Stemming Malay Text and Its Application in Automatic Text Categorization

Michiko YASUKAWA †(a), Hui Tian LIM ††, Nonmembers, and Hidetoshi YOKOO ††, Member

SUMMARY In Malay language, there are no conjugations and declensions and affixes have important grammatical functions. In Malay, the same word may function as a noun, an adjective, an adverb, or a verb, depending on its position in the sentence. Although extensively simple root words are used in informal conversations, it is essential to use the precise words in formal speech or written texts. In Malay, to make sentences clear, derivative words are used. Derivation is achieved mainly by the use of affixes. There are approximately a hundred possible derivative forms of a root word in written language of the educated Malay. Therefore, the composition of Malay words may be complicated. Although there are several types of stemming algorithms available for text processing in English and some other languages, they cannot be used to overcome the difficulties in Malay word stemming. Stemming is the process of reducing various words to their root forms in order to improve the effectiveness of text processing in information systems. It is essential to avoid both over-stemming and under-stemming errors. We have developed a new Malay stemmer (stemming algorithm) for removing inflectional and derivational affixes. Our stemmer uses a set of affix rules and two types of dictionaries: a root-word dictionary and a derivative-word dictionary. The use of set of rules is aimed at reducing the occurrence of under-stemming errors, while that of the dictionaries is believed to reduce the occurrence of over-stemming errors. We performed an experiment to evaluate the application of our stemmer in text mining software. For the experiment, test data used were actual web pages collected from the World Wide Web to demonstrate the effectiveness of our Malay stemming algorithm. The experimental results showed that our stemmer can effectively increase the precision of the extracted Boolean expressions for text categorization.

key words: Malay language, stemmer, stemming, affix rule, text mining

1. Introduction

Recently, the volumes of written text available in a digital format are increasing. The written text can include a large number of research papers, news articles, web pages, and blog entries. Manually analyzing such a large volume of electronic documents with written texts is cumbersome. Hence, the analysis of these documents using computer software has become imperative. This process is called text mining. Research on text mining has been conducted in several languages including Malay. However, the research on text mining in Malay is still relatively new. We have been engaged on the study of preprocessing for text mining in Malay. For the purpose of this paper, we focus on the method for extracting a set of Boolean expressions from documents written in Malay. We categorize newly-added documents automatically with the extracted Boolean expressions. In this categorization process, the precision is more important than the recall. This is because many general users on the World Wide Web complain about the nebulosity of resultant documents with unwanted documents while they look at only a few documents among the resultant documents [1]. Hence, a system with high precision is considered to be more advantageous. In our research, the Boolean expressions selected are a list of conjunctions of important words from the documents. These expressions should be able to summarize the documents. At the same time, this list of conjunctions is used to achieve automatic text categorization. In order to extract these expressions, the documents are required to be in a spreadsheet format. This format is called the document-word-matrix. In this format, the documents and words are related to each other. An element in the matrix corresponds to the frequency of the occurrence of a word in a document. In general, documents contain a lot of words that are not essential to the topic of the document, called unimportant words. Such words can lead to the extraction of unnecessary conjunctions. Therefore, it is important to filter out such words and select only the most important words. These important words are referred to as "index words."

In order to select index words, it is essential to enumerate the words in the document. The complexity of this enumeration process depends on the characteristics of the corresponding language. In English, for example, it is not so difficult to extract words from documents. Usually, English words are separated by a blank space. In contrast, it is difficult to extract words from Japanese documents. In Japanese, there is no rule that requires each word to be separated. Because one or more words are concatenated in a sentence without any space, the boundaries of the words are not distinct. In order to distinguish the individual words, morphological analysis of the document is required in the case of languages such as Japanese.

Since Malay is written by using the English alphabet, the processing of Malay documents should be comparable to that of English documents. However, the grammatical elements of Malay language have a unique feature; in this, a variety of affixes play an important role in the construction of sentences. In Japanese, the meaning of a sentence is clarified with the help of sentence particles. These particles are concatenated to words without space. Similarly, in Malay, affixes are used to make the meaning of sentences clear and
are concatenated to Malay stem words without space. The importance of affixes in Malay is no more than that of particles in Japanese. Therefore, it is important to identify and eliminate such unimportant linguistic fragments in Malay as well as in Japanese. The identifying process used in Malay is not morphological analysis but a process referred to as stemming. In this paper, we propose a new Malay stemmer (stemming algorithm). This proposed algorithm employs a set of Malay affix rules and two types of dictionaries: stem-word dictionary and derivative-word dictionary. The former is employed to ensure that under-stemming does not occur. The latter is used to ensure that over-stemming does not occur. We evaluate the effectiveness of the use of the proposed algorithm in text mining. The result of this evaluation is also discussed in this paper.

