VOL. 10, NO. 3, FEBRUARY 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

PARSE TREE VISUALIZATION FOR MALAY SENTENCE (BMTutor)

Yusnita binti Muhamad Noor and Zulikha binti Jamaludin
School of Computing, College of Arts and Sciences, Universiti Utara Malaysia, Sintok, Kedah, Malaysia.
E-Mail: s92715@student.uum.edu.my

ABSTRACT

In Malaysia, various efforts have been made by the government and language researchers in improving student's ability of mastering Malay language (BM) due to their poor ability in grammar and sentence structure. In terms of technology, to date, there is no computer software or a prototype that is available that can help students in learning the BM sentence structure. Thus, BMTutor is introduced as a solution to this problem. BMTutor is a prototype for visualizing Malay sentence combined with sentence checker, sentence correction and word attribute components. BMTutor is intended to facilitate the learning process of sentence construction and grammatical structure in BM. It is also to enhance the learning process in BM that can be used by communities, especially students. An algorithm in designing BMTutor is discussed in this paper. The algorithm of the software is done sequentially as followed: 1) tokenizing 2) checking the number of words, 3) searching and comparing process to check the spelling or conjunctions, 4) assigning each word with a certain word class, 5) matching with rules, and 6) delivering/producing output (sentence correction or parse tree visualization, word attribute components, and parse tree from sentence examples). Based on the testing conducted, output from the development process shows that the prototype can correct all 15 invalid sentences and can produce parse tree visualization for all 20 sentences.

Keywords: BMTutor, BM sentence structure, parse tree visualization, sentence checker, sentence correction, word, attribute components.

INTRODUCTION

In Malay language (BM), lack of proficiency in understanding grammatical sentences among the community is nothing new, especially among students (Zaharani, 2012; Nor Hashimah, 2010). As described by Bagavathy (2005), the issue of poor understanding of BM grammar among students (especially school students) is caused by the difficulty in understanding the grammatical structure of the sentence. Moreover, Daing Melebek (2004) stated that school students are facing a problem in understanding BM sentence structure because they do not understand and do not know how to use the correct combination of phrases and word classes. There are some students that do not understand grammar until they have graduated. The students also faced a problem in forming a correct sentence and they cannot differentiate the type of word classes which can be seen from the assigned essay writing (Daing Melebek, 2004; Nawi, 2003). In terms of technology, to date, there is no computer software or a prototype that is available that can help students in learning the BM sentence structure.

Therefore, BMTutor is introduced to help Malay speakers, especially secondary school students, to undertstand BM sentence structure and word classes. Having the BMTutor, students have the opportunity to explore and learn about the structure of BM sentences by learning the phrase structure formation and the word attribute through a computerized visualization method.

BMTutor is a prototype for visualizing a BM sentence in parse tree combined with sentence checker, sentence correction and word attribute components. Parse tree visualization is used in explaining a sentence structure because this method is also used by linguists in understanding the structure of a sentence. It is a tool to increase the correct use of BM, as well as to help the community to enhance their BM, especially among the students.

The BMTutor only focuses on declarative sentence that does not exceed 14 words. Sample of such sentences were collected from the secondary school BM textbooks used by the Form 1 until Form 5 students. The collected sentences were analyzed to determine the BM syntax rules for the development of BMTutor. The scope of this study is limited only for a sentence that does not exceed 14 words The number of words is based on the recommendation by Abdullah (2008) who mentioned that an understandable sentence is made up of no more than 14 words. Furthermore, the prototype is intended to help students in understanding the sentence structure from the most basic level.

BMTUTOR COMPONENTS

BMTutor is a parse tree visualization package combined with sentence checker, sentence correction, parse tree visualization, word attribute components and parse tree visualization for sentence examples. The sentence checker needs to be included to produce a parse tree only for grammatical sentence. After the checking process, a sentence correction will be proposed for any invalid sentence entered. On the other hand, for each correct sentence, a parse tree visualization will be displayed. Each node in the parse tree has a hyperlink, which will display the word attribute components (word class, word derivation, translation, image and sentence examples) for each elected word. Each sentence example will also have a hyperlink to visualize a new parse tree. An error message will be produced for any incorrect sentence entered before a new sentence is produced, and the error message will also be produced if there is any unmatched word with the database.

