Jawi Stemmer: Evaluation of Stemmers Based on Strength and Accuracy

Suliana Sulaiman, Khairuddin Omar, Nazlia Omar, Mohd Zamri Murah, Hamdan Abdul Rahman

Abstract
Stemmers are important for Information Retrieval and essential for Natural Language Processing. Many researchers have focused on the effect of stemming on document retrieval and measured their stemmers with recall and precision. Applications in text mining (i.e., document indexing and clustering) require great accuracy in index terms. The aim of this paper is to evaluate a Jawi stemmer based on its strength and accuracy. To achieve the result, a control experiment was carried out. Using the results obtained from the experiments conducted in this study, the Jawi stemmer was deemed to be the best stemmer (accuracy = 85.08%), followed by the Abdullah stemmer (accuracy = 81.17%), and the Ahmad stemmer (accuracy = 78.67%).

Keywords: Stemmer Strength, Malay Stemmer

1. Introduction
A stemmer is a process used to reduce a derived word into its own root word [1]. A derivative word can be described as a word containing an affix and a root word to form another word. These words share a similar meaning. For example, the derived word لرالی (brlari) can be stemmed to its root word لر (lari). When searching an online document, search engines use stemming methods to produce the root word and match with the user’s queries [2]. Most stemmers show significant improvements for document retrieval compared to no stemming at all. These improvements can be seen in recall level [3]. In other research, Md Zahurul, 2007 [4] proposed an inexpensive way of improving the Bengali spelling checker, using a stemmer instead of morphological analysis. The Bengali spelling checker finds misspelled suffix words and the stemmer eliminates the suffix and leaving the stemmed word (which was misspelled by the user). The spelling checker checks the stemmed word with the stem dictionary and finds the best implication for the stemmed word. The suffix was then added before the word listed in the suggestion list. This spelling checker produced 90.8% accuracy when tested with 13000 Bengali words. For Malay transliteration, to transliterate a Rumi document into Jawi, a stemmer was used to divide the root word from the affixes [5]. This method was proposed by Yon Hendri, 2009 [5] and showed the highest result amongst other techniques.

Most Malay stemming algorithms for Rumi are rule based. In document collection, the stemmers help to reduce the size and complexity of data, and also helpful to improve the performance of Information Retrieval. However, document indexing and clustering for text mining applications require great accuracy in index terms rather than index compression. Therefore, in order to produce the best stemmer, it is essential for us to not only evaluate the performance of the stemmer on IR, but to evaluate the performance of the stemmer on strength and accuracy. This paper is an extension for new Jawi Stemmer results, as presented in [6] and [7], and implies further results on its strength using Frakes and Fox’s, 2003 [8] proposed method. The evaluation’s results using IR will be discussed in another paper. This paper is divided into seven sections. Section 2 describes the Malay morphology. Section 3 presents an overview of related work. Section 4 describes the Jawi stemmer. Section 5 presents the stemmer performance. Section 6 discusses the result analysis; and finally, Section 7 presents the conclusion and future work.
2. Malay morphology

Affixation in Malay is divided into four types: prefix, suffix, infix, and circumfix. Besides the original Malay affix, some foreign affixes from English, Sanskrit, and Arabic languages have also been adopted and used as Malay affixes. These foreign affixes can be in either prefix or suffix form [9].

1. Prefix:
The Prefix can be found before the root word. Prefixes are found in derivative nouns, derivative verbs, and derivative adjectives [9]. They are known as noun prefix, verb prefix, and adjective prefix. Table 1 shows examples of Prefix di-, ke-, se-, and beR-.

