A Minimally-Supervised Malay Affix Learner

Yee Lin Tan
Swarthmore College
Swarthmore, PA 19081
yeelin@cs.swarthmore.edu

Abstract

This paper presents a minimally-supervised system capable of learning Malay affixation. In particular, the algorithm we describe focuses on identifying *p*-similar words, and building an affix inventory using a semantic-based approach. We believe that orthographic and semantic analyzes play complementary roles in extracting morphological relationships from text corpora. Using a limited Malay corpus, the system achieved F-scores of 36% and 86% on prefix and suffix identification. We are confident that results would improve given a larger Malay corpus. In future work, we plan to extend our algorithm to include automatic discovery of morphological rules.

1 Introduction

1.1 Overview

There are over 18 million speakers of Malay in United Arab Emirates, the US, and southeast Asian countries such as Malaysia, Indonesia, Brunei, Singapore, Thailand, and Myanmar (Ethnologue, 2002). Malay uses both Roman and Arabic scripts, and belongs to the Western Malayo-Polynesian group of languages in the giant Austronesian family of over 1200 languages (Ethnologue, 2002).

1.2 Malay Morphology

Malay morphological processes include affixation (prefixal, suffixal, and infixal) as well as reduplication; however, prefixation is one of the most productive of these processes. There is a total of 21 prefixes in Malay (Tatabahasa Dewan, 1993) and the more common ones include *men*-, *pen*-, *ber*-, *se*-, *ter*-, and *di*-. (See Appendix for the full list.) With the exception of *men*- and *pen*-, prefixes typically do not result in changes to the stem. However, prefixes *men*-, *pen*-, and their allomorphs (which we will denote as *meN*- and *peN*- respectively), take on different forms depending on the initial letter of the stem. The allomorphs of the prefix *meN*- are *me*-, *mem*-, *men*-, *meny*-, *meng*-, and *menge*-. Similarly, the allomorphs of *peN*- are *pe*-, *pem*-, *pen*-, *peny*-, *peng*-, and *penge*-. The use of the allomorphs of *meN*-, which parallels that of *peN*-, is illustrated as follows:

(a) *me*- is typically used with stems that begin with the letters l, m, n, ng, ny, r, w or y. For example, *me*- + 'nanti' (wait) = '*me*nanti' (to wait).
(b) *mem*- is typically used with stems that begin with the letter b, or cognate verbs that begin with f, p, or v. For example, *mem*- + 'beri' (give) = '*mem*beri' (to give), and *mem*- + 'proses' (process) = '*mem*proses' (to process).
(c) *men*- is typically used with stems that begin with the letters c, d, j, sy, z or cognates that begin with the letters t or s. For example, *men*- + 'cari' (search) = '*men*cari' (to search), and *men*- + 'sintesis' (synthesis) = '*men*sintesis' (to synthesize).
(d) *meng*- is typically used with stems that begin with vowels, the letters g, gh, kh, or cognates that begin with k. For example, *meng*- + 'ambil' (take) = '*meng*ambil' (to take), and *meng*- + 'kritik' (critique) = '*meng*kritik' (to criticize).
(e) *meny*- is typically used with stems that begin with the letter s, which is dropped in the inflected form. For example, *meny*- + 'sumpah' (swear) = '*meny*umpah' (to swear).
(f) *menge*- is used with monosyllabic stems. For example, *menge*- + 'cat' (paint) = '*menge*cat' (to paint), and *menge*- + 'kod' (code) = '*menge*kod' (to encode).

Unlike prefixation, suffixation in Malay never results in a change to the stem. -*i*, -*an*, and -*kan* are the only three suffixes in the language. These suffixes can be attached to words to form nouns, verbs, and adjectives. For example, 'air' (water) + -*i* = 'air*i*' (to irrigate), 'lukis' (draw) + -*an* = 'lukis*an*' (drawing), and 'kirim' (send) + -*kan* = 'kirim*kan*' (to send). Suffixes can also be combined with prefixes. *peN*- and *ke*- are frequently combined with -*an* to form nouns, while combinations like *meN-...-i*, *meN-...-kan*, *di-...-i*, *di-...-kan*, *ber-...-kan*, and *ber-...-an* generally form verbs.