2. Stemming in Malay

Malay language [2] is spoken by more than eighty million people in the Netherlands East Indies, the Malay Peninusla, and adjacent territories. Indonesian language [3] is based on a version of Malay. Hence, Malay and Indonesian have common linguistic features. In Malay, the writing system is based on the use of the Latin alphabet. Therefore, written Malay text is similar to English texts. A sentence written in Japanese, English, and Malay is shown below as an example:

(1) Karewa, honwoyonda. (Japanese)
(2) He read a book. (English)
(3) Dia membaca buku. (Malay)

In the example, the Japanese sentence (1) is translated into English (2) and Malay (3). In sentence (2), we can distinctly recognize the individual English terms because blank spaces separated each word from the other. Therefore, the words “He,” “read,” “a,” and “book” can be extracted from sentence (2). However, it is difficult to distinguish between individual words in each component in the Japanese sentence (1) and the Malay sentence (3). Here a component refers to a series of successive characters, which does not contain any separator. The detailed analysis of Japanese and Malay components will be discussed later. In general, written texts contain a lot of words which have similar meanings but different spellings. It is desirable to identify these words so that the performance of text mining can be enhanced.

For instance, written English texts may contain words like “read” and “reads.” Similarly, written Japanese texts may contain “yomu” (the original form of verb “read”) and “yon” (a conjugation of the verb “read”). Further, written Malay texts may contain “baca” (the stem word for “to read”) and “membaca” (a derivative word of the stem word for “to read,” which implies “read <something> for <someone>”). Since the words “read” and “reads” have the same semantic meaning, they should be identified as the same word in text mining. If these words “read” and “reads” are to have different meanings, the rules governing the words would be semantically incorrect. Therefore, it is important to reduce the inflected or derived words to their original forms.

In English, stemming algorithms are applied to inflected or derived words. There are different approaches to achieve stemming in English. A popular algorithm carries out the stripping of word endings. In this algorithm, if a word ends in “ed,” it is removed from the main word. If the word ends in “s,” the “s” is removed. In English, the plural of the noun “book” is “books.” Similarly, the word “read” takes the conjugated form “reads” in the third-person singular present tense. In the English stemming algorithm, “books” and “reads” would be reduced to “book” and “read,” respectively. English stemming algorithms such as Porter [4] and Lovins [5] are widely employed in the research and development of English text processing.

In Japanese text mining, morphological analyzers are applied to sentences. Morphological analysis does not perform stemming but carries out a type of word segmentation. In this analysis, the input sentences are divided into small linguistic units and, their parts-of-speech are determined. These units are called morphemes. By identifying the parts-of-speech of the morphemes in sentence (3), we can filter out the semantically trivial morphemes such as the particle “wo” or the auxiliary “da.” Then, the non-trivial words such as the noun “hon” or the verb “yomu” can be obtained. Morphological analyzers ChaSen [6] and MeCab [7] are widely employed in the research and development of Japanese text processing.

To carry out Malay text processing, the use of an effective stemming algorithm is essential. However, a standard Malay stemmer has not been defined thus far. As mentioned earlier the linguistic features of Malay are similar to those of Japanese. In Table 1, the Japanese sentence (1) is divided into morphemes using the Japanese morphological analyzer ChaSen [6]. In Table 2, the Malay sentence (3) is analyzed by a native Malay speaker. Each morpheme in the inflected or derived words to their original forms.

In English, stemming algorithms are applied to inflected or derived words. There are different approaches to achieve stemming in English. A popular algorithm carries out the stripping of word endings. In this algorithm, if a word ends in “ed,” it is removed from the main word. If the word ends in “s,” the “s” is removed. In English, the plural of the noun “book” is “books.” Similarly, the word “read” takes the conjugated form “reads” in the third-person singular present tense. In the English stemming algorithm, “books” and “reads” would be reduced to “book” and “read,” respectively. English stemming algorithms such as Porter [4] and Lovins [5] are widely employed in the research and development of English text processing.

In Japanese text mining, morphological analyzers are applied to sentences. Morphological analysis does not perform stemming but carries out a type of word segmentation. In this analysis, the input sentences are divided into small linguistic units and, their parts-of-speech are determined. These units are called morphemes. By identifying the parts-of-speech of the morphemes in sentence (3), we can filter out the semantically trivial morphemes such as the particle “wo” or the auxiliary “da.” Then, the non-trivial words such as the noun “hon” or the verb “yomu” can be obtained. Morphological analyzers ChaSen [6] and MeCab [7] are widely employed in the research and development of Japanese text processing.

To carry out Malay text processing, the use of an effective stemming algorithm is essential. However, a standard Malay stemmer has not been defined thus far. As mentioned earlier the linguistic features of Malay are similar to those of Japanese. In Table 1, the Japanese sentence (1) is divided into morphemes using the Japanese morphological analyzer ChaSen [6]. In Table 2, the Malay sentence (3) is analyzed by a native Malay speaker. Each morpheme in

| Table 1 | Morphological analysis of a Japanese sentence. |
|---------|--------------------------------------------|
| Appeared form | Original form | Part-of-speech |
| kare (彼) | kare (彼) | Pronoun |
| wa (は) | wa (は) | Particle |
| . | . | Punctuation |
| hon (本) | hon (本) | Noun (book) |
| wo (を) | wo (を) | Particle |
| yon (読む) | yomu (読む) | Verb (read) |
| da (だ) | da (だ) | Auxiliary |
| . | . | Punctuation |

| Table 2 | Stemming a Malay sentence by a native Malay speaker. |
|---------|-------------------------------------------------|
| Malay | English | Japanese | Part-of-speech |
| Dia | he/she | 彼/彼女 | Pronoun |
| mem+ | を+する | Affix |
| baca | read | 読むこと | Stem (read) |
| buku | book | 一冊の本 | Stem (book) |
| . | . | Punctuation |

