EXHAUSTIVE AFFIX STRIPPING AND A MALAY WORD REGISTER TO SOLVE STEMMING ERRORS AND AMBIGUITY PROBLEM IN MALAY STEMMERS

Salhana Amad Darwis[1], Rukaini Abdullah[2], Norisma Idris[3]
Department of Artificial Intelligence
[2,3]Faculty of Computer Science and Information Technology
University of Malaya, 50603 Kuala Lumpur, Malaysia.
Email : [1]salha81@yahoo.com; [2]rukaini@um.edu.my; [3]norisma@um.edu.my;

ABSTRACT

Stemmers or word stemming algorithms reduce a derivative word to its root word by removing all the affixes. The complexity of Malay Language (ML) morphological rules and Malay lexicon make stemming Malay words difficult. There is no fixed method to determine the affix to be removed from a derivative word to produce the correct root word. Furthermore, a derivative word could contain one or more valid root words. Stemming errors still exist in the previous Malay Language Stemmers (MLS). Regardless of the approaches used, they rely on the first affix matched or the first root word found. Hence, some words were under stemmed or over stemmed while words with many valid root words were not stemmed to reveal the correct root word. This multiple root words or ambiguity problem, however, has never been addressed by previous MLS. To solve the over stemming and under stemming errors, we propose an approach that exhaustively strips all matched affixes to ensure that a valid root word will be extracted. In addition, we also propose the use of a Malay Word Register to address the ambiguity problem of determining the correct root word. We tested the proposed approach with words from newspaper articles, Malay translation of the Quran, History essays and incorrectly stemmed words from the previous stemmers. The results reveal this stemmer is successful with 99.8% accuracy. There were no stemming errors. The imperfect accuracy is due to the ambiguity problem approach.

Keywords: Malay language, stemming, Malay Language stemmers, Malay word register, ambiguity problem, under stemming, over stemming

1.0 INTRODUCTION

Stemmers or word stemming algorithms reduce a derivative word to its root word by removing all the prefixes, suffixes and confixes. The size of documents for processing is also reduced since only words with similar concepts are processed irrespective of their forms. Thus, applications like automatic essay grading systems, text summarization, text translations, document categorization and cross language information retrieval use stemming as their pre-processing steps to increase their efficiency [1][2][3][4][5].

Stemming algorithms have been developed years ago to cater for different languages across the globe. The role or purpose of the stemmers and the differences in the linguistic aspects are the reasons why stemming algorithms are developed specifically for certain languages.

The earliest work on English language stemmer was reported by Lovins in 1968 [6] followed by Dawson in 1974 [7] and Porter in 1980 [8]. Since the stemmers were designed to cater for different purposes, there were differences in the stemming methods used. For example, Porter stemmer used partial matching concept to reduce the complexity of the stemming algorithm although the stemmer may not necessarily produce a root word [8]. It may be suitable for certain information retrieval applications as long as it can map related words to the same stem. Lovins and Dawson stemmers on the other hand aimed for a more generic usage suitable for any computational linguistics field by finding the linguistic root of the derivative word. Hence, morphological rule based method is used in their stemmers to ensure the stemming process follow proper morphological rules. This method involves the use of dictionaries, spelling exception rules, order class iteration, longest and shortest match and recoding technique to validate the morphological rules [6],[7]. The results were reliable since each step performed in the stemming process resembles the method as advocated by linguists.

Besides English, there are also stemmers developed for other languages such as Spanish [9] and Bengal [10]. However, stemming Bengal words is challenging since Bengal language has poor linguistic rules due to improper documentation of the rules. As a conclusion, the selection of stemming method in a particular language is influenced by a few factors such as the characters and morphological structure of the language, the purpose of stemming, and the suitability of techniques on a particular language.

The remainder of this paper is organized as follows: Section 2 describes the Malay morphological rules and stemming difficulties followed by previous work on Malay stemmers. Section 3 explains the structure of the proposed Malay stemmer, section 4 presents the results and Section 5 concludes the paper.

2.0 MALAY STEMMER

2.1 The Malay Morphological Rules

Malay language (ML), a national language of Malaysia, is a major language in the Austronesian family language. It is spoken in several countries in South East Asia such as Malaysia, Singapore, Indonesia and Brunei as a communication medium as well as an official language. Malay morphological rules consist of the list of affixes in Malay language and methods to attach the affixes to root words to form derivative words. There are 3 types of affixes in ML [11]. They are:

a. **The Prefix, '*Awalan'*** – prefixes are words attached to the beginning or left of a root word to form a meaningful derivative word. The prefix type will determine the part of speech of the derivative word such as noun, verb or adjective which represents the role of the derivative word. Some rules need to be adhered to during the construction of the derivative words. For instance, some prefixes can only be appended to root words that begin with certain alphabets. This is known as the first character validation rule. Another rule named as recoding requires the removal of the root word first character and replacement of additional letter to the prefix during words construction.

b. **The Suffix, *'Akhiran'*** - suffixes are words attached at the end of a root word during words construction. There are three categories of suffixes. Derivational suffixes produce derivative words with different part of speech from the root word. For instance, the verb '*makan*' (eat) will become a noun when attached to '*an*' in '*makan+an'* (food). Another group of suffixes which gives emphasis to the word are known as particle suffix or *'kata penegas'*. An example would be '*jangan+lah*' (don't!) The last type of suffix when attached to a root word will show possession such as '*buku+nya'* (its book). Similar concepts of suffixes have been introduced in Indonesian stemmers [2, 3].

c. **The Confix** - confixes refer to the combination of prefixes and suffixes that could be attached to a root word.

The older rules of Malay word construction had included infix. However, Malay linguists have decided the use of infix is impractical and have now considered and accepted words with infixes as valid root words [11]. An example of a word with infix but now considered as a root word is '*telapak*'(palm). It was previously derived from '*tapak*' (palm) + infix '*el*' as in '*t+el+apak*'.