In addition to prefixes and suffixes, Malay has four infixes, namely -*el*-, -*em*-, -*er*-, and -*in*-. Compared to the other two affix categories, infixes are used relatively

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 55–62
Computer Science Department, Swarthmore College

infrequently since only a very small subset of words in Malay take infixes. The following are examples of such words: 'tunjuk' + *-el-* = 't*el*unjuk' (index finger), 'gilang' + *-em-* = 'g*em*ilang' (splendid), 'gigi' + *-er-* = 'g*er*igi' (serrated), and 'sambung' + *-in-* = 's*in*ambung' (continue).

Nested affixation occurs in Malay as well. Fortunately, unlike some agglutinative languages, no more than three layers of affixation is allowed in Malay. For example, the stem 'orang' (person) can be prepended with the prefix *se-* to form the word '*se*orang' (alone), followed by the layer *ke-...-an*, resulting in '*kese*orang*an*' (loneliness), and finally the prefix *ber-* to form the word '*berkese*orang*an*' (to suffer from loneliness). Similarly, the word 'kesinambungan' (continuity) can be decomposed to *ke-* + s + *-in-* + ambung + *-an*, in which the stem 'sambung' (continue) undergoes two layers of affixation.

Aside from nested affixation, reduplication is also common to Malay. Reduplication is the process of repeating phonological segments of a word. There are three types of reduplication in Malay, namely full and partial reduplication, and reduplication that results in a certain rhythmic phonetic change (Tatabahasa Dewan, 1993). The first two processes typically produce indefinite plurals and words that convey a sense of resemblance or homogenity, while the latter usually results in words that describe repetitive or continuous actions, heterogenity, and level of intensity or extensiveness. For example, 'pulau-*pulau*' (islands) results from the full duplication of the word 'pulau' (island) while '*se*siku'(triangle/drawing tool) results from the partial reduplication of the word 'siku' (elbow). Partial reduplication is not limited to the front segment of a word, as duplicated phonetic segments can be added to the end of a word as well (e.g. 'berlari-*lari*' and 'kasih-*mengasihi*'). In rhythmic reduplication, the entire stem is repeated but with phonetical changes that can either be a free phonetic change or involve rhythmic vowel and consonant repetition. The following examples illustrate the different types of rhythmic reduplication (Tatabahasa Dewan, 1993).
Vowel reduplication: 'sayur-*mayur*' (vegetables)
Consonant reduplication: 'gunung-*ganang*' (mountains)
Free reduplication: 'saudara-*mara*' (relatives)

1.3 Motivation and Goals

Automated morphological analysis can be incorporated into information retrieval systems as well as grammar and spell checkers. With the increase in the number of computer and internet users in southeast Asia, performance of such systems is becoming increasingly important. According to a 1999 International Data Corporation (IDC) report, internet users in the Asia Pacific region show preference for viewing the World Wide Web in their native language, especially when English is not their native tongue. Nevertheless, a recent check on Google revealed that there is still no option to limit a search query to only webpages written in Malay. Furthermore, while a Malay grammar and spell checker is currently available on Microsoft Word, a quick check showed that it does not catch errors that pertain to word order or incorrectly inflected words.

In this paper, we propose an algorithm that automatically induces a subset of Malay morphology. In particular, this algorithm takes as input a text corpus and produces as output an affix inventory of prefixes and suffixes. This system ignores infixes since they are not productive in modern Malay and their use is limited to a very small subset of words (Tatabahasa Dewan, 1993).

Although Malay is a reduplicative language, word reduplication will be ignored here as well since the goal of this system is to obtain an affix inventory for a highly prefixal language, not to perform a complete morphological analysis of Malay. The proposed algorithm can be used as part of the design of a complete morphological analyzer. Since Malay morphology is similar to that of Indonesian, this algorithm is likely to be portable to Indonesian as well.

2 Related Work

Most of the existing morphological analyzers focus on suffixal languages. With the exception of Schone and Jurafsky (2001), whose work we will describe in Section 2.1, few have considered prefixes, circumfixes, infixes, or languages that are agglutinative or reduplicative. Previous unsupervised morphology induction systems can be divided into two main categories based on whether the goal is to obtain an affix inventory or to perform a more comprehensive morphological analysis.

