC_ [DT] [NN] SENT]

NC _ VC _ NC _ PPC _ NC _ [SENT]

NC _ VC _ NC _ PPC _ NC _ END

Chunk Based Sentence Pattern:

S = NC_VC_NC_PPC_NC_END

There are currently 1000 trained sentence patterns for our system. Sample trained sentence patterns are as follows:

NC_RC_NC_VC_END=S

NC_RC_VC_INFC_COC_VC_NC_END=S

NC_VC_NC_AC_INFC_AC_NC_END=S

NC_COC_NC_COC_NC_VC_AC_END=S

NC_COC_NC_RC_VC_PTC_PPC_NC_END=S

NC_COC_NC_VC_INFC_INFC_NC_END=S

NC_RC_VC_INFC_AC_INFC_NC_PPC_END=S

QC_AC_VC_NC_END=S

QC_NC_VC_AC_END=S

4.3. Detecting and Analyzing Errors by using Hybrid Model

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

4.3.1. Trigram Language Model

The simplest models of natural language are n- gram Markov models. N-gram models are the examples of statistical model. 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, $\{c_0, c_1, c_2, \ldots, c_n\}$. This is a set of chunks in our training sentences, e.g., {NC, VC, AC,…, END}. In N-gram language model, each chunk depends probabilistically on the n-1 preceding chunks. This is expressed as shown in (1).

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

where (c_i) is the current chunk of the input sentence and it depends on the previous chunks.

This is called trigram language model, where each chunk (c_i)depends probabilistically on previous two chunks (c_{i-1}, c_{i-2}) and is shown in equation (2) [9].

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

This system uses trigram language model of chunks (*n=3*) to get the good results for detecting chunk errors. Trigram language model is most suitable due to the capacity, coverage and computational power [4]. This model makes use of the history events in assigning the current event some probability value and therefore, it suits for our approach.

4.3.2. Rule-Based Model

Rule-based model has successfully used to develop natural language processing tools and applications. Rule-based system is more transparent and errors are easier to diagnose and debug. It 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 [11].

To determine needed helping verb or preposition, our system checks POS tags and root words within the chunk. There are about 200 grammar rules for analyzing chunk errors. When the sentence patterns increased, the grammar rules will be improved. These rules can determine the chunk structures, syntactic structure and ensure the agreement relations between various chunks in the sentence. The accuracy of translation system can be increased by the product of the rule based model.

Some rules for subject verb agreement are described as follows:

If previous tag is "NNS" And current tag is "VBZ" Then display incorrect subject verb agreement. If previous tag is "NN" And current tag is "VBP" Then display incorrect subject verb agreement.

4.3.3. Example

For example, an incorrect translated sentence *"A man a woman came to our house"*, the following sentence pattern and probability values are resulted.

POS Tagging	A[DT] man[NN] a[DT] woman[NN] came[VBD] to[TO] our[PP$] house[NN] .[SENT]
Making Chunks	NC[DT_ NN] => [A man] NC[DT_NN] => [a woman] VC[VBD] => [came] INFC[TO] => [to] NC[PP$_NN]=> [our house] END[SENT] => [.]
Chunk based Sentence	NC_NC_VC_INFC_NC_END
Probabilities of each chunk based on trained sentences	P(NC/none, none)= 0.586 P(NC/none, NC)= 0.0 P(VC/NC, NC)= 0.0 P(INFC/NC, VC)= 0.483 P(NC/VC, INFC)= 0.364 P(END/INFC, NC)= 0.675

P(S)=P(NC/none,none).P(NC/none,NC).
P(VC/NC,NC).P(INFC/NC,VC).
P(NC/VC,INFC).(END/INFC,NC)
=0.586 * 0.0 * 0.0 * 0.483 * 0.364 * 0.675 =0.0

The product of the whole sentence P(S) is 0.0 according to the trigram model in (2). In this case, the proposed system searches the possible chunks for the second place by using these probability values:

P(VC/none, NC)=0.54
P(RC/none, NC)=0.01
P(COC/none, NC)= 0.01

Verb chunk (VC) which has maximum probability is firstly substituted in the second place. Then, the sentence rule NC_VC_NC_VC_INFC_NC_END is obtained. However, this rule is incorrect by comparing the sentence rules. Therefore, RC and COC are also substituted for missing chunk. When COC is substituted in the second place, the correct sentence rule NC_COC_NC_VC_INFC_NC_END is resulted.

