A SHALLOW PARSER-BASED GRAMMAR CHECK WITH COMPLEX SENTENCES AND CLAUSES

***Thandar Htay*[1], *Moe Thida*[2], *Myint Myint Khaing*[3]**

[1]Application Department, University of Computer Studies, Myitkyina, Myanmar
[2]Application Department, University of Computer Studies, Myitkyina, Myanmar
[3]University of Computer Studies, Pin Long, Myanmar

Abstract

Many word processing systems today include grammar checker which can be used to point out various grammatical problem in a text. This paper describes a grammar checker that uses the shallow parser based on analysis of words and POS tags to decide whether the sentence is grammatically correct or not. This paper is aimed to implement Shallow Parser Based Grammar Check with Complex Sentences and Clauses for the English Language. This system can be built starting with just one rule and then extending it rule by rules. It can check the spelling error and tenses of sentence with the complex sentences and clauses according to the pattern rules. Grammar rules are used to describe the correct syntax of a language. This paper provides the grammar checking software developed for detecting the spelling error and the grammatical error on English texts.

Keyword: Shallow Parsing, Grammar Checking, Spelling Checking

1.INTRODUCTION

Natural language processing programs have been applied in several areas. Most of the word processing systems available in the market incorporate spelling, grammar and style-checking systems for other English and other foreign language, one such rule-based grammar checking system for English. Grammar checking is the task of detection and correction of grammatical errors in the text.

In this system, first, the user can enter input sentence or paragraph and then an input sentence is split into words such as tokens. Spell checker check the spelling error using the lexicon as a kind of dictionary. Dictionary has to be usually large to analyze a wide range of business and technical communications. Each word of the sentence is assigned its corresponding Part-Of-Speech (POS) tag. The system can perform sentence analysis by using shallow parser with IOB tags. Finally, it can display correct grammatical sentence by grammar checker using grammatical rules. This system used thousands of rules.

The remaining of the paper is organized as follows: related work is proposed in section 2. Section 3 represents grammar checking system with complex sentences and clauses for English language based on Shallow Parser. In section 4, the paper describes the process of shallow parser. Section 5 represented pattern matching process and generate grammatical sentence described in section 6. Finally, conclusion comes in section 7.

2.RELATED WORK

Grammar checking is a major part of Natural Language Processing (NLP) whose applications ranges from proofreading to language learning. Madhvi Soni, Jitendra Singh Thakur proposed a comprehensive study of English grammar checking techniques highlighting the capabilities and challenges associated with them [5]. Tin Muyar Win, Zin Mar Than described rule-based chunk level grammar checking with parallel approach and it involves tokenization, rules-based tagging, and phrase chunking with chunk level grammar checking and clause segmentation [7]. Daniel Naber described about a rule-based style and grammar checker system [2]. It proposed to detect errors in a sentence. Each word of the text is assigned. Its part-of-speech tag and sentence is split into chunk, such as noun phrase, verb phrase.

3.GRAMMAR CHECKING FOR COMPLEX SENTENCES AND CLAUSES USING SHALLOW PARSER

Grammar checking is the fundamental application and it checks correctness of input sentence, which has a strong effect on other NLP application [1]. This paper consists of five main parts: Tokenization, Spell Checking, POS tagging, Shallow Parser for phrase chunking and grammar checking using rule-based techniques as shown in figure 1.

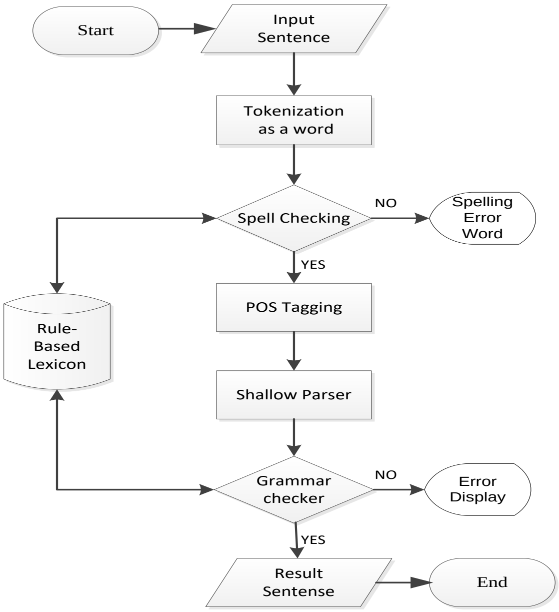

Figure 1 Proposed system for shallow parser-based grammar checking

First step is entering any document, paragraph or sentence and an input sentence is split into words such as tokens. And then Spell checkers search, suggest and correct or incorrect each words within sentences in second step. Spell checking using a lexicon as a dictionary. Third step is tagging the tokenizing words corresponding to their parts of speech. In step four, this step is syntax analyzing by using shallow parser. It captures structure relationships between words and phrases. The tokenized sentence is divided into phrase chunks as noun phrase, verb phrase, etc. Chunker is a division of the text's sentence into series of words as the following:

