Journal of Fundamental and Applied Sciences

ISSN 1112-9867

Available online at http://www.jfas.info

Research Article

Special Issue

# PENALIZING UNKNOWN WORDS' EMISSIONS IN HMM POS TAGGER BASED ON MALAY AFFIX MORPHEMES

H. Mohamed[1,*], N. Omar[2] and M. J. A. Aziz[2]

[1]Cyber Security Centre, UniverisitiPertahananNasional Malaysia, Sungai Besi Camp, 57000 Kuala Lumpur, Malaysia

[2]Knowledge Tech. Group, Centre for AI Technology (CAIT), UniversitiKebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia

Published online: 10 September 2017

## ABSTRACT

The challenge in unsupervised Hidden Markov Model (HMM) training for a POS tagger is that the training depends on an untagged corpus; the only supervised data limiting possible tagging of words is a dictionary. Therefore, training cannot properly map possible tags. The exact morphemes of prefixes, suffixes and circumfixes in the agglutinative Malay language is examined to assign unknown words' probable tags based on linguistically meaningful affixes using a morpheme-based POS guessing algorithm for tagging. The algorithm has been integrated into Viterbi algorithm which uses HMM trained parameters for tagging new sentences. In the experiment, this tagger is first, uses character-based prediction to handle unknown words; next, uses morpheme-based POS guessing algorithm; lastly, combination of the first and second.

**Keywords:** Malay POS tagger; morpheme-based; HMM.

Author Correspondence, e-mail: hassan@upnm.edu.my

doi: http://dx.doi.org/10.4314/jfas.v9i3s.36

## 1. INTRODUCTION

Recently, there is some interest to move further in research of Malay Natural Language Processing. For example, a Malay POS Tagger based-on supervised training reported by [1]. POS tagging is very important as it is a low-level parsing of natural language to build many Natural Language (NLP) applications. Another example is a parallel language corpus as a resource for Malay language reported by [2] and a basic tokenizer tool for Jawi writing reported by [3].

The Malay language is categorised as an agglutinative or derivative language where most of the words are formed by merging affixes with root words [4-5]. Affixation is performed by adding the affix at the beginning (prefixes), middle (infixes) or at both the ends (circumfixes) of the root word. Due to the well-defined affixation rules, the word class of Malay derivative words can be intuitively guessed. This paper examined the effectiveness of using Malay affix morphemes for handling unknown words in the unsupervised Hidden Markov Model (HMM) POS tagging. For under-resourced languages such as Malay using the unsupervised training method helps to avoid the need of a large annotated corpus which is labour intensive, time consuming and high in costs.

Several researchers have conducted the efforts to train unsupervised HMM POS tagger to cater for words that are not listed in the dictionary and ambiguous words as well.Among the common methods used are exploiting words' ending to enlarge the training dictionaries [6-7] or directly estimating the initial emissions for unknown words [8-9] or directly estimating the lexical probabilities for ambiguous words [10].

Other researches focus on building an annotated corpus automatically and training the HMM using the supervised approach [11-12]. This paper emphasises on the morphological characteristics of the Malay origin as opposed to the traditional basic statistical POS tagging, which is linguistically independent and does not explicitly include linguistic features. This study aims to examine the effectiveness of using actual affix morphemes of the Malay language rather than the use of the words' ending characters as features for predicting the POS of unknown words. Therefore, a suitable combination of methods in the Viterbi tagging for Malay is identified. The similar works are in [13-14]. In [13]applies affixation and word

relation rules which is clearly rule-based approach. On the other hand, in [14] uses morphological analyser and applies machine learning technique. The other related work is in [15], which applies statistical unsupervised method using N-gram and Dice Coefficient for similarity measurement purpose. The other proposed methods for Malay POS tagging are based on supervised methods [16-17] and syntactic drift with data-driven approach [18-19].

### 1.1.Malay Tag Set

To define a linguistically motivated Malay tag set, while at the same time determining a suitable number of tags for the statistical approach, the classification of Malay words described in Malay grammar textbooks written by many scholars [4, 20-21] are refined. The descriptions provide guidance in designing a tag set, rather than trying to adopt from English or other languages.The tag-sets built for English do not cover all the characteristics of the Malay language; some English tags are not useful for Malay and there is a lack of tags to cover some Malay word classes, such as numeral classifiers [22].

Accordingly, many monolingual or bilingual Malay dictionaries already associated word classes of entries, which coincided with Malay grammar in textbooks [23-25]. A complete set of Malay tags are listed in Table 1, considering the proposed modifications on the tag sets of the dictionary to suite with HMM training. The modifications are as follows:

### 1.2.Auxiliary Words (*Kata Bantu*)

There are two type of auxiliary words in Malay i.e. modal auxiliary (*Kata Bantu Ragam*) and aspectual auxiliary (*Kata Bantu Aspek*). The modal auxiliary illustratesthe mood of the act on the verbs; for example, *hendak*(want), *mahu*(wish), *harus*(should), *mesti* (must), *boleh* (can) and *dapat* (can). There are no clear verb tenses in Malay as opposed to English. Therefore, aspectual auxiliaries are used to indicate whether the state of the verb; past, still on-going or yet to be done. For example, *telah* (already past), *sudah* (already past), *pernah* (ever), *sedang* (still), *masih* (still), *akan* (will) and *belum* (not yet).

### 1.3.Function Words (*Kata Tugas*)

Function words in Malay are limited but they significantly play different roles in a text. They are used in a sentence or phrase as grammatical functions. Their role includes determiners (*Kata Penentu*), imperative words (*Kata Perintah*), discourse markers (*PenandaWacana*),

affirmative words (*Kata Pembenar*), directional words (*Kata Arah*), assertion words (*Kata Penekan*) and nominalisers (*Kata Pembenda*). Therefore, function words (*Kata Tugas*) are detailed-up as per roles.

### 1.4.Existential or *Kewujudan* Tag

A new tag is created to differentiate between verbs in dominantly Subject-Verb-Object Malay sentence patterns with another special case verb, ada (exist), in the sentence pattern of Verb-Subject which is very rare. For example, the sentence '*ada lima puluhekorkambing'* (there are fifty goats) complies with the Verb-Subject pattern as compared to '*diaada lima puluhekorkambing*' (he/she has fifty goats), which complies with the Subject-Verb-Object pattern.

### 1.5.Relative Pronouns or *GantiNamaRelatif* for '*yang*' (which, that)

Many Malay grammar textbooks classify the word yang as a relative subordinating conjunction (kata hubungpancanganrelatif), which is the subclass of kata hubung (conjunction). Our corpus indicates that the yang is the most frequently used word. It is good to classify such words into a single class to avoid a skewed emission probability of HMM being affected due to high usage of certain words in a class.

