International Journal of Engineering & Technology, 7 (4.29) (2018) 9-12

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

The Effectiveness of Using Malay Affixes for Handling Unknown Words In Unsupervised HMM POS Tagger

Hassan Mohamed[1]*, Nazlia Omar[2], Mohd Juzaiddin Ab Aziz[3]

[1]Cyber Security Centre, National Defence University of Malaysia (NDUM),
Sungai Besi Camp,57000 Kuala Lumpur, Malaysia
[2,3]Knowledge Tech. Group, Centre for AI Technology (CAIT),
Universiti Kebangsaan Malaysia (UKM), Bangi, 43600, Malaysia
**Corresponding author E-mail: hassan@upnm.edu.my*

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. A morpheme-based POS guessing algorithm has been introduced to assign unknown words' probable tags based on linguistically meaningful affixes. Therefore, the exact morphemes of prefixes, suffixes and circumfixes in the agglutinative Malay language is examined before giving tags to unknown words. The algorithm has been integrated into HMM tagger which uses HMM trained parameters for tagging new sentences. However, for unknown words their parameters are absent. Therefore, the algorithm applies two methods for assigning unknown words' emission to HMM tagger, first is based on uniform distribution of all possible tags; and second, is based on marginal proportionate distribution of tags. The effective method is proven to be using morpheme-based POS guessing with unknown word emissions substituted by a value proportionate to the marginal distribution of tags.

Keywords: *Malay; POS tagger; unsupervised HMM.*

1. Introduction

Part of speech (POS) tagging is very important as it is a low-level parsing of natural language to build many Natural Language (NLP) applications. The POS tagger assigns a unique grammatical class to each word according to the context in a sentence. Therefore, a word can have different POS tags based on their meaning which reflects to ambiguity problem. The other issue is guessing unknown words POS to any unseen words. The ambiguity and guessing unknown words problems make the POS tagging a non-trivial process since context or 'meaning' interpretation must be considered before assigning the probable tag for a given word.

There is some interest to move further in Malay Natural Language Processing research especially in POS tagging. Most of the previous works on Malay POS tagging was based on supervised training. The recent is in [1] called Mi-POS, was developed using maximum entropy approach implemented using OpenNLP. A Malay POS tagger that used context information (i.e. surrounding tags) and prefix/suffix information was developed in [2] to resolve ambiguous tags and unknown word's tag using trigram Hidden Markov Model (HMM). Other POS tagger in [3] is for Bahasa Indonesia developed using two machine learning methods namely Conditional Random Fields (CRF) and Maximum Entropy (MaxEnt). The one that use unsupervised training was in [4] using N-gram and Dice Coefficient approaches for similarity measurement purposes for projecting from English tags to Malay words. Furthermore, the rule-based method for Malay POS tagger was developed by [5] called RPOS which applies Malay affixation rules and word relation to determine word category. POS tagging based-on Malay affixation was also reported in [6], where the tag of a word was determined by the result of inferences of morphological analysis rules. On the other hand, a syntactic drift and data-driven approach to identify the Malay grammar class appeared in [7] where a Malay tagset is derived through the analysis of syntactic structure. Furthermore, this tagset has been used for annotating four Dewan Bahasa dan Pustaka (DBP) novels.

Malay language is a derivative language where most of the words are formed by merging affixes with root words [8], [9]. Affixation is accomplished by either adding an affix at the beginning, middle or the ends of the root word or combination of both begins and ends. Due to the well-defined affixation rules, the word class of Malay derivative words can be intuitively guessed. Therefore, analysis on Malay morphology for POS prediction was done in [10] from the views of computational linguistics using two machine learning algorithms i.e. Decision Tree (J48) and Nearest neighbour (kNN). The verb category (KK) firmly classified by J48 algorithm. This idea inspires to examine the effectiveness of using Malay affix morphemes for handling unknown words in the unsupervised Hidden Markov Model (HMM) POS tagging and hence improve our tagging accuracy in Malay POS Tagger (MyPOST). This method 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.

2. Morpheme-based POS Guessing

Malay is a language which belongs to agglutinative language family [8] such that affixation forms many derivative words. The way of forming derivative words is accomplished by merging root words with affixes. Affixation could be any one of the following processes i.e. prefixation, suffixation, circumfixation (prefix and suffix), and infixation. Prefixation only involves prefixes, which is affix concatenated preceding a root word. Suffixation involves suffixes, which is affix concatenated succeeding a root word. Circumfixation is a process which involved both prefix and suffix to a root word. While infixation is affixation process which inserts an affix within the root word.

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 1.

Table 1: 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 for an algorithm to choose the linguistic rules, as presented in Table 1, to determine the precedence of affixes for the best guessing of word classes. There are two criteria can be used for this purpose. First, the number of letters in each affix. The longest affix string can be higher precedence because they are mostly superset to the shortest. For example, the prefix *pe-...* in *Rule 1* is a subset to the prefix *per-..., penge-..., peng-..., pen-..., pem-...* and *pel-...*. Second, the number of affixes in each category. The number of prefixes is higher than suffixes, so that prefixes could be set as higher precedence rather than suffixes.