[NP Amber] [VP was] [NP a great king]

In final step, the grammar checker uses a rule-based lexicon to detect grammatical errors in the text. And then it generates suggestions to correct those errors. Grammar Checking rules, it will identify the correct sentence by matching rules in the rules directory which depend on POS tags. The most important part of a rule is pattern.

3.1. Tokenization as a word

Tokenization is the process of split a sequence into words, punctuations and other symbols. These words and expression sequences are called tokens. The tools performing such tokenization are tokenizer. The following example illustrates the basic function of a tokenizer.

Input complex sentence: Playing hard is all fun and games until someone loses an eye.

Tokenized sentence:

Playing	hard	is	all	fun	and	games
until	someone	loses	an	eye	.	

3.2. Spell Checking

Spelling and grammar checkers are widely-used tools in NLP. It aims to help in detecting and correcting various writing errors. This system used lexicon for spell checking. The lexicon consists of the correct spelling of each word. For words that can have more than one meaning, the lexicon lists all of the various meanings permitted by the system. Input text to spell checker contains extra characters such as hyphen, colon, embedded helping characters like parenthesis, quotes etc. and abbreviation [4]. Typically, spell checkers have three main tasks. First, they search errors by checking and validating whether a word is correct or has been misspelled. Ssecondly, it generates candidate corrections and finally, it enlists the most likely candidate corrections as suggestions to the user.

3.3. Part-of-Speech Tagging

POS Tagging is the process of assign a part-of-speech label or other lexical class marker to each of a sequence of words reflecting their syntactic category. POS tag of a word can be one of major word groups and can be used in stemming for information retrieval on morphological affixes. POS tagging identifies the words in a given sentence corresponding to their parts of speech. Eight parts of speech are noun, verb, pronoun, adjective, adverb, preposition, conjunction and interjection.

Noun	<NN>
Verb	<VB>
Pronoun	<Pro N>
Adjective	<JJ><ADJ>
Adverb	<ADV>
Preposition	<PRP>
Conjunction	<COJ>
Interjection	<IJ>

These tags depend on definition and context (i.e. relationship with adjacent words in the sentences).

E.g. The girl wrote a letter to her cousin.

(DT The (NN) girl (VB) wrote (DT) a (NN) letter (PRP) to (Pro N) her (NN) cousin.

4.SHALLOW PARSER

Shallow parsing is important for many NLP applications, particularly for information extraction, information retrieval, cross linguistic information access, question answering, etc. Shallow parsing consists of part of speech tagging and chunking. These methods can do firstly segmentation and labeling sequences of characters and then chunking. In this system, shallow parser method used for syntax. Shallow Parsing is the task of recovering only a limited amount of syntactic information from NL sentences. Shallow parser has proved to be a useful technology for written and spoken language domain. Shallow parser is a division of words that together consist of grammatical unit.

4.1. Architecture of shallow parser

Architecture of shallow parser consists of three main parts. This parser can do firstly input sentence segmentation and labeling sequences of characters (i.e. noun, verb, adverb, pronoun, etc.) and then chunking.

The chunk identification is a fundamental task for shallow parsing. A chunk is a textual unit of adjacent word tokens. Chunks are non-overlapping regions of a text and cannot contain other chunks. Some words in a sentence may not be grouped into a group. Relation finding is the final step of shallow parser as shown in Figure 2.