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Condition</th>
<th>Derived Word</th>
</tr>
</thead>
<tbody>
<tr>
<td>di-</td>
<td>Add di-</td>
<td>di+ambil (pick) = diambil (picked)</td>
</tr>
<tr>
<td>ke-</td>
<td>Add ke-</td>
<td>ke+arah (fold) = kelipat</td>
</tr>
<tr>
<td>se-</td>
<td>Add se-</td>
<td>se+jauh (far) = Sejauh (as far)</td>
</tr>
<tr>
<td>beR-</td>
<td>-Use be-, bel-, ber- if the first word character = r. Otherwise, use be- if the first word character != r.</td>
<td>beR + rasa (feel) = herasa (feeling)</td>
</tr>
<tr>
<td></td>
<td>*Exception use bel- for ajar (teach) and unjur (project)</td>
<td>beR + lalu (walk) = belalau (walking)</td>
</tr>
<tr>
<td></td>
<td>-When the word does not start with the character r, use prefix ber-</td>
<td>beR + layer (sail) = belayar (sailing)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>beR + ganti (change) = berganti (changing)</td>
</tr>
</tbody>
</table>

Table 2 shows examples of prefix meN- and peN-.

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Prefix</th>
<th>First letter</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>meN-</td>
<td>me-</td>
<td>-i, m, n, ng, ny, r, w, y</td>
<td>-Add prefix to the root word. For example: melawan (fighting)</td>
</tr>
<tr>
<td>/peN-</td>
<td>/pe-</td>
<td>-l, m, n, ng, ny, r, w, y</td>
<td>-Add prefix to the root word. For example: membasuh (washing)</td>
</tr>
<tr>
<td>mem-</td>
<td>/mem-</td>
<td>-b, v</td>
<td>-Add prefix and remove 1st character in the root word. For example: memasti (certainty)</td>
</tr>
<tr>
<td>/men-</td>
<td>-b, v</td>
<td>-f, p</td>
<td>-Add prefix to the root word. For example: memasuki (entering)</td>
</tr>
<tr>
<td>men-</td>
<td>/men-</td>
<td>-c, d, j, sy, z</td>
<td>-Add prefix to the root word. For example: mencuci (cleaning)</td>
</tr>
<tr>
<td>/meng-</td>
<td>-c, d, j, sy, z</td>
<td>-t</td>
<td>-Add prefix and remove 1st character in the root word. For example: menarik (tempting)</td>
</tr>
<tr>
<td>meng-</td>
<td>/meng-</td>
<td>-a, e, g, h, i, o, q, u, x</td>
<td>-Add prefix to the root word. For example: mengajar (teaching)</td>
</tr>
<tr>
<td>-k</td>
<td>-Add prefix and remove 1st character in the root word. For example: mengayuh (cycling)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. Suffix
Affixes that are found in the last position of the root word are called suffixes. Similar to prefixes, suffixes can be single or double. In the Malay language (Rumi script), a suffix does not require any changes when it is applied to a root word to form a derived word. In Malay, a suffix can be found as a noun suffix and a verb suffix. Example: atas (up) + an = atasan (upper).

3. Infix
An Affix within a root word is called an infix. A small amount of infixes appear in the Malay language. Most of the root words that contain infixes are used as the root word for them. Example: gigi (teeth) + er = gerigi (sharp teethes).

4. Circumfix
Root words that contain an affix at both the beginning and the end of a word are called Circumfix. Circumfixes can be found as a derivative noun, a derivative verb, or a derivative adjective (Nik et al, 1993b). The spelling method for a circumfix follows the spelling method for both prefix and suffix. For prefix, some of the word must be removed to form the correct word. Example: pel+ ajar (educate) + an = pelajaran (education). Table 3 shows the most common Malay circumfixes.
2.1 Jawi versus Roman (Rumi)

Jawi and Roman (Rumi) differ from one another in three aspects. The first is between the Jawi and Roman glyphs. Due to the influence of the Islamic culture, Arabic characters have been absorbed into the Malay writing style and reconciled using six extra characters to make them suitable for Malay speech [10]. After the British came to Malaya in 1511 (now known as Malaysia), they introduced Roman characters to the locals. Since that time, Malaysians have used Roman (Rumi) characters in their writing [11]. Even though Jawi and Rumi have many distinctions, such as the characters themselves, some Jawi glyphs are the same; like Arabic glyph, but have six additional characters that are not covered as Arabic characters. Many outsiders thought that Jawi was an Arabic language. However, the truth is that Jawi belongs to the Malay language. Because Jawi imitates Arabic characters, it does not have upper or lower cases for each character like the Roman (Rumi) style. For this reason, it can be difficult to determine whether nouns in Jawi are proper or common. Jawi characters also have different glyphs for their isolated, initial, medial, and final forms.

