Malay Declarative Sentence: Visualization and Sentence Correction

Yusnita binti Muhamad Noor
School of Computing
College of Arts and Sciences
Universiti Utara Malaysia
06010 Sintok, Kedah
s92715@student.uum.edu.my

Zulikha binti Jamaludin
School of Computing
College of Arts and Sciences
Universiti Utara Malaysia
06010 Sintok, Kedah
zulie@uum.edu.my

Abstract — Language researchers introduced sentence parse tree visualizations to help in understanding sentence structure, especially in English. Among the applications introduced, phpSyntaxTree and RSyntaxTree give users the opportunity to visualize an English sentence through online interaction. In Malaysia, language research in sentence parse tree visualization for Malay (BM) still hasn't attracted enough researchers to produce a prototype as has been done in English. However, several parsers for BM sentences have been introduced. The parsers will produce a parse tree as an output from the parsing process. Based on the parsers, methods can be extended to produce sentence parse tree visualizations for BM. Parsers for checking sentence structure need to be included in visualization methods. Visualization methods involved consist of 1) tokenizing, 2) checking the number of words, 3) assigning word class, 4) checking spelling or conjunctions, 5) checking and matching with formula, 6) suggestion or visualization 7) word attributes and 8) visualization from a corpus. A prototype for the methods introduced is still under the development and improvement process. However output from the development process in validating the sentence, giving corrections for incorrect sentences and creating a parse tree has had good output results.

Keywords — *BM sentence parser, parser with suggestions, BM sentence parse tree visualization, parse tree, method in BM sentence parse tree visualization.*

I. INTRODUCTION

Language researchers seeking to produce standard rules for using a language have conducted many studies. For example, Chomsky (1957) introduced the theory of transformational generative grammar to help understand English sentence structure. He used parse tree representation in creating that theory. Until today, this theory has been used in both traditional and automatic-based language studies. It also led to the development of parse tree visualization especially in English, as have been produced in systems like RSyntaxTree [1], SynView [2], VAST [3], phpSyntaxTree [4] and Link Grammar [5].

Unfortunately, research in parse tree visualization In Malay language (BM) studies, still hasn't attracted enough researchers to produce an automatic-based tool in presenting a sentence. However, sentence parsers, one of the needed tools in sentence parse tree visualization in BM, have been introduced in [6] and [7]. Both parsers will analyze a sentence according to BM sentence rules and a correct grammatical sentence will produce a parse tree as the output. The purpose of the parser is to validate a sentence structure in terms of syntax and semantics.

In making BM more progressive in computer-based processing, this paper will discuss methods involved in the development of BM sentence parse tree visualization. To visualize a grammatical sentence, a sentence parser also needs to be included. Hence, methods used by previous BM parsers will be followed and extended.

In Section 2, this paper reviews previous BM sentence parsers. In Section 3, BM sentence parse tree visualization is discussed, and two of the related BM parsers are included. Next, in Section 4, methods involved in BM parse tree visualization are introduced. Finally, Section 5 discusses the differences between previous BM parsers and the proposed BM sentence parse tree visualization.

II. BM SENTENCE PARSER

References [6], [7], [8] and [9] have conducted studies on a sentence parser for BM. The studies will parse the sentence following rules provided in the system and the output produced will inform the user about whether the input sentence is categorized correctly or not based on the theory of transformational-generative grammar. Output from [6] and [7] will be produced in the form of parse tree. Each parser was developed for a different purpose. References [8] and [9] developed the parser to analyze the validity of a sentence in term of syntax structure. The parser in [6] will check the sentence's validity in term of syntax and semantics. The statistical parser in [7] aims to reduce structural ambiguity in a sentence by introducing the probabilistic approach in parsing

the sentence. The processes involved in all BM parsers can be seen to feature three approaches. These are:

1. After part-of-speech (POS) tagging, the applications will match each word with the lexicon to validate the order of words or word classes according to the rules;
2. Either output (parse tree or Parlog clauses) or error messages will be produced according to the rules; and
3. The applications also use a top-down approach, which is a recursive descent parser.

