A Comparative Study of Stemming Techniques on the Malay Text

Rosmayati Mohemad, Nazratul Naziah Mohd Muhait, Noor Maizura Mohamad Noor, Nur Fadilla Akma Mamat
Faculty of Ocean Engineering Technology and Informatics, Universiti Malaysia Terengganu,
21030 Kuala Nerus, Terengganu, Malaysia

Abstract—**Text stemming, an essential preprocessing step in the development of Natural Language Processing (NLP) applications, involves the transformation of various word forms into their root words. Stemming plays a critical role in decreasing the volume of text, thereby enhancing the efficiency of various computational tasks such as information retrieval, text classification, and text clustering. Stemming is a rule-based approach. On the other hand, it frequently suffers affixation errors that result in under-stemming, over-stemming, or both, as well as unstemmed or spelling exceptions. Every language has different stemming techniques, and among the most well-known Malay stemming algorithms are the Othman and Ahmad algorithms. Therefore, this study aims to compare the performance of the stemming errors between the Othman and Ahmad algorithms in stemming Malay text, particularly on two different domains of textual datasets, which are the course summaries of the education domain and housebreaking crime reports of the crime domain. The Othman algorithm presents a set of 121 stemming rules (set A). In the meantime, Ahmad's algorithm proposes two distinct sets of stemming rules, comprising 432 (set B) and 561 rules (set C), respectively. Based on the experiment results with 100 course summaries, the Ahmad algorithm (Set B) obtained a higher accuracy rate of 93.61%. The second highest is the Ahmad algorithm (Set C) with 93.53%. The Othman algorithm achieved the lowest accuracy with 86.04% compared to the other two algorithms. Meanwhile, findings from the experiment with 100 housebreaking crime reports show similar results, with the Ahmad algorithm (Set C) achieving the highest stemming accuracy of approximately 93.80% and the Othman algorithm producing the lowest stemming accuracy (83.09%). The result indicates that stemming accuracy is consistent across different types of datasets.**

Keywords—Algorithm; ahmad algorithm; malay language; othman algorithm; rule-based; stemming; stemmer

I. INTRODUCTION

Due to the explosive growth of textual information that has continuously been generated from electronic media over the past ten years, text analytics has received a lot of attention and research [1], [2]. Text analytics, sometimes called text mining, is the integration of linguistic, computational statistics, and computer science techniques [3]. The research on text analytics has been conducted in a variety of languages, including English [4], Arabic [5], Russian [6], Indian [7], Chinese [8], and Thailand [9] to solve various problems in text classification, text clustering, sentiment analysis, and topic modeling. Text analytics have a significant impact on data science [10], [11]. The significant contribution of text analytics in terms of providing enriched and structured datasets that enable in-depth analysis. This has led researchers to explore various text mining techniques, which involve the extraction of valuable information and insights from unstructured textual data. Unstructured textual information requires preprocessing, involving the removal of irrelevant terms, before it can be used in text analytics tasks. Preprocessing is important for decreasing data sparsity, reducing data dimensionality, increasing the quantity of captured semantic information, and ensuring data consistency [12]. The common steps in text preprocessing are tokenization (splitting text into words or phrases), stop word removal, and stemming [13], [14]. Working with a large volume of dimensional text could negatively affect the performance of text analytics. Therefore, stemming is critical for improving the effectiveness of text analytics.

Stemming is one of the basic and essential steps in text preprocessing. It is a natural language processing (NLP) technique used to match the numerous inflectional and derivational morphological forms of a word to its stem or root word. For example, stemming matches the words *maintaining*, *maintained*, and *maintenance* to their root word, *maintain*. Meanwhile, stemmers are the programs that do stemming [15]. In Malay, there are seven-word patterns: affixation, reduplication, compounding, blending, clipping, abbreviation, and borrowing [16]. Malay morphology is recognised for having extremely complex morphological features that are used to construct different word patterns. For instance, the addition of the prefix "pe" to the root word "makan" (eat) results in the word "pemakan" (eater), thereby modifying the meaning of the root word. Thus, it is crucial to understand the morphological structure of the Malay language to reduce derived words into their respective root words.