†The original Japanese sentence “彼は、本を読んだ。” is transliterated into Romaji.
Table 3  Malay affixes and these examples.

| Affix type | Stem word (English word) | Derivative word (English word) | Part-of-speech | Composition                        |
|------------|--------------------------|--------------------------------|----------------|-----------------------------------|
| Prefix     | baca (read)              | membaca (read <something>+)   | verb           | prefix + stem                     |
| Suffix     | baca (read)              | baca+kan (reading, reader)    | noun           | stem + suffix                     |
| Infix      | gigi (tooth)             | gerigi (saw, handsaw)         | noun           | (half-stem) + infix + (half-stem) |
| Confix     | baca (read)              | membacakan (read <something> for <someone>) | verb               | prefix + stem + suffix            |

Table 4  Prefix and first letter of stem.

| Prefix      | First letter |
|-------------|--------------|
| me+, pe+    | l, m, n, r, w|
| mem+, pem+  | b, f, p, v   |
| men+, pen+  | c, d, j, t, z|
| meng+, peng+| a, e, g, h, i, k, o, u|
| meny+, peny+| s            |

Table 1 is attached to its part-of-speech. Hence, by identifying its part-of-speech, the non-trivial words such as “hon” or “yomu” can be extracted from the sentence. In Table 2, Malay components are analyzed by a native Malay speaker. By extracting the stem words from the components, the non-trivial words such as “baca” or “buku” are obtained.

Conjugations and declensions, which are found in English and other languages, do not exist in Malay. As an example of declension, the English verb “retrieved” is “retrieves” in the third-person singular present tense, and the noun “book” is “books” in the plural form. In Malay, the same word can function as a noun, an adjective, an adverb, or a verb, depending on its position in the sentence. There is no inflection to identify the number, gender, or the case of the word. The tense of the sentence, such as past tense and future tense, is indicated by an adverb (e.g., “besok” (tomorrow)). In order to elucidate the meaning of written Malay texts, derivative words are used. Derivative words are derived from the stem words by a process called derivation. Derivation is influenced by the presence of affixes and the reduplication and compounding of stem words. Reduplication refers to the use of a reduplicated element. An example of reduplication is the plural form of the Malay word “buku” (book), which is “buku-buku” (books). In Malay, identifying stem words in reduplications and compound words is rather easy. However, identifying stem words in their derivative forms is difficult. This identification process in Malay is referred to as Malay stemming, which is the objective of our algorithm. Even though it is easy for a native Malay speaker to recognize a stem word in its derivative form, it is too complicated to stem Malay derivative words by a step-by-step procedure. Furthermore, some stemming processes can be difficult to understand even by native Malay speakers. These processes require the knowledge of the language as that available in Malay dictionaries. Therefore, it is essential to develop a reliable Malay stemmer. Using such a Malay stemmer, a large volume of text can be automatically processed by computers instead of humans. This will be a great advantage for text mining in Malay. The design of the Malay stemmer is closely related to the use of Malay affixes. In the next paragraph, the role of affixes attached to stem words in Malay is described.

There are four types of affixes in Malay: prefix, suffix, infix, and confix. Table 3 shows some examples of these affixes. In Malay, infixes are regarded as a peculiar type of affixes. The stem words that can be attached to infixes are limited. In Japanese, there are no infixes. In contrast, confixes can be used with many stem words. A confix is also called a circumfix. It is a compound affix of a prefix and a suffix. In Japanese, confixes such as “o+suru” or “o+ninaru” are used in honorific expressions. For example, “otsutsuesuru” or “oyasumininaru” is used to address someone with respect. In Malay, a great variety of confixes are commonly used, such as the confix “me+kan,” which converts the stem words to function as a verb to express causation. In general, when affixes are attached to stem words, the form of the affix remains unchanged. However, the forms of affixes may change under special circumstances. For example, prefixes “me+” and “pe+” change according to the stem words to which they are attached. The changing rules of these prefixes are shown in Table 4. As shown in the table, prefix “me+” changes to “mem+,” “men+,” “meng+,” or “meny+,” depending on the first letter of the corresponding stem word. When prefix “me+” is attached to the stem word “baca,” “me+” changes to “mem+” because the first letter of the stem word is “b.” Thus, the derivative word “membaca” is obtained (see example in Table 3). In Table 4, the letters shown in bold font, f, p, t, k, and s call for a special rule. When the prefixes “me+” or “pe+” is attached to a stem word having one of these five letters as the first letter, the letter is omitted from the word. For example, when stem word “sedia” (prepare) is attached to the confix “meny+kan,” the derivative word is not “meny sedia kan” but “meny edia kan.”