**Table 1.** List of Malay tag set

| Tags | Description | Examples (The Translations Provided Are In Lateral) |
|---|---|---|
| ADA | Existential | *ada*(exist) |
| GEL | Title | *Datuk, Haji* |
| GNR | Relative pronoun | *yang* (which, that) |
| JDH | Numeral classifier or *penjodohbilangan* | *orang* (people),*buah*(fruit) |
| KA | Adjective | *pandai*(clever),*bodoh*(stupid) |
| KAD | Adverb | *sekarang*(now),*tadi*(just now) |
| KAR | Directional Word | *bawah*(under),*tepi*(side) |
| KBIL | Numeral | *satu*(one), *100* |
| KBR | Modal Auxiliary | *mahu*(wish),*harus*(should) |

| | | |
|---|---|---|
| KEP | Abbreviation | *UKM* |
| KGN | Pronoun | *kamu*(you),*awak*(you) |
| KH | Conjunction | *dan*(and),*lalu*(then) |
| KK | Verb | *pulang*(return),*tidur*(sleep) |
| KN | Common Noun | *rumah*(house),*kambing*(goat) |
| KNF | Negative Word | *bukan*(not) *and tidak*(no/not) |
| KNK | Proper Noun | New York, Pasir Mas |
| KP | Intensifier | *Sungguh*(true/exact) |
| KPB | Nominalizer | *lajunya*(its speed), *sakitnya*(painfulness) |
| KPM | Narrator | *ialah*(is),*adalah*(is) |
| KPN | Emphatic Words | *juga*(also),*jua, pun* |
| KPR | Affirmative Word | *ya*(yes),*benar*(true) |
| KPT | Assertion Word | *nampaknya*(it seems),*bahawasanya* |
| KS | Preposition | *dari*(from),*pada*(at) |
| KSR | Interjection | *amboi*(wow),*bedebah*(ah) |
| KTP | Imperative Word | *sila*(please),*jemput*(invite) |
| KTY | Interrogative Word | *berapa*(how),*bila*(when) |
| PIN | Foreign Word | *university* |
| PW | Discourse Marker | *kalakian*(urging),*maka*(then) |
| TEN | Determiner | *ini*(this) *and itu*(that) |
| EMAIL | Email Address/ Web Site | hassan.dbangi@yahoo.com |
| $ | Dollar Sign | $ RM |
| # | Pound Sign | # £ |
| “ | Left Quote | ‘ “ |
| ( | Left Parenthesis | ( [ { < |
| ) | Right Parenthesis | ) ] } > |
| , | Comma | , |
| . | Sentence-Final Punctuation | ! ? . |
| : | Mid-Sentence Punctuation | - ... ; : |

| | | |
|---|---|---|
| SYM | Any Symbols | ` ^ _ @ * / \ & % + = \| ~ |

## 2. RESULTS AND DISCUSSION

The accuracy of the tagging denotes the percentage of the words correctly assigned with tags as compared to the tagged corpus [33]. Therefore, the tagging performance is often measured by the overall tagging, known word and unknown word tagging accuracies [28, 34]. Known words refer to words present in the training corpus and vice-versa. However, in our case, the definition of unknown words is extended to include the words that may exist in the training corpus but not listed in the dictionary. Therefore, the accuracy in our evaluation is termed into five types of accuracies to ease the analysis of tagging.

- Overall-the overall performance of the tagger.
- Seen word with unique tag-the performance of tagging words presents in the training that exist in the dictionary with only one tag.
- Seen words with ambiguous tags-the performance of tagging words presents in the training that exist in the dictionary with more than one tag.
- Seen words not existing in the dictionary-the performance of tagging words not listed in the dictionary but seen in the training.
- Unseen words-the performance of tagging words absent in the training corpus.

Each accuracy is calculated as the ratio of number of correctly tagged words of the related accuracy in the test corpus to the total number of tagged words of the related accuracy in the test corpus. Table 2 presents the results of the experiments.

**Table 2.** Tagging performance

| **Methods** | **Overall** | **Seen Words** | | | **Unseen Words** |
|---|---|---|---|---|---|
| | | **Exist in Dictionary** | | **Not Exist in Dictionary** | |
| | | **Unique Tag** | **Ambiguous Tags** | | |
| *1* | 38.50 | 42.30 | 7.08 | 40.31 | 30.10 |
| *2* | 81.81 | 92.00 | 75.78 | 38.97 | 33.42 |
| *3* | 81.71 | 92.00 | 75.83 | 38.33 | 32.22 |

| | | | | | |
|---|---|---|---|---|---|
| *4* | 82.25 | 92.00 | 75.52 | 42.90 | 31.22 |
| *5* | 82.28 | 92.00 | 76.04 | 42.52 | 31.94 |
| *6* | 82.53 | 92.00 | 76.19 | 43.93 | 33.72 |
| *7* | 82.59 | 92.00 | 76.14 | 44.56 | 33.40 |
| *8* | 82.58 | 92.00 | 76.13 | 44.32 | 34.20 |

**Legend of the Methods**

1. *Baseline*
2. *Viterbi (training iteration = 2) with words' starting (max. length = 4)*
3. *Viterbi (training iteration = 3) with words' ending (max. length = 8)*
4. *Viterbi (training iteration = 2) with morpheme (uniform distribution)*
5. *Viterbi (training iteration = 2) with morpheme (proportionate distribution)*
6. *Viterbi (training iteration = 2) with combination of morphemes and words' starting (max. length = 8)*
7. *Viterbi (training iteration = 2) with combination of morphemes and words' ending (max. length = 5)*
8. *Viterbi (training iteration = 2) with combination of morphemes and words' starting (max. length = 1) with successive abstraction smoothing ignores words have affixes*
9. *Viterbi (training iteration = 2) with combination of morphemes and words' ending (max. length = 6) with successive abstraction smoothing ignores words have affixes*

**2.1.Viterbi Tagging with a Words' Starting or Words' Ending**

This method of tagging unknown words can be looked as a character-based POS prediction. There are two possibilities that influence this results, i.e. the number of training iterations and the maximum predefined length of characters used in the words' starting or ending.Due to this reason, the experiment must be repeated for each predefined length of characters (ranging from one to twelve characters) for the words' starting and ending methods with various numbers of iterations (ranging from one to ten iterations). The iteration and the length of

characters that gives the highest overall performance is considered as the best performance. The experiments revealed the best performance at the second iteration of HMM training with a maximum predefined length of 4 characters for words' starting method. The overall accuracy drops to 81.81% (second row of Table 6). On the other hand, the overall accuracy drops to 81.71% (third row of Table 6) on the third iteration of HMM training with 8 characters maximum predefined length of words' ending method. Although the percentages' do not present significant difference, there is a difference in terms of the number of tokens with 0.1% of accuracy, reflecting about 121 tokens (out of 121,090 test tokens).

The tagging accuracy for unseen words by using a words' starting information is 39.42% on the fourth iteration of HMM training. On the other hand, using a words' ending information, the accuracy is 33.22% on the second iteration.The different percentage of 7.20%, indicates that using a words' starting information is slightly more accurate than using a words' ending information. Furthermore, using a words' ending information requires more subsequent characters.

Tagging seen words not listed in the dictionary using a words' starting information outperformed the use of a words' ending information. The tagging accuracy for using a words' starting information is 39.02% on the third iteration as compared to using a words' ending information, which is 38.36% on the fourth iteration.The difference of 0.66% reflects about 105 tokens (out of 15,882). This finding strengthens the argument to use words' starting information for character-based prediction of unknown words' POS.