NLP employs various stemming approaches such as rule-based, dictionary-based, statistical-based, and hybrid stemming. The best approach is determined by the language involved and the nature of the textual dataset. The stemming strategy of rule-based affix elimination is used in this paper, which eliminates the prefix, suffix, and circumfix infix. Othman [17] and Ahmad [18] algorithms are the most pioneering rule-based Malay stemmers. Even though there are plenty of rule-based stemming approaches for Malay that have been improved by the previous researchers since then, they still suffer from affixation errors, including over-stemming, under-stemming, unchanged, and spelling exceptions [19], [20], [21]. The major causes of this stemming error are the affix removal method, the similarity of the root word with the affixation word, and exception rules in prefixation and confixation [22],[23].

The quality of stemming algorithms is typically measured by how accurately they map the variant forms of a word to the same stem. In addition, the presence of diverse morphological structures within a textual dataset, the proper use of appropriate vocabularies, and the word patterns in the textual datasets also play a significant role in contributing to the accuracy of stemming. To the best of our knowledge, there is limited study on the comparative analysis of stemming algorithms for Malay. The most recent works were published in [24] and [25]. However, the purpose of this paper is not to discuss any improvements in terms of the morphological rules of Malays languages. Nevertheless, the purpose of this research is to determine the degree to which the abundance and consistency of precise vocabulary and grammar usage in the textual dataset influence the efficacy of the stemming procedure. Therefore, the objective of this study is to conduct a comparative analysis of the Othman and Ahmad algorithms in terms of their effectiveness in stemming textual data from two distinct domains: education and crime. The analysis of the error rate in stemming and the accuracy of performance for different textual datasets are performed. The analysis is conducted utilising the 121 rules of the Othman algorithm, referred to as set A, as well as the Ahmad algorithm for both set B (comprising 432 rules) and set C (comprising 561 rules). A total of 100 Malay documents for each domain is randomly selected in order to assess the performance of this study. The best result was obtained by the Ahmad algorithm for both datasets, with a stemming accuracy of 93.61% for the education dataset and 93.80% for the crime dataset. Meanwhile, the Othman algorithm attains a stemming accuracy of 86.04% for the education dataset and 83.09% for the crime dataset.

This paper is organised into five sections. Section II discusses the related works on the existing Malay word stemmer. Meanwhile, Section III describes the research methodology used to compare Malay stemming algorithms. Section IV presents the experiment's results and discussion. Finally, Section V concludes the paper by summarising the main achievements and making future recommendations.

II. Related Work

This section discusses a selection of recent studies on Malay stemming algorithms that have employed a rule-based affix elimination approach, which involves the removal of prefixes, suffixes, circumfixes, and infixes. A prefix is a type of affix that is attached to the beginning of a root word, while a suffix is a specific type of affix that is appended to the last position of a root word. In the realm of linguistics, it is worth noting that a linguistic element known as a circumfix, or more formally referred to as a prefix-suffix, is an affix that is comprised of two parts. Both of these parts are strategically positioned, with one located at the beginning of the root word, while the other is attached to the end of the root word. An infix is a type of affix that is inserted in the middle of a root word.

The most pioneering stemmer for Malay is the Othman algorithm, which was proposed in 1993 [22]. This algorithm makes use of Kamus Dewan 1991, a Malaysian dictionary. The use of the dictionary facilitates the identification of the root word after the removal of the affixes, provided that the affixes are matched to the stemming rules. By using the pattern matching rules, the affixes of the word are eliminated once the rule is matched. However, this stemmer has caused over-stemming errors because it does not consider searching for the word in the dictionary before performing the stemming process. Following this, Ahmad et al. [23] proposed a modified version of the Othman algorithm, known as the Ahmad algorithm, in which the algorithm considers dictionary lookups before proceeding to the stemming process and improves the order of applied morphological rules. In addition to enhancing stemming performance, two sets of new rules are developed, each consisting of 432 rules and 561 rules. This algorithm, which uses the rule application order, has been performed on two datasets of ten chapters of the Quran and 10 research abstracts. A series of empirical experiments are run, and the results reveal that the best order for rule-based affix elimination is prefix, circumfix, suffix, and infix, whereas the Ahmad algorithm produces a better performance compared to the Othman algorithm.