We have described the process of attaching affixes to stem words. Malay stemmer performs the above processes in the reverse order; that is, it separates the attached affixes, recodes an omitted letter, if any, and obtains the stem word. Let us consider some examples: derivative word “membacakan” is separated as “mem+kan” to obtain the stem “baca.” Similarly, the derivative word “menyediakan” is separated as “meny+kan,” the first letter of the stem word, “s,” is recoded, and the stem word “sedia” is obtained. The purpose of our stemming is to reducing the derived words to their original stem words. It should be noted that extracted stem words are important, while the removed affixes are deleted. When Malay stemming is being carried out by humans, humans are familiar with affixes and stem words. Because of this knowledge, stem words are precisely extracted from the derivative words. In computer processing, the stemming algorithms should have a high
level of precision and low computing cost, which can be achieved by using predefined language resources. Most of the previously reported studies conducted on Malay stemmer aimed at performance improvement of information retrieval systems [8]–[11]. The information provided by the evaluation experiments in those studies is limited in terms of data amount and document topics. Our aim is to develop a stemmer that can be applied to the text mining of a number of documents on a variety of topics. As an augmentation of the existent Malay stemming algorithms, our stemmer is expected to provide an effective way of stemming for text mining in Malay.

3. Development of Malay Stemmer

Our goal is to extract a set of Boolean expressions from documents. Boolean expressions are supposed to provide a summary of the documents. Therefore, the extracted Boolean expressions are useful in text mining. Besides, these expressions are represented by a list of conjunctions of words. These conjunctions can be used in text categorization. This is another reason why Boolean expressions are useful in text mining. The aim of our research is to enhance the performance of text categorization.

First, the conjunctions are extracted from labeled documents, which have been created in advance. For simplicity, we consider binary classifiers. As a result, the labels are either positive or negative. Let us consider the topic “baca.” The documents relevant to “baca” should belong to the positive class. The documents irrelevant to “baca” should belong to the negative class. Figure 1 shows an example of a set of Boolean expressions that are extracted without stemming. If a document contains the conjunctions (1-1), (1-2), or (1-3), it is categorized into the positive class, since it is relevant to “baca.” However, other documents, although relevant to “baca,” might not be categorized into the positive class because the conjunctions list does not include other derivatives of “baca,” such as “pembaca” or “pembacaan.” If stemming is applied to the extraction process, the extracted Boolean expressions would be simple and comprehensive. Figure 2 shows an example of a set of Boolean expressions that are extracted with stemming. By the introduction of the conjunction (2-1), it is ensured that all documents relevant to “baca” are categorized into the positive class.

The extraction process of a set of Boolean expressions

from documents is given as follows:

1. Extract components from documents. (Components are separated by blank space characters in Malay.)
2. Apply the stemmer to each component. (The stemmer deletes the unimportant affixes from the components and extracts the important stem words.)
3. Index the documents. Select significant stem words from extracted stem words. Count the frequency of each word in each document. Create a document-word-matrix.
4. Extract Boolean expressions from the document-word-matrix.

In this process, the Boolean expressions may be ineffective if the stemmer deletes too many or too few affixes. If too few affixes are removed from each derivative word, the extracted Boolean expressions would be complicated and non-exhaustive. This may decrease the recall of text categorization. If too many affixes are deleted from each derivative word, it would be difficult to differentiate between the documents in the positive class and the documents in the negative class. This may decrease the precision of text categorization. There are two types of errors that can occur in stemming:

**Type1**: under-stemming  
**Type2**: over-stemming

Type1 is said to be an error because in this, the affixes that should be deleted are retained. Type2 is an error because in this, the affixes that need to be retained are deleted. Over-stemming is further classified into two types:

**Type2-1**: over-stemming of a stem word  
**Type2-2**: over-stemming of a derivative word

In type2-1 error, one or more letters of a stem word are incorrectly removed. This may change one stem word to another. In type2-2 error, one or more significant affixes of a derivative word are incorrectly removed. This may change the meaning of the derivative word. In order to reduce the occurrence of both type1 and type2 errors, our stemmer follows the given procedure:

1. Employ a set of Malay affix rules, which can reduce the occurrence of type1 errors.
2. Employ a stem-word dictionary, which can reduce the occurrence of type2-1 errors.
3. Employ a derivative-word dictionary, which can reduce the occurrence of Type2-2 errors.

The flow of the proposed stemming algorithm is shown in Fig. 3. Our approach is described in detail in the following subsection.

3.1 Application of Affix Rules

Our stemmer uses a list of Malay affix rules. Some of these rules were originally defined by Othman and augmented by Ahmad. The list consists of 436 rules. They are listed in

![Fig. 1 Example of Boolean expressions (without stemming).](image1)

![Fig. 2 Example of Boolean expressions (with stemming).](image2)
Appendices A and B of [8]. Ahmad also proposed another list of affix rules for new derivative words (Appendix C of [8]). However, these rules can contradict those governing the traditional derivative words. Hence, our stemmer does not use the rules for new derivative words.

In the previous studies on Malay stemmer, the changing form of affixes was considered. However, they did not include recoding of the omitted first letter of the stem word. This led to the occurrence of a lot of stemming errors. In a previous study on an Indonesian stemmer, the stemming errors were shown to increase when the extraction was carried out without recoding the first letter of the stem in a derivative word [12]. Since Indonesian is linguistically close to Malay, we adopt the same recoding process. The recoding process in our stemmer is based on customary rules of Malay affixes shown in Table 4.