The BMTutor comprises of five components: 1) parse tree visualization, 2) sentence checker, 3) sentence correction, 4) word attribute components, and 5) error message.

Parse tree visualization: The parse tree is divided into subject and predicate in a hierarchical design. There are two categories of parse tree visualization which are through input sentence and through sentence examples.

A. Parse tree for input sentence

Parse tree will be visualized only when the entered sentence is correct according to the BM sentence rules for declarative sentence that does not exceed 14 words. The processes involved in visualizing a sentence are:

1. Sentence checker will start by counting the number of words according to the rules provided.
2. Each word will be matched with an appropriate word class.
3. For any unmatched word, the spell checker will check the word according to the rules provided. If the spell checker failed to find the appropriate solution, an error message will be displayed.
4. Sentence condition will be checked for each word and its appropriate word class to ensure that the sentence received for further analysis is only for declarative sentence. If not, an error message will be displayed.
5. Syntax checking is performed by matching each syntax in the input sentence with the appropriate rules provided.
6. Matched syntax will display parse tree visualization, otherwise a sentence correction will be displayed.
7. Each node in the parse tree will have a hyperlink to view word attribute components.

B. Parse tree from sentence examples

A list of sentence examples is included in the word attribute components page. The sentences are retrieved according to the words choice in the parse tree for input sentence. Each sentence has a hyperlink to make new parse tree visualization.

Sentence checker: To ensure that the sentence received is under the research scope, the sentence checker will play its role by performing word class and rules match.

Sentence correction: If the entered sentence does not match with the appropriate rules provided, a sentence correction will be proposed. The combination of rules used in the input sentence will be matched with the rules provided in the database, and there will be some changes in the input sentence.

Word attribute components: As mentioned previously, each node in the parse tree visualization for the input sentence will have a hyperlink to the word attribute components page. The attributes include word class, word derivation, translation, image and sentence examples. Each attribute is displayed according to the word choice in the parse tree. Sentence examples will also be retrieved according to the word. From the sentence examples, user may view different types of sentences in different context for each word in the input sentence. Each sentence has a hyperlink to make new parse tree visualization.

Error message: Error message will be displayed for any invalid sentence entered, such as when the input sentence is not within the research scope, and if the word entered is not available in the database.

BM SENTENCE STRUCTURE

The Malay language is a context-free grammar, where there is a subject and a predicate in a sentence (Mohd Juzaiddin, 2005). It requires a set of grammar rules, also known as context-free grammar (CFG) in English or phrase structure rules (RSF) in BM (Mohd Juzaiddin, 2007). Every sentence used in a language is constructed according to the CFG, especially in BM. For this reason, there is a lot of researches that have been conducted in language studies to produce a good sentence structure, especially in BM. Sentence parser or sentence checker is one of the technology tools that can be used to validate a sentence in order to produce a good sentence structure. The tool parses the sentence according to the CFG provided. Its function is to validate the construction of words used in a sentence as in BMTutor. If a sentence is structured according to the rules of CFG, the parser will classify the sentence as true. In the BMTutor, the basic CFG rules for BM provided by Nik Safiah, Farid, Hashim and Abdul Hamid (2009) are used as a reference.

Based on the basic rules, this study conducted several activities in gathering specific rules according to the research scope. The activities were collecting sentence, sketching the parse tree for each collected sentence and verifying the validity of the rules collected.

Collecting sentence: Sentences were collected from the BM textbooks for Form 1 until Form 5 students according to the research scope. The sentences were then divided into several categories based on the number of words.

Sketching parse tree: For each collected sentence, a parse tree was sketched on the paper as shown in Figure-1. The rules were then collected from the sketching process. An example of how BMTutor collects the rules is shown in Figure-2.

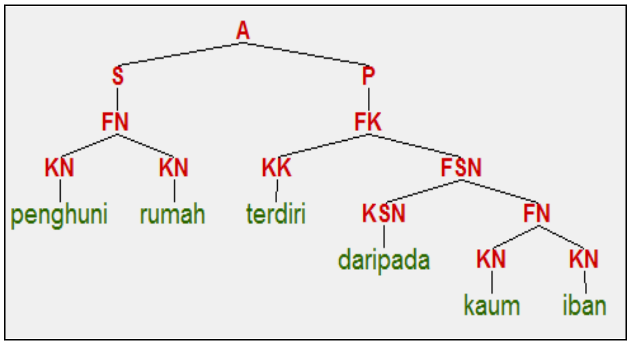


Figure-1. Parse tree.