To date, no parser that can propose a correction in terms of sentence structure has been created for any language, including BM. Therefore, the proposed method for correcting a sentence in this study contributes a new idea to language-based studies generally, but especially to BM. This method will not issue a proposed correction for a sentence, which cannot be analyzed by the system due to an incorrect use of language or the use of a severe sentence structure according to BM context-free grammar (CFG). Rather an error message will be displayed so that the user will enter the sentence again.

III. BM SENTENCE PARSE TREE VISUALIZATION

In explaining a language structure, language researchers use a parse tree representation, which involves grammar formation among the words. In BM, the same approach can be seen in [10], [11] and [12] and in other BM sentence studies. Because no computer-based system has been introduced thus far, the representations were done in a paper-based format. This format is time consuming for research work and also needs more space to explain the structure. Users also cannot get a full understanding when referring to the limited representations.

Sentence parse tree representation or visualization need phrase structure rules, which are known as context-free grammar (CFG). A BM sentence is categorized as a context-free grammar in which there is a subject and a predicate in a sentence [13]. A BM sentence consists of four basic CFG rules, which are a combination of noun phrase, verb phrase, adjective phrase and prepositional phrase.

A BM sentence parser as in [6] has resulted in a parse tree representation for the correct use of a sentence. It will check both syntax and semantics. The data stored in the lexicon are divided according to human or animal for the purpose in semantic analysis. Fig. 1 shows the example of tree structures generated by the prototype for the BM input sentences "*Sekawan lembu sedang melintasi jalanraya tersebut"* and "*Fatimah mengusahakan perniagaan di Ipoh*". Each generated tree structure is based on the sentence applicability to either an animal or a human.

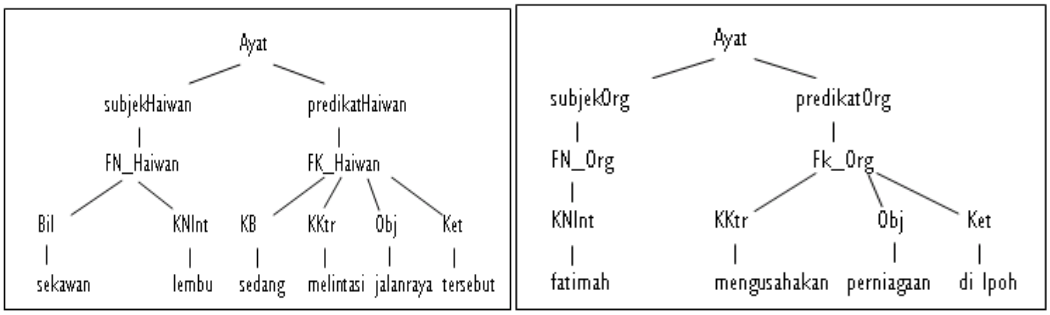


Figure 1. Parse tree produced in [6]

A statistical parser as in [7] will automatically assign a probability value for words in a sentence according to the value assigned in the database. The purpose is to reduce sentence structure ambiguity in the parse tree. An ambiguous parse tree will have more than one parse tree visualization. For example, the BM sentence "*nasi godak kenduri sangat sedap*" can produce two different parse trees, which are shown in Fig. 2. After the parse tree is generated, a message box showing the probability of each parse tree is produced. The higher probability value can be considered to have the most accurate parse tree structure.

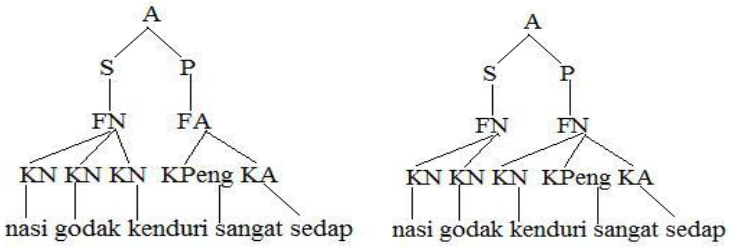


Figure 2. Parse tree in statistical parser as in [7]

IV. METHOD IN BM SENTENCE PARSE TREE VISUALIZATION

As mentioned, methods in BM sentence parsers consist of 1) POS tagging or matching each word with the lexicon, 2) matching with rules and 3) output. The same methods will be used in this study to produce a sentence parser with a few additions.