A component in Malay sentence must include at least one stem word. One or more affixes can be concatenated to the stem word. Some stem words contain a substring that is coincidentally identical to the affix. Furthermore, some affixes contain a substring that is identical to other affixes. For this reason, one or more stem words can be obtained from a component by using the Malay stemmer, even though it represents the same derivative word. The order of removing the affixes effectively from the components must be considered. In the previous Malay stemmer, the removal order was determined on the basis of the input data. They performed experiments using 10 chapters of the Koran and 10 articles of research abstracts. According to this study, the occurrence of stemming errors would be reduced if the removal order is changed on the basis of the input data. Our stemmer should be applicable to a number of documents covering a variety of topics. Therefore, we require a general removal order that can be applied to any document.

In Malay, infixes are rarely used. When infixes are removed from the components, there is a high probability that the remaining strings are incorrect or meaningless. In addition, it relatively difficult to remove shorter affixes with accuracy. If shorter affixes are removed, the remaining strings will be incorrect in high probability. On the basis of these points, the removal order in our stemmer is such that the affixes are removed in the order of confixes, prefixes, suffixes, and finally, infixes. Confixes and prefixes are the most commonly used affixes in Malay. On an average, confixes are longer than prefixes because confixes are a combination of prefixes and suffixes.

The 436 affix rules are categorized into confixes, prefixes, suffixes, and infixes. Then, affixes under each category are sorted in a long-before-short order. This order is selected to avoid the removal of wrong affixes. Although this approach is mostly advantageous, for some components, affix rules with a short-before-long order should be applied. Therefore, our stemmer applies the longer affix rules at first, and only if the stem word is not obtained, the shorter affix rules are applied.

### 3.2 Referring to Stem-Word Dictionary

Previous studies proposed that in order to prevent the occurrence of type2-1 errors, a stem word dictionary must be referred to. Without such referencing, over-stemming of stem words is unavoidable [8]. Our stemmer employs a printed dictionary [13], which comprises of 5,774 entries; each entry is a basic Malay stem word. It is important that the dictionary contains a substantial number of entries of stem words. From the experimental results of [8], the use of a comprehensive dictionary causes a number of under-stemming errors (type-1). This experiment had employed a Malay dictionary [14] with 22,393 entries. In general, Malay dictionaries include only stem words as their entries. When referring to dictionaries, Malay natives need to reduce a derivative word to the stem word to find it in the dictionary. However, for the sake of convenience, some dictionaries include considerable number of derivative words in their entries. In order to avoid the occurrence of under-stemming errors, we choose a basic dictionary rather than a comprehensive dictionary. The dictionary we employed in our algorithm is referred to as “KD” for simplicity.

### 3.3 Refering to Derivative-Word Dictionary

In the previous studies on English stemmer [15], the adverse effects of over-stemming have been discussed. In the experiments conducted using TREC-7 and TREC-8, it was observed that over-stemming of some derivative words caused such adverse effects. Although this issue has not been discussed in the studies conducted on Malay stemmer, it should be avoided in Malay stemming. Consider the two Malay derivative words “warisan” (heritage) and “mewarisi” (inherit), which have the same Malay stem word.
“waris” (heir/heiress). If the affixes are removed from these words, we obtain semantically different words with identical spelling. However, because our purpose is not stemming itself but text mining, reduction of the derivative words to give inappropriate index words should be avoided. Even if the resultant stem after stemming is the correct origin of the derivative word, the stemming should be invalidated if it does not satisfy our purpose. Therefore, we employ a derivative-word dictionary. Since there was no existent dictionary that is suitable for use in our stemmer, we created a dictionary using the Open-Office spell-checker dictionaries [16]. These dictionaries include popular derivative words. Our stemmer is supposed to be applied to Malay texts. However, we need to take into account the Indonesian language because Malay and Indonesian have common vocabularies and common grammatical rules. In fact, web-search results obtained by entering Malay keywords usually include many Indonesian documents. Hence, overstemming in Malay as well as that in Indonesian should be avoided. Thus, we created the derivative-word dictionary from the Malay spell-checker dictionary and the Indonesian spell-checker dictionary. For convenience, we call the created dictionary MYID. A brief statistics on MYID is as follows:

- 36,709 words in total
- 24,240 words are Malay words
- 23,417 words are Indonesian words
- 10,948 words are common to Malay and Indonesian

The words in MYID are composed of the Latin alphabet only. Words with a numeral character, punctuation mark (.), or a hyphen mark (-) are excluded. These exceptional words are out of the scope of our study because they require special preprocess other than stemming.

4. Evaluation Experiment

In order to perform an evaluation experiment on our proposed Malay stemmer, a stemming program is implemented using PHP [20]. The dataset for this experiment and the details of the experiment are described in this section.