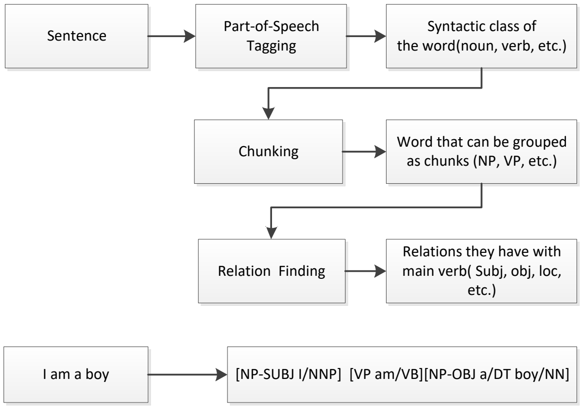
Sentence
Part-of-Speech Tagging
Syntactic class of the word(noun, verb, etc.)
Chunking
Word that can be grouped as chunks (NP, VP, etc.)
Relation Finding
Relations they have with main verb(Subj, obj, loc, etc.)
I am a boy
[NP-SUBJ I/NNP] [VP am/VB][NP-OBJ a/DT boy/NN]

Figure 2Typical shallow parser architecture

4.2. Phrase chunking with shallow parer

Phrase chunking assigns a tag to word sequence. Typical phrase chunking consists of noun phrase and verb phrase.

4.2.1. Noun Phrase chunking

A noun phrase (NP) is a phrase that head is a noun or pronoun by a set of modifiers. Noun phrase typically consist of determiner, adjectives and noun or pronoun.

NP	<PRP> <DT> <NN>
NP	<PRP> <JJ> <NN>
NP	<PRP> <POS> <NN>
NP	<POS> <JJ> <PRP> <DT> <NN>
NP	<NN> <PRP> <POS> <NN>
NP	<JJ> <PRP> <NN>
NP	<JJ> <DT> <NN> <PRP> <Pro N>

Table 1. Sample of Noun Phrase Chunking

Rules for Rewriting Noun Phrases

It can characterize certain types of NP.
Noun (N):
mya mya, walking, kindness, apple.
Determiner (DT) + (N):
The sum, some, some people, a god.
Determiner (DT) + Adjective (ADJ) + Noun (N):
An ancient city, a young boy, the first person, these beautiful flowers.
In this paper, the following NP patterns are regarded as the rules of phrase structure:
NP   N → (NP includes N).
NP   DT→N (NP includes DT and N).
NP   DT→ADJ N (NP includes DT and ADJ and N).
These three rules can be formed into a single rule.
NP (DT)   →(ADJ) (N)
This rule can be expanded into the three separate rules. The single rule has one expansion that it did not expect because DT and ADJ are optional; it can rewrite NP as follows;
NP   ADJ → N
If there are well-formed NP structures that consist of an adjective and a noun, then it would have to revise it to exclude structure that is not well formed. Of course, English indeed permit NPs that consist of ADJ and N.

4.2.2. Verb Phrase chunking

A single verb constructed into a verb phrase. The verb phrase includes various combinations of the main verb and any auxiliary verb, complements, and adjuncts.

VP	<V>
VP	<V> <PP>
VP	<V> <Pre P> <NP>
VP	<V> <NP> <PP>

Table 2. Sample of Verb Phrase Chunking

Rules for Rewriting Sentences and Verb Phrases

In rewriting sentences and verb phrases have two basic constituent parts.
S → NP VP
The following expansions for identifying verb phrases reveal that the structures on the right are VPs. The labels under parts of the VPs indicate the group constituents of those structures.
(a) Tom jogged.
(b) Tom won a bicycle.
(c) Tom won the bike in May.
The notation PP stands for prepositional phrase. Every preposition includes a preposition and a noun phrase. The rule for PP is this:
PP → Prep NP
It can formulate by using the following verb phrases.
VP → V
VP → V + NP
VP → V + PP
By using this rule, sentence can produce the following example:

(a) Jane swan in the pool.
(b) Aung Aung ran around the track.
(c) She flew to Japan.

Sentence has formulated by using the following phrase structure rules:
S → NP
NP → (DT) (ADJ) (N)
VP → V (NP) (PP)
PP → Prep NP
These rules describe every sentence has a NP and a VP. That every NP has an N, every VP has a V and every PP has Prep and a NP.

4.3. Shallow Parser Algorithm

Shallow Parsing is based on partial parsing and identifies the grammatical relations between separate chunks. Then the head and dependent chunks are specified for each relation [3]. To determine the grammatical function, the system use shallow parsing approach which is done by inducing a Begin (B), Inside (I) and Outside (O) tags. For instance, if it tries to chunk a sentence into Noun Phrase, Verb Phrase, and Preposition Phrase chunks, it represented the following tags:

B
The word begins a chunk of type (NP, VP, PP, etc.).

I
The word belongs to a chunk of type but does not begin it.

O
The word does not belong any chunk.