.

2.2 The Complexity in Malay Morphological Rules : Stemming Malay words

In ML, a root word can be derived to form another word known as derivative word by attaching the root word to a prefix, suffix or confix. The morphological structure of Malay derivative words can be summarized as follows:

a) [Prefix] $_+$ Root Word

b) [Multiple Prefix] + Root Word

c) [Prefix] + Root Word + [Suffix]

d) [Prefix] + Root Word + [Multiple Suffix]

e) [Multiple Prefix] + Root Word + [Suffix]

f) [Multiple Prefix] + Root Word + [Multiple Suffix]

g) Root Word + [Suffix]

h) Root Word + [Multiple Suffix]

Stemming Malay words is difficult due to the complexity of the morphological rules and Malay lexicon. There is no specific rule to determine which affix *s*hould be removed first or in which order they should be removed to reveal the correct root word even though the derivative word forms are similar. A derivative word that contains a root word which could be stemmed further to another root word adds to this stemming difficulty. Examples are shown in Table 1.

Table 1 : Stemming difficulties

Similar word forms : Derivative word to be stemmed	Possible detachment during stemming	Result after affix removal
dirasmi (officiated)	1. *di+ rasmi* 2. *dirasm + i* 3. *di + rasm+i*	1 *rasmi (*official*)* : root word 2 *dirasm* : invalid root word 3 *rasm* : invalid root word
didiami (inhabited)	1. *di + diami* 2. *didiam +i* 3. *di + diam + i*	1. *diami*: invalid root word 2. *didiam* : invalid root word 3. *diam* (silent) : root word
digilai(adored)	1. *di-gilai* 2. *di+gila+i*	1. *gilai* : invalid root word 2. *gila* (crazy): root word
dinilai(evaluated)	1. *di+nilai* 2. *di+nila+i*	1. *nilai* (value) : root word 2. *nila* (blue dye) : wrong root word
perundingan (consultancy)	1. *pe+runding+an* 2. *per+unding+an*	1. *runding* (consult) : root word 2. *unding* : invalid root word
perundangan (legislation)	1. *pe+rundang+an* 2. *per+undang+an*	1. *rundang* : invalid root word 2. *undang* (law) : root word
menyapu (sweep)	1. *meny+[s]+apu*	1. *sapu* (sweep) : *root word*
menyanyi (sing)	1. *meny+[s]+anyi*	2. *sanyi* : invalid root word
A root word within a root word : *menyekolahkan* (send to school)	1. *meny+[s]ekolah+kan* 2. *meny+se+kolah+kan*	1. *sekolah* (school) : root word 2. *kolah* (tub) : root word
penyakit (disease)	1. *penyakit* 2. *peny +[s]+akit*	1. *penyakit* (disease) : root word 2 *sakit* (pain) : root word
*mereka(*they*)*	1. *mereka* 2. *me+reka*	1. *mereka (*they*) :* root word 2. *reka (*create*) :* root word

In addition, complex morphological rules also contribute to the ambiguity problem where more than one root word can be produced as a result of stemming as shown in Table 2 below.

Table 2 : Stemming difficulties :ambiguity problem : selecting from multiple root words

Derivative word to be stemmed	**Detachment during stemming**	**Result after affix removal**
perangkaan (statistics)	1. *pe + rangka + an* 2. *per + angka + an*	1. *rangka* (skeleton) : root word 2. *angka* (numbers) : the correct root word
perancangan (plan)	*1. pe+rancang+an* *2. per+ancang+an*	1. *rancang* (plans): correct root word 2. *ancang*(preparation) : root word

2.3 Previous Works on Malay Stemmers

A few stemmers have been developed for the ML such as Othman's [12], Sembok's [12][13], Idris' [14] and Yasukawa's [15]. The details of each stemmer are as follows:

2.3.1 Othman's Stemmer

As the pioneer in Malay stemming, Othman introduced a basic design to stem derivative words. This stemmer was meant for general text and applications in Malay language. It adopted the rule-based approach using 121 affixes consisting of prefixes, infixes and prefix-suffix pairs as follows [12]:

a) Prefix rules format : Prefix+, e.g. *ber+*
b) Suffix rules format: +suffix, e.g. *+kan*
c) Infix rules format: +infix+, e.g. *+el+*
d) Prefix-Suffix pair rules format: prefix + suffix, e.g. *di+kan*

Some of the limitations of this first stemmer are :

1) Over stemming and under stemming errors.
   i. The obvious limitation is that this stemmer does not have a dictionary check as the first step in the stemming process. Due to this, a derivative word that is already a root word is unnecessarily stemmed which leads to over stemming. For instance, the word '*mati'*(die), is over stemmed to '*mat+i*', '*masalah'*(problem) to '*masa+lah'*(time is) while '*sekolah'*(school) is stemmed as '*se+ko+lah'*, where '*ko'* is incorrectly produced as the root word. In some cases, the stemmer will not return a root word if a matched affix is not found [12]. There were also cases of under stemmed words such as '*peringatan*'(reminder). Instead of '*ingat*'(remember), it was incorrectly stemmed to '*peringat*'.
   ii. Another contributor to these errors is that this stemmer is dependent on the order of affixes presented which is prefix-suffix (ps), prefix (pr), suffix (su) and infix (in) respectively. This fixed order of affixes does not guarantee a correct result for derivative words with similar forms as shown in the examples in Table 1. Additionally, there was no clear justification why the affixes were arranged in such order.