The second difference between Jawi and Roman is the spelling method [12]. In Jawi, syllables and characters determine the use of vowels; which is not the case in Rumi. Each syllable in a Rumi word must have a vowel. For example, the word ‘they’ is spelt مرهک/ mrek/ in Jawi; but in Rumi, it is spelt ‘mereka’/mereka/. The word ‘mereka’ contains three syllables in Roman: [e], [e], and [a]. The word مرهک /mrek/ in Jawi only contains the vowel <表决> /e/. Another example is the derivation word بوكاان/ /bukaaan/, which comes from the root word بوك /bak/. This particular root word ends with the vowel [a], preceded by the consonant [k]. The final syllable [ka] should only be spelt with the letter <表决> /k/. Before the suffix {an} /an/ is fixed, the root should be added to the letter <表决>. Only then can the suffix {an} /an/ be fixed [12]. Figure 1 shows another difference of the spelling method between these two scripts.

The third difference between Jawi and Rumi is that a word derived from both English and Arabic is spelt differently from the local Malay word [12]. For example, the English-derived word ‘golf’ should be written as ‘دوف’ /golf/ and not as ‘دوف’ /golf/. Another example is the Arabic-derived word ‘دم’, which contains the short vowel [a]. The vowel [a], which can be represented by the letter <表决>, must be eliminated and spelt ‘담’ /dm/ in Jawi and not as ‘담’ /dam/. An Arabic-derived word that consists of a long vowel is similar to the word ‘ام’. To spell this word, the letter <表决> must be presented as the vowel [a]. Therefore, in this case, ‘ام’ will be spelt ‘امة’ /am/.

3. Related Work

3.1 Stemming Algorithm

The Stemming algorithm is vital because it increases the efficiency of document retrieval systems and reduces the size of the index file. This happens because stemming is able to group many morphological term variants into one single stem. Frakes 1992 [13] found that by storing stems instead of terms, a
single stem usually corresponds to several full terms; this means that compression factors of over 50% can be achieved [14]. Stemming has several disadvantages, such as the information about full terms will be lost or extra storage is necessary to store stemmed and unstemmed forms [13]. Based on Frakes and Yates 1992 work [15], stemming algorithms are divided into four types: Table lookup, N-gram, successor variety, and affix removal.

Many stemmers have been proposed for various languages. The most well-known stemmer for English is the Porter stemmer. Using a set of rules in English, Porter removes the prefix and the suffix [8]. The prefix and suffix rule was changed to accommodate these languages. The Dutch Porter [3] stemmer was also based on the Porter stemmer. This stemmer uses a simple suffix stripper, is very robust, and is easy to implement. One disadvantage of this stemmer is that it is unable handle the irregular word forms of the Dutch language [3]. The Khoja stemmer [16] was developed to remove the longest prefix and suffix. The remaining word was matched with verb and noun patterns to get the correct root using a general rule. The Khoja stemmer [16] is a fast and highly accurate Arabic stemmer. In the Malay language, affixation is divided into four types: prefix, suffix, circumfix, and infix. Sometimes, we need to add another character after the prefix and the circumfix is removed.

3.2 Malay Stemmer

The first Malay stemmer was the Othman stemmer [17]. It was tested on Roman characters and contains 121 deaffixation rules. Asim only tested the accuracy of the stemmer using the accuracy percentage. No further experiments have been reported regarding the stemmer’s strength.

Later, Ahmad [18] improved the rules presented by Othman [17] and tested the stemmer for precision and recall in document retrieval. Ahmad [18] was the first to test the data set from ‘Tafsir al-quran’ for precision and recall, in order to measure the performance of the stemmer. The Ahmad’s [19] stemming algorithm used the root word dictionary to check the generated results after the affixes were removed. The Rule Application Order (RAO) was introduced and the stemmer’s algorithm [19] suggested that the best sequence to apply the deaffixation rules was as follows: prefix, circumfix, suffix, and infix.