NPs and VPs are non-overlapping and non-recursive these three tags suffice to chunk a sentence. For example,
[NP He] [VP saw] [NP the big dog].
The sentence should be tagged as:
(B-NP)He (B-VP) saw (B-NP) the (I-NP) big (I-NP) dog (O).

Phrases	Chunk tags
Noun Phrase-NP	Begin chunk tag B-NP
	Internal chunk tag I-NP
Verb Phrase-VP	Begin chunk tag B-VP
	Internal chunk tag I-VP
Conjunction Phrase-CP	Begin chunk tag B-CP
	Internal chunk tag I-CP
Adjective Phrase-ADJP	Begin chunk tag B-ADJP
	Internal chunk tag I-ADJP
Adverb Phrase-ADVP	Begin chunk tag B-ADVP
	Internal chunk tag I-ADVP
Outside	Outside chunk tag O

Table 3. IOB Tag of Shallow Parser

5.PATTERN MATCHING PROCESS

To check sentence construction error, the system uses pattern-matching process and chunking process together with partial knowledge bases such as subject, verb, object, place, manner, time and reason knowledge bases. Pattern matching process is to achieve error detection in grammar analysis.

In this step, input text is divided into segments which correspond to certain syntactically unit. When the system encounters the sentence construction errors or grammar error such as subject-verb agreement error (they plays), determiner and noun disagreement error (a boys), and tense misuses error (he played the ball tomorrow.), the system can correct automatically and can only correct this type of errors.

For auto verb form correction, the system uses subject and verb agreement knowledge base. First, it tags the subject phrase as plural or singular subject and also verb phrase tag like that and then checks they agree or not .And also the system can correct verb form for tense mismatch. The system can check many types of complex sentences and clauses construction in English effectively and correctly with the help of pattern-matching process.

6.GENERATING GRAMMTICAL SENTENCE

Generating grammatical sentence is displayed the correct sentence for output. The paper supported to generate correct English writing using rule-based lexicon which stores any English alphabet words. Then error message is given by the grammar checker when there is no in fact no error in the text.

The Rule-based approach uses predefined grammar rules for the sentences. The random sentence is parsed against these rules and then the grammar checking analysis is applied on this sentence [6]. It works on complete sentences but not on single word. Grammar checking rules that will identify the correct sentence by matching rules in the rules directory which depend on POS tags. The pattern is a sequence of words, POS tags or chunk. If this sequence pattern is found in pattern matching, it can be declared as a correct sentence. This system used thousands of rules.

7.CONCLUSION

The paper can provide effective and efficient services to improve grammar checking process and can understanding the knowledge of shallow parser. This system can check the spelling error, the complex sentences and clauses according to the pattern rules. By using shallow parser, the system can shrink the searching time and memory spaces for storing words and can tag 94% accuracy rate for ambiguous words, 96% for sentence construction error checking. The result of the system can be used not only in grammar analyzer weather it is rule-based or shallow parsing but also in machine translation system, summarization and information extraction system. At last, the system the efficient and reliable output sentences with high accuracy.

8.ACKNOWLEDGEMENT

I would like to express my gratitude to all persons who directly or indirectly contributed towards the success of the paper. The authors would like to thank to all teacher from Centre of Excellence for Software Development and Training, Myanmar, who gives suggestions and advices for submission of paper. I would like to express grateful thanks to Dr. Myint Myint Khaing, Pro Rector of University of Computer Studies (Pin Long), for providing her guidance and necessary instructions throughout the development of this paper.

REFERENCES

[1] Nivedita S. Bhirud, R.P. Bhavsar, B.V. Pawar, "Grammar Checkers for Natural Languages: A Review", 2017.

[2] D. Naber, "A Rule-Based Style and Grammar Checker", 28 August 2003.

[3] Sneha Asopa, Pooja Asopa, Iti Mathur and Nisheeth Joshi, "A Shallow Parsing Model for Hindi Using Conditional Random Field", 2018.

[4] GurjitKaur, Kamaldeep Kaur, Parminder Singh, "Spell Checker for Punjabi Language Using Deep Neural Network", 2019.

[5] Madhvi Soni, Jitendra Singh Thakur,"A Systematic Review of Automated Grammar Checking in English Language", 2018.

[6] Roshan Fernandes, Dr. Rio D'Souza G. L., "Semantic Analysis of Reviews Provided by Mobile Web Services Using Rule Based and Supervised Machine Learning Techniques**",** 2018.

[7] T.M. Win, Z.M. Than, "Rule-based Chunk Level Grammar Checking", 2008.