The Malay morphological rules in Table 1 is integrated into HMM POS tagger for guessing unknown words, in which the sequence of letters in affixes is modelled using directed graph. Fig. 1 represents the prefixes of Rule 1, 2, 3 and 5; Fig. 2 represents the suffixes of Rule 1 and 2; and Fig. 3 represents the circumfixes of Rule 1, 2 and 4. An algorithm to guess the POS of unknown words using this model is given in Table 2. The algorithm examines the existence of Malay affix morphemes in unknown words and then predict the POS by tracing the graphs according to character sequence in the word. Any prefixes, suffixes or circumfixes are successfully examined if the tracking encounter at the determinant node whereby the predicted POSs and probability values (estimated word's emission) are stored.

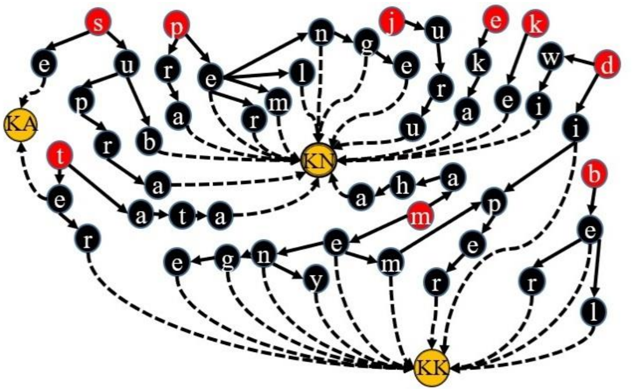

Fig. 1: Prefix graph. Yellow nodes are determinant node which contains either KA, KN or KK POS tag. The red nodes are indicating the start for traversing the graph.

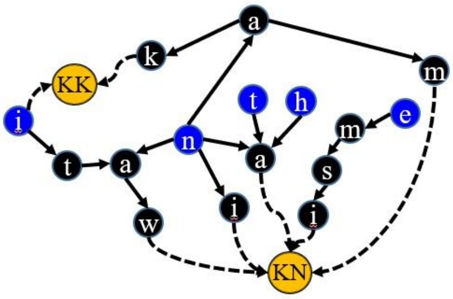

Fig. 2: Suffix graph. Yellow nodes are determinant node which contains either KN or KK POS tag. The blue nodes are indicating the start for traversing the graph.

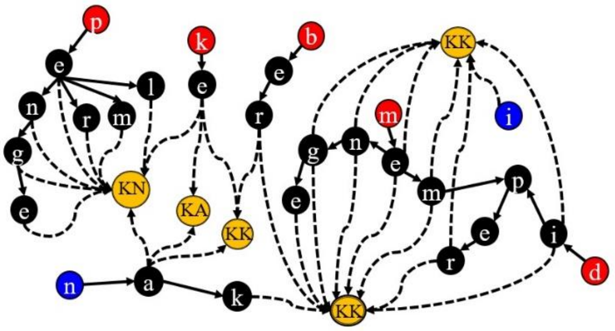

Fig. 3: Circumfix graph. Yellow nodes are determinant node which contains either KK, KN or KA POS tag. The red nodes are indicating the first start for traversing prefix while the blue nodes are for the second start of suffix.