Sock et al., [20] proposed another stemmer for Malay words. The stemmer started to stem when it found possible affixes in a word. Heuristics were used as the other criteria in order to minimize errors. The data set was selected based on three Malay websites: http://dbp.gov.my/, http://www.lib.usm.my/press, and http://cyberita.asia1.com.sg. The stemmer was investigated for retrieval performance by calculating precision and recall values. Idris [21] modified the algorithm developed by Ahmad [19]. She proposed extra dictionaries which improved the results of the stemmer. The data set was taken from historical subjects. Therefore, the proposed dictionary was historical and contained terms involved with history. Errors were reduced by both altering the deaffixation rule from Ahmad [19] and using the history dictionary.

Abdullah [22] presented a stemming algorithm and a Rule Frequency Order (RFO). Abdullah [22] also tested his stemmer based on precision and recall in document retrieval. Instead of using the deaffixation sequence suggested by Ahmad [19], Abdullah [22] stemmed each word using its frequency. Although the Abdullah [22] stemmer performed better than the other stemmers, it was still not suitable to be used with Jawi-derived words [7]. The rules produced for the Jawi stemmer and the Rumi stemmer were completely different.

Several researchers have tested stemmer accuracy in Information Retrieval. Ahmad [19] and Abdullah [22] have addressed the subject. Other works, [17], [19], [20], [21], [22], and [23], have only reported on the accuracy of the stemmer, and not the stemmer’s strength. Ahmad [19] and Abdullah’s [22] work has shown that the Malay stemmer can improve stemmer precision and recall in terms of document retrieval.

3.3 Stemming Effectiveness

Stemming effectiveness can be categorised into direct and indirect methods. Direct methods require significant work, which can be described as the measurement of stemming errors, based on a set of data. However, for indirect methods, a particular scope is specified for use with the stemmer: for example, when using a stemmer for document retrieval. Frakes and Fox [8] stated that a stronger stemmer may improve recall, reduce precision, and increase index compression. Frakes and Fox [8] proposed a
method of measuring stemmer strength, which was used by Al-Kabi [4] as a benchmark to assess stemmer performance by Al-Kabi and Al-Mustafa [4], Taghva [26] and Al-Sarhan et al. [8]. From the results produced in [4], Al-Kabi ranked the Rabab’ah stemmer as the best, followed by Al-Sarhan [8], Al-Mustafa [24], and Taghva [26]; based on the accuracy and strength tests of Frakes and Fox [8].

**Frakes Evaluation Metrics**

Frakes and Fox [8] proposed evaluation metrics to calculate stemmer weight and distinguish between stemmers. The following paragraphs describe the details of this method. The mean number of words per conflations class is the average size of the words that match to the same stem for a corpus. Normally, a higher value of mean number of words per conflations class (MWC) signifies the higher accuracy of a stemmer. To calculate ‘the mean number of words per conflations class’ we use the following formula:

\[
MWC = \frac{N}{S}
\]  

(8)

Where, \(N\) refers to the total number of unique corpus words and \(S\) refers to the number of unique root words after the stemming process has taken place. For example, if the corpus contains unique words, such as “لاجك” /brjln/, “لاجك” /pjln/, “لاجك” /jalann/ and “لاجك” /jalnkn/. All of these words were stemmed to the root word “لاجك” /jaln/. Using the equation from (8), the MWC is four.

The Index Compression Factor (ICF) of the stemmer can be computed using Equation (9) below.

\[
ICF = \frac{UW - S}{UW}
\]

(9)

Where, \(UW\) is the total number of unique words before the stemming process, and \(S\) is the number of unique root words after the stemming process has taken place. The value of the ICF represents the number of words that can be compressed by the stemmer. A higher ICF value reflects the greater strength of the stemmer [8].