4.1 Dataset for Experiments

We obtained the required web pages using the search engine API [17]. First, the 10 keywords shown in Table 5 were used in the web search. For each keyword, we selected as many search results as possible (up to 1000). From these search results, we obtained as many web pages (cache pages) as possible. Almost half of the URLs searched directed to invalid web pages. After obtaining the web pages, a portion of the text data was taken from each page and saved as a separate document. In order to apply text categorization, the documents are labeled according to their topics. A document is relevant to the keyword that was used during its web search. If the web search for two or more different keywords gives the same document, then the document might not be relevant to the keywords. In fact, a majority of such duplicated documents of different topics were the top page or the menu page of a portal site. These duplicated documents were removed from the experimental dataset. The rest of 3,825 documents were labeled by their corresponding topics. The labeled documents are regarded as relevant to their corresponding topic. The number of URLs searched, web pages obtained, and relevant documents per topic are shown in Fig. 4. Our constructed dataset is called MYTEXT.

4.2 Correctness of Affix Removal

By our experiment, the correctness of affix removal is evaluated. The affix removal corresponds to the steps 3 to 9 in Fig. 3. In this experiment, we temporarily discontinue the use of the derivative-word dictionary MYID so that our stemmer can process any derivative word. In MYTEXT, there are a large number of words. Most of them are useless or not so useful in text mining. In this experiment, we need to verify the stemming correctness of important derivative words in MYTEXT. We select 945 most important derivative words from MYTEXT. These derivative words necessarily contain the important keywords. The important keywords are the words listed in Table 5 or Table 6. In Table 6, the words are selected by a native speaker of Malay and are all closely related to the given topic words. The selected derivative words are processed by our Malay stemmer. The resultant stems were checked by a native Malay speaker. The result gave 14 over-stemming errors and 6 over+under-stemming errors (shown in Table 7), and 13 under-stemming errors (shown in Table 8). Here, over+under refers to a

| Table 5 | Topics in Malay for experimental data. |
|---------|--------------------------------------|
| agama (religion), budaya (culture), ekonomi (economy), hiburan (entertainment), kesehatan (health), komputer (computer), makanan (food), pendidikan (education), politik (political), sukan (supports) |
Table 7 Stemming errors of stemmed derivative words.

| No. | Derivative word | Stem word (English translation) | Stemming result (English translation) | Error type |
|-----|-----------------|---------------------------------|---------------------------------------|------------|
| 1   | antipoligami    | poligami (polygamy)            | gam (glue)                            | over       |
| 2   | bermaklumat     | maklumat (information)         | maklum (to know)                      | over       |
| 3   | berpasukan      | pasukan (team)                 | pasu (pot)                            | over       |
| 4   | berpoligami     | poligami (polygamy)            | gam (glue)                            | over       |
| 5   | bersepasukan    | pasukan (team)                 | pasu (pot)                            | over       |
| 6   | bertemakan      | tema (theme)                   | makan (eat)                           | over+under |
| 7   | dipoligami      | poligami (polygamy)            | gam (glue)                            | over       |
| 8   | gemakan         | gema (echo)                    | makan (eat)                           | over+under |
| 9   | kesukanan       | sukan (sport)                  | kanan (right)                         | over+under |
| 10  | makanan         | makan (eat)                    | niak (mother)                         | over       |
| 11  | memasukkan      | masuk (enter)                  | pasu (pot)                            | over       |
| 12  | memasukkan      | masuk (enter)                  | pasu (pot)                            | over       |
| 13  | pasukananya     | pasukan (team)                 | pasu (pot)                            | over       |
| 14  | pendidikan      | didik (to educate)             | di (at)                               | over       |
| 15  | peny (s) emakan | semak (to check)               | makan (eat)                           | over+under |
| 16  | poligami        | poligami (polygamy)            | gam (glue)                            | over       |
| 17  | polikas         | politik (political)            | tikus (rat)                           | over+under |
| 18  | polikaslah      | politik (political)            | tikus (rat)                           | over+under |
| 19  | ramanan         | ragam (manner)                 | raga (exhibit)                        | over       |
| 20  | sepasukan       | pasukan (team)                 | pasu (pot)                            | over       |

Table 8 Under-stemming errors of derivative words.

| No. | Derivative word | Stem word (English translation) | Error type |
|-----|-----------------|---------------------------------|------------|
| 1   | bervitamin      | vitamin (vitamin)               | under      |
| 2   | dinasihat       | nasihat (advice)                | under      |
| 3   | internetku      | internet (internet)             | under      |
| 4   | internetnya     | internet (internet)             | under      |
| 5   | kliniklanya     | klinik (clinic)                 | under      |
| 6   | makanlah        | makan (eat)                     | under      |
| 7   | memaksimakan    | maksima (maximum)               | under      |
| 8   | meminimakan     | minima (minimum)                | under      |
| 9   | mengoptimakan   | optima (optimum)                | under      |
| 10  | muzikal         | muzik (music)                   | under      |
| 11  | penasihat       | nasihat (advice)                | under      |
| 12  | politikal       | politik (political)             | under      |
| 13  | seninya         | seni (art)                      | under      |

The composite error of over-stemming and under-stemming errors. The error rate for processing the experimental derivative words was approximately 3.5%. In the experiment in [9], they used a part of the Malay translation of the first only 2 chapters of the Koran. As the order in which the affix rules were applied was changed, the error rates varied from 1.4% to 4.4%. Our dataset is different from that used in the above experiment. Our experiment is more extensive in terms of the data topics (10 topics) and the volume of data (3,825 documents). Although the error rate of our stemmer should not be directly compared to that of the above study, the resultant error rate can be regarded as being within the range of the reliable results.