For given example, the missing chunk (COC) represents POS tag (CC) which corresponds to English words ('and', 'or', ',') according to the chunk rules. Therefore, the correct sentence might include 'and', 'or' and ',' between two noun chunks ([NC] [A man], [NC] [a woman]) by grammar rules. As a result, by using trigram with rule-based model for our grammar checker can be successful to detect and analyze the errors.

5. Experimental Results

The proposed system is trained and tested on simple, compound and complex sentence types. The grammar errors mainly found in the tested sentences are subject verb agreement, missing chunks and incorrect verb form. This system currently considers only the syntactic structure of the sentence. It is also limited detection of the semantic errors and fully correction of syntactic errors. The system suggests the possible words for chunk errors. The performance of this approach is measured with precision and recall. Precision is the ratio of the number of correctly detected errors to the number of detected errors in (3). Recall is the ratio of number of correctly detected errors to the number of errors in (4).

The resulting precision and recall of detecting grammar errors on different sentence types are shown in Table 3.

$$\text{PRECISION} = \frac{NumberofCorrectlyDetectedErrors}{NumberofDetectedErrors} \times 100\% \quad (3)$$

$$\text{RECALL} = \frac{NumberofCorrectlyDetectedErrors}{NumberofErrors} \times 100\% \quad (4)$$

Table 3.Results of Detection Grammar Errors

Sentence Type	Tested Sentences	Detect	Correct	Precision	Recall
Simple	760	726	717	98.76%	94.34%
Compound	550	530	516	97.36%	93.82%
Complex	560	552	530	96.74%	94.64%

6. Conclusion

One challenge in Myanmar-English statistical machine translation system is that the output can often be ungrammatical. To address this issue, the proposed system has investigated the use of a grammar checker. This paper presents a Context Free Grammar based bottom up parsing, a hybrid approach (trigram-based markov model and rule-based model) to detect grammar errors of translated English sentences. Therefore, the proposed grammar checker can improve the effectiveness of translation quality.

References

[1] A.Sharma, N.Jaiswal, "Reducing Errors in Translation using Pre-editor for Indian English Sentences", *Proceedings of ASCNT–, CDAC*, Noida, India, pp. 70–76, 2010.

[2] B.Crysmann, N.Bertomeu, "Hybrid processing for grammar and style checking", *Proceedings of the 22nd International Conference on Computational Linguistics*, pp 153-160, Manchester, August 2008.

[3] B.M. Sagar, G. Shobha and P.R.Kumar, "Solving the Noun Phrase and Verb Phrase Agreement in Kannada Sentences", *International Journal of Computer Theory and Engineering*, Vol. 1, No. 3, 1793-8201, August, 2009.

[4] B.R.E.Charniak, "Measuring Efficiency in High-Accuracy, Broad-Coverage Statistical Parsing" *Proceedings of the COLING 2000, Workshop on Efficiency in Large-Scale Parsing Systems*, Pages 29-36, 2001.

[5] D.Naber, "*A Rule-Based Style and Grammar Checker*", 2003.

[6] K.F.McCoy, C.A.Pennington, L.Z.Suri**, "***English Error Correction: A Syntactic User Model Based on Principled Mal-Rule Scoring*". Computer and Information Sciences Department and Applied Science and Engineering Laboratories, University of Delaware.

[7] G.Adriaens, "Simplified English Grammar and Style Correction in an MT Framework", *Translation and the Computer 15*, Papers at a conference,8-19, (London:Aslib), November, 1993.

[8] L.Buscail, and P.S.Dizier, "Textual and Stylistic Error Detection and Correction: Categorization, Annotation and Correction Strategies", *IEEE English International Symposium on Natural Language Processing*, 2009.

[9] M Selvam, A M Natarajan, and R Thangarajan, "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.

[10] M.J.Alam, N.U.Zaman and M.Khan, "*N-gram based Statistical Grammar Checker for Bangla and English*", Center for Research on Bangla Language Processing, BRAC University, Dhaka, Bangladesh.

[11] P.Charoenpornsawat, V.Sornlertlamvanich, T.Charoenporn, "*Improving Translation Quality of Rule-Based Machine Translation*", Information Research and Development Division, National Electronics and Computer Technology Center, Thailand.

[12] R.Thurimella, "Context Free Grammars", 2005.

[13] S.Stymne, L.Ahrenberg, "*Using a Grammar Checker for Evaluation and Postprocessing of Statistical Machine Translation*", Department of Computer and Information Science Linkoping University, Sweden.