2.1 Morphological Analysis

Gaussier (1999) uses an inflectional lexicon to analyze derivational morphology. His system automatically induces suffixes by splitting words based on p-similarity, that is words that are similar in exactly the first p characters. Schone and Jurafsky (2000), on the other hand, extract affixes by inserting words into a trie, and observing places in the trie where branching occurs, an approach similar to identifying p-similar words. Using only the 200 most-frequent affixes, they generate a list of pairs of morphological variants (PPMVs). Their system then determines the semantic relationships between word pairs via Latent Semantic Analysis. Word pairs with high semantic correlations form conflation sets. Schone and Jurafsky (2001) extended their semantic-based algorithm to include orthographic and syntactic cues, and applied their algorithm to induce more extensive morphological relationships (prefixes as well as circumfixes) in German, Dutch, and English.

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 55–62
Computer Science Department, Swarthmore College

2.2 Affix Inventories

Brent *et al* (1995), uses a Minimum Description Length approach to obtain suffixes that result in the maximum compression for any given corpus. DéJean (1998) uses an algorithm that exploits the entropy of the next character in a word. His algorithm decomposes a word into stem and suffix when the number of possible characters following a sequence of characters in a word exceeds a certain threshold. Like Brent *et al*, his goal, too, was to obtain an affix inventory using statistical methods.

2.3 Previous Work in Malay Morphology

Very little work on morphology induction has been done in Malay. The most recent work with regard to Malay morphology is an automated stemmer proposed by Tai *et al* (2000) as part of the design of an information retrieval system for Malay. In addition to a set of heuristics, their system is given a list of prefixes and suffixes along with an explicit set of rules under which affixes may be removed from words. Their overall goal is different from ours: Tai *et al* seek an efficient, but highly supervised stemming system, while we seek a minimally-supervised system that is capable of inducing affixation in Malay via semantic-based analysis. The output of our system may be used to eliminate the need for an explicit affix list that is required by their stemming algorithm.

3 Current Approach

We propose to extend Schone and Jurafsky's semantic-based approach to analyzing a highly prefixal, agglutinative language. Like Schone and Jurafsky (2000), our algorithm can be decomposed into four phases, namely (1) building an initial affix inventory, (2) identifying pairs of potential morphological variants (PPMVs), (3) computing semantic correlation of PPMVs, and finally (4) identifying valid affixes by selecting morphological variants with high semantic correlation. We use a text corpus consisting of news articles from an online Malaysian newspaper, and words from an online Malay dictionary.

3.1 Phase 1: Selecting Potential Affixes

In the first phase of analysis, we build two tries via in-order and reverse order insertion of words from the corpus along with their frequencies. Before describing the algorithm that extracts potential prefixes from these tries, we define the following terms:
(1) type count: Each distinct word in the corpus is considered a unique type. Hence, type count refers to the frequency of occurrence of a unique word.
(2) branching factor: When two or more p-similar words are inserted into a trie, branching occurs at the p-th node. For instance, in the reverse trie shown in Figure 2, branching occurs at the fourth node from the root since

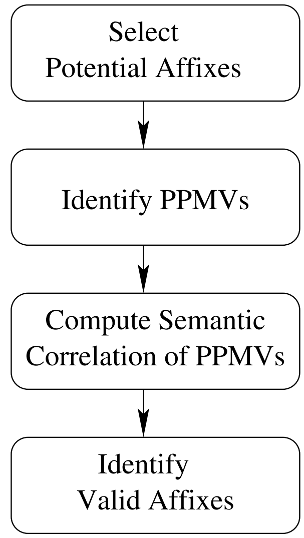

Figure 1: System architecture.

'cuba' (try), 'dicuba' (tried), and 'mencuba' (trying) are similar in exactly the last four characters. The branching factor is the number of branches that hang off a node in the trie. In the previous example, the branching factor at 'c' is 3.

To extract candidate prefixes from the reverse trie, the system needs to identify p-similar words. However, we believe that a constant p value is unsuitable for this task since errorneous splitting points may be proposed. Hence, we try to automatically induce an appropriate p value for different sets of words. To do this, we observe places in the trie where τ, the ratio between the branching factor and the type count is exactly 1. We call these places potential breaking points (PBPs). Splitting words into stem and affix when the τ ratio is 1 gives us an estimate of a suitable p value for any given subtrie.

