Chunk-based Grammar Checker for Detection Translated English Sentences

Nay Yee Lin
University of Computer Studies,
Yangon, Myanmar

Khin Mar Soe
Natural Language Processing Laboratory
University of Computer Studies,
Yangon, Myanmar

Ni Lar Thein
University of Computer Studies,
Yangon, Myanmar

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 target language output (English) can often be ungrammatical. To solve this need, we propose an ongoing chunk-based grammar checker for 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 sentences such as missing words. Therefore, we intend to develop a grammar checker by using trigram language model and rule based model. The system identifies the chunk types and generates context free grammar (CFG) rules for recognizing grammatical relations of chunks. In this paper, we present an overview of the current research being carried out three main functions: detecting sentence patterns in chunk types, analyzing chunk errors and correcting the errors. We hope that it encourages improving the translation quality of Myanmar to English.

General Terms

Statistical Machine Translation, Context Free Grammar, Trigram Language model, Rule based model

Keywords

Chunk-based Grammar Checker

1. INTRODUCTION

Grammar is the set of structural rules that govern the composition of clauses, phrases, and words in any given natural language. Grammar checking is one of the most widely used tools within natural language processing (NLP) applications. Grammar checkers check the grammatical structure of sentences based on morphological processing and syntactic processing. These two steps are part of natural language processing intending to understand natural languages. Morphological processing is the step where individual words are analyzed into their components and non-word tokens, such as punctuation, are separated from the words. Syntactic processing is the analysis where linear sequences of words are transformed into structures that show grammatical relationships between the words in the sentence [6].

Grammar checkers are most often implemented as a feature of a larger program, such as a word processor. However, such a feature is not available as a separate free program for machine translation. Therefore, we propose a grammar checker as a complement for machine translation.

Three main approaches 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].

Among these approaches, this paper presents a grammar checker by using statistical and rule based model. 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 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 trained sentence patterns. If the sentence pattern is incorrect, we analyze the chunk errors and then correct the errors.

The rest of the paper is organized as follows. Section 2 presents the related work of this paper. Section 3 describes the overview of Myanmar-English Statistical Machine Translation System. In section 4, the proposed chunk based grammar checker is explained. Section 5 reports the experimental results of our proposed system and finally section 6 concludes the paper.

2. RELATED WORK

Many researchers have been worked grammar checking in NLP for various languages. In the following paragraphs, we discuss briefly some of the related work.

N-gram statistical grammar checker for both Bangla and English is proposed in [8]. 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 checked 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 and the mainly rule-based Swedish grammar checker Granska [15].

In [10], 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.

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 [7].

An 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. In most of the Indian languages a verb ends with a token which indicates the gender of the person (Noun/ Pronoun). They showed the implementation of this agreement using Context Free Grammar. Around 200 sample sentences have taken to test the agreement.

In [11], 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 MACHINE TRANSLATION SYSTEM

Input for Myanmar-English statistical machine translation system (SMT) is Myanmar sentence and the target output is English sentence. In this system, source language model, alignment model, translation model and target language model are required to complete translation as shown in Figure 1. Our proposed system is concerned with the target language model to check the grammar errors for translated English sentences.

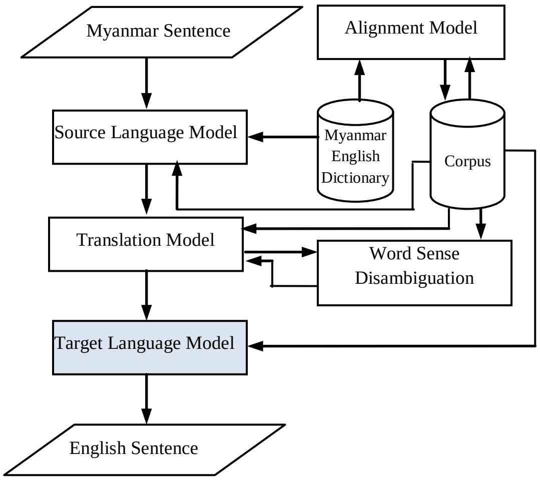

Figure 1. Myanmar-English Statistical Machine Translation System

The input sentence has been processed in three models (source language model, alignment model and translation model), the 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 "ပန်းခြံထဲမှာ သစ်ပင်များ ရှိကြသည်။", the translated English sentence might be "*are trees in park.*". This sentence has missing words "*There*" and "*the*" for correct English sentence "*There are trees in the park.*". As an another input "သူသည် လက်ဖက်ရည်တစ်ခွက် သောက်နေသည်။", the translated output is "*He is drinking a cup tea.*". In this sentence, "of" (preposition) is omitted from "*a cup of tea*". These examples are just simple sentence errors. When the sentence types are more complex, grammar errors detection and correction are more needed.