**2.2 Viterbi Tagging with Unknown Words' POS Predicted through Malay Affix Morphemes**

The number of training iterations can influence the results. Therefore, the experiments are repeated for each iteration ranging from one to ten. The best overall performance from those iterations is considered the best result.Table 6 depicts the results of tagging performance using a combination of Viterbi with handling unknown words using morpheme-based POS guessing (stated in row 4 and 5). The best overall tagging accuracy is 82.28% when the unknown words' emission is substituted by a value proportionate to the marginal distribution of tags. The results are slightly better than the results of the experiment done on Viterbi with words'

starting or ending methods. However, tagging unseen words is less accurate in Viterbi tagging with morpheme-based POS guessing (31.94%) compared to the results of the experiment done on Viterbi with words' starting method (33.42%), with a difference of about 1.48% (see row 2 and 5 of Table 6). This indicates that the affixation rules do not enhance the accuracy of POS guessing of unseen words. However, Viterbi tagging with morpheme-based POS guessing enhances the accuracy of tagging words absent in the dictionary by 42.52%; better than the baseline.

**2.3 Combination of Words' Starting or Ending Methods with Affix Morphemes**

Three factors influence the results, i.e. the number of training iterations, the maximum predefined length of characters used in a words' starting or ending methods and the number of joint "word-tag" and word types used in the successive abstraction smoothing. Words that contained affixes are ignored when counting the total number of joint "word-tag" that shared the same sequence of words' starting or ending used in Equation (13) and the total number of word types that shares the same sequence of words' starting or ending used in Equation (14). The experiment is repeated for each maximum predefined length of characters, ranging from one Equation (1) to twelve Equation (12) characters for a words' starting and ending methods with various numbers of iterations, ranging from one Equation (1) to ten Equation (10).For each iteration, there are four sets of probabilities:

- *SET 1*: The probability of a tag $t_i$ given the first $m$ letters as formulated in Equation (9).
- *SET 2*: The probability of a tag $t_i$ given the last $m$ letters (using the reverse order of characters).
- *SET 3*: The probability of a tag $t_i$ given the first $m$ letters, ignoring the affixed words.
- *SET 4*: The probability of a tag $t_i$ given the last $m$ letters, ignoring the affixed words.

The tagger checks whether the unknown word contains an affix morpheme. If present, the words' emission is substituted by a value proportionate to the marginal distribution of tags. If the affix morpheme is absent, the emission is substituted by either the probability in *SET 1* to *SET 4* above. The results are shown in the last four rows in Table 6 (row 6-9). Combined methods of affix morpheme POS guessing tend to produce better results. The emission is substituted by a value proportionate to the marginal distribution of tags and words' ending

information. The probability of a tag $t_i$ is given by the six-predefined length of letters with successive abstraction, smoothing ignored affixed words (using the probability in *SET 4*). This combination method performs overall tagging with 82.72% of accuracy, the highest among all combinations whilst maintaining high accuracy in guessing tags for words that do not exist in the dictionary.Without a combination method, i.e. the Viterbi using a words' starting information (maximum 4 characters of predefined length) is good for tagging unseen words.

**2.4.Analysis on Malaya Affixes**

A words' starting and ending predictions model implicitly includes Malay linguistics, which is affix information. It extends the paradigm of affixations in linguistic meaning. Malay affixes have some significant statistical distribution. The distribution of words containing circumfixes, prefixes or suffixes in the Malay language is almost consistent for different corpus size. Fig. 1, 2 and 3 show the distribution of affixes found in the training corpus (995,240 tokens) and test corpus (121,090 tokens). The test corpus has 17,818 tokens of unknown words not listed in the dictionary, (17.45%) implying that 44.46% of words containing affixes.Therefore, 45.13% of tagging accuracy for words not in the dictionary using the combination methods of affix morpheme for POS guessing and words' ending with smoothing ignored affixed words in Table 6 (row 9) is near to the percentage of words not listed in the dictionary with affixes (44.46%). It is expected that 97.13% words would be correctly tagged using a combination of morpheme-based POS guessing and words' ending with smoothing ignored affixed words by focusing only on derivative words. This percentage indicates that using morpheme-based POS guessing for tagging affixed words is effective.