4.3 Effectiveness of Our Stemmer

We have conducted an experiment on the application of text categorization. In [18], a method for text categorization was proposed. In this method, Boolean expressions are extracted from the labeled documents. The effectiveness of this method has already been established. If a decent number of documents and a good set of index words are given, Boolean expressions can be satisfactorily extracted from the document using this method. A software tool kit called RIKTEXT based on this method is also available [19]. In order to evaluate the performance of our stemmer in a text categorization application, labeled documents are required to be classified into test data and learning data. For each topic, 10 documents were selected from MYTEXT to be used as test data. The remaining documents were used as learning data. Specifically, we made a set of test data that consists of 100 documents (10 relevant documents and 90 irrelevant documents) for each topic. Then, we conducted categorization sessions for 10 topics. In our evaluation, 1,000 documents in total were judged. One of the important features of our stemmer is the referencing of the derivative-word dictionary MYID. In the previous study, the effect of the use of derivative-word dictionaries was not discussed. We aim at elucidating this. In this experiment, we compared the following three methods.

1. Text categorization without stemming. (baseline1)
2. Text categorization with stemming but without the use of MYID. (baseline2)
3. Text categorization with stemming using MYID. (proposed)

The precision and recall results of the experiments are shown in Fig. 5 and Fig. 6, respectively. From Fig. 5, it is seen that the proposed stemmer can effectively increase the precision of text categorization. This is because the proposed method can avoid the occurrence of over-stemming of the derivative words. However, in the case of the topic "agama" (religion), the result for without stemming is better than that of the proposed stemmer. One reason for this poor performance of the stemmer is that this particular topic leads to confusion in text categorization. In Malay culture, "agama" (religion) and "budaya" (culture) are closely re-
lated. Therefore, the documents related to these topics may not be categorized correctly. For example, documents on Buddhism are related to “agama” (religion) as well as “budaya” (culture) in Malay culture and hence maybe categorized incorrectly. With the exception of the topic “agama” (religion), the proposed stemmer is observed to be effective for all the topics.

The average precision, recall, and F-measure are shown in Table 9. The effectiveness of the proposed method is not so significant with respect to its F-measure. However, the results show that the precision of text categorization increases by the application of the proposed stemmer. Therefore, applying the proposed stemmer is advantageous to many general users on the World Wide Web because they are considered to prefer looking at a small number of relevant documents and they prefer not looking at irrelevant documents.

In the experiment, baseline1 yields under-stemming because it does not carry out any stemming. In contrast, baseline2 yields over-stemming of derivative words because it does not employ any derivative-word dictionary. On the other hand, since the proposed stemmer employs a derivative-word dictionary (MYID), it can reduce over-stemming of derivative words. Native speakers of Malay are able to remove an affix from a derivative word while they are also able to leave a derivative word unchanged if it should not be stemmed. Hence, they can naturally avoid over-stemming of derivative words. To the best of our knowledge, reducing over-stemming of derivative words has not been discussed in previous studies on Malay stemmers. Our proposed algorithm, which employs a derivative-word dictionary, is supposed to be applicable to Malay stemmers not only for text categorization but also for information retrieval or machine translation. Evaluation of the stemmer in other applications is left for our future work.

4.4 Efficiency of Our Stemmer

When the proposed stemmer is put to practical use, the processing time of the stemmer is needed to be taken into account. First, we show the algorithmic complexity of the proposed stemmer. As the stemming algorithm (See Fig. 3) shows, if an input component is included in the dictionary, it skips the steps from Step-5 to Step-7. Hence, the stemming process on such components requires the minimum processing time. The time complexity is $O(m+n)$, where the number of the words in the dictionary is $m$ and the number of input components is $n$. If an input component contains as many affixes as possible, the loop iteration from Step-5 to Step-7 is maximized. Hence, the stemming process on such components requires the maximum processing time. The time complexity is $O(m + nk)$, where the number of the words in the dictionary is $m$, the number of the input components is $n$, and the number of the affix rules is $k$.

Although $m$, $n$, and $k$ can be large, the actual stemming process is not cumbersome on today’s personal computers. Next, we discuss the actual processing time of the stemming process. We ran the stemmer on a Linux PC with a 2.4 GHz Intel Core2-Quad processor and 2 GB of memory. For the measurement, we extracted the most frequent 10,000 Malay components from the dataset (See 4.1). The measured time is 2 min 2.162 s for 10,000 components, i.e., an average of 0.0122 s per component.

Finally, we discuss the processing time in terms of text categorization applications. If a stemmer requires too much processing time in its application, users would not use the stemmer. The text categorization process consists of two phases:

**Phase-1** Extracting a set of Boolean expressions from labeled documents. (preprocess)

**Phase-2** Categorizing unlabeled documents with the extracted expressions. (main process)

In the evaluation experiment of text categorization (See 4.3), each evaluation session involved 10,000 Malay components and 3,825 Malay documents. In the experiment, Phase-1 required 15 min 34.267 s and Phase-2 required 0.574 s on average. While the main process should be performed immediately at the user’s request, the preprocess can be performed during the idle time of the processor. Stemming is an additional process to the preprocess. While the stemmer
does not affect the processing time of the main process, it can be effective to increase the precision of the text categorization result. The processing time of stemming is much shorter than the whole processing time of text categorization (preprocess and main process). Therefore, our stemmer is considered to be applicable for increasing the precision of text categorization.

