Malay Semantic Text Processing Engine

Benjamin Chu Min Xian
MIMOS Berhad
Kuala Lumpur, Malaysia
mx.chu@mimos.my

Liu Qiang
MIMOS Berhad
Kuala Lumpur, Malaysia
qiang.liu@mimos.my

Rohana Mahmud
University of Malaya
Kuala Lumpur, Malaysia
rohanamahmud@um.edu.my

Arun Anand Sadanandan
MIMOS Berhad
Kuala Lumpur, Malaysia
arun.anand@mimos.my

Kow Weng Onn
MIMOS Berhad
Kuala Lumpur, Malaysia
kwonn@mimos.my

Dickson Lukose
MIMOS Berhad
Kuala Lumpur, Malaysia
dickson.lukose@mimos.my

***Abstract*—Semantic Text Understanding is a process that transforms text into conceptual representation. In this paper, we propose a Text Understanding System for Malay Language. The system comprises of two components: Morphology Analyzer and Semantic Text Interpreter. Some initial evaluation experiments were conducted on these components to gain explanatory insights into its performance. All the current text processing systems we reviewed are focused on preliminary algorithms and rules associated to lexical, morphological and syntax analysis. In our paper, we developed an integrated approach for a text understanding system that has the ability to represent the semantics of the text.**

Keywords-Natural Language Processing; Semantic Text Understanding; Morphology Analysis; Semantic Text Interpretation.

I. INTRODUCTION

The development of fast algorithms to understand and exploit the content of a document, and extracting useful information is very critical. In recent years, development in the area of semantic analysis of natural language text has triggered many applications in Text Mining, Summarization, Text Understanding, Information Retrieval and Extraction. Extracting actionable insight from large highly dimensional data sets, and its use for more effective decision-making, has become a pervasive problem across many fields in research and industry.

Extracting meaningful information from natural language text is the essential challenge that needs to be addressed. In developing these systems for main languages (e.g., English), the researchers have addressed several computational linguistic challenges including lexical, morphological, syntax and semantic processing. There are several fundamental challenges to semantic processing. Essentially, an extensive knowledge base is needed to process the text. Moreover, the complexity of defining rules for different languages when designing algorithms need to be addressed [1].

In this paper, our research focus on a Malay Language Text Understanding (MLTU) for standard Malaysian formal language, known as *Bahasa Malaysia (BM)* or the *Malay language*. Although, a wide demand and usage for the Malay language with a population of more than 28 million speakers, text processing systems geared for this language is still lagging behind.

This paper is structured as follows: Section 2 describes the related work on existing text understanding systems for Malay language; Section 3 describes our Semantic Text Processor system; Section 4 evaluates the performance of the system. Finally, Section 5 concludes this paper with a discussion on the overall outcome achieved and future research directions.

II. RELATED WORK

Several existing techniques in the current state-of-the art for text understanding generally aimed at constructing the syntax and semantic structures from texts. The main challenges for opened and natural language text understanding are caused by the ambiguity of natural language. As Malay native speakers, we will easily be able to understand the semantics of the following example sentence.

"Ali melihat Aminah dengan sebuah teleskop dan dia memanggilnya kuat-kuat" [Malay]

"Ali saw Aminah with a telescope and he is calling her loudly" [English translation]

However, the sentence itself for a machine to comprehend the meaning is quite difficult, as it lacks both the background knowledge and issues with the ambiguity of complex linguistic structures. Extracting meaningful information from natural language text is the essential challenge that should be addressed. In the existing systems, several Computational Linguistic challenges have been addressed focusing more on lexical, morphological and syntax analysis while lesser emphasis on semantic processing.

Many previous researchers in Natural Language Processing (NLP) had attempted to develop a Malay Morphology Analyser and Syntax parsers of speech tagger and parsers [2][3][4][5][6][7][8]. However, most works claimed the difficulties in resolving the stemming issues [9][10][11][12].