Once a potential breaking point is identified, each candidate prefix that hangs off that PBP is checked for its overall frequency in the corpus. Only the T most frequent candidate prefixes, as determined by their frequencies in the forward trie, are selected as potential prefixes, and thus, added to the potential prefix inventory.

A reverse selection process is performed to determine potential suffixes. That is, candidate suffixes are identified from PBPs in the forward trie, and depending on their overall frequencies in the reverse trie, the system decides whether or not to add these candidate suffixes to the potential suffix inventory.

3.2 Phase 2: Identifying PPMVs

Pairs of potential morphological variants are constructed from words that descend from the same root node in the trie, share a common PBP, and contain a potential affix

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 55–62
Computer Science Department, Swarthmore College

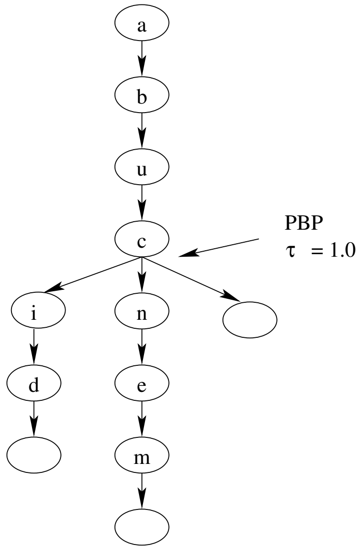

Figure 2: Structure of reverse trie with the words 'dicuba', 'mencuba', and 'cuba' inserted. The empty nodes represent end-of-word markers.

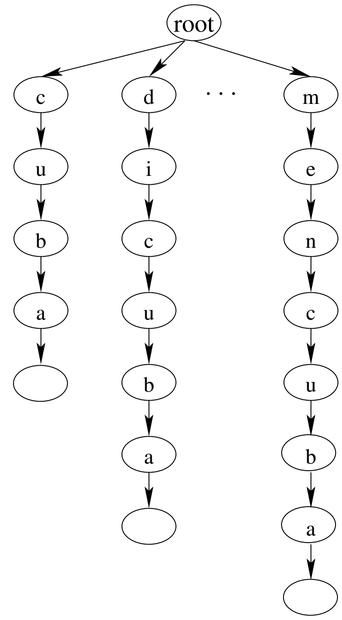

Figure 3: Structure of forward trie with the words 'dicuba', 'mencuba', and 'cuba' inserted.

in the initial inventory. For instance, if *di-*, *men-*, and *NULL* were candidate prefixes that were added to the inventory at the PBP shown in Figure 2, then the pairs of morphological variants would be ('dicuba', 'mencuba'), ('dicuba', 'cuba'), and ('mencuba', 'cuba'). The three affixes $\{$*di-*, *men-*, *NULL*$\}$ form what we call the affix set for the stem 'cuba'. The same construction process is repeated to obtain PPMVs for words containing candidate suffixes.

3.3 Phase 3: Computing Semantic Correlation of PPMVs

Morphologically-related words frequently share similar semantics. Accordingly, we determine the validity of candidate affixes in the potential affix inventory by computing the semantic correlation of the PPMVs. The correlation score of each PPMV gives us an estimate of the validity of the two affixes it contributed to the initial inventory. For this purpose, we construct a co-occurrence vector with a $\pm$ 5-word window for each word in the PPMV using the corpus from Phase 1. We then compute the cosine of the angle between the two vectors using the standard formula:

$$\cos(\vec{v}_1, \vec{v}_2) = \frac{\vec{v}_1 \cdot \vec{v}_2}{|\vec{v}_1||\vec{v}_2|}$$

The dot product is the projection of one vector onto the other, and is thus a measure of the similarity, or more accurately, the co-directionality of two vectors. In view of this, the cosine of the angle between two co-occurrence vectors is commonly used as a measure of the semantic correlation of two words. Ideally, a pair of morphologically-related words would have a large dot product, and thus, a high cosine score.