A- S P
S- FN
FN-KN KN
P-FK
FK-KK FSN
FSN- KSN FN
FN- KN KN

A = Sentence
S= Subject
P= Predicate
FN= Noun phrase
FK= Verb phrase
FSN= Prepositional phrase
KN-Noun
KK= Verb

Figure-2. CFG for a sentence.

The sketch was designed by the parse tree in a hierarchical display. Similar to the parse tree, the display divided the sentence into subject and predicate. Each subject and predicate would have different combination of phrases and word classes. The display of the phrase structure and word classes depends on the rules involved in a sentence. After the CFG was collected, each phrase and word classes were separated as shown in Figure 3. The rules collected in Figure-3 were coded into a programming style (Python was used in this study) so that the BMTutor can analyze the sentence.

Phrase structure
FN- KN KN, FK- KK FSN, FSN- KSN FN
Word class: KN, KK, KSN

Figure-3. Rules for a sentence.

Rules validation

The collected rules were validated by a member of the Dewan Munsyi (DM). The member of the DM is an expert in BM as approved by Dewan Bahasa dan Pustaka. The expert analyzed the collected sentences, parse tree and rules to ensure that the collected rules were correct. The BMTutor was developed based on these rules. The following are the processes of performing the validation method:

1. Checking the type of sentences for all the collected sentences within the scope of the study, namely the Malay declarative sentences. There are some isolated sentences which are difficult to analyze as well as other clauses and sentences with no subject and sentences containing discourse within the scope of this study.
2. Reviewing the part-of-speech (POS) tagging for each word in the sentence as sketched in the parse tree.
3. Reviewing the structure of the parse tree sketches (subject-predicate division, phrase, word and word class), especially in terms of the distribution of phrases in order to comply with the formation of correct sentence structure.
4. After reviewing the parse tree sketches and sentences, the results of the review are sent back to the researcher. Once the researcher had made the necessary corrections, the expert will review the order of the collected rules by the division of phrases in accordance with the BM sentence patterns such as a noun phrase + noun phrase, noun phrase + verb phrase, noun phrase + adjective phrase and noun phrase + prepositional phrase.
5. Confirming the order of the collected rules according to the BM sentence patterns in the correct order of BM CFG.

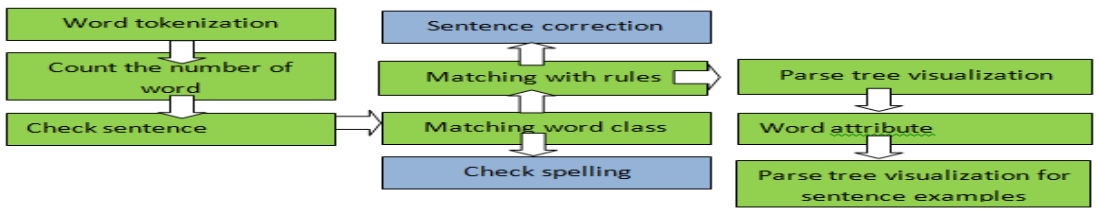


Figure-4. BMTutor design algorithms.

The algorithm for designing the BMTutor as shown in Figure-4 can be described according to the following pseudo code:

Steps

```
1.Input (sentence)
2.Token each word in the sentence
3.Count the number of word
3.1    If the number of word less than two, print
       "Please enter a sentence", Back to (1)"
```

```
3.2      If the number of word is more than 14, print
         “Sentence with more than 14 words are
         accepted”, Back to (1)
3.3      If not, proceed to (4)
4.Check sentence condition
4.1      If there is a conjunction, print "compound
         sentences are not analyzed", End.
4.2      Else, proceed to (5)
5.Matching word class
5.1      Match each word with CFG
5.1.1    For unmatched word, proceed to (5.2)
5.1.2    If it is a success, proceed to (6)
5.2      Check spelling
5.2.1    Matching with format (address/date/special
         noun), token as noun, repeat step (5.1)
5.2.2    Else, print “Sorry, the word " " is not in the
         database", End.
6.Matching with rules
6.1      Match each word class with CFG
6.1.1    If it is a success, display parse tree
         visualization with a hyperlink
6.1.1.1  Give suggestion to the user to click on the
         node. If user click the node, Proceed to (7)
6.1.2    Else, print error message.
6.1.2.1  Give sentence suggestion
6.1.2.1.1   Check CFG-like matching
6.1.2.1.2   Change the input words according to the
            CFG
6.1.2.1.3   Print new sentence suggestion
6.1.2.1.4   Else, print error message, End.
7.Word attribute components
7.1      Print all the components for the selected
         word.
7.2      Give suggestion to the user to choose the
         sentence examples.
7.2.1    If user click the sentence, go to (8)
7.2.2    Else, end.
8.Parse tree visualization for sentence examples
8.1      Repeat step (6.1.1)
9.End.
```

The algorithms used can also be described in details as follows:

1. Token and count the number of words

$$A = P_2 \leq P_n \leq P_{14} \qquad (1)$$

A sentence (A) is accepted for more than one word and must not exceed 14 word (P) based on the research scope. Otherwise, an error message will be produced. Then, the tagging step will proceed.

2. Check sentence condition

$$KK = \{KN, KK, KA, KS, KT\}$$
$$JA = \{AP, AT, AS, APE, AM\}$$
$$SA = \{KK_5 \in JA \setminus AT, AS, APE, AM\} \qquad (2)$$

The word class (KK) is divided into noun (KN), verb (KK), adjective (KA), preposition (KS) and others, named as function words (KT). The types of sentence (JA) in BM consist of declarative sentences (AP), interrogative sentences (AT), exclamation sentences (AS), command sentences (APE) and compound sentences (AM). To check the sentence condition (SA), all words that are matched with KK will be accepted except (\) for those that matched with the word class under AT, AS, APE and AM.

3. Tagging or matching word class

$$P_i \in KK_{5,}\ i = 2,...,14 \qquad (3)$$

The words in a sentence (P), which are between 2 to 14 words, must be within the word class (KK) as provided in the database.

4. Spell checker

date {dd/mm/yy}
address {no, kg..00000,..}
proper noun {‘capital letter...’}
KN = {address, date, proper noun, number}
$$SE = \{P_i \in KK_5 \cap P_i \in KN\},\ i = 1, ...14 \qquad (4)$$

The spell checker (SE) will analyze a word (P) which is not in the word class (KK) list. The word will be matched with four formats in the spell checking process. If it matched with any format specified, the word will be categorized as a noun (KN). Otherwise, an error message will be displayed.

5. Matching with rules

$$RSF = \{514\}$$
$$SS = \{SA \in RSF\},\ \text{where } {}^{n}(RSF) = 514 \qquad (5)$$

There are about 514 rules (CFG) collected. In performing the syntax checking (SS), the sentence condition (SA) must be within the rules stated in the database. After the syntax checking, the output in the parse tree visualization will be produced for correct sentence identified in the syntax checking process. Otherwise, the output for the sentence correction is produced as shown in equation (6). In the parse tree visualization, user can choose each of the parse tree node to visualize a set of word attribute components and do the parse tree visualization for sentence examples.

6. Sentence correction

$$RSF = \{514\}$$
$$A \in (KK_5 \cap RSF) \qquad (6)$$

Each word with its associated word class (KK) will be matched with RSF or CFG as listed in the database. Any relatively close RSF with the order of KK will be retrieved and the words associated with the KK will be changed.

OUTPUT FROM PROTOTYPE DEVELOPMENT

To test the developed prototype, a total of 50 sentences types were selected randomly from the Form 1 Malay textbook for the purpose of collecting the related rules. The rules were obtained through the paper-based parse tree sketch of each sentence.

A. Test invalid sentence

The prototype was tested by using 15 invalid sentences. The words in the collected sentences were repositioned in order to create invalid sentences. The results indicated that the prototype can produce a good output with correct sentence correction for all the 15 sentences. This study does not focus on the semantic issues. A sentence with invalid use of semantic element is considered a correct sentence if it follows the rules provided. The results are shown in Table-1.

Table-1. Suggested sentences produced to the users for invalid sentences entered.