2) Ambiguity problem.
   The instance when a matched affix is removed from the derivative word and the root word exists in the global or stem dictionary, this algorithm stops and returns the root word immediately which may not be the correct root word. The words '*perangkaan*' and '*perancangan*' in Table 2 illustrate this problem clearly. The algorithm does not check other possible affix to be removed from the derivative word. This means the stemmer could not identify other root words that could be extracted from the derivative word. For derivative words with multiple or ambiguous root words, the stemmer has to select the correct root word. This ambiguity problem was not addressed by this stemmer.

2.3.2 Sembok's Stemmer

Sembok's improvements [12] on Othman's stemmer include :

1) Adding the stem dictionary look-up step at the start of the algorithm to avoid stemming a word that is already a root word which could produce over stemming errors.
2) Adding more lists of affixes such as Set A consisting of Othman's original collection, Set B that contains a list of affixes that exist in the Quran and Set C as the collection of the Modern Malay list of affixes.
3) Arranging the affixes in the input file in this order of Set A, B and C where he believes plays an important role in determining the accuracy of the stemmer. To determine which order of the affixes gives the best result, some experiments involving different orders of affixes were conducted as follows:
   Test1: pr-ps-su-in;
   Test2: pr-su-ps-in;
   Test3: ps-pr-su-in;
   Test4: ps-su-pr-in;
   Test5: su-pr-ps-in;
   Test6: su-ps-pr-in; where pr = prefix; ps = prefix + suffix; su = suffix ; in =infix

With these three improvements, Sembok's stemmer solved some problems in Othman's. However, over stemming and under stemming errors still exist. The fixed order of affixes did not always produce correct root words. For example, '*Mem+i'*, '*Me+i', 'i'*. If the former is chosen first, then the stemmer will produce a correct root word, but if the latter is selected first, an incorrect root word will be produced. So, there is still no solution on which affix should be given priority to ensure a correct root word is produced each time. The same problem will occur with other ambiguous words such as '*pengakuan'*(confession) which can be stemmed to *'aku'*(me) *or 'kaku'*(stiff), and *'memalingkan'*(turn away) which can be stemmed to *'paling'*(turn) or '*maling*'(thief).

Sembok claimed that most of the remaining errors were due to the precise order in which to apply the different classes of affix. To find the best order was still a problem to them at that time and it was left for future research. This is however not the sole reason why an incorrect root word was produced by this stemmer. Similar to Othman's stemmer, Sembok's algorithm does not check for other possible combinations of affix that may return other root words which then requires a method to choose the correct one. Hence this stemmer still has under stemming and over stemming errors and unsolved ambiguity problem.

2.3.3 Idris' Stemmer

This stemmer was developed for an automated essay grading system for History Malay text [14]. The basic features in this stemmer are quite similar to Sembok's stemmer and the expected output is to produce a root word. However, in Idris' stemmer, modifications were done to suit the History text. The stemmer also changed the order of the affixes to be processed and include them as steps in the algorithm instead of looping through the affix input file. This stemmer improved recall in automated History essay grading system since the concept of local dictionary was introduced to cater for special terms such as place, country, event, and person's name. Despite these improvements, problems still exist. In terms of solving under stemming and over stemming errors, Idris' stemmer does not improve much from the previous Malay stemmers. Even though the prefix and suffix checking is done in the algorithm rather than running through the affix list in the input files as in Sembok's, the same issues of deciding the best order of the prefix and suffix that need to be followed and which affix should be applied first were still unanswered.

Similar to previous stemmers, Idris's stemmer stops and returns a root word after the matched affix is found. However, unlike previous stemmers, in cases where the root word is not found after removing the matched affix, the algorithm does not loop back to search the next possible matched affix. In this case, it may not return any root word in the end.

Again similar to others, the algorithm does not check exhaustively for other possible combinations of affix that may be removed to return other root words. In conclusion, Idris' stemmer still has stemming errors and the ambiguity problem was also not addressed.

2.3.4 Sembok's RFO Stemmer

To minimize the stemming errors, Sembok included the Rules Frequency Order, RFO, in his new stemmer in 2005 [13]. He found that by arranging the prefix and suffix order according to the most frequently occurred affix in the sample data, most of the over stemming and under stemming errors could be solved. Based on the experiments conducted using the Malay translation of the two chapters of the Quran, the stemming accuracy increased to 98.6%. The improvements of this new stemmer involve :

1. Adding more affix rules to the original set including the spelling variation and recoding rules.
2. Determining the best order of the affixes.
    i. The list of affix is no longer arranged according to the original sequence of *'pr', 'su', 'ps'* and *'in'* as done in Test 1 to 7 in Sembok 1996 stemmer [12].
    ii. For the first pass in the algorithm, the affixes must be sorted in alphabetical order.
    iii. For the second pass, the affixes are arranged according to the frequency of occurrences of each rule in the data set. Hence, affixes frequently found in the sample data were given priority to be removed first.

Despite those improvements, this stemmer still does not address the ambiguity problem because after removing the matched affix, the algorithm stops, return the root word and does not check for other possible combination of affixes that could be removed to reveal other root words.

2.3.5 Yasukawa's Stemmer

This stemmer is for an automated text categorization in Malay language [15]. It uses a derivative word dictionary, stem word dictionary and two affix lists. The derivative word dictionary contains derivative words from the Microsoft Malay Spell Checker. Initial check of this dictionary is done before the affix removal process starts. This is to ensure that a derivative word which exists in the derivative dictionary does not get stemmed for the purpose of word classification. Hence the algorithm was designed as not to necessarily return a root word.

The affixes were sorted according to the longest match and shortest match list. If no root word is found after removing the affix found in the longest match list, the stemmer will then consider the removal of the affix in the shortest match list. The following are limitations found in this stemmer.

i) The ambiguity problem is still not addressed. The algorithm is dependent on the affix order similar to previous stemmers. Although it considers two different affix lists, longest match and shortest match, it stops after the stemmed word is found in the stem dictionary. It does not check other possible combinations of affix that may return other root words.
ii) This algorithm is more suitable for text categorization applications which do not necessarily need a root word, as long as the classification of the given input is found. This is because a derivative word which exists in the derivative word dictionary will not be stemmed. The algorithm will return a value even though the word is not yet checked against the stem dictionary to confirm if it is a root word.