This study aims to produce a prototype that can check a sentence structure according to BM CFG and will produce a parse tree visualization. In the process of parsing (syntax checking), a sentence with structure correction will be proposed to the user if the input sentence does not follow CFG rules. For a correct sentence, a parse tree visualization will be produced. Each node in the parse tree will have a link to the abbreviated information or to view the related word attributes. Nodes for an input sentence will have a link to a word attributes page that will list the attributes of word class, word derivation, word translation, audio and image of the selected word. It also will display a list of sentence examples that will be retrieved from a corpus repository. The sentence retrieved is according to the selected word. The purpose in showing the sentence is to help users understand more about each word used in the input sentence. Then, each sentence will have a link to a new parse tree visualization page.

As compared to the [6] and [7] studies, the methods introduced in this paper will be limited in scope only to syntax analysis and the basic sentence. Semantic factors and sentence structure ambiguity will not be involved. This means that, if a sentence follows the CFG rules but is incorrect in terms of semantics, that sentence still can produce a parse tree, and it is considered to be a correct sentence. Additionally, an ambiguous word in a sentence will have more than one POS which will produce more than one parse tree visualization depends on the CFG.

The methods involved are shown in Fig. 3. The methods can be divided into three phrases, which are the parsing process, the suggestion process and the parse tree process. After receiving input, the parsing process will start by tokenizing the sentence into words until the rules match. The suggestion process will play its role when there is a problem in matching the syntax structure. Lastly, the parse tree will be produced for a correct grammatical sentence as well as the word attributes page and parse tree visualization for the sentence example.

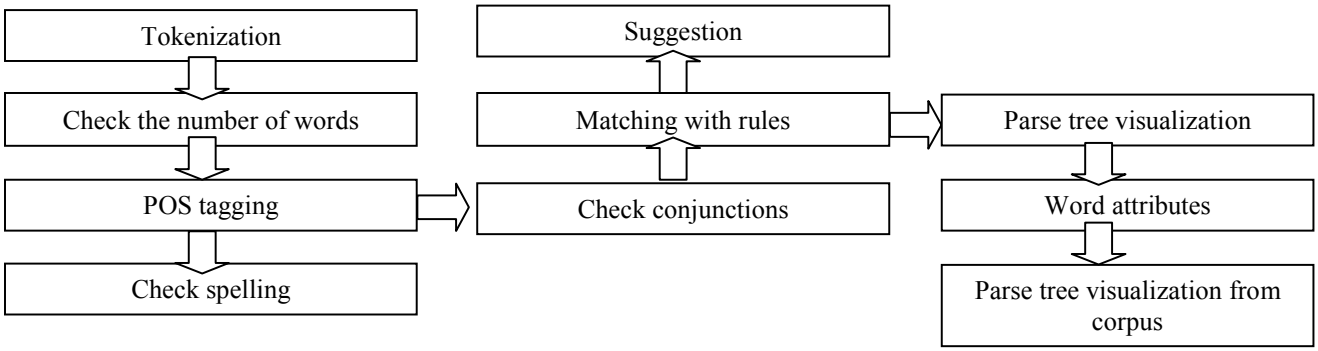


Figure 3. Methods in visualizing BM sentence parse tree

Processes involved in Figure 3 are described below:

1) Token sentence into words.
2) Sentence conditions are reviewed to ensure that the input sentence is more than one word.
3) Each word will be matched with the appropriate POS as provided in the repository.
4) For unmatched words with appropriate POS, the spell checking process will check if the word is categorized under special noun words, date, number or address. If the spelling checking process still can't assign the appropriate POS, an error message will be displayed. Otherwise, the sentence condition will be checked. This step is to determine that the sentence is not a compound sentence by checking the conjunctions. If there is a conjunction in the words list, the system will produce an error message because only a basic sentence will be processed.
5) Determination of the validity of the input sentence is decided by matching the structure of input sentence or the order of its word classes with CFG. CFG produced as in [11] is used as a reference. When the matching is successful, the parser will continue to display the output. Otherwise it would require the proposed method of correction to be carried out.
6) In the suggestion stage, similar CFG for the input sentence will be searched according to the order of word classes listed for input and CFG. Only one similar CFG will be taken. Replacement will be done by changing the position of the words (input sentence) according to the word class order in CFG that have been retrieved. Hence, the proposed sentence will be displayed to the user. However, for

sentences that are too difficult to change, it only will give an error message

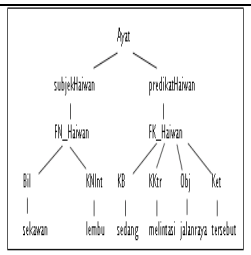
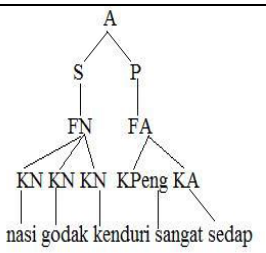
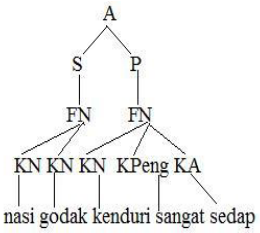
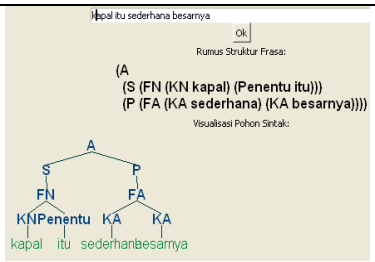
For a correct sentence as determined by the parser, it will display output in CFG (order of CFG) and parse tree visualization. Each node in the parse tree will have a link. The node for the abbreviation word classes will show the meaning of word class and nodes for words in input sentence will be linked to a word attributes page.

7. The word attributes page will display the attributes for the selected word, which will consist of the word class, word derivation, word translation in BI, audio, image and a list of sentence examples. All sentences that consist of the selected word will be retrieved from the corpus. Each sentence example will have a link to a new parse tree visualization page.
8. The selected sentence from the word attributes page will visualize a syntax tree in a new page.

V. BM SENTENCE PARSER VS BM SENTENCE PARSE TREE VISUALIZATION

A parser and parse tree visualization are two different tools in language processing. Most parser prototypes that have been produced will display the output in parse tree representation to show that the parser has correctly parsed the sentence. In this study, in visualizing the parse tree, methods introduced require a parser with sentence correction, word attributes and parse tree visualization from the corpus. The differences between previous BM parsers and the proposed BM sentence parse tree visualization are shown in Table 1.

Table 1. Previous BM parsers vs. the proposed BM sentence parse tree visualization

Characteristics	**Rosmah's parser [8]**	**Suzaimah's parser [9]**	**Ahmad Izuddin's parser [6]**	**Noor Hafhizah's parser [7]**	**Proposed BM parse tree visualization**
Focus analysis	Syntax	Syntax	Syntax and semantic	Syntax for sentence with structural ambiguity	Syntax
Output	Subject: NF (*Cakera liut, itu*) Predicate : A (*murah*)	For example, input [*ali, makan*] will produce the following output: *P=ayat(frasa_nama(ung_nama(ali))), frasa_kerja(ung_kerja(kata_kerja(makan))) succeeded.*		 	 Word attributes Selected word: *kapal* Word class: noun Derivation words: - Translation: ship Image: Audio: "click to hear" Sentence examples: 1. *Kapal besar* 2. Suggestion: Sentence examples above have a link to new parse tree
Method	1. After POS tagging, the applications will match each word with the lexicon to validate the order of words or word classes according to the rules. 2. If it fits the rules, either output (parse tree or Parlog clauses) or an error message will be produced.				1) Tokenization 2) Check the number of words 3) Assign word class 4) Check spelling or check conjunctions 5) Checking and matching with rules 6) Suggestion or parse tree visualization

		7) Word attributes 8) Parse tree visualization from corpus

VI. CONCLUSION

Sentence parse tree visualization can be utilized in different ways. It can be used for understanding a sentence structure as is done in SynView. It can be used for syntax or sentence checking. It also can be used to show the relationship between the phrase and word classes, or it can be used to show the relationship between the grammars as was done in Link Grammar system.