There are many English grammar errors to correct ungrammatical sentences. This grammar checker currently detects and provides the following errors:

- 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. Verbs vary in form 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.

4. CHUNK-BASED ENGLISH GRAMMAR CHECKER

In SMT system, there are very few spelling errors in the translation output, because all words are come from the corpus. Therefore, this system proposes a target-dominant grammar checker for Myanmar-English machine translation system as shown in Figure 2.

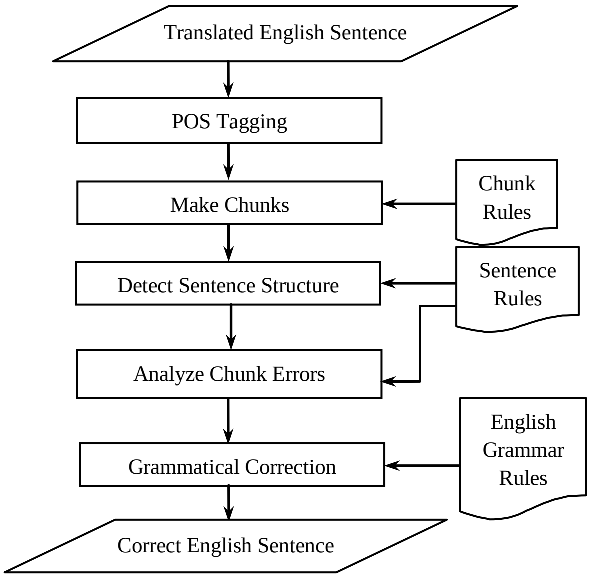

Figure 2. Overview of proposed system

4.1 Part-of-Speech tagging

Part of Speech (POS) tagging is the main process of making up the chunks in a sentence as corresponding to a particular part of speech. 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.

There are many approaches to automated part of speech tagging. In this system, each word is tagged by using Tree Tagger which is a Java based open source tagger. However, Tree Tagger often fails to tag correctly some words when one word has more than one POS tag. For example, POS tags of the word "sweet" are "JJ" and "NN". In this case, refinement of the POS tags for these words is made by using the rules based on the position of the neighbor words' POS tags. The example for refinement tags is shown in Table 1.

Table 1. Refinement tags

Sentence	Incorrect Tag	Neighbor word's POS tag	Refine Tag
He eats a ***sweet.***	***sweet***=JJ	Previous Tag =DT	***sweet***=NN
He is a ***tailor***.	***tailor***=VB	Previous tag is DT	***tailor***=NN
He ***bit*** a rope.	***bit***=NN	Previous tag is PP	***bit***=VBD

4.2 Making Chunks

Making chunks is a process to parse the sentence into a form that is a chunk based sentence structure. A chunk is a textual unit of adjacent POS tags which display the relations between their internal words. 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.

4.2.1 Context Free Grammar (CFG)

CFGs 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 $A \rightarrow \alpha$.

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 [14].

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. Top-down parsing begins with the start symbol and attempt to derive the input sentence by substituting the right hand side of productions for non terminals. Bottom-up (shift–reduce) parsing begins with the input sentence and combines words into higher-level chunks until the unit finally becomes a sentence. Bottom-up parsers handle a large class of grammars [9]. In this system, Bottom-up parsing is used to parse the sentences.

4.2.2 Parsing chunks by using CFG

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

Table 2. Proposed Chunk Types

Chunk Types	Description	Example
NC	Noun Chunk	a young boy, the girls
VC	Verb Chunk	is playing, goes, went
TC	Time Chunk	tomorrow, yesterday
COC	Conjunction Chunk	and, or, but
INFC	Infinitive Chunk	to
AC	Adjective Chunk	more beautiful, younger, old
RC	Adverb Chunk	usually, quickly
PTC	Particle Chunk	up, down
PPC	Prepositional Chunk	at, on, in, under
QC	Question Chunk	Where, Who, When

In our proposed system, sample Context Free Grammar G = (V, T, S, P) is described as follows:
S= S
V={S, NC, VC, PPC, TC, DT, JJ, NN, VBZ…}
T= {a, young, man, is, reading, in, writes,…}
P =

S	=>	NC_VC_NC_PPC_NC_TC_END
S	=>	NC_VC_NC_END
S	=>	QC_AC_VC_NC_END
NC	=>	DT_JJ_NN
NC	=>	DT_NNS
NC	=>	PP
VC	=>	VBZ_VBG
VC	=>	VBP
VC	=>	VBD
PPC	=>	IN
DT	=>	A,The,This,…
JJ	=>	young, tall, clever,…
NN	=>	man, apple, book,…
PP	=>	He, She, They,…
VBZ	=>	is, writes, reads,…
VBG	=>	reading, playing,…
VBZ_VBG	=>	is playing, is writing,…