In summary, stemming errors still exist in previous Malay Language Stemmers. Regardless of the approach used, they rely on the first affix matched or the first root word found. Hence some words were under or over stemmed. As such they did not encounter derivative words that could be stemmed to multiple root words. To eliminate the under-stemming and over-stemming errors, we propose to exhaustively apply all relevant affixes instead of stopping at the first affix matched or first root word found which could result in multiple or ambiguous root words. To address the ambiguity problem where the selection of the correct root word is required, we propose the use of a Malay Word Register.

3.0 STRUCTURE OF THE PROPOSED STEMMER

This stemmer uses a rule based approach to produce a linguistic root word. Effective methods and techniques from previous stemmers are collected and applied. However, there are two new elements included in the proposed structure. One enables the exhaustive affix stripping approach to be implemented and the other introduces the use of a Malay Word Register. The proposed stemmer consists of 3 components: the input files, **i*n***, the process engine, **p*n*** and the output files, **o*n,*** as illustrated in Fig. 1. The following subsections describe the components and functionality.

3.1 Input files

The input files are further classified into 3 categories as follows:

i. **Derivative words to be stemmed (i1)** – This file stores the derivative words to be stemmed by the process, p1, to reveal its root word.

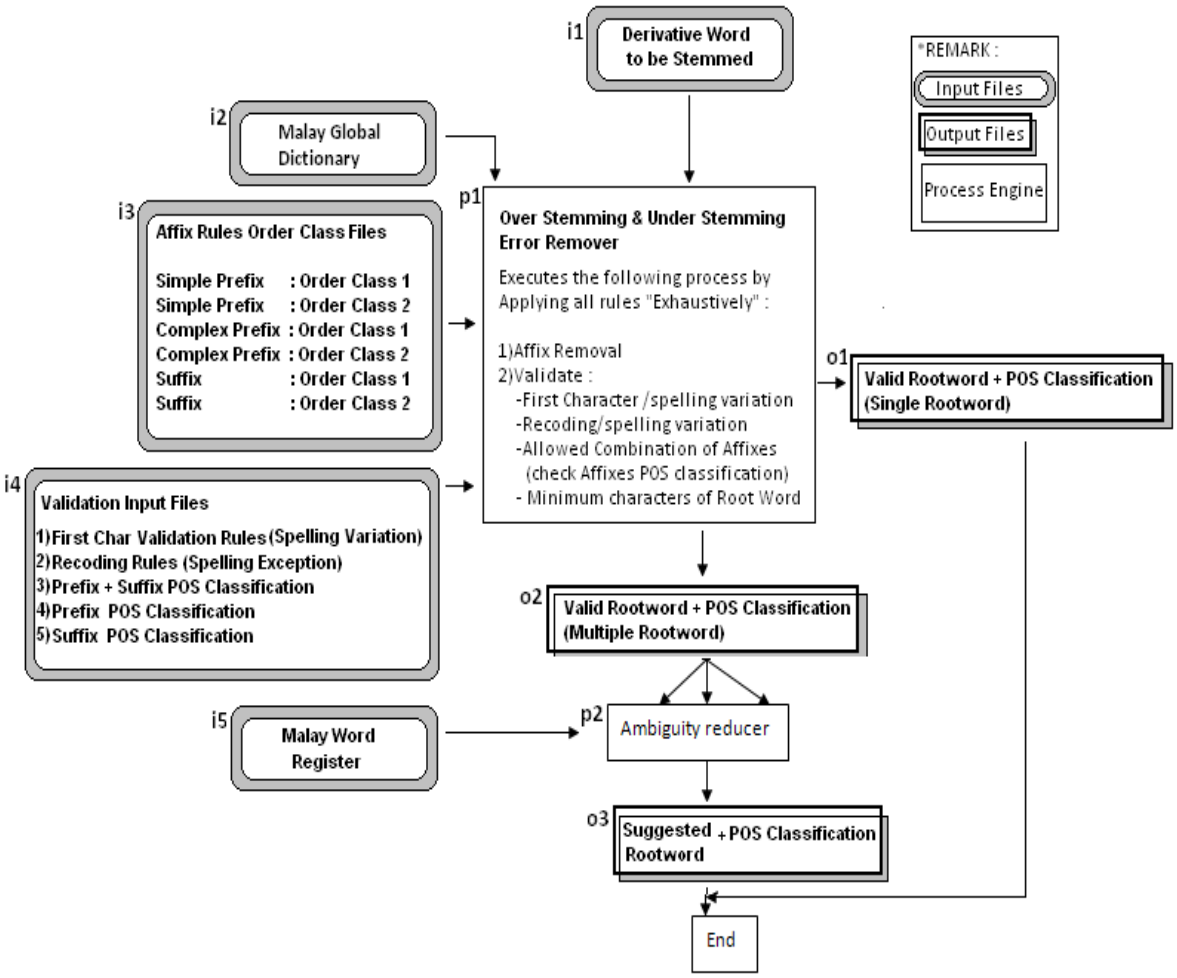

Fig. 1: Structure of the Proposed Stemmer

ii. **The Corpus (i2 and i5)** – There are two types of corpus used in the proposed stemmer :