For example, the affixation method will derive various words that changed their syntactic class category from the original word (i.e., compared to English, which is forming a new word using inflection method; but, usually, the syntactic class category remains the same). For instance, the word makan (verb - purposely) becomes makanan (noun), when adding the suffix 'an'; becomes pemakanan (adjective), when adding circumfirxes 'pe…an', and becomes termakan (verb - unintentionally), when adding prefix 'ter'. Another major method of forming Malay language that is hardly found in other Languages is reduplication method, which can be full-duplication, such as the word kuat-kuat, or the partial duplication, such as lelaki (i.e., laki-laki).

All these characteristics and word formation issues create many problems for morphology analysis in Malay. Although the issues of labeling the morpheme and the dynamic nature of the syntactic category have been highlighted in MALEX [2][3] and MALIM [4], under-stemming and over-stemming problems remain unresolved [9][10][11][12].

All the systems we reviewed above are focused on preliminary algorithms and rules associated to syntax and morphology analysis. None are focused on developing an integrated approach for Malay Semantic Text Understanding. The ability to represent the semantics of the text is the most essential aspect of this approach. In the following section, we will describe the components of our Malay Semantic Text Understanding System.

III. SYSTEM DESCRIPTION

A. *Morphology Analyzer*

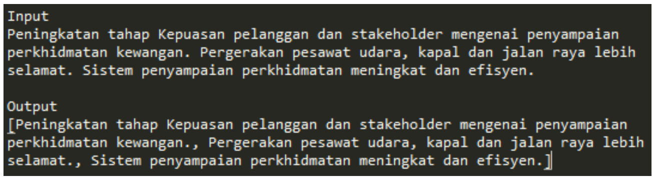
In the English morphology analyzer, stemming and lemmatization are the important task to allow the system to identify the root words. In Table I, the English verb for the different tenses may appear in different forms of spelling. For example, the verb 'walk', it will be appended with an affix's' in simple present tense, it spells as 'walks'; in present progressive tense it is appended with an affix 'ing' is appended, it spells as 'walking'; in simple past tense an affix 'ed' is appended and it spells as 'walked'. The verb 'eat' will change its spelling in various forms in different tenses: in simple past tense it spells as 'ate'; in present perfect tense it spells as 'eaten'. The English verbs will be changed in form spelling according to the tense. In the Malay language perspective, there will not be any spelling changes in the word for each grammar tense in Malay language; in the most of situation, an additional word will be added in front of the word to fulfill the grammar tense issue. As we observed, it is possible to perform Malay language analysis without stemming and lemmatization. As mentioned above, we will only be focusing on the Part-Of-Speech (POS) in Malay morphology analyzer in our initial system.

TABLE I. STEMMING AND LEMMATIZATION

English	Malay
walk	**berjalan**
walks = walk + s	**berjalan**
walked = walk + ed	**telan berjalan**
walking = walk + ing	**sedang berjalan**
eat	**makan**
ate	**sudah makan**
eaten	**telah makan**
beautiful	**cantik**
beautifully = beautiful + ly	**dengan cantik**

In the Malay POS module, we use Apache OpenNLP library [13] to perform Malay POS tagging task. The OpenNLP POS tagging module is language dependent and only performs well if the model language matches the language of the input text. Currently, it supports mainly for European languages. The Apache OpenNLP library is a machine learning based toolkit. We need to prepare for the Malay POS annotated corpus to train the OpenNLP POS tagger module for Malay language. In this experiment, we have collected about 2000 Malay sentences. We use of the Malay WordNet [14] to annotate the POS with each token of the sentences and validated by the Malay native speakers. After the corpus is annotated, 80% of the corpus is used for training and 20% of the corpus is used for evaluation. We are able to get very high accurate from the evaluation for Malay POS tagging with the new trained Malay POS module with the known words. Dataset preparation and evaluation results will be elaborated further in details in the following section of the experiment and evaluation in. There are three OpenNLP modules used to perform POS tagging: Sentence Detector, Tokenizer and Part-Of-Speech Tagger.

The OpenNLP Sentence Detector is able to detect punctuation characters to determine the end of a sentence. Malay and English language share the same alphanumeric and punctuation characters. Therefore, it is possible to directly use the existing English sentence module for the Malay language sentence detection task. The sentence detector can be easily integrated into our application through OpenNLP API. As shown in Fig. 1, the input of the sentence Detector is a text string and the output is an array of Strings, where each string is one sentence.