3.4 Phase 4: Identifying Valid Affixes

Using the cosine scores of the PPMVs computed in the previous phase, we determine the "goodness" and "badness" of each candidate affix in the initial inventory. For every PPMV with a cosine score above the cosine threshold, *C*, we increment the "goodness" of the affixes corresponding to that PPMV by 1. Likewise, for every score below the threshold, the "badness" of the corresponding affixes is incremented by 1. For instance, assuming a cosine threshold of 0.2, the goodness of *di-* and *men-* corresponding to the PPMV ('dicuba', 'mencuba') from Table 1 will be incremented by 1 each. Similarly, the goodness of *men-* and *NULL* is incremented since the pair ('mencuba', 'cuba') has a cosine score greater than the threshold we defined earlier. However, the cosine score for ('dicuba', 'cuba') is below the threshold; consequently, the affixes *di-* and *NULL* will have their

PPMV	Cosine score
('dicuba', 'mencuba')	0.22
('dicuba', 'cuba')	0.19
('mencuba', 'cuba')	0.32

Table 1: Cosine scores of PPMVs formed from the stem 'cuba' and affixes in the set $\{$*di-*, *men-*, *NULL*$\}$.

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 55–62
Computer Science Department, Swarthmore College

badness scores incremented by 1. The goodness and badness scores of each candidate affix in the affix set {*di-*, *men-*, *NULL*} corresponding to the stem 'cuba' are summarized in Table 2.

A new inventory is constructed from candidate affixes in the initial inventory whose goodness scores are greater than or equal to their badness scores. From the previous example, both *di-* and *men-* would be considered valid affixes for the stem 'cuba', and hence, added to the new inventory.

Affix	Goodness	Badness
di-	1	1
men-	2	0
NULL	1	1

Table 2: Scores of affixes in the set {*di-*, *men-*, *NULL*} corresponding to the stem 'cuba'. These scores were obtained using the validity heuristic described in Phase 4.

4 Results

4.1 Prefixes

Before Semantic Analysis: In order to determine a reasonable value for *T*, frequency thresholds were varied between 0 and 500 in increments 5 (with the exception of the interval between 35 and 40 in which *T* was incremented by 1), and proposed affix inventories were evaluated for recall and precision. Figures 4 and 5 summarize the results of this evaluation. Since we valued recall over precision in the initial phase, and did not wish to lose any correctly identified affixes prior to semantic analysis, we fixed *T* at 36. The later phases would serve to increase precision by eliminating incorrectly hypothesized prefixes. Thus, with a *T* value of 36, the system achieved 100% recall, 7.95% precision, and 14.74% on F-measure.

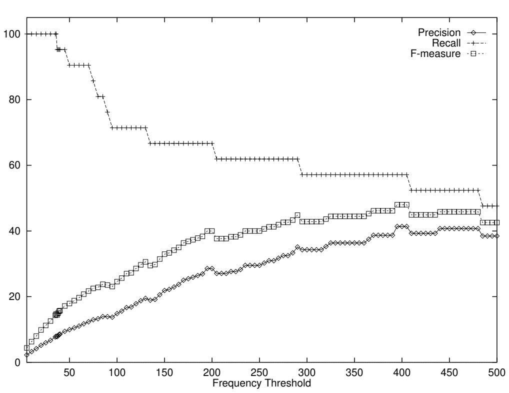

Figure 4: Precision, recall, and F-measure as a function of the frequency threshold, *T*, in the initial phase of prefix identification. Recall is highest for $0 \leq T \leq 36$.

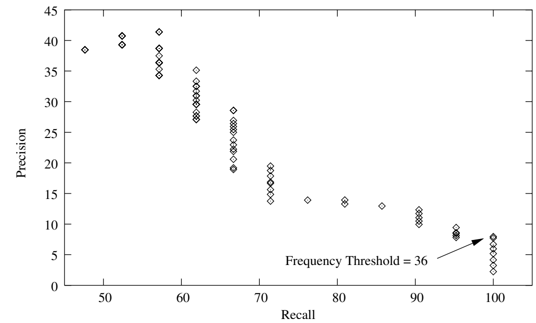

Figure 5: Precision versus recall for prefixes. At 100% recall, precision is highest at a frequency threshold *T* of 36.