- **Malay Global Dictionary, *i2***, '*Kamus Dewan'*, [16] - contains all root word entries defined for Malay language. As used in the previous English and Malay stemmers, the global dictionary is required to check if a root word is produced after the removal of an affix.
- **Malay Word Register, *i5,* '***Daftar Kata B. Melayu'*,** [17] - is a collection of root words and their respective derived words. The root words and their derivative words were compiled by linguists from the *Dewan Bahasa Pustaka*, DBP (The Institute of Language and Literature) according to current usage and practicality in the Malay Language. Grammatically, all words can be combined with any affix as long as they adhere to the morphological rules. However not all of the combination will form a derivative word that is acceptable in the current usage of the language. Fig. 1: i5 illustrates where the word register is used in the stemmer. In this work, this input file is developed to contain sample entries as listed in the Malay Word Register. An example of how the pair of root words and their respective derivative words is stored in this input file is shown in the box labeled as i5 in Fig. 2. Thus, the word register, *i5* is used as the final reference to select the correct root word from the many root words extracted from a derivative word. The approach of using a Malay word register to address the ambiguity problem has never been applied in any of the previous Malay stemmers.

iii. **Morphological Validation rules files (i3 and i4)** - It is a crucial component used by the process, *p1* to perform validation as shown in Fig. 1. This is divided into two sub types labeled as Affix order classes ,*i3* and Validation input files, *i4* as described below :

- **Affix order classes, i3**: There are six different classes. Two represent the simple prefix. Another two refer to the complex prefix and the last two classes contain the suffix. Complex prefix order classes consist of prefixes that require further validation before their removal from a derivative word unlike simple prefix. The different order classes are important to ensure affixes that are extensions of others will not be in the same class. For example, the prefixes *'Pe+'* and *'Pen+'* are placed in separate classes. The segregation of affixes according to order classes enables the implementation of the exhaustive affix stripping approach. The order class concept was implemented in the English stemmer but not in the previous Malay stemmers.

- **Validation input files, i4**: There are a total of 5 files that contain these validation rules: first character validation rules, recoding/spelling exception rules, prefix and its part-of-speech (POS) category, suffix and its POS category and prefix+suffix with its POS category. The rules ensure the root word and the removed affix are valid according to the Malay morphological rules. The validation is done immediately after the removal of affix and the resulting root word is found in the global dictionary. It is important to ensure only valid root words are considered as the correct root word before the stemmer returns the final result. The following explains the validation rules input files.

  i. **First Character Validation (spelling variation):** This input file contains a list of affixes and allowable first character of the resulting root word. This ensures the morphological rule where some prefix can only be attached to root words that start with certain letters is adhered to in order to produce a valid root word. For example the prefix '*me+*' can only be attached to words that start with the letters m, n, ng, ny, r, l, w or y.

  ii. **Recoding (spelling exception):** This input file contains a list of recoding rules also known as spelling exception rules as defined by Malay morphological rules. The recoding rules require the root word spelling adjustment depending on the removed affix. This process is important to restore the original spelling of the root word which may change during word construction. For example, the root word for *'mengetawakan'*(laughed at) is *'ketawa'* (laugh) as the recoding rules require that 'k' be restored after removing '*meng'* and *'kan'* from *'meng+etawa+kan'* to produce '*[k]+etawa'*.

  iii. **Suffix, Prefix and Prefix + suffix POS classification files:** These input files contain lists of affixes (Suffix , prefix, and prefix + suffix) and their part of speech (POS) as defined in the Malay morphological rules. For example, '*men+*' is a Verb-prefix while '*ke + an'* is a Noun-confix. This means the derivative word that contains those prefix and confix is a verb and a noun respectively. For example : '*men+didik*' (educate) is a verb and '*ke+pasti+an'*(certainty) is a noun. After the removal of the affix, if the root word exists in the global dictionary and the POS of the removed affix is found, then the stemmer will indicate that the affix combination of confix like '*ke + an*' is valid and hence the resulting root word is also valid. On the other hand, if the POS is not found for an affix combination such as '*men+an'*, it indicates that the affix combination is not defined in Malay morphological rules. Hence, the resulting root word will be indicated as invalid. This step is useful in two ways. Firstly, to ensure the resulting root word is valid. Secondly, the stemmer can determine the part of speech of the derivative root word before stemming which may be useful if semantic information of the derivative word is needed in text processing applications.

The morphological validation components have been used in previous Malay stemmers except for the validation of minimum characters of a root word and valid combination of affixes which were used in the Indonesian's CS stemmer [18][19]. In this proposed stemmer, all these rules are incorporated to ensure complete morphological rule validation. The samples of these input files are shown in Fig. 2.