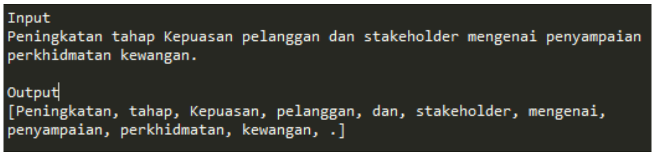


```
Input
Peningkatan tahap Kepuasan pelanggan dan stakeholder mengenai penyampaian
perkhidmatan kewangan. Pergerakan pesawat udara, kapal dan jalan raya lebih
selamat. Sistem penyampaian perkhidmatan meningkat dan efisyen.

Output
[Peningkatan tahap Kepuasan pelanggan dan stakeholder mengenai penyampaian
perkhidmatan kewangan., Pergerakan pesawat udara, kapal dan jalan raya lebih
selamat., Sistem penyampaian perkhidmatan meningkat dan efisyen.]
```

Figure 1. Sentence Detector Input and Output

The OpenNLP Tokenizer segments the input character sequence into tokens. Tokens are usually words, punctuation, and numbers. The tokenizer module expects an input string, which contains the untokenized text. If possible, one sentence will be best input string for the tokenization module. In this experiment, the input array of the sentences is provided from the output of the Sentence Detector. The sample result is shown in Fig. 2. Tokenizer returns an array of strings where each string is one token.

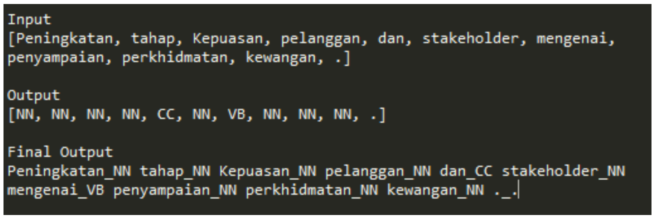


```
Input
Peningkatan tahap Kepuasan pelanggan dan stakeholder mengenai penyampaian
perkhidmatan kewangan.

Output
[Peningkatan, tahap, Kepuasan, pelanggan, dan, stakeholder, mengenai,
penyampaian, perkhidmatan, kewangan, .]
```

Figure 2. Tokenizer Input and Output

The POS Tagger marks the input tokens with their corresponding POS tag based on the token itself and the context of the token. A token can possibly have multiple POS tags, the POS tagger uses maximum entropy probability model to predict the correct POS tag from the tag set. A tag dictionary is used by the POS tagger to limit the possible tags for a token; this will also increase the POS tagger tagging accuracy and performance. As shown in Fig. 3, the expected input of the POS tagger is a tokenized sentence in the form of string array where each of the strings is a token. The output is a tag array; it contains one POS tag for each token for the input array. The corresponding tag can be found at the same index of the tag array. The final output of the POS tagger will be a sentence where token and tag pairs are concatenated with an underscore, "_".



```
Input
[Peningkatan, tahap, Kepuasan, pelanggan, dan, stakeholder, mengenai,
penyampaian, perkhidmatan, kewangan, .]

Output
[NN, NN, NN, NN, CC, NN, VB, NN, NN, NN, .]

Final Output
Peningkatan_NN tahap_NN Kepuasan_NN pelanggan_NN dan_CC stakeholder_NN
mengenai_VB penyampaian_NN perkhidmatan_NN kewangan_NN ._.
```

Figure 3. POS Tagger Input and Output

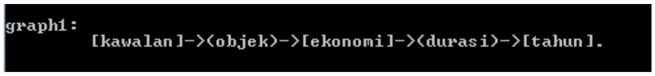
B. *Semantic Interpreter*

For this module, we have extracted the grammatical rules from [19] and we have defined all of these programmatically for each of the thematic roles listed in Table II. Semantic Interpreter will use the rules defined to generate the semantic representation of the sentence. In this case, the semantic representation is in the form of Conceptual Graphs (CG).