Table 2: POS Guesser Algorithm using Affix Morphemes

For each unknown word, find their affix morpheme using the following steps:

```
1. Traverse the circumfix graph
   If meet determinant node, then
   Return POS set embedded to the node
2. Else traverse 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. Assigning unknown word's emission to HMM tagger

Whenever the tagger encounters unknown words, the POS guesser algorithm would propose possible tags. Due to the proposed tags are ambiguous, HMM tagger (i.e. Viterbi algorithm) needs to disambiguate and assign the most possible POS tags as per word context through words' emission probability. 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, it is assigned according to uniform distribution of all possible tags given in (1). Second, it is assigned according to marginal proportionate distribution of tags given in (2).

$$P(w|t) \cong \begin{cases} \frac{1+\delta}{|X|+\delta|T|} & \text{if } t \in X \\ \frac{\delta}{|X|+\delta|T|} & \text{if } t \notin X \end{cases} \tag{1}$$

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 δ 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 3 depicts the different values given to δ against the accuracies of tagging the unknown words in each partition.

Table 3: Observation results for tagging unknown words in each partition against different given δ values

Observing corpus	Given δ 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%

$$P(w|t) \cong \begin{cases} \frac{P(t)+\delta}{Y}, & \text{if } t \in X \\ \frac{\delta}{Y}, & \text{if } t \notin X \end{cases} \tag{2}$$

where P(t) is the probability of tag; Y is the normalisation factor; and δ is the smoothing factor defined as the lowest P(t) for t in X multiply by coefficient ϵ (ϵ = 0.1 is the best value determined by a cross validation observation). Table 4 depicts the different values given to ϵ against the accuracies of tagging the unknown words in each partition.

4. Results

The accuracy of the tagging denotes the percentage of the words correctly assigned with tags as compared to the tagged corpus [11]. Therefore, the tagging performance is often measured by the overall tagging, known word and unknown word tagging accuracies [12], [13]. 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 present in the training that exist in the dictionary with only one tag.
- Seen words with ambiguous tags – the performance of tagging words present 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.

Table 4: Observation results for tagging unknown words in each partition against different given ϵ values

Observing corpus	Given ϵ 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%

Table 5 presents the results of the experiments. The number of HMM training iterations would influence the results. Therefore, the experiments are repeated for each iteration and the best overall performance from those iterations is considered the best result. 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. Furthermore, tagging words absent in the dictionary is 42.52% which is better than the baseline.

Malay affixes have some significant statistical distribution whereby the distribution of circumfixes, prefixes or suffixes in the Malay language is almost consistent for any different corpus size. The test corpus has 17,818 (17.45%) tokens of unknown words not listed in the dictionary, implying that 44.46% of words containing affixes. From the analysis, 42.90% of tagging accuracy for words not in the dictionary using affix morpheme is near to the percentage of words not listed in the dictionary with affixes (44.46%). Therefore, it is expected that 97.13% is correctly tagged for any unknown words containing Malay affixes using morpheme-based POS guessing in HMM tagger. This result indicates that using morpheme-based POS guessing for tagging affixed words in HMM tagger is very effective.

Table 5: 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	82.25	92.00	75.52	42.90	31.22
3	82.28	92.00	76.04	42.52	31.94

Legend of the Methods

1. Baseline
2. HMM (training iteration = 2) with morpheme (uniform distribution)
3. HMM (training iteration = 2) with morpheme (proportionate distribution)

5. Conclusion

The dictionary does not include all words found in the corpus, especially derivative words such as passive verbs and derivative nouns. Therefore, the training HMM tagger 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. The best overall performance of HMM tagging with morpheme-based POS guessing with unknown word emissions substituted by the value proportionate to marginal distribution of possible tags of unknown words (82.28%) showed the effectiveness of tagging unknown words.

Acknowledgement

This research material is based on work supported by the Universiti Kebangsaan Malaysia (UKM) under Fundamental Research Grant Scheme FRGS/1/2012/SG05/UKM/02/13. The authors would like to thank National Defence University of Malaysia (NDUM) for this publication support.

References

[1] Xian BCM., Lubani M, Ping LK, Bouzekri K, Mahmud R & Lukose D (2016), "Bechmarking Mi-POS: Malay Part-of-Speech Tagger", *International Journal of Knowledge Engineering*, Vol. 2(3) 115-121

[2] Mohamed H, Omar N & Aziz MJA (2011), "Statistical Malay part-of-speech (POS) tagger using Hidden Markov approach", *Proceeding of International Conference on Semantic Technology and Information Retrieval*, pp: 231-236

[3] Pisceldo F, Adriani M, & Manurung R (2009), "Probabilistic part of speech tagging for Bahasa Indonesia", *Proceeding of MALINDO'09*

[4] Zamin N, Oxley A, Bakar ZA & Farhan YA (2012), Ed., *A lazy man's way to part-of-speech tagging*, ser. Lecture Notes in Computer Science, Berlin, Heidelberg, Springer, Vol. 7457, pp: 106-117

[5] Alfred R, Mujat A & Obit JH (2013), Ed., *A ruled-based part of speech (RPOS) tagger for Malay text articles*, ser. Lecture Notes in Computer Science, Berlin, Heidelberg, Springer, Vol. 7803, pp: 50-59

[6] Ranaivo-Malançon B (2005), Approach for a Malay Morphosyntactic Tagging (Approche pour un etiquetage morphosyntaxique du malais). *Proceedings of the Traitement Automatique des Langues Naturelles*, Dourdan, France, available online: https://taln.limsi.fr/tome2/P138.pdf, last visit: 29.07.2015

[7] Knowles G. & Don ZM (2003), "Tagging a corpus of Malay texts, and coping with syntactic drift", *Proceeding of the Corpus Linguistics 2003 Conference*, Lancaster, 2003, pp. 422-428

[8] Karim NS, Farid OM, Hashim M & Hamid MA, *Tatabahasa dewan edisi ketiga*, Kuala Lumpur, Malaysia: Dewan Bahasa dan Pustaka, (2010)

[9] Abdullah H, *Morfologi siri pengajaran dan pembelajaran bahasa Melayu*, Kuala Lumpur, Malaysia: PTS Professional, (2006)

[10] Bakar J, Omar K, Nasrudin MF & Murah MZ, "Morphology Analysis in Malay POS Prediction", *Proceeding of AICS'13*, (2013), pp. 112-119

[11] Schröder I, "A Case Study in Part-of-Speech Tagging Using the ICOPOST Toolkit", Department of Computer Science, University of Hamburg, Technical report FBI-HH-M-314/02, (2002)

[12] Dandapat S, "Part-of-Speech Tagging for Bengali", MSc thesis, Indian Institute of Technology, Department of Computer Science and Engineering, Kharagpur, India, Jan. 2009

[13] Giesbrecht E & Evert S, (2009), "Is Part-of-Speech Tagging a Solved Task? An Evaluation of POS Taggers for the German Web as Corpus", *Proceeding of. Web as Corpus Workshop (WAC5)*, pp. 27-35