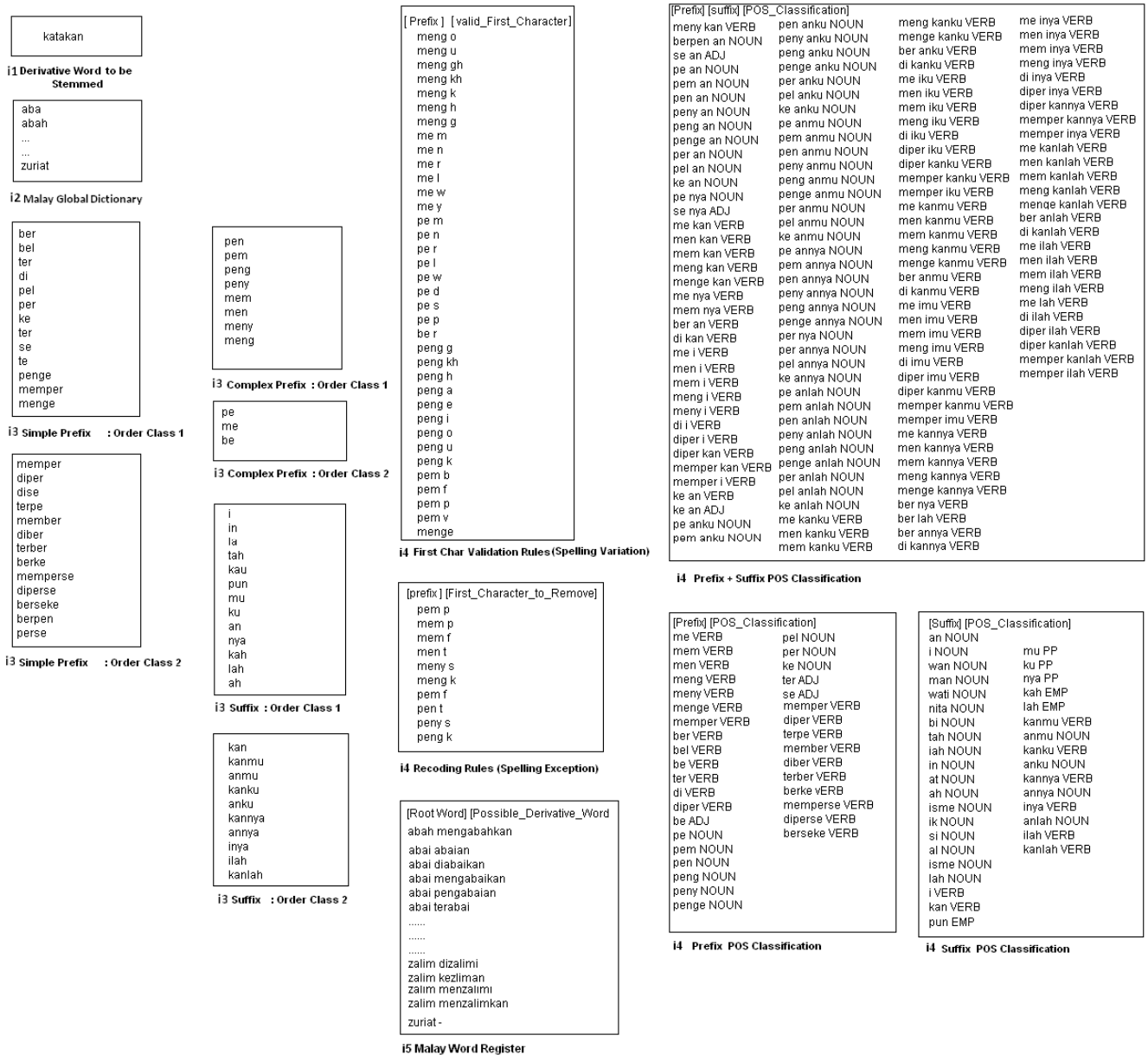

Fig. 2 : The Input Files – i1, i2, i3, i4, i5

3.2 Process Engine

This main component of the stemmer consists of two processes. The first, labelled as p1 in Fig. 1 removes under stemming and over stemming errors. The second labelled as p2, solves the ambiguity problem by suggesting the correct root word. The following explains the two processes.

i. **Over Stemming and Under Stemming Error Remover**

This process removes the under stemming and over stemming errors. It implements the exhaustive application of affix approach where it looks for possible affix to be removed in all order classes by

considering all possible derivative word morphologies such as [prefix+root], [root+suffix] or [prefix+root+suffix]. It does not stop when the first matched affix is found. It will instead search for the next possible affix to be removed. The general process is given below.

1. Reads a derivative word to be stemmed.
2. Checks the Malay global dictionary.
3. Checks the affix order classes.
4. Performs affix removal if matched.
5. Applies the validation rules to ensure a valid root word.
    a. valid first character
    b. recoding/replacement of first character
    c. valid affix combination
    d. valid minimum of three character root words
6. Stores the valid root word with its POS classification in an array.
7. Repeat steps 3 to 6 till all relevant affix have been accounted for.
8. Accepts single root word as the final root word and stores in file o1.
9. Stores multiple root words in file o2 and activates the ambiguity reducer process.
10. Ambiguity reducer uses the Word Register to select the correct root word.
11. End

In the case where only a single root word is found from the exhaustive search, the process will return the single root word as the final root word. This is written to the output file, *o1*. If multiple root words are produced, then all the root words will be written in the output file, *o2*. The multiple root words entry in o2 indicates that ambiguity exists for that derivative word. Hence, the stemmer will then execute *p2* which is the ambiguity reducer process.

This exhaustive search approach is important and useful especially to identify derivative words that have multiple root words. Previous ML stemmers did not identify such derivative words. Hence this ambiguity problem has never been addressed until now.

ii. Ambiguity Reducer

The Ambiguity Reducer process, *p2* will be activated if multiple valid root words are listed in the output file, *o2*. The structure of *o2* is shown in section 3.3 (ii). The general process of *p*2 is listed below.

1. Reads a root word from o2
2. Writes the pair of the derivative word and the resulting root word to output file, o3
3. If the pair exists in the Malay Word register, tags it as [suggested].
4. Repeat step 1 to 3 until all entries in o2 are accounted for.

If *o2* contains the following derivative word and two resulting root words,

a) '*Katakan*' (Derivative) *'Kata'* (resulting root word) VERB (root+*'kan'*)
b) *'Katakan'* (Derivative) *'Katak'* (resulting root word) NOUN (root+'*an'*)

and the combination of '*Katakan'* (say) and *'Kata'* (tell) is found in the Malay Word Register, this pair will be written in output file, *o3* and tagged as [suggested] which indicates '*kata'* (tell) is the correct root word of '*katakan*'. On the other hand, if the pair '*Katakan'* and *'Katak' (frog)* is not found in the Word Register, this means that in the Malay language context, it is incorrect to use the root word '*katak'* to form the word '*katak+an'* which has illogical meaning even though it satisfies the morphological rules. This pair will be copied to the *o3* file without the tag.

This ambiguity reducer process addresses the ambiguity problem of derivative words with multiple root words by referring to the Malay Word Register to determine the correct root word. This approach of using a Malay word register has not been used in previous Malay stemmers.