No.	Input sentence	Sentence correction	Rules for sentence correction	Correct (✓), Incorrect (✗)	Target sentence	Rules for target sentence
1.	di sana baginda	baginda di sana	KN KSN KN	✓	baginda di sana	KN KSN KN
2.	di baginda sana	sana di baginda	KN KSN KN	✓	baginda di sana	KN KSN KN
3.	baginda sana di	sana di baginda	KN KSN KN	✓	baginda di sana	KN KSN KN
4.	ditangkap penjahat	penjahat ditangkap	KN KK	✓	penjahat ditangkap	KN KK
5.	membantu kami sedia	kami sedia membantu	KN KK KK	✓	kami sedia membantu	KN KK KK
6.	membantu sedia kami	kami sedia membantu	KN KK KK	✓	kami sedia membantu	KN KK KK
7.	saya ayah dengan	ayah dengan saya	KN KSN KN	✓	saya dengan ayah	KN KSN KN
8.	dengan saya ayah	ayah dengan saya	KN KSN KN	✓	saya dengan ayah	KN KSN KN
9.	itu makmal saiz	saiz makmal itu	KN KN PENT	✓	saiz makmal itu	KN KN PENT
10.	di sini baru saya	saya di sini baru	KN KSN KN KA	✓	saya baru di sini	KN KA KSN KN
11.	sini saya di	saya di sini	KN KSN KN	✓	saya di sini	KN KSN KN
12.	saya sini di	saya di sini	KN KSN KN	✓	saya di sini	KN KSN KN
13.	di saya sini	sini di saya	KN KSN KN	✓	saya di sini	KN KSN KN
14.	berpengalaman Mahzan	mahzan berpengalaman	KN KK	✓	mahzan berpengalaman	KN KK
15.	berpengalaman Datuk Mahzan	datuk mahzan berpengalaman	GEL KN KK	✓	datuk mahzan berpengalaman	GEL KN KK

The abbreviations used in Table-1 are shown in Table-2.

Table 2. Abbreviations and their meaning.

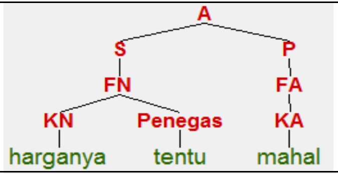
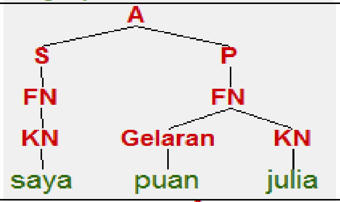
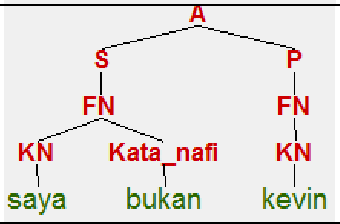
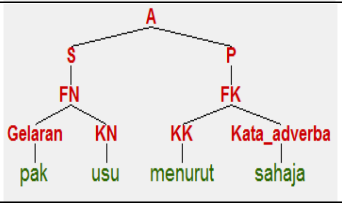
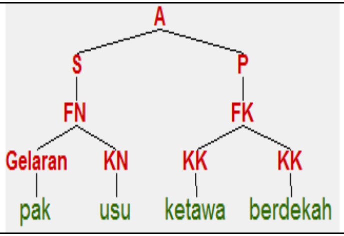
Abbreviation	Meaning
KN	Noun
KSN	Prepositional
KK	Verb
KA	Adjective
PENT	Determinant

When comparing the number 2, 3, 7, 8, 10, and 13 sentences to the sentence examples, the output indicate that the sentences differed in terms of the words positions. However, in terms of the sequence of the word classes, the respective suggested sentences did meet the rules, thus they are considered correct. Nine of these sentences proposed sentence correction according to the target sentence, while the other six produced sentence corrections that are different from the target sentence. The weakness of these sentences is that it can confuse users if the word substitution produces incorrect sentence structure especially in terms of the meaning because semantic aspect is not considered.

B. Parse tree visualization

20 testing sentences were collected randomly from 50 collected sentences. The purpose of testing these sentences is to ensure that the prototype is able to analyse a sentence by producing a parse tree if the sentence entered is classified valid. The testing showed that the prototype can produce a good output, and all the sentences entered were successfully analyzed. As an example, Table-3 gives 5 out of 20 parse tree visualizations done by the prototype.