After Semantic Analysis: Cosine thresholds were varied between 0.2 and 1.0, and the new prefix inventories were re-evaluated. Figure 6 shows that, at a cosine threshold *C* of 0.45, our system obtained 150.62% relative increase in precision but suffered 19.05% relative decrease in recall. The F-measure climbed 170% after semantic analysis.

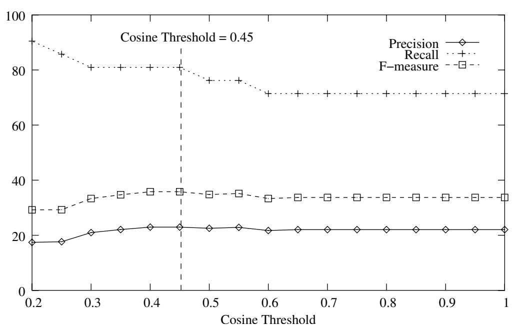

Figure 6: Precision, recall, and F-measure as a function of the cosine threshold, *C*, in prefix identification. F-measure is highest at *C* = 0.45.

4.2 Suffixes

Before Semantic Analysis: As a consequence of the low precision in prefix identification, our system did not attempt to remove prefixes from words in the corpus before they were reinserted into the tries for potential suffix selection, as suggested by Schone and Jurafsky (2001). To identify candidates for the initial suffix inventory, we employed the method described for prefixes, that is, we varied the value of the frequency threshold *T* between 0 and 2000 in increments of 5, and evaluated the proposed inventories for recall and precision. Figure 7 shows the evaluation results. The system achieved 100% recall, 60% precision, and 75% on F-measure for *T* values between 1500 and 1745.

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 55–62
Computer Science Department, Swarthmore College

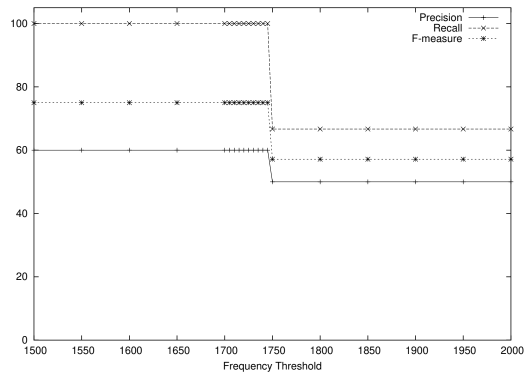

Figure 7: Precision, recall, and F-measure as a function of the frequency threshold, T, in the initial phase of suffix identification. At 100% recall, precision is highest for $1500 \leq T \leq 1745$.

After Semantic Analysis: The new suffix inventories that were obtained with cosine thresholds varied between 0.2 and 1.0 were re-evaluated as described in section 4.1. At a cosine threshold of 0.65 (see Figure 8), the system was able to achieve 80% precision and 10.71% increase in F-measure while maintaining recall at 100%. The identified suffixes were *-i*, *-an*, *-kan*, and *-a*. Of these, the first three were correct.

Table 3 provides a summary of the precision, recall and unweighted F-scores obtained by the system before and after semantic analysis.

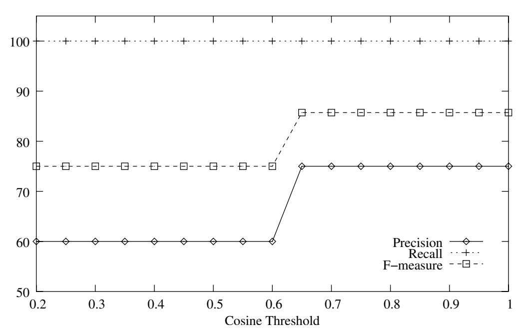

Figure 8: Precision, recall, and F-measure as a function of the cosine threshold, C, in suffix identification. F-measure is highest for $0.65 \leq C \leq 1.0$.

5 Discussion

5.1 Corpus Size

The results from prefix identification were rather disappointing. However, we believe that this is not a shortcoming of our algorithm, but rather of the training corpus itself. While it is typically easy to find unlabelled corpora in English, building a large corpus in Malay proved rather difficult. Our corpus contains just over 22,000 unique words, compiled from two online sources: a Malaysian newspaper and a Malay dictionary. Since English is widely spoken in Malaysia, English words frequently find their way into Malaysian news articles, and thus our corpus. Extending the news corpus to include entries from the dictionary increased the number of prefixes found in the initial phase, but doing so presented a significant problem in the semantic analysis phase because many of the words from the dictionary were not found in the news articles.