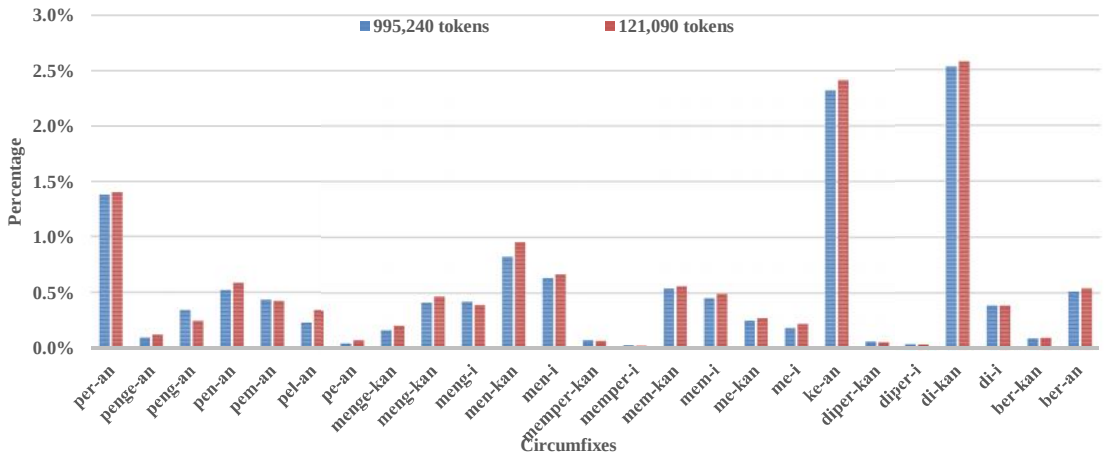

**Fig.1.** The distribution of Malay circumfixes in two different corpuses

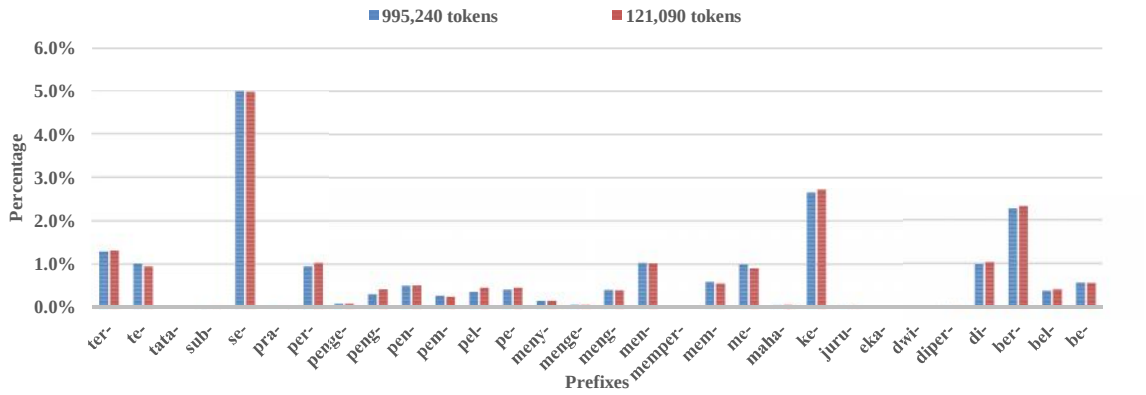


**Fig.2.** The distribution of Malay prefixes in two different corpuses

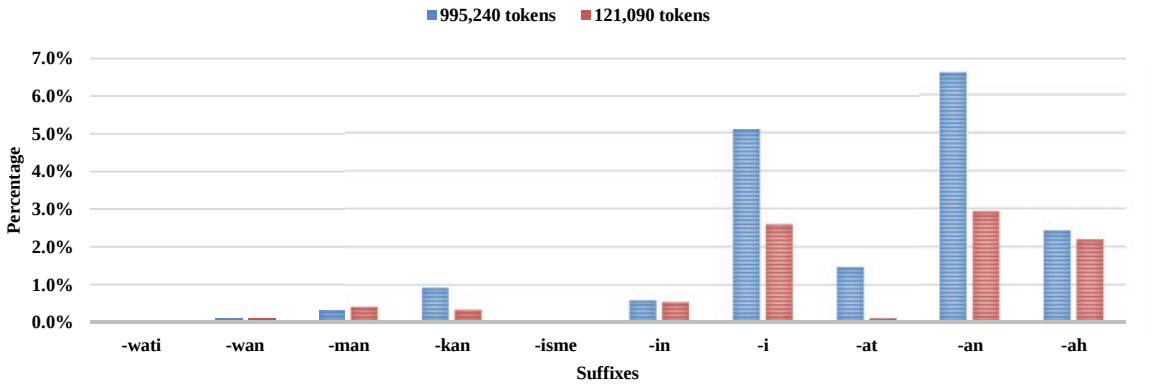


**Fig.3.**The distribution of Malay suffixes in two different corpuses

The perfect guessing by affix morphemes is shown by circumfixes*meng-...-kan* and *memper-...-i* with error percentage of zero. However, if only reliable data is considered, which is data that occurred more than 100 times, the circumfix*di-...-kan* is the best affix morpheme for guessing because it is repeated 1,176 times (appeared in unknown words) in the tests corpus with tagging error as low as 5.36%. The bad guessing is identified as unknown words containing the morpheme *ke-...-an*, with the error percentage of 28.08%, over 381 unknown words. The poor result is expected because the morpheme *ke-...-an* has ambiguous tags that are KN, KK and KA.

Table 3 depicts the significance of Malay circumfixes based-on the lowest error rate when morphemes are used for guessing the unknown words' POS. This analysis is based on the result in Table 2 (row 5) in which unknown words are handled by Malay affix morphemes by replacing the unknown words' emissions by marginal proportionate distribution of tags. As

per observations, Malay words that contain the circumfix*di-...-kan* is rarely listed as entries in the dictionary. The circumfix*di-...-kan* is used to derive passive verbs, which is quite similar to suffix …-*ed* in English, for indicating the past tense. Therefore, our method to penalise the emission probabilities of unknown words using Malay affix morphemes is effective for certain morphemes.

**Table 3.** The best circumfixes for guessing unknown words' POS

| Circumfixes | Percentage of Errors |
|---|---|
| di-...-kan | 5.36% |
| per-...-an | 5.56% |
| di-...-i | 8.13% |
| mem-...-kan | 11.27% |
| mem-...-i | 14.83% |
| pen-...-an | 17.30% |
| ber-...-an | 21.09% |
| pem-...-an | 21.95% |

**Table 4.** The best prefixes for guessing unknown words' POS

| Prefixes | Percentage of Errors |
|---|---|
| men- | 4.63% |
| se- | 7.45% |
| di- | 9.29% |
| ber- | 20.20% |
| peng- | 21.15% |
| ter- | 27.02% |
| ke- | 28.27% |

Table 4 shows the significance of Malay prefixes based on the lowest error rate when the regarding morphemes are used for guessing the unknown words' POS. This analysis is also based on the result in Table 2 (row 5). According to the condition, whereby only some amount of words is considered as reliable data (words that occur more than 100 times), the prefix *di-* shows the best guessing with an error rate of 9.29% from 990 words.The bad guessing is

identified as unknown words that contain prefix *be-...* with an error rate of 66.67% from 105 words. The reason for this, is that the prefix *be-...* clashes with Malay words that originally begin with *be-…* (such as *begitulah, benar-benar, benihnya*, etc.) in a way that the words become unknown because their word-form orthographically changed after adding particle *lah*, clitic*nya* or hyphen.

According to the condition, whereby only some amount of words are reliable data (words that occur more than 100 times), the suffix …*-i*shows the best guessing with an error rate of 15.76% from 590 words. The reason for this high error rate is because the morpheme …*-i* clashes with Malay words that originally end up with letter *i* such as *ahli-ahli, saksi-saksi, Hilmi, Fahmi, koboi*, etc. In general, guessing the POS using Malay suffixes give inaccurate results; for example, the suffix …*-an* has successfully guessed only half of the 644 words (error rate 49.69%). The other prefixes show an error rate higher than 50%.The original Malay prefixes in the rules are only *-an, -kan* and *-i*; the others are from use in foreign words, such as …*-in*, …*-ah*, …*-at* from Arabic and …*-isme* from English.

## 3. EXPERIMENTAL

Unknown words often play an important role in describing the meaning of a sentence because an unknown word is mostly a special word that carries more semantic information than a known word [26]. Most of the unknown words can be assigned with an open class, such as nouns or verbs, by the assumption that they are impossible to exist in close class category such as determiners or prepositions. Handling unknown words is a key to improve the performance of POS taggers [27].Any POS tagging models which handle unknown words are often used and adapted for tagging under-resourced languages [28].

The term "unknown word" in statistical POS tagging refers to words that are absent in the training corpus or dictionary. In case of unsupervised HMM POS tagging, words that are not in the training corpus are known as “unseen words”. The unseen words lead to the problem of inexistence of their emission probabilities, which requires the Viterbi tagging to approximate the value with some mechanism.Furthermore, absence of words in the dictionary implies that such words in the corpus cannot be matched to their tags. The initial emission probabilities of

such words must be approximated with some mechanism. In this case, the substitution method can be used to replace emission probability of words not listed in the dictionary, i.e. $P(w|t_{unknown})$with some value produced using the mechanism of handling unknown words.

**3.1.Assigning the Initial Emissions of HMM Training**

Prior to the initialisation, all word types in the corpus can be grouped into their equivalent classes after being matched with POS tags referred in the dictionary. For example, words that are categorised as only noun are pooled into one equivalent class, and words that are categorised as either verb or adjective are form in another class, and so on. Grouping the words into their equivalent classes tremendously reduces the number of emission parameters, giving an advantage in estimating the transitions more reliably [29-30]. Furthermore, a word that occurred in the corpus more than 100 times should individually group into a single class to avoid a skewed probability of high frequency with low frequency words within the same class. Each group is treated as a metaword, $u_L$, where $L$ is a subset of the integers from 1 to $T$ and $T$ is the number representing the tags.

In our Malay POS tagger development, the HMM training employs an untagged Malay corpus containing 995,240 tokens which consisting of 30,640 word types including symbols. Out of 30,640 word types, 14,068 words are not listed in the dictionary. After completing the training, the metaword belonging to 'unknown tag' group also has an emission probability, which is the probability of the metaword given by an 'unknown tag', $P(u_L|t_{unknown})$.The value of this probability is produced in conjunction with other emission values by training. Nevertheless, this value must be substituted with certain probabilistic measures during Viterbi tagging.