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 has been found. For example, a simple sentence "*The students are playing football in the playground.*" is chunked as follows:

POS Tagging:

The	[DT]
students	[NNS]
are	[VBP]
playing	[VBG]
football	[NN]
in	[IN]
the	[DT]
playground	[NN]
.	[SENT]

Making Chunks:

NC [DT_ NNS] => [The students]
VC [VBP_VBG] => [are playing]
NC [NN] => [football]
PPC [IN] => [in]
NC [DT_NN] => [the playground]
END [SENT] => [.]

Chunk Based Sentence Pattern:

S = NC_VC_NC_PPC_NC_END

4.3 Detecting 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.

4.3.1 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 [12]. N grams are traditionally presented as an approximation to a distribution of strings of fixed length. N-grams of words or chunks are used in the type of patterns used to continuous sequential patterns allowing arbitrary gaps between words.

According to the n-gram language model, a sentence has a fixed set of chunks,{ C_0 , C_1 , C_2 ,… 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 words. This is expressed as shown in (1).

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

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 (2) [16].

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

Given a sentence, a trigram is a sequence of three chunks (C_i , C_{i+1} , C_{i+2}) where a generic chunk C_i is either the i-th chunk of the sentence.

Trigram language model is most suitable due to the capacity, coverage and computational power [4]. The trigram language 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.

4.3.2 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.

It relies on hand-constructed rules that are collected 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 [13]. Therefore, the accuracy of translation system can be increased by the product of the rule based correcting ungrammatical sentences.

4.4 Grammar Error Correction

The final step of our proposed system is controlled by grammar rules to determine proper corrections. These rules can determine syntactic structure and ensure the agreement relations between various chunks in the sentence. POS tags for each chunk are used to correct grammar errors. There are about 100 sentence patterns and 1300 English grammar rules for correction at present. When the sentence patterns increased, the grammar rules will be improved. Some rules for correcting subject-verb agreement are presented in Table 3.

Table 3. Sample Rules

Rules (NC_VC)	Example
NNS +VBP	We go
NNS +VBD	We went
NNS +VBP_VBG	We are going
NNS +VBD_VBG	They were going
NNS +VBP_VBD	They have worked
NNS +MD_VB	They will come
NN +VBZ	She goes
NN +VBD	She went
NN +VBZ_VBG	She is going
NN +VBD_VBG	She was going
NN +VBZ_VBD	He has walked
NN +MD_VB	He will come

4.5 Evaluation of Proposed System

For an incorrect translated sentence "A boy a girl went to their school.", the following sentence pattern and probability values are obtained.

POS Tagging :

A[DT] boy[NN] a[DT] girl[NN] went[VBD] to[TO] their[PP$] school[NN] .[SENT]

Making Chunks :

NC [DT_ NN] => [A boy]
NC [DT_NN] => [a girl]
VC [VBD] => [went]
INFC [TO] => [to]
NC [PP$_NN] => [their school]
END [SENT] => [.]

Chunk based Sentence Pattern:

NC_NC_VC_INFC_NC_END

Probabilities of each chunk from 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) =0.586 * 0.0 * 0.0 * 0.483 * 0.364 *0.675 =0.0

The product of the whole sentence is 0.0 by equation (2). In this case, we search the sequence of chunks P(NC/none, NC) which has zero probability. We get the probability values for possible chunks depend on previous chunks (none, NC) as follows:

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

According to these probabilities, RC, VC and COC can be in the second place. Firstly, VC (verb chunk) is substituted as the maximum probability. Then the sentence pattern NC_VC_NC_VC_INFC_NC_END is obtained. However, this rule is incorrect by comparing the trained sentence patterns. Therefore, RC and COC are also substituted. When COC is substituted, the correct sentence rule NC_COC_NC_VC_INFC_NC_END is resulted for our system. As a consequence of this example, the proposed system can search the correct chunk type (COC).

Thereafter, the proposed system fills up a word in the missing place depending on rules to correct the grammar error. The missing chunk (COC) represents POS tag CC which corresponds to English words (‘and’, ‘or’, ‘,’) according to the chunk rules. The correct sentence pattern might include ‘and’ between two noun chunks ([NC_COC_NC] [A boy and a girl]) according to the grammar rules.

5. EXPERIMENTAL RESULTS

The proposed system is tested on about 1800 number of sentences. 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. It currently detects the syntactic structure of the sentence and limits the detection of semantic errors.

