Developing a Chunk-based Grammar Checker for Translated English Sentences

Nay Yee Lin[a], Khin Mar Soe[b], and Ni Lar Thein[c]

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

Abstract. Machine Translation systems expect target language output to be grammatically correct. In Myanmar-English statistical machine translation system, target language output (English) can often be ungrammatical. To address this issue, we propose an ongoing chunk-based grammar checker 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 levels and generate context free grammar (CFG) rules for recognizing grammatical relations of chunks. There are three main processes to build a grammar checker: checking the sentence patterns in chunk level, analyzing the chunk errors and correcting the errors. According to experimental results, this checker can detect simple, compound and complex sentence types for declarative and interrogative sentences. This system is useful for reducing grammar errors of target language in Myanmar-English machine translation system.

Keywords: Statistical Machine Translation System, Trigram Language Model, Rule based Model.

1 Introduction

A language checker, typically, has two basic components, a spell-checker and a grammar checker. Whereas the spell-checker, usually, limits its operation to the inspection and correction of individual text words, the grammar checker has to cope with errors that can only be detected in contexts that are larger than the word (Anna).

Grammar is the set of structural rules that govern the composition of clauses, phrases, chunks 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 to understand the 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 (Rich and Knight 1991).

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 (Naber, 2003).

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 of Myanmar-English machine translation by using trigram language model 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 using English grammar rules.

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

This section presents the related works of the grammar checking in natural language processing for many languages.

Alam et al. (2006) proposed an approach which based on n-gram statistical grammar checker for both Bangla and English. It considered the n-gram based analysis of words and POS tags to decide whether the sentence is grammatically correct or not. Sharma and Jaiswal (2010) developed a model for reducing errors in translation using Pre-editor for Indian English Sentences. They have used a major corpus in tourism and health domains. This was incorporated in the AnglaBharti Engine and gave significant improvement in the Machine Translation output.

A user model 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. Together these models affect scores on a set of grammar rules which are used to produce a “best interpretation” of the user’s input (McCoy et al., 1996).

Stymne and Ahrenberg (2010) 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 (Koehn et al., 2007) and the mainly rule-based Swedish grammar checker Granska (Domeij et al., 2000; Knutsson, 2001)..

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 (Adriaens,1993).

An approach based on hybrid approach 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 2 (Crysmann et al., 2008).

Buscail and Dizier (2009) 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

Input for Myanmar-English statistical machine translation system (SMT) is Myanmar sentence and the target output is English sentence. Myanmar-English statistical machine translation system has developed source language model, alignment model, translation model and target language model to complete translation.

- **The source language model** includes making Part-of-Speech (POS) tags and function tags for each Myanmar word and searching grammatical relations of Myanmar sentence.
- **The translation model** includes phrase extraction, translation from Myanmar sentences to English sentences by using Myanmar-English bilingual corpus. This model also interacts with Word Sense Disambiguation (WSD) system to solve ambiguities when a phrase of a Myanmar sentence has more than one sense.
- **The alignment model** is working parallel with the other models. Its main work is to build the word and phrase aligned Myanmar-English bilingual corpus.
- **The target language model** includes two parts such as reordering the translated English sentences and smoothing it by using English grammar checker to reduce grammar errors.

Our proposed system is concerned with the target language model to check the grammar errors for translated English sentences. After 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. 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။", "pan chan htae hmar thet pin myar shi kya thi", 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။", "thu thi laphet yae ta khwit thauk nay thi", 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 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 statistical machine translation system as shown in Figure 1.