**3.2.Estimating the Number of Joint "Word-Tag" $C(w_k, t_i)$**

For every iteration in the HMM training, the trained emission probability $P(u_L|t_i)$ can be proportionate to the conditional probability of the word given a tag $P(w|t_i)$ by the assumption $P(w|t_i) = \frac{C(w)}{C(u_L)}P(u_L|t_i)$ if $w \in u_L$ where $C(w)$ is the number of token $w$ and $C(u_L)$ is the total token accumulated in meta-word $u_L$. Hence, the joint probability of metawords $u_L$ with tag $t_i$ is estimated as follows:

$$P(u_L, t_i) = P(u_l|t_i)P(t_i) \quad (1)$$

The marginal $P(t_i)$ is estimated using the following equation:

$$P(t_i) = \frac{\sum_{r=1}^{R} \gamma_r(t_i)}{\sum_{j=1}^{T} \sum_{r=1}^{R} \gamma_r(t_j)} (2)$$

$\gamma_r(t_i)$ is the probability of being in state $t_i$ at observation $r$ for a given observation sequence in the HMM model.The probability of a metaword $P(u_L)$ is calculated after grouping the words into metawords and dividing the number of a metaword over all metawords:

$$P(u_L) = \frac{C(u_L)}{\sum_{u_{L'}} C(u_{L'})} (3)$$

A reverse conditional probability $P(t_i|u_L)$ is counted as follows:

$$P(t_i|u_L) = \frac{P(u_L, t_i)}{P(u_L)} (4)$$

The number of joint "word-tag" is estimated as follows:

$$C(w_k, t_i) = P(t_i|w_k)C(w_k) (5)$$

The $P(t_i|w_k)$ in Equation (5) can be substituted by $P(t_i|u_L)$ for every $w_k \in u_L$ as per assumption $P(w|t_i) = \frac{C(w)}{C(u_L)} P(u_L|t_i)$ mention above. Therefore:

$$C(w_k, t_i) = \frac{P(t_i|u_L)C(w_k)^2}{C(u_L)}; \quad \forall w_k \in u_L (6)$$

### 3.3.Morpheme-Based POS Guessing

The way of forming derivative words in Malay is accomplished by merging root words with affixes. For example, a root word *serap* (absorb) can produce new words such as *menyerap* (absorb), *menyerapkan* (induct), *diserapkan* (inducted), menyerapi (permeated), diserapi (be permeated), *penyerap* (absorber), *penyerapan* (absorption), *terserap* (absorbed), *terserapkan* (absorbable), *serapan* (absorption), *keterserapan* (absorptive), *dayaserap* (absorptive) and *kedayaserapan* (absorptiveness).Affixes are considered bound morpheme as opposed to root words, which are unbound that can receive affixations. Therefore, affixes cannot present alone in a sentence (for example, *ber-*, *ter-*, *ke-*, *me-*, *-nya*, *-kah*, *-lah*, *-pun*, *-an*), they must be affixed to root words. Affixes can be categorised into three types, i.e. prefixes, suffixes and circumfixes.

The part of speech (POS) of many derivative words formed by Malay morphological rules are

predictable such as derivative nouns classified as *Kata Nama* (Noun) or KN, derivative verbs classified as *Kata Kerja* (Verb) or KK and derivative adjective classified as *Kata Adjektif* (Adjective) or KA. The morphological rules are represented in Table 5.

**Table 5.** Malay morphological rules

Rule 1:

POS = {'KN'} if the derivative word has any following affixes:

1. Circumfixes: { *per-...-an, penge-...-an, peng-...-an, pen-...-an, pem-...-an, pel-...-an, pe-...-an* }
2. Prefixes: { *tata-..., supra-..., sub-..., pra-..., per-..., penge-..., peng-..., pen-..., pem-..., pel-..., pe-..., maha-..., ke-..., juru-..., eka-..., dwi-...* }
3. Suffixes: { *...-wati, ...-wan, ...-man, ...-isme, ...-in, ...-at, ...-an, ...-ah* }

Rule 2:

POS = {'KK'} if the derivative word has any following affixes:

1. Circumfixes: {*menge-...-kan, meng-...-kan, meng-...-i, men-...-kan, men-...-i, memper-...-kan, memper-...-i, mem-...-kan, mem-...-i, me-...-kan, me-...-i, ke-...-an, diper-...-kan, diper-...-i, di-...-kan, di-...-i, ber-...-kan, ber-...-an* }
2. Prefixes: { *meny-..., menge-..., meng-..., men-..., memper-..., mem-..., me-..., diper-..., di-..., ber-..., bel-..., be-...* }
3. Suffixes: { *...-kan, …-i* }

Rule 3:

POS = {'KA'} if the derivative word has any following prefixes:

1. Prefixes: { *te-…, se-…* }

Rule 4:

POS = {'KN', 'KA'} if the derivative word has the following circumfix:

1. Circumfix: { *ke-...-an* }

Rule 5:

POS = {'KK', 'KA'} if the derivative word has the following prefix:

1. Prefix: { *ter-...* }

It is critical to apply the linguistic rules presented in Table 5 to evaluate the precedence of affixes for the best guessing of word classes. However, by examining the letters in each affix, the longest affix string becomes a superset to the shorter one. For example, the prefix pe-… in Rule 1 is a subset to the prefix *per-..., penge-..., peng-..., pen-..., pem-...* and *pel-*….Therefore, the longest affixes are always the highest precedence. The circumfixes are made up of both certain prefixes and suffixes, in which both prefixes and suffixes are subsets to circumfixes. For example, the circumfix*diper-…-kan* is made up of a combination of the prefix *diper-*… and suffix …*-kan*. Therefore, the circumfix becomes the highest precedence followed by prefixes and suffixes. The suffixes have lower precedence compared to prefixes because they are fewer in numbers.

The directed graphs are used to integrate the Malay morphological rules into HMM POS tagger. Fig. 4 represents the circumfixes of Rule 1, 2 and 4; Fig. 5 represents the prefixes of Rule 1, 2, 3 and 5; and Fig. 6 represents the suffixes of Rule 1 and 2. The red nodes indicate the start of tracking prefixes in Fig. 5, whereas the blue nodes indicate the start of tracking suffixes in Fig. 6. An algorithm to guess the POS of unknown words using morphological rules is given in Table 7. This algorithm is used to examine the existence of Malay affix morphemes in unknown words and then predict their POS. The algorithm concludes a word is having affixes if the graph either in Fig. 4, 5 and 6 is traceable according to character sequence in the word. Circumfixes in Fig. 4 are successfully concluded if tracking both prefixes and suffixes encounter at the determinant nodes indicated by the orange colour in which the predicted POSs are stored in the node.Similarly, prefixes and suffixes are successfully concluded if the tracking encounter at the determinant node (orange colour) in Fig. 4 and 5 respectively.The algorithm is treated as baseline tagging based on morphological rules and does not involve any training corpus or tagged dictionary.