In BM, no developed prototypes have been produced that focus on parse tree visualization. BM parsers as in [6] and [7] aim to check the sentence and to reduce the structural ambiguity in the parse tree. The proposed parse tree visualization in this study refers to both studies as guides in producing a BM parser with sentence correction as a needed sub-tool as well as word attributes. Methods involved in the prototype development are discussed in this paper.

The process involved in presenting a word attributes and sentence examples in the parse tree visualization will be discussed in our future work.

REFERENCES

[1] Y. Hasebe. (2012, April, 08). RSyntaxTree [online]. Available: http://yohasebe.com/rsyntaxtree/

[2] C. Behrenberg. (2009). SynView v0.3 user's manual [Online]. Retrieved Dec 22, 2010, Available: http://www.christianbehrenberg.de/files/SynView/SynView_source.rar,

[3] F. J. Almeida-Martinez, J. Urquiza-Fuentes, and A. Velzquez-Iturbide, "Visualization of Syntax Trees for Language Processing Courses." *Journal of Universal Computer Science*, vol. 15(7), pp. 1546-1561. 2009.

[4] M. Eisenbach and A. Eisenbach. (2003). phpSyntaxTree-drawing syntax trees made easy

[Online]. Retrieved Dec 20, 2010, Available: http://www.ironcreek.net/phpsyntaxtree/

[5] D. Sleator and D. Temperley. (1993). Parsing English with a link grammar [Online]. Retrieved Dec 29, 2010, Available:http://www.cs.cmu.edu/afs/cs.cmu.edu/project/link/pub/www/papers/ps/LG-IWPT93.pdf

[6] A. I. Zainal Abidin, S. P. Yong, R. Kasbon, and H. Azman. "Utilizing top-down parsing technique in the development of a Malay language sentence parser, "in *Proc. of the 2nd International Conference on Informatics*, 2007, pp. 128-134.

[7] N. F. Abd. Rahim, "A statistical parser to reduce structural ambiguity in Malay grammar rules," M.S. Thesis, Universiti Malaya, Kuala Lumpur, 2011.

[8] R. Abdul Latif, "Penyemak Sintaksis Ayat Bahasa Malaysia," M.S. thesis, Universiti Kebangsaan Malaysia, Bangi, 1995.

[9] S. Ramli, "Reka bentuk dan implementasi suatu penghurai bahasa Melayu menggunakan sistem logik selari," M.S. thesis, Universiti Putra Malaysia, Selangor, 2002.

[10] Z. Yusoff, *Cintailah bahasa kita, suatu tanggapan linguistik berkomputer*. Universiti Sains Malaysia: Pulau Pinang, Malaysia, 1998.

[11] N. S. Karim, F. M. Onn, H. Musa, and A. H. Mahmood, *Tatabahasa Dewan Edisi Ketiga*, Dewan Bahasa dan Pustaka: Kuala Lumpur, 2009.

[12] A. Hassan, S. L. Jaya Rohani, R. Ayob, and Z. Osman, *Sintaksis, Siri pengajaran dan pembelajaran Bahasa Melayu*. PTS Professional Publishing Sdn. Bhd.: Kuala Lumpur, Malaysia (2006).

[13] M. J. Ab Aziz, F. Dato' Ahmad, and A. A. Abdul Ghani. "Pola Grammar Technique to Identify Subject and Predicate in Malaysian Language," *The Second International Joint Conference on Natural Language Processing*, 2005, pp. 185-190.