		Prefix	Suffix
Recall	Before	100.0	100.0
	After	80.95	100.0
Precision	Before	7.95	22.97
	After	22.97	75.0
F-measure	Before	14.74	75.0
	After	35.70	85.71

Table 3: Summary of precision, recall, and unweighted F-measure before and after semantic analysis.

Although the results from suffix identification could improve with a larger corpus as well, the size of the training corpus was not an issue in the case of suffixes. This is because the size of the corpus relative to the number of suffixes the system had to identify was approximately 7,000 to 1, while the ratio was only 1,000 to 1 in the case of prefixes.

5.2 Potential Issues with Validity Heuristic

While the validity heuristic employed in Phase 4 generally works well, it has potential issues that are not addressed in current work. A problem arises when there are many spurious PPMVs associated with a given stem. Consider the addition of *s-* to the affix set {*di-*, *men-*, *NULL*} in our earlier example. We present the corresponding PPMVs and their cosine scores in Table 4.

PPMV	Cosine score
('dicuba', 'mencuba')	0.22
('dicuba', 'cuba')	0.19
('mencuba', 'cuba')	0.32
('dicuba', 'scuba')	< 0.01
('mencuba', 'scuba')	< 0.01
('cuba', 'scuba')	< 0.01

Table 4: Cosine scores of PPMVs constructed from the affix set {*di-*, *men-*, *NULL*, *s-*} and the stem 'cuba'.

As mentioned earlier, our corpus has the potential of containing English words, thus there is a chance that a

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 55–62
Computer Science Department, Swarthmore College

word like 'scuba' may appear in the training corpus. Because of the presence of spurious PPMVs (due to the addition of *s*-), the hypothesized badness of each affix in the original set {*di*-, *men*-, *NULL*} has increased. In Table 5, we list the new goodness and badness scores with the addition of candidate prefix *s*-.

Affix	Goodness	Badness
di-	1	2
men-	2	1
NULL	1	2
s-	0	3

Table 5: Validity scores of candidate affixes in {*di*-, *men*-, *NULL*, *s*-}, assuming a cosine threshold of 0.2.

Although few instances like this were encountered in our current work, it is conceivable that such a problem would be significantly detrimental to our system's performance, especially in future work when a larger corpus is used. A more robust solution would be to compute the goodness and badness of each candidate in the affix set, remove any affix with a goodness score of 0, and then recompute the validity of each affix in that set by decrementing each of their badness scores by 1.

With *s*- removed, and the badness scores recomputed, the validity of *di*-, *men*-, and *NULL* would be restored to their original values as shown in Table 2.

This method of determining affix validity suffers from another drawback in that it would incorrectly identify affixes as invalid if there is a partition within an affix set associated with a given stem. A partition exists in an affix set if the PPMVs that are constructed from those affixes belong to two disjoint, morphologically-unrelated sets. Although we did not find an example like this in the Malay corpus, such a phenomenon occurs in languages like French. Consider the two verbs 'fonder' and 'fondre' whose simple past forms are {'fondai', 'fonda'} and {'fondis', 'fondit'} respectively. On seeing these four inflected words, our system would propose {*-ai*, *-a*, *-is*, *-it*} as the affixes associated with the stem 'fond'. The problem arises from the fact that the four words 'fondai', 'fonda', 'fondis', and 'fondit' belong to two morphologically-unrelated sets. Consequently, our validity heuristic would propose the scores shown in Table 6. Since none of the affixes have goodness scores greater than or equal to their badness scores, all of them would be erroneously discarded by our algorithm.

Fortunately, this phenomenon rarely occurs in most languages. Even in cases where it does, it is highly likely that the affixes, which are mistakenly discarded by the system, would be associated with other stems that do not suffer from the same problem.

Affix	Goodness	Badness
-ai	1	2
-a	1	2
-is	1	2
-it	1	2

Table 6: Validity scores of suffixes from the example in French.