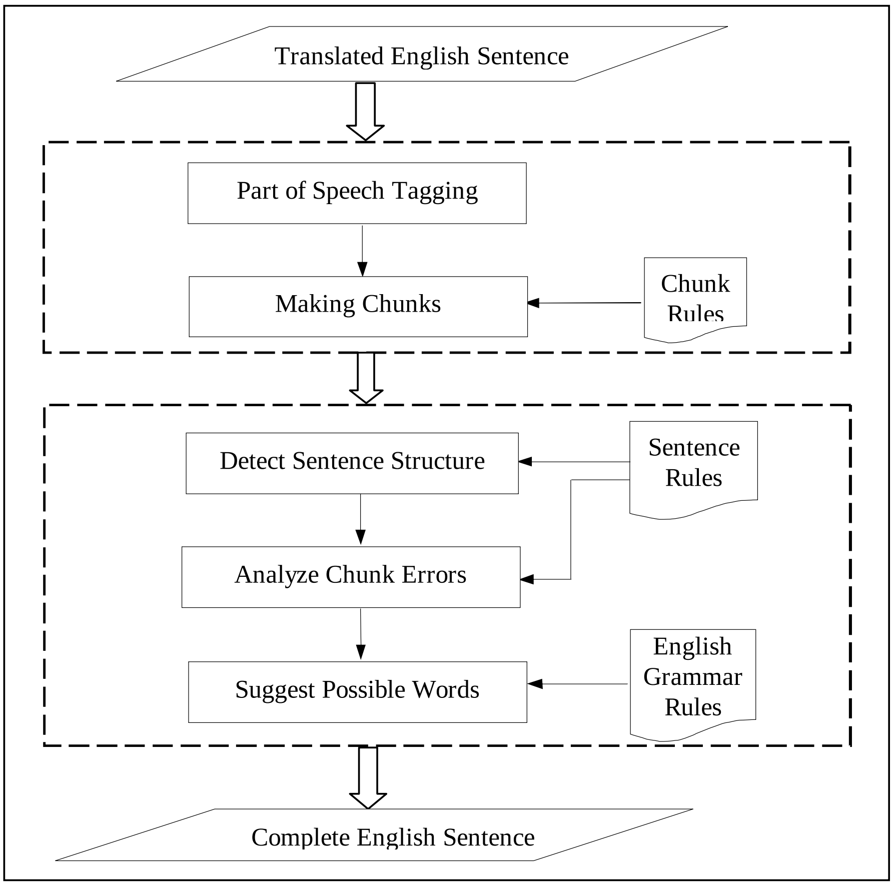

Figure 1: Overview of Proposed System.

4.1 Part-of-Speech (POS) Tagging

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: Example of refinement tags

Sentence	POS Tagging	Refine Tag by rules
He eats a ***sweet***.	He[PP] eats[VBZ] a[DT] ***sweet[JJ]*** . [SENT]	If previous tag is “DT”, current tag is “JJ” And current word is “sweet”, then change sweet[JJ] to ***sweet[NN].***
He is a ***tailor***.	He[PP] is[VBZ] a[DT] ***tailor[VB]*** .[SENT]	If previous tag is “DT”, current tag is “VB” And current word is “tailor”, then change tailor[VB] to ***tailor[NN].***

He ***bit*** a rope.	He[PP] ***bit[RB]*** a[DT] rope[NN] .[SENT]	If previous tag is "PP", current tag is "JJ" And current word is "bit", then change bit[RB] to ***bit[VBD].***

4.2 Making Chunk-based Sentence Patterns

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.

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 α$\in$(V$\cup$T)* is a sequence of terminals and variables, is written as A→α. 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 (Thurimella, 2005).

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 (Cooper et al., 2003). In this system, Bottom-up parsing is used to parse the sentences.

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 (Abney, 1996). 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: Chunk Types

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

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:

NC_VC_NC_PPC_NC_END (Chunk-based Sentence Pattern)

NC_VC_NC_PPC_NC_[SENT]

NC_VC_NC_PPC_[DT_NN][SENT]

NC_VC_NC_[IN][DT][NN][SENT]

NC_VC_[NN][IN][DT][NN][SENT]

NC_ [VBP_VBG][NN][IN][DT][NN][SENT]

[DT_NNS][VBP][VBG][NN][IN][DT][NN][SENT]

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

Chunk-based sentence patterns are widely used in this system for detection sentence patterns. The larger the trained sentence patterns, the better the detection errors. The system has currently trained on about 6000 number of sentence patterns for simple, compound and complex sentence types. Some sample sentence rules are shown in Table 3.

Table 3: Chunk-based Sentence Patterns

NC_VC_END=S
NC_VC_NC_END=S
NC_VC_AC_PPC_NC_END=S
NC_RC_VC_PTC_RC_END=S
NCS_PRV2_RC_NCB2_END_END=S
VC_NC_PPC_NC_END=S
VC_NC_VC_TC_END=S
VC_NC_VC_TO_NC_END=S
QC_VC_NC_VC_IEND=S
QC_VC_NC _IEND=S
QC_VC_NC_AC_PPC _IEND=S
QC_VC_NC_VC_PPC _IEND=S
QC_VC_NC_PPC_TC _IEND=S

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.

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 (Saul and Pereira, 1997). 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.

N-grams of words or POSs are widely used but are not the only type of patterns used in previous work. Sun et al. (2007) extended n-grams to non continuous sequential patterns allowing arbitrary gaps between words. Sjöbergh (2006) used sequences of chunk types, for example, "NP_VC_PP." The parse trees returned by a statistical parser are used by Lee and Seneff (2008) to detect verb form errors.

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 words. This is expressed as shown in equation 1.

$$p(c_{o,n}) = \prod_{i=0}^{n-1} p(c_i | c_{i-n+1},..., c_{i-1}) \qquad (1)$$

where (c_i) 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.

$$p(c_{o,n}) = \prod_{i=0}^{n-1} p(c_i | c_{i-1}, c_{i-2}) \qquad (2)$$

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 (3 Roark and Chamiak, 2000). 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.

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 (Charoenpornsawat et al., 2002). 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 type are used to correct grammar errors. There are about 1800 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 4.

Table 4: Some Rules for Subject Verb Agreement

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 Example

For an incorrect translated sentence *"A man a woman went to their house"*, the following sentence pattern and probability values are obtained.