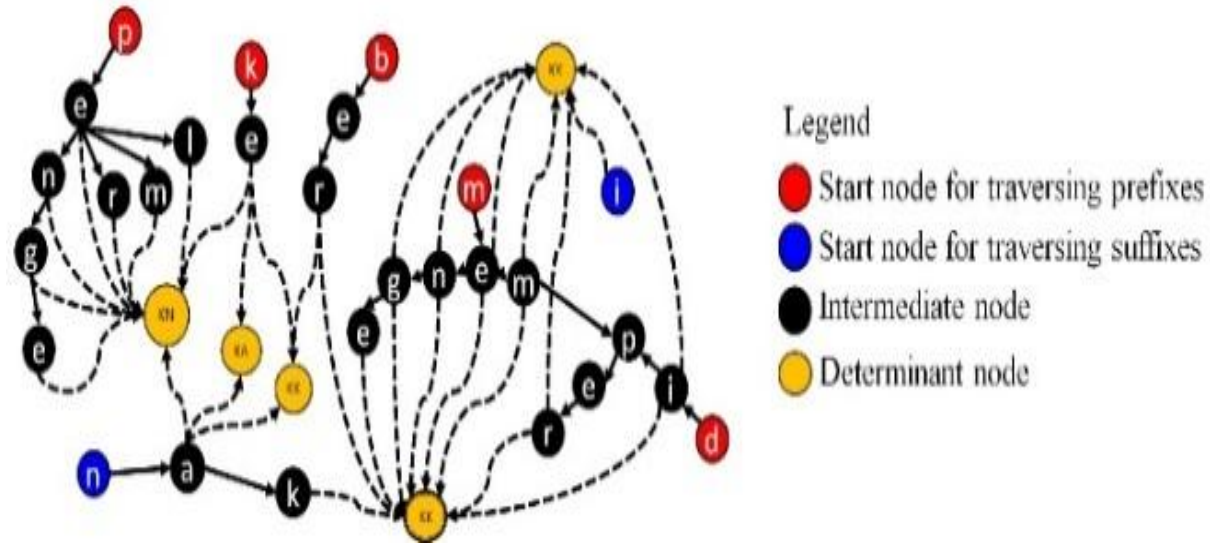


**Fig.4.**Circumfix graph

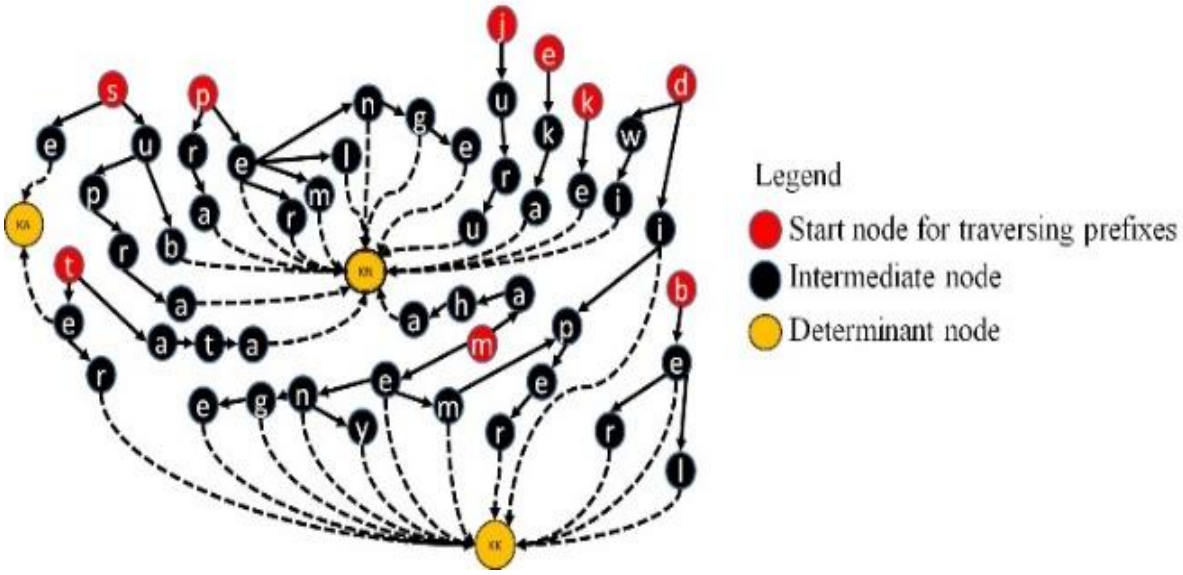


**Fig.5.** Prefix graph

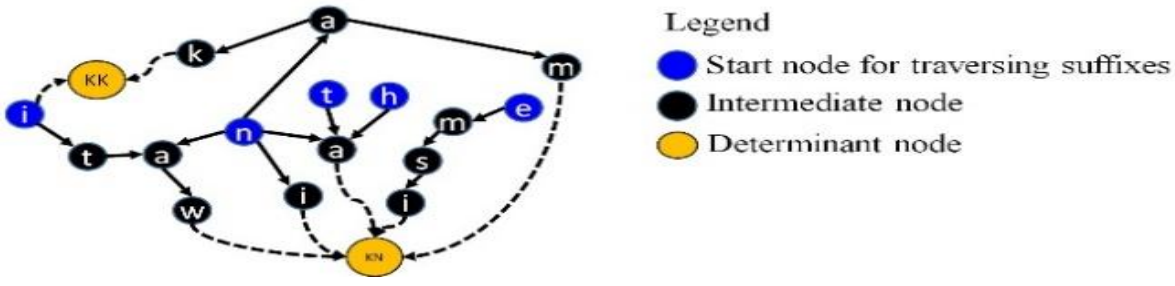


**Fig.6.** Suffix graph

**Table 3.** POS guesser algorithm using affix morphemes

| For each unknown word, find their affix morpheme using the following steps: |
|---|
| 1. Travers the circumfix graph |

If meet determinant node, then

*Return POS set embedded to the node*

2. Else travers prefix graph

   If meet determinant node, then

   *Return POS set embedded to the node*

3. Else travers suffix graph

   If meet determinant node, then

   *Return POS set embedded to the node*

4. Else

   *Return POS set = { 'KN', 'KNK', 'KK' }*

### 3.4. Penalizing Unknown Words' Emissions

Whenever the tagger encounters unknown words, the tags are allocated with possible tags given by the POS guesser algorithm. Due to this, Viterbi tagging needs words' emission probability to disambiguate and assign the most possible POS tags as per word context. Since unknown words are absent in the training corpus, such emission values are found missing.

To resolve this issue, the emission probabilities are estimated in two ways. First, the emission probabilities are assigned according to uniform distribution of all possible tags given in Equation 7. Second, the emission probabilities are assigned according to marginal proportionate distribution of tags produced during HMM training given in Equation 8.

### 3.5. Emission Probabilities by Uniform Distribution of all Possible Tags

$$P(w|t) \cong \begin{cases} \dfrac{1+\delta}{|X|+\delta|T|} & if\ t \in X \\ \dfrac{\delta}{|X|+\delta|T|} & if\ t \notin X \end{cases} \quad (7)$$

where $X$ is a set of possible POS of the unknown word returned by the POS guesser algorithm, $|T|$ is the number of all tags ($|T|$ = 40) and $\delta$ is a smoothing factor in which the best value is 0.01. The value comes from cross-validation result using the development corpus (30,017 tagged-tokens). The cross-validation observation is done by partitioning the development corpus into ten partitions with similar size (about 3K each). Nine of them are merged back and used for training and the rest is used for testing observation. This process is repeated ten

times, such that each partition is used for training and observation. Table 6 depicts the different values given to $\delta$ against the accuracies of tagging the unknown words in each partition.