The Word Change Factor (WCF) is the difference between the number of words before stemming and the number of words after stemming. Sometimes the stemmer will leave words unchanged. \(N\) is the number of unique words and \(C\) is the number of unchanged words. \(N-C\) is the number of words and stems that differ after the stemming has taken place [8]. The WCF is calculated using the formula below:

\[
WCF = \frac{N - C}{N}
\]

(10)

The mean number of characters removed (mean CR) counts the number of characters that are successfully removed by the stemmer. The best stemmer removes more characters, in order to indicate a better root word. For example, the words “لاجك” (brjln), “لاجك” (pjln), “لاجك” (jalann) and “لاجك” (jalnkn) will be stemmed to the root word “لاجك” (jaln). Therefore, the ‘mean CR’ can be computed as \(2 + 2 + 2 + 1\) / \(4\) = 1.75 characters. However, the weakness of this metric is that it does not measure the transformations of stem endings [8].

4. **Jawi Stemmer**

The aim of the Jawi stemmer is to stem local Malay derivatives and focus more on the new Jawi spelling method. This stemmer produces an accuracy of 84.89%, as reported in [7]. The Jawi stemmer [7] was built based on the Jawi rule and the ‘Spelling Error Detector Rule’ (SEDR) [6]. The stemmer used the ‘recording’ technique to cater for spelling exception problems. However, it differs from the Rumi stemmer; especially in the case of the suffix rule, due to the differences between spelling in Rumi and Jawi. For example, in Jawi, the suffix ‘-an’ can be represented by {으-}, {한-}, {원-} or {안-},
depending on the position of the vowel <ٔ> (alif) and its syllable [12]. The first one is the pre-processing process. This process eliminates unnecessary words, such as stop words, punctuation, duplications, and root words. At this time, a list of common root words is used. Next, the deaffixation rule is executed. The Jawi stemmer [6] is different from the Ahmad [19] stemmer in this respect. The best sequence to stem a Jawi-derived word is as follows: circumfix, prefix, suffix, and infix. The last process is the Spelling Error Detector Rule (SEDR). After the deaffixation rule is executed, the result is checked with the SEDR for words with 1, 2, 3, and 4 syllables. Here, SEDR checks the spelling of the stemmed word after the affix is eliminated. The correctly stemmed word is output as a root word. The SEDR was tested using 3018 pieces of data in Jawi, and the accuracy obtained by the experiment was 97.8% [6]. Figure 2 shows the framework for the Jawi stemmer.

5. Evaluating the performance of a stemmer

This experiment was divided into two tests, namely TEST A and TEST B. TEST A was a control experiment, while TEST B applied the SEDR. In test A, all of the stemmers were tested using the same rule and dataset. The root word from the dictionary had not been used in all of the stemmers. Meanwhile in TEST B, the stemmers were tested using the same rule and dataset but this time the SEDR was used to check the stemmed word. 1200 unique derivative words were used as a data set. The data was chosen randomly from an online newspaper [27] [28] and was transliterated using the Transliteration Engine Rumi-Jawi (TERUJA) [29]. Then it was checked by Hamdan Abdul Rahman, a Jawi expert, to ensure there were no errors. The accuracy of all stemmers was calculated from the experiment. From Table 4, the Jawi Stemmer TEST B showed a higher accuracy (85.08%) than Abdullah TEST B (81.17%) and Ahmad TEST B (78.67%) [7].

Table 4. The accuracy of all stemmers

<table>
<thead>
<tr>
<th>TEST A</th>
<th>TEST B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct Root Word</td>
<td>540</td>
</tr>
<tr>
<td>Error</td>
<td>660</td>
</tr>
<tr>
<td>Accuracy (%)</td>
<td>45.00</td>
</tr>
</tbody>
</table>

Figure 3 shows stemmer errors for TEST A and TEST B. Ahmad and Abdullah reported a high number of errors for understemming and overstemming for Test A. Meanwhile, in Test B,
understemming errors decreased from 537 to 147 for Ahmad and from 493 to 135 for Abdullah. Ahmad, Abdullah, and Jawi stemmers performed best in Test B.

The Frakes and Fox [2003] method was used to investigate the stemmer’s performance in TEST A and TEST B. Frakes and Fox [2003] suggested other evaluation metrics to assess the stemmer’s strength, such as mean number per conflation class, word changed factor, index compression factor, and mean number of character removal. Stemmer strength is a degree to which a stemmer changes the words that it stems [8]. Stemming performance is shown as in Table 5.