3.3 The Output Files

This section describes the output files as used in the stemmer. As illustrated in Fig. 1, there are three types of output files.

i. Single root word + POS classification, *o1*

This file stores the single root word that resulted from the *p1* process after the exhaustive application of affix is completed. It holds the derivative word, resulting root word and its affix POS classification. This indicates that only a single root word is returned as a valid root word without any ambiguities. The logical format of *o1* is as follows:

[Derivative word] [Single Root Word] [POS]
where POS is the part of speech of the pre-stemmed derivative word.

ii.* Multiple root words + POS classification, *o2

This file stores the multiple root words that resulted from *p1*. The entries are sorted as an array which lists the derivative word, the multiple root words and their POS classification as follows:

[Derivative word] [Root Word 1] [POS]
[Derivative word] [Root Word 2] [POS]
……..
[Derivative word] [Root Word n] [POS]

Where,

- POS is the part of speech of the derivative word.
- *n* represents the number of valid root words returned by *p1* and the number of possible root words that could be morphologically constructed to form that derivative word.

If the output file, *o2* contains many root words, then this indicate that ambiguity exists for that derivative word. Hence further processing by the Ambiguity reducer, *p2* will be carried out.

iii. Suggested root word + POS classification, *o3*

All the pair of derivative words and their respective root words in o2 will be written again in this file, *03* but this time, the tag 'suggested' will be put next to the pair that is found in the Malay Word Register. The task of checking their occurrence in the Malay Word Register is done by the Ambiguity reducer, *p2*. If the pair is not found, then the tag will not be included. The format of output file, *o3* is as follows, and the sample is shown in Fig. 3: *o3*.

[Derivative word] [Root Word 1] [POS][SUGGESTED]
[Derivative word] [Root Word 2] [POS]
……..
[Derivative word] [Root Word n] [POS]

For example:
(a) Katakan Kata VERB SUGGESTED
(b) Katakan Katak NOUN

The two root words extracted from the derivative word '*katakan*' are *'kata'* and '*katak'*. But since only the first pair (*a*) '*katakan*', '*kata*' is found in the Malay Word Register, the first pair is tagged with 'SUGGESTED'. Hence the correct root word for *'katakan'* is *'kata'* not *'katak'*. A sample of entries in the three output files are shown in Fig. 3 : *o1, o2, o3.*

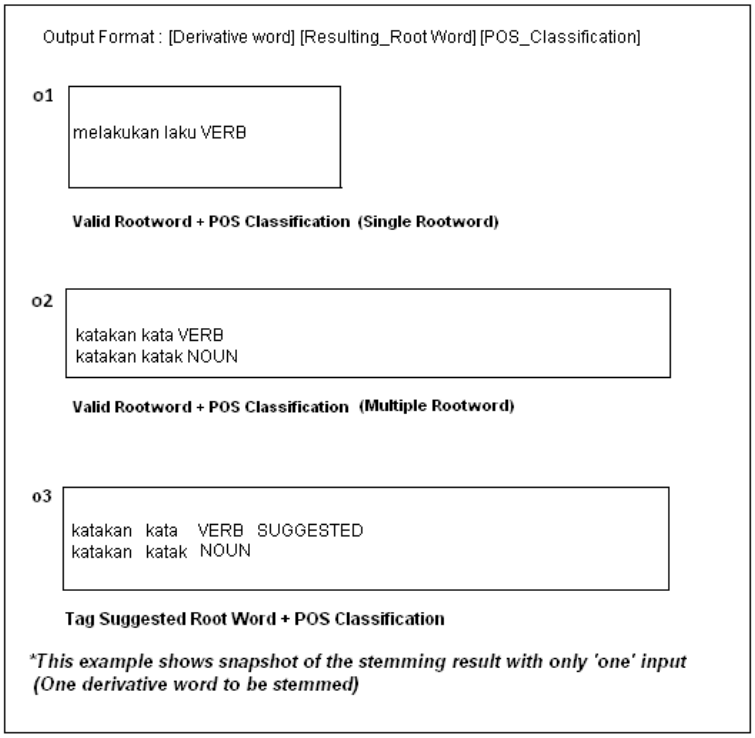

Fig. 3: The output files: o1, o2, o3

4.0 RESULTS

To evaluate the proposed stemmer, testing was done on two sets of data. The first data set contains words from the Malay Translation of the first two chapters of Quran, high school History essays and newspaper articles. The selected words encompass those with various morphological complexities. The second data set consists of words that were incorrectly stemmed by previous Malay stemmers. Table 3 below shows the number of words used in the datasets and the results.

The results from the stemmer were compared with the results from two native speakers. For derivative words with unambiguous or single root word such as '*me+lukis*' (draws) or '*beri+lah*'(give), both the stemmer and native speakers identified the exact root words. However, for derivative words with ambiguous or multiple root words, a few of the stemmer suggested root words differed from the speakers as shown in column D. This is because the pair of the root word and the derivative word could not be found in the Malay Word Register as it may not include all possible derivative words despite their practical usage. In particular, those derivative words with particles suffix like '*+lah*' to show emphasis. An example of such derivative word is '*merekalah*'(it's them!) which was correctly stemmed to '*reka*' (design) and '*mereka*'(them/they) as the root words. However, the pair of '*reka*' and '*merekalah*' or '*mereka'* and *'merekalah'* was not listed in the word register. Hence, none of the root words was suggested as the correct one even though semantically, it should be '*mereka*'. Such cases contribute to the 99.8% accuracy of this stemmer.