**Table 6.** Observation results for tagging unknown words in each partition against different given $\delta$ values

| ObservingCorpus | Given $\delta$ Values | | | | |
|---|---|---|---|---|---|
| | $10^{-1}$ | $10^{-2}$ | $10^{-3}$ | $10^{-4}$ | $10^{-5}$ |
| Partition 1 | 32.14% | 38.74% | 37.62% | 37.61% | 37.61% |
| Partition 2 | 31.22% | 37.74% | 36.01% | 35.32% | 35.10% |
| Partition 3 | 32.32% | 38.05% | 37.01% | 36.82% | 36.70% |
| Partition 4 | 33.00% | 38.73% | 37.62% | 37.61% | 37.61% |
| Partition 5 | 31.82% | 38.64% | 38.00% | 37.80% | 37.80% |
| Partition 6 | 32.00% | 37.84% | 36.61% | 35.82% | 35.80% |
| Partition 7 | 31.00% | 37.54% | 36.91% | 35.72% | 35.50% |
| Partition 8 | 31.23% | 37.94% | 36.91% | 36.32% | 36.10% |
| Partition 9 | 32.24% | 38.77% | 37.52% | 37.51% | 37.51% |
| Partition 10 | 32.10% | 38.70% | 37.82% | 37.31% | 37.11% |

### 3.6.Emission Probabilities by Marginal Proportionate Distribution of Tags

$$P(w|t) \cong \begin{cases} \frac{P(t) + \delta}{Y}, & if\ t \in X \\ \frac{\delta}{Y}, & if\ t \notin X \end{cases} \quad (8)$$

where $P(t)$ is the probability of tag, $Y$ is the normalisation factor and $\delta$ is the smoothing factor defined as the lowest $P(t)$ for $t$ in $X$ multiply by coefficient $\epsilon$ ($\epsilon = 0.1$ is the best value observed). This observed value is also determined by a cross validation observation. Table 7 presents the different values given to $\epsilon$ against the accuracies of tagging the unknown words in each partition.

**Table 7.** Observation result for tagging unknown words in each partition against different given $\epsilon$values

| Observing Corpus | Given $\epsilon$ Values | | | | |
|---|---|---|---|---|---|
| | $10^{-1}$ | $10^{-2}$ | $10^{-3}$ | $10^{-4}$ | $10^{-5}$ |
| Partition 1 | 39.31% | 38.26% | 37.86% | 37.82% | 37.80% |
| Partition 2 | 38.50% | 37.87% | 37.18% | 37.11% | 37.08% |
| Partition 3 | 39.51% | 38.37% | 37.97% | 37.90% | 37.85% |
| Partition 4 | 39.61% | 38.36% | 37.96% | 37.83% | 37.80% |
| Partition 5 | 38.90% | 38.01% | 37.52% | 37.08% | 37.00% |
| Partition 6 | 38.90% | 38.00% | 37.55% | 37.49% | 37.45% |
| Partition 7 | 38.40% | 37.87% | 37.17% | 37.10% | 37.08% |
| Partition 8 | 38.80% | 37.81% | 37.56% | 37.49% | 37.40% |
| Partition 9 | 39.50% | 38.10% | 37.25% | 37.20% | 37.19% |
| Partition 10 | 39.00% | 38.00% | 37.20% | 37.19% | 37.15% |

### 3.7. Predicting POS through a Words Starting

The term “words’ starting” is the sequence of characters that begin a word string. For example, the word “*hasut*” (instigate) can have a word starting set of {“*h*”, “*ha*”, “*has*”, “*hasu*”} for a predefined length of four characters. Intuitively, the longer the sequence of characters, the stronger the judgment in predicting a words’ tag. For substantial amounts of this information alternative emission probability values of unknown words can be estimated.For example, the probability of a tag given a words’ starting is estimated based on the statistical data available for words that begin with the same sequence of letters. Therefore, the probability distribution can be generated from all words in the training corpus that share the same sequence of letters for some predefined length. This model implicitly embedded the linguistic knowledge of Malay affixes. The probability of a tag $t_i$ given the first $m$ letters $l_1 l_2 \dots l_m$ of the letter sequence in a word is estimated and smoothed using successive abstraction [31-32].

This estimation is recursively calculated by considering the marginal distribution of tags $P(t_i)$ produced by HMM training, formulated in Equation (2) and the standard division in Equation (15) to every successive character.

$$\hat{P}(t_i|l_1 l_2 \dots l_m) = \frac{P(t_i|l_1 l_2 \dots l_m) + \sigma \hat{P}(t_i|l_1 l_2 \dots l_{m-1})}{1 + \sigma} (9)$$

$$\hat{P}(t_i|l_1) = \frac{P(t_i|l_1) + \sigma \hat{P}(t_i)}{1 + \sigma} (10)$$

$$P(t_i|l_1 l_2 \dots l_m) = \frac{C(t_i, l_1 l_2 \dots l_m)}{C(l_1 l_2 \dots l_m)} (11)$$

$$\hat{P}(t_i) = P(t_i) (12)$$

For any defined length of $m$ and $m > 0$, $C(t_i, l_1 l_2 \dots l_m)$ is the total number of joint "word-tag" that shares the same words' starting $l_1 l_2 \dots l_m$ with tag $t_i$; $C(l_1 l_2 \dots l_m)$ is the total number of word types that shares the same words' starting $l_1 l_2 \dots l_m$. Therefore:

$$C(t_i, l_1 l_2 \dots l_m) = \sum_{w \subset l_1 l_2 \dots l_m} C(w, t_i) (13)$$

$$C(l_1 l_2 \dots l_m) = \sum_{w \subset l_1 l_2 \dots l_m} C(w) (14)$$

The number of word type $C(w)$ can be counted based on the word type's frequency in the training corpus. The number of joint "word-tag" $C(w, t_i)$ is estimated using Equation (6) and the value of $\sigma$ is a standard deviation of the marginal distribution of tags Equation (15) produced in each iteration of HMM training.

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (P(t_i) - \bar{P})^2}, \bar{P} = \frac{1}{N} \sum_{i=1}^{N} P(t_i) (15)$$

### 3.8.Predicting POS through a Words' Ending

By looking at the backward sequence of characters of a word, a similar concept to predicting the POS through a words' starting is used for words' ending. For example, the word "*hasut*" can have a word ending set of {*"t"*, *"ut"*, *"sut"*, *"asut"*} for a predefined length of four characters.

The probability distribution for a words' ending is generated from all word types in the training corpus that share the same words' ending of some predefined length. The probability of tag t_i given the last m letters l_(n-m+1) l_(n-m+2)…l_n of a word is recursively estimated and smoothed, like predicting POS for unknown words through a words' starting in the above section, but treated in reverse order of characters

## 4. CONCLUSION

This work presented a Malay POS tagger based on the unsupervised Hidden Markov Model (HMM). The dependence of training on an untagged corpus and limitation of unsupervised training limited to a published dictionary is a major challenge in the work. The dictionary does not include all words found in the corpus, especially derivative words such as passive verbs and derivative nouns. Therefore, the training outcome has a problem with unknown words, not just words absent in the corpus, but also words that appeared but are not listed in the dictionary.

Effort has been made for finding the exact morphemes of prefixes, suffixes and circumfixes in the agglutinative Malay language.When tagging a new sentence, words in the sentence identified as not listed in the dictionary are assigned with probable tags based on linguistically meaningful affixes, as defined in morphological rules through the morpheme-based POS guessing algorithm.