Meanwhile, Sankupellay & Valliapan [24] developed the Mangalam algorithm, where they adopted the Porter stemming algorithm to stem Malay documents. Although the Porter stemmer is commonly used to stem English words, the algorithm is adaptable to handle dual words, or “kata ganda" from Malay documents. The Porter stemmer was successfully adopted in stemming Indonesian text [25]. The stemmer used a root word dictionary to validate the four categories of affixes, including inflection particles, possessive pronouns, derivation suffixes, and derivation prefixes. Some of the studies conducted by Rosid et al. [26] used the Sastrawi library in their studies to examine the comparison of stemming result against Tala porter stemmer. Tala porter stemmer was developed by Fadilah Z. Tala in 2003 using five steps in Porter stemmer by imitating how words are derived and inflected [27]. This study used 50 of the Indonesian student complaint documents. The findings of the study indicate that employing the Sastrawi dictionary yields superior results in comparison to utilising the Tala Porter dictionary. The result shows 92% accuracy when they used the Sastrawi dictionary, and 82% when they just used the Tala Porter stemmer. The processing speed for Sastrawi libraries is faster than Tala Porter with 0.6 seconds compared to 241.6 seconds for Tala Porter.

III. Research Methodology

The overall framework of research methodology in this study is depicted in Fig. 1. There are three main phases, including textual data collection, text preprocessing, and performance evaluation.

A. Phase 1: Textual Data Collection

This study makes use of textual datasets from the following two primary domains: education and housebreaking crime. The first textual dataset employed in this study consists of a compilation of course summaries in the Malay language. Meanwhile, the second dataset comprises a collection of housebreaking crime reports ranging from 2010 to 2013, also in the Malay language. The two datasets were obtained from a higher education institution and the Royal Malaysia Police Department, and they are both closed domain datasets. Table I shows the detailed descriptions of both textual datasets. For the purposes of this study, a sample of 100 course summaries and 100 housebreaking crime reports was randomly chosen and stored in Excel format. The collection of 100 documents of course summaries comprises a cumulative total of 5,520 words, whereas the set of 100 housebreaking crime reports contains a total of 3,530 words. The range of word lengths observed in the course summaries documents spans from 24 to 135 words, while the housebreaking crime reports consist of approximately 22 to 155 words.