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, recall and F-score according to equation 3, 4 and 5. The resulting precision, recall and F-score of chunk-based grammar checker on different sentence types are shown in Table 4.

$$\text{PRECISION}=\frac{NumberofCorrectly\ ReducedErrors}{Numberof\ \ ReducedErrors}\times 100\% \quad (3)$$

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

$$\text{F score}=2\times\frac{Precision\times Recall}{Precision+Recall} \quad (5)$$

Table 4. Experimental Results

Sentence Type	Actual	Check	Correct	Precision	Recall	F-score
Simple	758	659	588	89.23%	77.57%	82.99%
Compound	630	570	487	85.43%	77.3%	81.17%
Complex	670	620	520	83.87%	77.62%	80.62%

6. CONCLUSION

A chunk-based grammar checker for translated English sentences which uses of trigram language model and rule based model. CFG rules are also used for identifying the sentence patterns and to divide a text into segments which correspond to certain syntactic units. It is expected that this ongoing research will be benefit for Myanmar-English machine translation system. Moreover, we plan to improve the accuracies of detection, analyzing and correction grammar errors. Future work is needed to expand the sentence rules to fully assess all sentence types and detect the semantic errors.

7. REFERENCES

[1] Anuradha, S., Nishtha, J., 2010. “Reducing Errors in Translation using Pre-editor for Indian English Sentences”, Proceedings of Annual Seminar of CDAC-Noida Technologies, Noida:70-76, India.

[2] Berthold, C., Nuria, B., Peter, A., Dan, F., Tina, K., August, 2008. “Hybrid processing for grammar and style checking”, 22nd International Conference on Computational Linguistics:153-160, Manchester.

[3] Sagar, B.M., Shobha, G., and Ramakanth, K.P, August, 2009. “Solving the Noun Phrase and Verb Phrase Agreement in Kannada Sentences”, International Journal of Computer Theory and Engineering (IJCTE) ISSN:1793-8201, Vol. 1, No.3.

[4] Brian, R., Eugene, C., 2000. “Measuring Efficiency in High-Accuracy, Broad-Coverage Statistical Parsing”, In Proceedings of the International Conference on Computational Linguistics,Workshop on Efficiency in Large-Scale Parsing Systems: 29-36.

[5] Daniel, N., 2003. A Rule-Based Style and Grammar Checker.

[6] Elaine, R., Kevin, K., 1991. Artificial Intelligent. Second edition. New York: McGraw Hill, Inc.

[7] Geert, A., November, 1993. "Simplified English Grammar and Style Correction in an MT Framework", Translation and the Computer 15 conference, 8-19.

[8] Jahangir, A., Naushad, U., and Mumit, K., 2006. "N-gram based Statistical Grammar Checker for Bangla and English", Proceedings of 9th International Conference on Computer and Information Technology (ICCIT 2006), Dhaka, Bangladesh.

[9] Keith, D. C., Ken, K., and Linda, T., 2003. Bottom-up Parsing.

[10] Kathleen, F. M., Christopher, P., Linda, Z. S., January, 1996. "English Error Correction: A Syntactic User Model Based on Principled Mal-Rule Scoring", In Proceedings of the Fifth International Conference on User Modeling, Kailua-Kona, Hawaii.

[11] Laurie, B., and Patrick, S., 2009. "Textual and Stylistic Error Detection and Correction: Categorization, Annotation and Correction Strategies", IEEE English International Symposium on Natural Language Processing.

[12] Lawrence, S. and Fernando, P., 1997. "Aggregate and mixed order Markov models for statistical language processing", Proceedings of the Second Conference on Empirical Methods in Natural Language Processing, 81-89. ACM Press, New York.

[13] Paisarn, C., Virach, S., and Thatsanee, C., 2002. "Improving Translation Quality of Rule-based Machine Translation", In 19th International Conference on Computational Linguistics : Workshop on Machine Translation in Asia. Taipei, Taiwan.

[14] Ramki, T., 2005. Context Free Grammars.

[15] Sara, S., Lars, A., 2010. "Using a Grammar Checker for Evaluation and Postprocessing of Statistical Machine Translation", In Proceedings of the International Conference on Language Resources and Evaluation (LREC) 2010. Valetta, Malta.

[16] Selvam, M., Natarajan, A. M., and Thangarajan, R., 2008. "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

[17] Steven, A., 1996. Tagging and Partial Parsing**,** In: Ken Church, Steve Young, and Gerrit Bloothooft (eds.), Corpus-Based Methods in Language and Speech. Kluwer Academic Publishers, Dordrecht.