Viterbi tagging with words' starting information is better than using a words' ending information for guessing unknown words' POS. A good overall accuracy is achieved using a words' starting information with the need to check a maximum of four characters in predefined length (81.81%) compared to using a words' ending information, where a maximum of six characters predefined length (81.71%) is necessary.The overall performance of Viterbi tagging with morpheme-based POS guessing shows that the unknown word emissions substituted by the value proportionate to marginal distribution of possible tags of unknown words (82.28%) is better than the words' emission substituted by the equal distribution of all possible tags of the unknown word (82.25%). However, emissions substituted by the equal distribution of all possible tags are good for tagging words not listed in the dictionary (42.90%). On the other hand, emissions substituted by a value proportionate to the marginal distribution are good for tagging unseen words (31.94%). Viterbi tagging with morpheme-based POS guessing (good overall accuracy was 82.28%) is better than HMM-Viterbi tagging with a words' starting information prediction (good overall accuracy was 81.81%). Therefore, tagging unknown words identified as not existing in the dictionary is better with the assistance of morpheme-based POS guessing (42.52%)

**5. ACKNOWLEDGEMENTS**

We would like to thank Assoc. Prof. Dr. Norsimah Mat Awal and her students for verifying the development and test corpus of this project through three workshop series and MrAzharJaludin who assisted in obtaining the raw data. This material is based on work supported by the UniversitiKebangsaan Malaysia (UKM) under Fundamental Research Grant Scheme (FRGS/1/2012/SG05/UKM/02/13), provided by the Malaysian Ministry of Education (MOE).

**6. REFERENCES**

[1] Benjamin C M X, Mohamed L, Liew K P, Khalil B, Rohana M,Dicson L.BechmarkingMi-POS: Malay part-of-speech tagger. International Journal of Knowledge Engineering, 2016, 2(3):1101-1112

[2] Juhaida A B, Khairuddin O, Mohammad F N,Mohd Z M. NUWT: Jawi-specific buckwalter corpus for Malay word tokenization. Journal of Information and Communication Technology, 2016, 15(1):107-131

[3] Juhaida A B, Khairuddin O, Mohammad F N,Mohd Z M.Tokenizer for Malay language using pattern matching. In IEEE14th International Conference onIntelligent Systems Design and Applications, 2014, pp. 140-144

[4] Safiah K. N., Farid O. M., Hashim M., Hamid M. A.Tatabahasadewanedisiketiga. Kuala Lumpur: DewanBahasadanPustaka, 2010

[5] Abdullah H.Morfologisiripengajarandanpembelajaranbahasa Melayu. Kuala Lumpur: PTS Professional, 2006

[6] Miller J E, Torii M, Vijay-Shanker K. Adaptation of POS tagging for multiple biomedical domains. In Workshop onBiological, Translational, and Clinical Language Processing, 2007, pp. 179-180

[7] Shell M. IEEEtran homepage on CTAN. Utah: Comprehensive TEX Archive Network, 2002

[8] Cucerzan S,Yarowsky D. Language independent, minimally supervised induction of lexical probabilities. In 38th Annual Meeting on Association for Computational Linguistics,

2000, pp. 270-277

[9] Garrette D,Baldridge J. Type-Supervised Hidden Markov Models for part-of-speech tagging with incomplete tag dictionaries. In Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, 2012, pp. 821-831

[10] Goldberg Y, Adler M,Elhadad M. EM can find pretty good HMM pos-taggers (when given a good start). In Association for Computational Linguistics, 2008, pp. 746-754

[11] Garrette D,Baldridge J. Learning a part-of-speech tagger from two hours of annotation. In Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2013, pp. 138-147

[12] Garrette D, Mielens J,Baldridge J. Real-world semi-supervised learning of POS-taggers for low-resource languages. In 51st Annual Meeting of the Association for Computational Linguistics, 2013, pp. 583-592

[13] Alfred R, Mujat A, Obit J H. A ruled-based part of speech (RPOS) tagger for Malay text articles. In A. Selamat, N. T. Nguyen, & H. Haron (Eds.), Lecture Intelligent Information and Database Systems: 5th Asian Conference, ACIIDS 2013, Kuala Lumpur, Malaysia, March 18-20, 2013, Proceedings, Part 2. Berlin: Springer,2013, pp. 50-59

[14] Juhaida A B, Khairuddin O, Mohammad F N,Mohd Z M. Morphology analysis in Malay POS prediction. In International Conference on Artificial Intelligence in Computer Science and ICT, 2013, pp. 112-119

[15] Zamin N, Oxley A, Bakar Z A, Farhan Y A. A lazy man's way to part-of-speech tagging. In D. Richards, & B. H. Kang (Eds.),Knowledge management and acquisition for intelligent systems. Berlin: Springer, 2012, pp. 106-117

[16] Hassan M, Omar N, Aziz M J A. Statistical Malay part-of-speech (POS) tagger using Hidden Markov approach. In International Conference onSemantic Technology and Information Retrieval,2011, pp. 231-236

[17] Pisceldo F, Adriani M, Manurung R. Probabilistic part of speech tagging for BahasaIndonesi. In 3rd International MALINDO Workshop, 2009, pp. 1-6

[18] Knowles G, Don, Z M. Tagging a corpus of Malay texts, and coping with syntactic drift.

In Corpus Linguistics, 2003, pp. 422-428

[19] Don Z M. Processing natural Malay texts: A data-driven approach. TRAMES, 2010, 14(1):90-103

[20] Abdullah H., Seri L. J. R., Razali A.,Zulkifli O.SintaksissiripengajarandanpembelajaranbahasaMelayu.Kuala Lumpur: PTS Professional, 2010

[21] Asmah O.NahuMelayumutakhir.Kuala Lumpur: DewanBahasadanPustaka, 2009

[22] Ranaivo-Malançon B. Issues in building a Malay part of speech tag-set. In International MALINDO Workshop,2008, pp. 104-108

[23] Hock O. Y.Kamusdwibahasa. Selangor: Pearson Longman, 2009

[24] Hawkins M. J.KamusdwibahasaBahasaInggeris-Bahasa Malaysia.Selangor: Oxford Fajar, 2008

[25] Arbak O.KamuskomprehensifbahasaMelayu.Selangor: Oxford Fajar, 2005

[26] Vadas D, Curran J. Tagging unknown words with raw text features. In Australasian Language Technology Workshop, 2005, pp. 32-39

[27] Hall J. A probabilistic part-of-speech tagger with suffix probabilities. Master thesis, Sweden: Växjö University, 2009

[28] Dandapat S. Part-of-speech tagging for Bengali. Master thesis, Kharagpur: Indian Institute of Technology, 2009

[29] Banko M, Moore R C. Part of speech tagging in context. In 20th International Conference on Computational Linguistics, 2004, pp. 556-561

[30] Kupiec J. Robust part-of-speech tagging using a hidden Markov model. Computer Speech and Language, 1992, 6(3):225-242

[31] Brants T. TnT: A statistical part-of-speech tagger. In 6th Conference on Applied Natural Language Processing, 2002, pp. 224-231

[32] Samuelsson C. Handling sparse data by successive abstraction. In 16th Conference on Computational Linguistics-Volume 2, 1996, pp. 895-900

[33] Schröder I. A case study in part-of-speech tagging using the ICOPOST toolkit. Technical Report FBI-HH-M-314/02, Germany: University of Hamburg, 2002

[34] Giesbrecht E, Evert S. Is part-of-speech tagging a solved task? An evaluation of POS taggers for the German web as corpus. In 5th Web as Corpus Workshop, 2009, pp. 27-35

**How to cite this article**:
Mohamed H, Omar N, Aziz M J A. Penalizing unknown words' emissions in hmm pos tagger based on malay affix morphemes. J. Fundam. Appl. Sci., 2017, 9*(3S), 457-483.*