For example, we can have a sentence as the input to this module, "*Kawalan ekonomi sepanjang tahun*" which means "*Economy restraint throughout the year*". From the previous module, this sentence will be annotated to produce the conceptual graph, which is shown in Fig. 4, as follows:

Annotated sentence:
Kawalan_NN ekonomi_NN sepanjang_IN tahun_NN

CG:



```
graph1:
        [kawalan]->(objek)->[ekonomi]->(durasi)->[tahun].
```

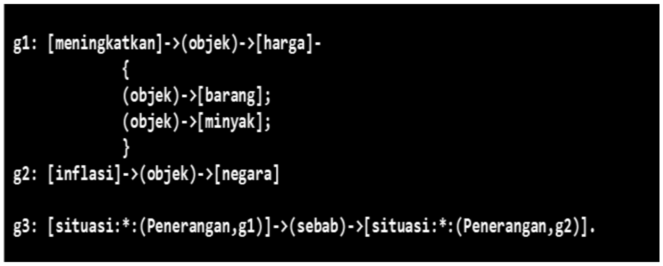
Figure 4. Simple Conceptual Graph in Malay

As shown in Fig. 4, this is a simple graph representing the meaning of the text. Moreover, we have defined rules to produce nested graphs for several sentence cases as shown below.

Sentence:
Meningkatkan harga barang dan minyak kerana inflasi negara.
English translation:
Increase the price of goods and oil due to the country's inflation

Annotated sentence:
Meningkatkan_VB harga_NN barang_NN dan_CC minyak_NN kerana_CC inflasi_NN negara_NN

CG:



```
g1: [meningkatkan]->(objek)->[harga]-
          {
          (objek)->[barang];
          (objek)->[minyak];
          }
g2: [inflasi]->(objek)->[negara]

g3: [situasi:*:(Penerangan,g1)]->(sebab)->[situasi:*:(Penerangan,g2)].
```

Figure 5. Nested Conceptual Graph in Malay

An example of a nested graph is shown in Fig. 5. In g1, the concept [harga] is the object (objek) of the verb [meningkatkan]. The concept [harga] is linked by the object relation (objek) to both concepts [barang] and [minyak] due to the conjunction in the sentence. Similarly in g2, the concept [inflasi] is linked by the object relation (objek) to the concept [negara]. In g3, a situation described by g2 is caused by a situation expressed in g1. The relation "caused by" between these two situations is using the Malay thematic role "sebab". Table II shows the complete listing of the thematic roles used in Malay.

TABLE II. THEMATIC ROLES FOR MALAY

Malay	English translation
Pelaku	Agent
Alami	Experiencer
Alat	Instrument
Asal	Origin
Bilangan	Amount
Destinasi	Destination
Deritaan	Patient
Durasi	Duration
Gaya	Manner
Hasil	Result
Kepunyaan	Possession
Kesan	Effector
Manfaat	Beneficiary
Muasal	Matter
Objek	Object
Permulaan	Start
Penyertaan	Accompaniment
Perbandingan	Comparand
Sebab	Because
Sifat	Attribute
Tema	Theme
Tempat	Location
Tujuan	Purpose
Ukuran	Measurement
Waktu	PointinTime
Perhinggaan	Completion
Penafian	Negation
Jalan	Path

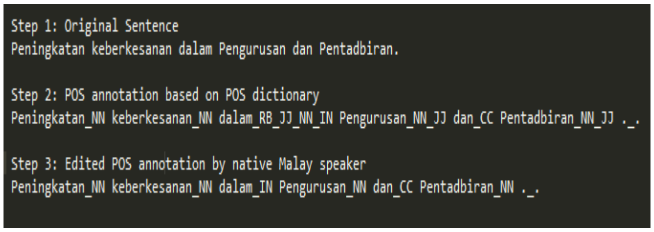
IV. EXPERIMENTAL EVALUATION

A. Datasets

In the current state of the art, there is no Malay language POS annotated corpus that is available to train the POS module for Malay language. Many previous attempts have been done to prepare the data manually [3][8][15]. With the unavailability of any Malay POS annotated corpus, data preparation is an important task in this initial research work.