POS Tagging :
A[DT] man[NN] a[DT] woman[NN] went[VBD] to[TO] their[PP$] house[NN] .[SENT]

Making Chunks :
NC [DT_ NN] => [A man]
NC [DT_NN] => [a woman]
VC [VBD] => [went]
INFC [TO] => [to]
NC [PP$_NN] => [their house]
END [SENT] => [.]

Chunk based Sentence :
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. From this example, the proposed system can search the correct chunk type (COC) by using trigram language model and rule based model.

Thereafter, the proposed system fills up a word in the missing place depending on grammar rules to correct the 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 man and a woman]) according to the English 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 and then described whether the sentence type is interrogative or declarative. 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 5.

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

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

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

Table 5: Experimental Results

Sentence Type	Actual	Reduce	Correct	Precision	Recall	F-score
Simple	650	570	512	89.83 %	78.77 %	83.94%
Compound	530	480	402	83.75 %	75.85 %	79.61%
Complex	560	530	440	83.02%	78.57%	80.73%

6 Conclusion

A chunk-based grammar checker for translated English sentences which makes use of trigram language model and rule based model. Context Free Grammar rules are also used for identifying the sentence patterns and to divide a text into chunk types which correspond to certain syntactic units. We use our own training sentence patterns. We expect this grammar checker will get the benefits for Myanmar-English machine translation system. Moreover, we plan to improve the accuracies of detection, analyzing and correction grammar errors. In the future, we will expand the sentence rules to fully assess all sentence types and detect the semantic errors.

References

Abney, S. 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.

Adriaens, G. November, 1993. Simplified English Grammar and Style Correction in an MT Framework, *Translation and the Computer 15 conference*, pp. 8-19, (London:Aslib).

Alam, J., N. UzZaman and M. Khan. 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.

Buscail, L. and P. Saint-Dizie. 2009. Textual and Stylistic Error Detection and Correction: Categorization, Annotation and Correction Strategies, *IEEE English International Symposium on Natural Language Processing.*

Charoenpornsawat, P., V. Sornlertlamvanich and T. Charoenporn. 2002. Improving Translation Quality of Rule-based Machine Translation, *In 19th International Conference on Computational Linguistics (Coling2002),* Workshop on Machine Translation in Asia. Taipei, Taiwan.

Cooper, K.D., K. Kennedy and L. Torczon. 2003. "*Bottom-up Parsing*".

Crysmann, B., N. Bertomeu, P. Adolphs, D. Flickinger and T. Kluwer. August 2008. Hybrid processing for grammar and style checking, *Proceedings of the 22nd International Conference on Computational Linguistics,* pp. 153-160, Manchester.

Domeij, R., O. Knutsson, J. Carlberger, and V. Kann. 2000. Granska-an efficient hybrid system for Swedish grammar checking, *In Proceedings of the 12th Nordic Conference on Computational Linguistics (Nodalida'99)*, pp.49-56, Trondheim, Norway.

Hein, A.S. "A Chart-Based Framework for Grammar Checking Initial Studies", Linguistics Department, Uppsala University, http://guagua.echo.lu/langeng/en/le3/scarrie/scarrie.html.

Koehn, P. and H. Hoang. 2007. Factored translation models, In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural language Processing and Computational Natural Language Learning, pp. 868-876, Prague, Czech Republic.

McCoy, K.F., C. Pennington and L.Z. Suri. 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.

Naber, D. 2003. "A Rule-Based Style and Grammar Checker" Faculty of Engineering, University of Bielefeld.

Rich, E. and K. Knight. 1991. *Artificial Intelligent.* Second edition. New York: McGraw Hill, Inc.

Roark, B. and E. Charniak. 2000. Measuring Efficiency in High-Accuracy, Broad-Coverage Statistical Parsing, *In Proceedings of the International Conference on Computational Linguistics (COLING) 2000* Workshop on Efficiency in Large-Scale Parsing Systems, pp. 29-36.

Saul, L. and F. Pereira. 1997. Aggregate and mixed order Markov models for statistical language processing, *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pp. 81-89. ACM Press, New York.

Sharma, A. and N. Jaiswal. 2010. Reducing Errors in Translation using Pre-editor for Indian English Sentences, *Proceedings of Annual Seminar of CDAC-Noida Technologies (ASCNT)*, Noida, pp.70-76, India.

Sjöbergh, J. 2006. Chunking: An unsupervised method to find errors in text, *In Proceedings of the 15th Nodalida Conference*, pp.180-185. Joensuu, Finland: University of Joensuu electronic publications in linguistics and language technology.

Stymne, S. and L. Ahrenberg. 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.

Sun, G., X. Liu, G. Cong, M. Zhou, Z. Xiong, J. Lee and C. Lin. 2007. Detecting erroneous sentences using automatically mined sequential patterns, *In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics,* pp. 81-88. Prague, Crech Republic: Association for Computational Linguistics.

Thurimella, R. 2005. "Context Free Grammars".