Table-3. Parse tree visualization.

No.	Test sentence	Parse tree visualization
1	Harganya tentu mahal	 A S P FN FA KN Penegas KA harganya tentu mahal
2	Saya puan Julia	 A S P FN FN KN Gelaran KN saya puan julia
3	Saya bukan Kevin	 A S P FN FN KN Kata_nafi KN saya bukan kevin
4	Pak Usu menurut sahaja	 A S P FN FK Gelaran KN KK Kata_adverba pak usu menurut sahaja
5	Pak Usu ketawa berdekah	 A S P FN FK Gelaran KN KK KK pak usu ketawa berdekah

The abbreviations used in Table-3 are shown in Table-4.

Table-4. Abbreviations and their meaning.

Abbreviation	Meaning
A	Sentence
S	Subject
P	Predicate
FN	Noun phrase
FA	Adjective phrase
FK	Verb phrase
KN	Noun
KK	Verb
KA	Adjective

CONCLUSIONS

Researches in the language field have grown in popularity among researchers from various countries. Therefore, the algorithm used in the development of BMTutor can be utilized to undertake studies related to language processing for any language. It will be useful as a sub-tool needed in semantic processing, machine translation and others. Even though research in BM processing is still at a moderate level relative to other languages as stated by Mohd Juzaiddin (2007) and Zuraidah (2010), there is a possibility of BM becoming one of the languages to be focused by future researchers.

Therefore, it can be concluded that this study has highlighted the algorithm and components involved in developing the BMTutor. BMTutor is intended to facilitate the learning process of sentence construction and grammatical structure in BM. It is also to enhance the learning process in BM that can be used by communities, especially students.

REFERENCES

Zaharani A., & Nor Hashimah, J., "Incorporating structural diversity in the Malay grammar." GEMA Online™ Journal of Language Studies, 12(1), Special Section, 17-34. 2012.

Nor Hashimah, J., Junaini, K., & Zaharani, A., "Sosiokognitif pelajar remaja terhadap Bahasa Melayu." GEMA Online™ Journal of Language Studies, 10(3), 67-87. 2010.

Bagavathy, A. C. (2005), "Mengatasi Kelemahan Murid Menguasai Aspek Tatabahasa Dalam Bahasa Melayu

Melalui Cara Permainan Bahasa," Prosiding seminar penyelidikan pendidikan IPBA, 2005, 50-58.

Daing Melebek, R., "Perubahan struktur kata tunggal Bahasa Melayu mengikut aliran," PhD. Thesis, Universiti Putra Malaysia, 2004.

Nawi, I. (2003). Budaya bangau oh bangau dalam Bahasa Melayu [Online]. Retrieved January 18, 2010, Available: http://www.oocities.com/pendidikmy/berita/berita42003.html

Abdullah, H. (2008). Tatabahasa Pedagogi untuk sekolah menengah [Online Google books]. Retrieved January 18, 2010,available: http://books.google.com.my/books?id=thNaJje24TsC&pg=PR12&dq=bahasa+melayu&hl=en&ei=SB41TeacOcrwrQfr56nZCA&sa=X&oi=book_result&ct=result&resnum=5&ved=0CDkQ6AEwBDg8#v=onepage&q=bahasa%20melayu&f=false

Mohd Juzaiddin A. A. et al. "Pola Grammar Technique to Identify Subject and Predicate in Malaysian Language," in Proc. The Second International Joint Conference on Natural Language Processing, 11-13 October 2005, pp. 185-190.

Mohd Juzaiddin A. A. (2007). Pengkomputeran Linguistik Bahasa Malaysia [Online]. Retrieved Dec 28, 2010, Available: http://www.ftsm.ukm.my/programming/prosiding-atur07/08-Juzaiddin.pdf

Nik Safiah, K., Farid, M. O., Hashim, M., & Abdul Hamid, M. (2009)., Tatabahasa Dewan Edisi Ketiga, Dewan Bahasa dan Pustaka: Kuala Lumpur.

Zuraidah M. D., "Processing natural Malay texts: A data-driven approach." TRAMES: A Journal of the Humanities & Social Sciences, Vol. 14(1), p.90. 2010.