5. Conclusion

In this paper, a new Malay stemming algorithm has been proposed. We implemented a stemming method based on the proposed algorithm. We also performed an evaluation experiment on the implemented method using a set of actual documents from the World Wide Web and the text mining software. The results of the experiment revealed that our algorithm can effectively increase the performance of text mining, particularly the precision of text categorization. In Malay, stemming is important in text processing because a lot of derivative words need to be reduced to their stem words. By effectively stemming the derivative words, the text mining software need not process the unimportant affixes. In the previously proposed Malay stemming algorithms, the effect of over-stemming of derivative words has not been investigated. Our stemmer employs a derivative-word dictionary as well as a stem-word dictionary and a set of Malay affix rules. By using these resources, our stemmer performs affix removal effectively.

The stemming algorithm applies affix rules in a static order. When one or more affix rules can be applied to a derivative word, the order of these rules needs to be optimized and should be flexible. In the future, we plan to work on the optimization of the order of affix rules. For this, learning the patterns of stems and affixes from a large volume of Malay corpus is considered to be a promising approach.

References

[1] N. Höchstötter and D. Lewandowski, “What users see - Structures in search engine results pages,” Information Sciences, vol.179, pp.1796–1812, 2009.
[2] E.F. Winckel, Handbook of The Malay Language, Winckel Press, 2007.
[3] J.N. Sneddon, Indonesian: A Comprehensive Grammar, Routledge, 2006.
[4] M.F. Porter, “An algorithm for suffix stripping,” Program, vol.14, no.3, pp.130–137, 1980.
[5] J.B. Lovins, “Development of a stemming algorithm,” Mechanical Translation and Computation Linguistics, vol.11, no.1, pp.23–31, 1968.
[6] ChaSen. http://chasen.aist-nara.ac.jp/
[7] McCab. http://mecab.sourceforge.net/
[8] F. Ahmad, M. Yusoff, and T.M.T. Sembok, “Experiments with a stemming algorithm for Malay words,” JASIS, vol.47, no.12, pp.909–918, 1996.
[9] M.T. Abdullah, F. Ahmad, M. Yusoff, R.Mahmod, and T.M.T. Sembok, “A stemming algorithm for Malay language,” Proc. ICTTA, pp.181–186, 2005.
[10] M.T. Abdullah, F. Ahmad, R. Mahmod, and T.M.T. Sembok, “Improvement of Malay information retrieval using local stop words,” Proc. IATC, pp.1–7, 2005.
[11] S.Y. Tai, C.S. Ong, and N.A. Abdullah, “On designing an automated malaysian stemer for the Malay language,” Proc. IWIRA, pp.207–208, 2000.
[12] M. Adriani, J. Asian, B. Nazief, S.M.M. Tahagghohi, and H.E. Williams, “Steering indonesian: A confix-stripping approach,” ACM TALIP, vol.6, issue.4, article no.13, pp.1–33, 2007.
[13] K.M. Hussain, “Kamus dwibahasa,” pelanduk, 1993.
[14] Dewan Bahasa dan Pustaka, Kamus Dewan, DBP, 1991.
[15] S. Honma, “An evaluation of English stemming data in full-text retrieval,” IPSJ SIG Notes, vol.2000, no.65, pp.55–62, 2000.
[16] OpenOffice.org, Dictionaries.
[17] Yahoo! JAPAN, Yahoo! JAPAN DEVELOPER NETWORK
[18] S.M. Weiss, N. Indurkhya, T. Zhang, and F.J. Damerau, Text Mining, Springer, 2005.
[19] N. Indurkhya, RIKTEXT: Rule induction kit for text, 2004.
[20] PHP: Hypertext Preprocessor, http://php.net/

Michiko Yasukawa received her Bachelor’s degree in Engineering from Ritsumeikan University in 1996. She received her Master’s degree and Doctoral degree in Informatics from Kyoto University in 2000 and 2003. She is currently an Assistant Professor at the Department of Computer Science, Gunma University. Her research interests include information retrieval and natural language processing. She is a member of IPSJ, IEEE Computer Society, and ACM.

Hui Tian Lim received the B.Eng. degree in computer science from Gunma University in 2008. He is interested in text mining in Malay.

Hidetoshi Yokoo was born in Yamagata, Japan, on April, 1954. He received the B.Eng. degree in instrumentation physics, the M.Eng. degree in information engineering, and the D.Eng. degree in mathematical engineering from the University of Tokyo, in 1978, 1980, and 1987, respectively. From 1980 to 1989 he was Research Associate at the Department of Electrical Engineering, Yamagata University, Japan. In 1989, he joined Gunma University, Kiryu, Japan, where he is currently Professor in the Department of Computer Science. His research interests are in data compression, text processing, and their applications to computer science in general. Dr. Yokoo is a member of the Information Processing Society of Japan, the Institute of Electronics, Information and Communication Engineers, the IEEE Computer Society, and the SITA.