As the first step, the POS data was extracted by utilizing both Malay WordNet [14] and Apertium [16] Malay to Indonesian translation dictionary. The Malay WordNet is a lexical dictionary (currently supports Malaysian and Indonesian). The dictionary comprises of 19,210 synsets, 48,110 senses and 19,460 unique words with POS tag in the Malay WordNet, where all the relations (hypernyms, meronyms etc.) are extracted from WordNet. This project was initiated by Francis Bond from Nanyang Technological University [17]. The project is inspired by Princeton WordNet since there is no lexical dictionary for Malay language. Apertium is a machine translation engine designed to translate closely related languages. The current Apertium engine supports language translation from Indonesian to Malay. In doing so, the engine uses POS information and translation rules for Malay and Indonesian words. We extracted this POS data Apertium, along with Malay WordNet, to build our POS annotation corpus.

In this research work, we collected about 2000 Malay sentences as our dataset. We also created a module to extract and combine the Malay POS data for the Malay WordNet and Apertium. Once Malay POS data dictionary is ready, we created another module to parse and annotate all possible POS for the Malay sentences base on the POS dictionary, as the result some of word may have annotated with multiple POS tag. The final step, native Malay speaker will need to validate and correct the tags for all the Malay sentences. Fig. 6 shows the annotation result for each the steps involved.



```
Step 1: Original Sentence
Peningkatan keberkesanan dalam Pengurusan dan Pentadbiran.

Step 2: POS annotation based on POS dictionary
Peningkatan_NN keberkesanan_NN dalam_RB_JJ_NN_IN Pengurusan_NN_JJ dan_CC Pentadbiran_NN_JJ ._.

Step 3: Edited POS annotation by native Malay speaker
Peningkatan_NN keberkesanan_NN dalam_IN Pengurusan_NN dan_CC Pentadbiran_NN ._.
```

Figure 6. Malay Part-Of-Speech Annotation Sample

During implementation and experiment, 80% of annotated sentences were used for POS module training data; the rest of the 20% were used as evaluation data.

B. Evaluation Results

Based on the methods described above, evaluation has been conducted to determine the accuracy of the two main modules; Morphology Analyzer module and the Semantic Interpreter module.

- *Morphology Analyzer*

The overall accuracy of the POS tagging was calculated as the ratio of correct POS tags found by the system over the total number of POS tags. The accuracy scores along with the corpus size are plotted in Fig. 7. Between Phase 1 and Phase 2, the inconsistencies in the POS annotations were fixed. For example, the Malay word "dan" was annotated as preposition "IN", in some sentences and as conjunction "CC" in other cases. In Phase 3, along with increasing the number of annotated sentences, a Tag dictionary is a word dictionary, which contains specified POS tags for the tokens. This ensured that inappropriate tags were assigned to tokens, which will result in better accuracy. Naturally, increasing the number of annotated sentences resulted in better accuracy, until a plateau was reached, at 2000 annotated sentences.

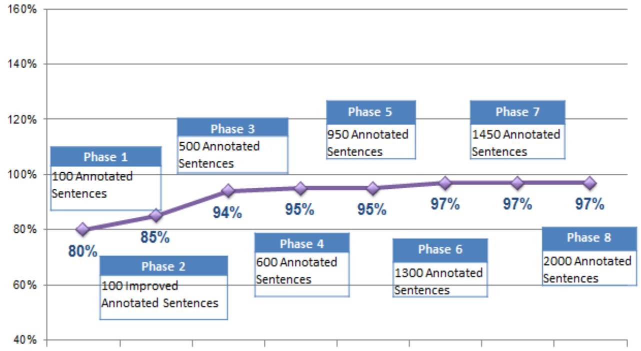


Figure 7. Morphology Analyzer Experimental Results

- *Semantic Interpreter*

In evaluating this module, we have created 70 graphs manually as the gold standard for our benchmark. The results produced by the system were classified as Correct (indicating full match), Partial (indicating a partial match) or Incorrect (indicating incorrect representation). As shown in Fig. 8, the results show that 62 graphs were classified as correct, 7 as incorrect and 1 partial match.