### Table 5. Summary of stemming performance

<table>
<thead>
<tr>
<th>Error count</th>
<th>Test A</th>
<th>Test B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean number of words per conflation class (MWC)</td>
<td>1.133</td>
<td>1.364</td>
</tr>
<tr>
<td>Word Changed Factor (WCF)</td>
<td>0.990</td>
<td>0.999</td>
</tr>
<tr>
<td>Index Compression Factor (ICF)</td>
<td>0.118</td>
<td>0.267</td>
</tr>
<tr>
<td>The mean number of character removal (mean CR)</td>
<td>2.070</td>
<td>3.055</td>
</tr>
</tbody>
</table>

### 6. Result Analysis

Table 5 shows an improvement in the results between TEST A (control experiment) and TEST B. The metric increased in TEST B because the SEDR was used to select the best-stemmed word after the deaffixation rule was applied.
Figure 4 shows the comparisons of MWC for TEST A and TEST B. The mean number of words per conflation class (MWC) shows a higher value for the Jawi stemmer in TEST A, while the strength value increased for the Jawi stemmer in TEST B. The highest value of MWC indicates the stemmer with the greatest strength.

Figure 5 shows the comparison of WCF for TEST A and TEST B. There is no difference in the Word Changed Factor (WCF) between the Ahmad, Abdullah, and Jawi stemmers, for both TEST A and TEST B. This is because the stemmers use the same rules in their stemming algorithms. The deaffixation rule tries to find as many possible affixations as it can. Because the same rules are used in all stemmers, the WCF values do not differ between the stemmers for each test. For TEST B, the use of SEDR increased the potential accuracy of the stemmed root word to be removed; therefore, the strength value of WCF in TEST B only has a small increment. Next, the Index Compression Factor (ICF) is calculated.
Figure 6. Comparison of the Index Compression Factor (ICF) of each stemmer.

Figure 6 represents a comparison of the Index Compression Factors (ICF) of each stemmer. In terms of the Index Compression Factor (ICF), the strength of each stemmer is nearly identical. However, the highest strength was seen in the Jawi stemmer [7] for TEST B. Again, the order of deaffixation rules applied and the SEDR affected the result of ICF. Next, the Mean Number of Character Removal (CR) is computed.

Figure 7. Comparison of the Mean Number of Character Removal (CR) of each stemmer.

Figure 7 represents a comparison of the Mean Number of Character Removal (CR) of each stemmer. Interims of the Mean Number of Character Removal (CR), the Jawi stemmer [Sulaiman et al., 2011] in TEST A, shows the highest value of all stemmers. When the root word dictionary is eliminated, the algorithm fails to check the correct stem word. In this case, overstemming occurs and the value of the Jawi stemmer [7] in TEST A is increased over the Jawi stemmer [7] in TEST B.

From the results reported Figures 4, 5, 6, and 7, we can state that based on the Frakes method [8], the best stemmer should be able to produce more words per conflation class, have a high compression factor, and remove more characters. According to Table 3, TEST B showed the highest values in MWC, ICF, and WCF. It can be concluded that the results presented in Table 3 show that the Jawi stemmer [7] is the best stemmer, followed by the Abdullah [22] stemmer, and the Ahmad [19] stemmer.

7. Conclusions and future works

This study shows that the strength of the Jawi stemmer can be calculated based on stemmer strength. Four of Frakes’ [8] evaluation metrics were selected for use, namely mean number of words per
conflation class (MWC), index compression factor (ICF), word change factor (WCF), and the mean number of character removal (mean CR). The results show that the Jawi stemmer [7] has the greatest strength, followed by the Abdullah [22] and Ahmad [19] stemmers. Using an appropriate deaffixation rule and SEDR can help to increase the accuracy of the stemmer (as shown in TEST B). The stemmer was also tested to see whether it could boost recall and precision in document retrieval and the further results included statistical tests will be reported in the next paper.