Table 3 : Stemming results

Source	Number of Words tested	Number of words correctly stemmed		Number of words with correctly suggested RW
		Resulting in a single root word (RW)	Resulting in multiple root words (MRW)	
	A	B	C	D
Quran (Al-Baqarah)	3,874	3,858	16	13
History Text (SPM Essay)	3,583	3,580	3	3
General Text (Newspaper Article)	1,708	1,699	9	9
Previous Stemmers				
Incorrectly stemmed words from Indonesian Stemmer	15	9	6	6
Incorrectly stemmed words from Sembok 1996 Stemmer	41	13	28	24
Incorrectly stemmed words form Sembok 2005	32	0	32	29
Incorrectly stemmed words from Idris' stemmer	51	37	14	13
TOTAL	**9304**	**9196**	**108**	**97**

5.0 CONCLUSION

In this paper, we described why stemming Malay words is difficult. We also explained why existing approaches of relying on first matched affix or a root word found still produce stemming errors. We proposed the exhaustive application of affixes approach implementable by separating the affixes into several order classes to remove the errors. As a result, all valid root words, single or multiple, in a derivative word could be extracted. To solve the ambiguity problem where selecting the correct root word is required, we introduced the use of a Malay Word Register. This stemmer performed well with 99.8% accuracy. Further works will focus on investigating other approaches to solve the ambiguity problem.

REFERENCES

[1] S.M.F.D. Syed Mustapha, N. Idris, Rukaini Abdullah. "Embedding Information Retrieval and Nearest-neighbour Algorithm into Automated Essay Grading System". *ICITA*, 2005 (2):169 – 172.

[2] H. Dalianis, M. Hassel. "Porting and Evaluation of Automatic Summarization", *Nordic Language Technology*, Norway, 2003.

[3] M. Fatah, F. Ren, S. Kuroiwa. "Stemming to Improve Translation Lexicon Creation Form Bi-texts". *Information Processing & Management,* 1996, 42(4): 1003 – 1016.

[4] R.G. Raj, S. Abdul-Kareem. "A Pattern Based Approach for the Derivation of Base Forms of Verbs from Participles and Tenses for Flexible NLP". *Malaysian Journal of Computer Science,* 2011, 24(2): 63 – 72.

[5] R.G. Raj, S. Abdul-Kareem. "Information Dissemination and Storage for Tele-Text Based Conversational Systems' Learning". *Malaysian Journal of Computer Science,* 2009, 22(2): 138 – 159.

[6] J.B. Lovins. "Development of a Stemming Algorithm". *Mechanical Translation and Computational Linguistics*, 1968, 11:22–31.

[7] J.L. Dawson, "Suffix and Word Conflation". *Literacy and Linguistic Centre Bulletin*, Cambridge, 1974.

[8] M.F. Porter, "An Algorithm for Suffix Stripping", 1980, *Program*, Vol. 14 , pp 130-137.

[9] C.G. Figuerola, R. Gómez-Díaz, Á.F. Zazo-Rodriguez, J.L. Alonso-Berrocal., "Stemming in Spanish: A First approach to its impact on information retrieval". In Results of the *CLEF 2001 Cross-Language System Evaluation Campaign*. Working Notes for the CLEF 2001 Workshop, pp.197-202, Darmstadt, Germany.

[10] P. Majumder, M. Mitra, S.K. Parui, G. Kole, P. Mitra, K. Dutta, "YASS: Yet another suffix stripper". *ACM Translations on Information Systems*, 2007, Vol. 25, No. 4, pp. 18-38.

[11] Nik Safiah Karim. *Tatabahasa Dewan*. Dewan Bahasa & Pustaka, 2004.

[12] Ahmad, Fatimah, M. Yusoff, T.M.T. Sembok, "Experiments with a Stemming Algorithm for Malay Words". *Journal of the American Society for Information Science*, 1996, 47(12) : 909-918, USA.

[13] Muhamad Taufik Abdullah, Fatimah Ahmad, Ramlan Mahmod, T. M. T. Sembok. "A Stemming Algorithm for Malay Language". *CITA 2005*: 181-186.

[14] N. Idris and S.M.F.D. Syed Mustapha. "Stemming for Term Conflation in Malay Texts". *International Conference of Artificial Intelligence (ICAI2001)*, Las Vegas, 2001, pp 1512 – 1517.

[15] M. Yasukawa, T. L. Hui, H. Yookoo. "Stemming Malay Text and Its Application in Automatic Text Categorization". *IEICE TRANS. INF & SYS*., 2009, VolE92-D, No. 12.

[16] Dewan Bahasa dan Pustaka, DBP.(2005). *Kamus Dewan Edisi keempat*.

[17] Dewan Bahasa dan Pustaka, Kuala Lumpur ,DBP. (2006). *Daftar Kata Bahasa Melayu* – Jilid 1&2.

[18] B.A.A, Nazief, M. Adriani, "Confix-Stripping: Approach to Stemming Algorithm for Bahasa Indonesia". *Internal Publication*, Faculty of Computer Science, Univ.of Indonesia, Depok, Jakarta, 1996.

[19] Jelita Asian, H. E. Williams, S. M. M. Tahaghoghi, "Stemming Indonesian". *28th Australasian Computer Science Conference (ACSC2005), Conference in Research and Practice in Information Technology*, Vol. 38, 2005.

BIOGRAPHY

Salhana Darwis obtained her Masters degree in Computer Science (Artificial Intelligence) from University of Malaya in 2011. Currently, she serves as Senior Analyst at the Standard Chartered Scope International.

Rukaini Abdullah is a Senior Lecturer at Faculty of Computer Science & Information Technology, University of Malaya. Malaysia. She obtained her PhD from University of Leeds. Her research interests include Natural Language Processing, Computer Based Learning and AI in Education.

Norisma Idris is a lecturer at Faculty of Computer Science and Information Technology of the University of Malaya in Kuala Lumpur. She received her PhD in Computer Science (Artificial Intelligence) from University of Malaya in 2011. She has several years teaching experience and currently teaches Artificial Intelligence courses such as Natural Language Processing and Applications and Prolog. Her research interests include Automated Summarization Assessment and Text Processing for Unknown Words.