V. DISCUSSION

Upon analyzing the results, it was found that the partial match was due to a missing concept in the knowledge base. Fig. 8 shows the knowledge base is based on the Malay WordNet with over 30,000 concepts. Although the partial match is only 1%, but extending and enriching this knowledge base with more concepts will further improve the interpretation accuracy. One of the reasons behind the incorrect results was found to be the lack of support for anaphora resolution. For example, this is shown in Fig. 9 where the pronoun ‘mereka’ is not being resolved to the noun ‘penduduk’.

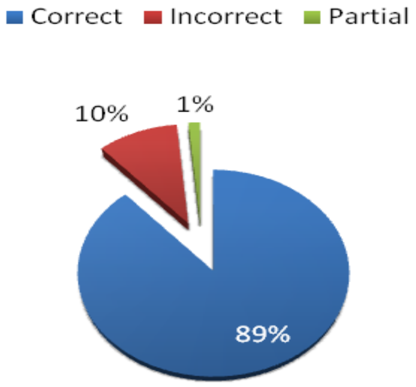
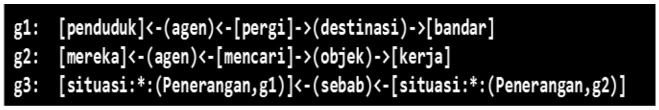


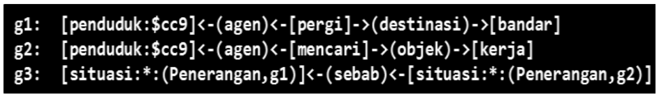
Figure 8. Semantic Interpreter Experimental Results



```
g1:  [penduduk]<-(agen)<-[pergi]->(destinasi)->[bandar]
g2:  [mereka]<-(agen)<-[mencari]->(objek)->[kerja]
g3:  [situasi:*:(Penerangan,g1)]<-(sebab)<-[situasi:*:(Penerangan,g2)]
```

Figure 9. Incorrect representation without anaphora resolution

Anaphora resolution [18][19] is the problem of identifying how contextual entities are referred within a single or several sentences, typically what a pronoun or a noun phrase is referring to. For example, from the sentences *john loves mary* and *he wishes to marry her*, the entity *john* is referred by *he* and *mary* is referred by *her*. Another example in Malay language can be seen in the following sentence. “Penduduk pergi ke bandar kerana mereka mencari kerja” is translated as "the villagers went to the city because they wanted to find a job”. Here, the word ‘mereka’ (they) is referring to a pronoun; therefore it should be resolved to “penduduk” (villagers). As shown in Fig. 10, the correct representation of the graph:



```
g1:  [penduduk:$cc9]<-(agen)<-[pergi]->(destinasi)->[bandar]
g2:  [penduduk:$cc9]<-(agen)<-[mencari]->(objek)->[kerja]
g3:  [situasi:*:(Penerangan,g1)]<-(sebab)<-[situasi:*:(Penerangan,g2)]
```

Figure 10. Correct representation with anaphora resolution

where the reference indicator *$cc9* would denote the coreference.

VI. CONCLUSION AND FUTURE WORK

State-of-the-art-text processing systems for Malay Language are still dealing with problems related to lexical, morphological and syntax analysis. Based on syntax analysis alone, meaning through syntax is still insufficient to explain the comprehension of natural language texts. Therefore, we proposed an integrated approach for Malay Text Understanding, which included both syntax and semantic processing.

In summary, we have developed Morphology Analyzer and Semantic Interpreter components. From a qualitative comparison perspective, we have evaluated both components on how well they can perform (this is quite subjective, and is based on our initial benchmarking exercise).

In future, we plan to enrich our Malay Linguistic knowledge base derived from Malay WordNet with other linguistic resources. We will continue to evaluate both of our components with a large news dataset to improve our semantic rules. Furthermore, we will also explore Coreference Resolution for Malay Language. Coreference Resolution will help to refine the semantic representation produced by resolving all anaphors and cataphors to their intended referents.