5.3 Unsupervised Selection of Thresholds

Although the values of the frequency and cosine thresholds in this experiment were hand-picked to obtain the best results, these values can be obtained automatically. The following is a potential algorithm for doing so:

(1) Set the frequency threshold T to a reasonably small number, say, 5, in order to eliminate potential typos as well as the possibility of foreign words in the corpus.
(2) Run Phase 1 with $T = 5$ to obtain an initial affix inventory, I.
(3) Build a vocabulary of all distinct words in the corpus. Attach each affix $a \in I$ to each word in the corpus. Check if we still have a valid word in the vocabulary. If we do, add a to the new inventory, I'.
(4) Next, run Phase 2 for each affix in I'.
(5) Now, run Phase 3 with varying cosine thresholds, starting at 0. With each different threshold, check to see if we have lost any affix in I'. Increase the threshold as long as we have 100% recall on the affixes in I'. Save the cosine threshold C' prior to the drop in recall on I'.

C' should give us a good estimate of the optimal cosine threshold for the initial inventory I. Since I' is a subset of I, we are guaranteed that recall on the affixes in I would drop before C'. Having estimated the value of the cosine threshold, we can now return to running Phase 2 with I, and Phases 3 and 4 with a cosine threshold of C'.

6 Conclusion

Despite relatively disappointing results, we are confident that this algorithm would be more successful on a larger corpus. Being one of the first systems built to analyze Malay affixation, this system shows promise of analyzing highly prefixal languages. More importantly, this system provides a starting point for future work in Malay morphological analysis.

Acknowledgements

Many thanks to Richard Wicentowski and five anonymous reviewers for useful suggestions and comments on a first version of this paper.

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 55–62
Computer Science Department, Swarthmore College

References

Michael R. Brent, Sreerama K. Murthy, Andrew Lundberg. 1995. Discovering Morphemic Suffixes: A Case Study In MDL Induction. *Proceedings of 5th International Workshop on Artificial Intelligence and Statistics*.

Hervé DéJean. 1998. Morphemes as Necessary Concept for Structures Discovery from Untagged Corpora. *Workshop on Paradigms and Grounding in Natural Language Learning*.

Ethnologue Report for Language Code: MLI. 2003. Ethnologue. *http://www.ethnologue.com/*.

Éric Gaussier. 1999. Unsupervised Learning of Derivational Morphology from Inflectional Lexicons. *ACL '99 Workshop Proceedings: Unsupervised Learning in Natural Language Processing*, University of Maryland.

Asia/Pacific's Internet Users Demand Localized Web Content, IDC Finds. 1999. Techmall. *http://www8.techmall.com/techdocs/TS991104-4.html*

Nik S. Karim, Farid M. Onn, Hashim Hj. Musa, Abdul H. Mahmood. 1993. *Tatabahasa Dewan* Edisi Baharu. Dewan Bahasa dan Pustaka, Kuala Lumpur, Malaysia.

Patrick Schone, Daniel Jurafsky. 2000. Knowledge-Free Induction of Morphology Using Latent Semantic Analysis. *Proceedings of the Computational Natural Language Learning Conference*.

Patrick Schone, Daniel Jurafsky. 2001. Knowledge-Free Induction of Inflectional Morphologies. *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Sock Y. Tai, Cheng S. Ong, Noor A. Abdullah. 2000. On Designing an Automated Malaysian Stemmer for the Malay language. *Proceedings of the 5th International Workshop International Retrieval with Asian Languages*.

Appendix: Affix List

Affix Type	Nouns	Verbs	Adjectives
Prefix	pe- pem- pen- peng- penge- pel- per- ke- juru-	me- mem- men- meng- menge- memper- di- diper- bel- ber- ter-	ter- se-
Suffix	-an	-kan -i	
Prefix and Suffix Combination	pe-...-an pem-...-an pen-...-an peng-...-an penge-...-an per-...-an pel-...-an ke-...-an	me-...-kan mem-...-kan men-...-kan meng-...-kan menge-...-kan me-...-i mem-...-i men-...-i meng-...-i menge-...-i memper-...-kan memper-...-i di-...-kan di-...-i diper-...-kan diper-...-i ber-...-kan ber-...-an ke-...-an	ke-...-an
Infix	-el- -er- -em-		-el- -er- -em- -in-

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 55–62
Computer Science Department, Swarthmore College