REFERENCES

[1] Benjamin Chu, Fadzly Zahari, and Dickson Lukose, Large-Scale Semantic Text Understanding. In Semantic Technology and Knowledge Engineering (STAKE) 2010 Conference Proceedings, Sep. 2010, pp. 28-39.

[2] Knowles, Gerald and Zuraidah Mohd Don, Tagging a corpus of Malay texts and coping with 'syntactic drift'. Proceedings of the corpus linguistics, 2003, pp. 422-428.

[3] Zuraidah Mohd Don, Processing Natural Malay Texts: A Data Driven Approach, TRAMES, Vol 14 (64/59), 2010, pp. 90-103.

[4] Mohd Yunus Sharum, Muhammad Taufik Abdullah, Mohd Nasir Sulaiman, Masrah Azrifah Azmi Murad, and Zaitul Azma Zainon Hamzah, MALIM- A new computational Approach of Malay Morphology, IEEE, Vol 2, 2010, pp. 837-843.

[5] Timothy Baldwin, and Suad Awab, Open Source Corpus Analysis Tool for Malay, Retrieved Nov. 2013, from: https://code.google.com/p/malay-toklem/

[6] Mat Awal, Norsimah Abu Bakar, Kesumawati and Abdul Hamid, Nor Zakiah Jalaluddin, and Nor Hashimah, Morphological Differences between Bahasa Melayu and English: Constraints in Students' Understanding, The Second Biennial International Conference on Teaching and Learning of English in Asia, 2007, pp.1-11.

[7] H. Mohamed, N. Omar Abd, and M. J. Aziz, Statistical Malay Part Of Speech Tagger using Hidden Markov Approach, International Conference on Semantic Technology and Information Retrieval, Putrajaya, June. 2011, pp. 231-236.

[8] Rayner Alfred, Adam Mujat, and Joe Henry Obit, A Rule-based Part Of Speech (RPOS) Tagger for Malay Text Articles; A. Selamat (Eds), ACIIDS 2013, Part 11, LNAI 7803, 2013, pp. 50-59.

[9] Salhana Amad Darwis, Rukaini Abdullah, and Norisma Idris, Exhaustive Affix Stripping and a Malay Word Register to solve stemming errors and ambiguity problem in Malay Stemmers, Malaysian Journal of Computer Sciences, Vol 25, 2012, pp. 213-218.

[10] Bali Ranaivo-Malancon, Computational Analysis of Affixed Words in Malay Language, Unit Terjemahan Melalui Komputer, USM, Technical Report, 2004.

[11] S. A. Fadzli, and A. K. Norsalehen, I. A. Syarilla, H. Hasni, and M. S. S. Dhalila, Simple Rules for Malay Stemmer, The International Conference on Informatics and Applications, Jun. 2012, pp. 28-35.

[12] Y. L. Tan, A Minimally-Supervised Malay Affix Learner, Proceedings of the Class 2003 Senior Conference, Swarthmore, 2003, pp. 55-62.

[13] OpenNLP, Retrieved Nov. 2013, from: http://opennlp.apache.org

[14] Malay Wordnet, Retrieved Dec. 2013, from:

http://wn-sa.sourceforge.net/index.eng.html

[15] Norshuhani Zamin, Alan Oxley, and Zainab Abu Bakar, A Lazy Man's Way to Part-Of-Speech Tagging, Knowledge Management and Acquisition for intelligent system lecture notes in cimputer science, vol 7457, 2012, pp. 106-117.

[16] Apertium a free/open-source machine translation platform, Retrieved Nov. 2013, from: http://www.apertium.org/

[17] Noor Nurril Hirfana, Bte Mohamed, Saquan Suerya, and Bond Francis, Creating the Open Wordnet Bahasa, Proceeding of the 25th Pacific Asia Conference on Language, information and Computation, 2011, pp. 255-264.

[18] Ruslan Mitkov, and Wv. Sb. Wolverhampton, Anaphora Resolution: the State of the Art. Computational Linguistics, 1999, pp. 1-34.

[19] Asmah Haji Omar, Nahu Melayu Mutakhir 5th Edtion. Kuala Lumpur: Dewan Bahasa Pustaka, 2009.

ISBN: 978-1-61208-329-2