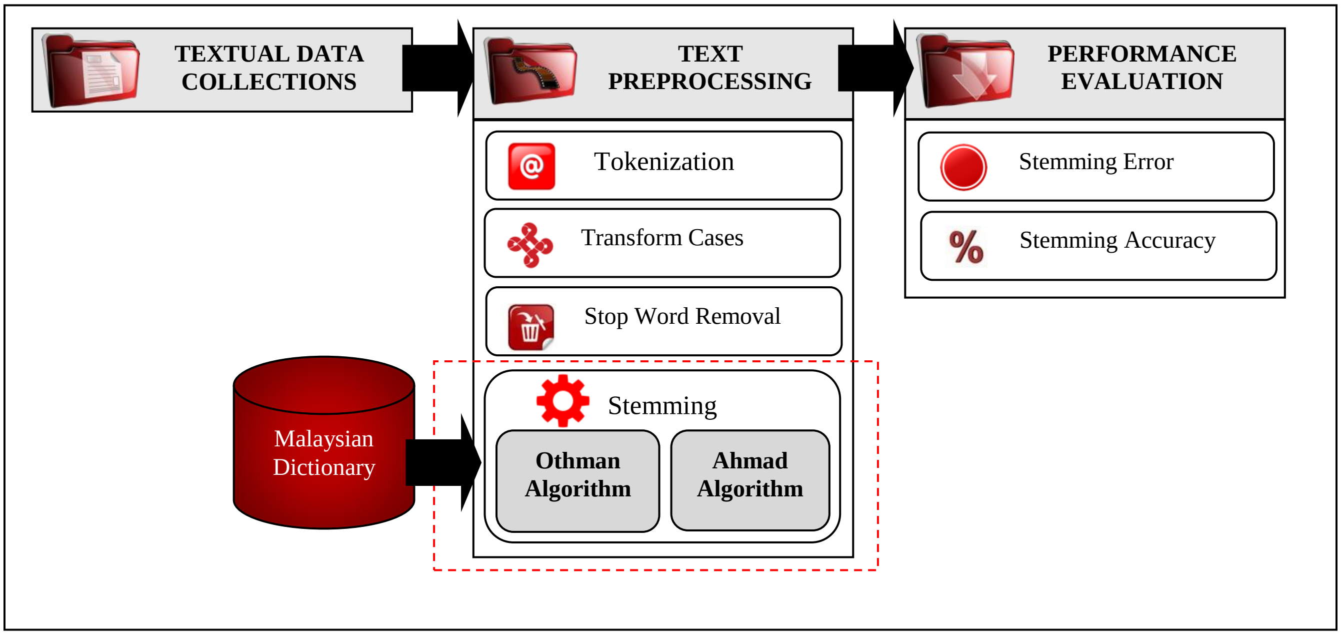

Fig. 1. Framework of research methodology.

TABLE I. DETAILS DESCRIPTION OF TEXTUAL DATASET FOR EDUCATION AND HOUSEBREAKING CRIME DOMAIN

Domain	Number of Documents	Total Number of Words	Range of Word Lengths
Education	100	5520	24-135
Housebreaking Crime	100	3530	22-155

B. *Phase 2: Text Preprocessing*

The text processing phase contains several steps, such as tokenization, transform cases, stop word removal, and stemming. These steps are essential to reducing the noise of the words in the raw dataset. It is crucial to reduce the document's feature size before proceeding to the next computational task. Tokenization processes are splitting the sentence or paragraph into single words, while transform cases are the process of converting all words to lowercase. A stop word refers to a word that is highly prevalent and frequently occurs in both written and spoken language but does not contribute significant semantic meaning to the overall document. Examples of stop words include prepositions, conjunctions, numbers, and punctuation marks. In Malay, examples of stop words are “di” (at), “ke” (to), “dengan” (with), and “dari” (from). In this experimental study, a total of 323 stop words were identified and subsequently employed to eliminate unnecessary words from the dataset. Table II presents a list of 100 stop words that were utilised in the context of this research.

TABLE II. AN EXAMPLE OF LIST OF STOP WORDS IN MALAY LANGUAGE

List of Stop Words				
ada	seandainya	kalau	apa-apa	sekitar
inikah	agar	sebelumnya	katakan	atau
sampai	janganlah	allah	segala	kepadaku
adakah	sebab	kami	apabila	selain
inilah	akan	sebenarnya	ke	ataukah
sana	jika	amat	sehingga	kepada
adakan	sebagai	kamikah	apakah	selalu
itu	aku	secara	kecuali	ataupun
sangat	jikalau	antara	sejak	kepadamu
adalah	keatas	kamipun	apapun	selama
itukah	akulah	sedang	kelak	bagaimana
sangatlah	jua	antaramu	sekalian	kepadanya
adanya	sebanyak	kamu	atas	diatas
itulah	akupun	sedangkan	kembali	di
saya	juapun	antaranya	sekalipun	samping
adapun	sebelum	kamukah	atasmu	seluruh
jadi	al	sedikit	kemudian	bagi
se	juga	apa	sekarang	kerana
agak	dari	kamupun	atasnya	seluruhnya
jangan	alangkah	sedikitpun	kepada	bagimu

The subsequent procedure involves the execution of the stemming process. Comparative experiments were conducted to evaluate the effectiveness of the Othman and Ahmad algorithms for stemming. Both algorithms are being implemented using the Java programming language. Fig. 2 depicts the Othman algorithm, while Fig. 3 depicts the Ahmad algorithm. The algorithms are combined with the proposed rules, with Othman employing 121 stemming rules (Set A) and Ahmad employing 432 (Set B) and 561 (Set C) stemming rules, respectively. Three distinct programmes have been developed to represent different sets of rules, in particular Othman Stemmer (Set A), Ahmad Stemmer (Set B), and Ahmad Stemmer (Set C). Meanwhile, six separate series of experiments were conducted to stem two different textual datasets from the education and housebreaking crime domains. The resulting stem words were stored in Excel files. The Malaysian dictionary used in this study consists of a collection of root words obtained from the Kamus Dewan Edisi Keempat.

```
Step-1: If there are no more words then stop, otherwise get the next word.
Step-2: If there are no more rules then accept the word as a root word and go
        to Step-1, otherwise get the next rule.
Step-3: Check the given pattern of the rule with the word: If the system finds
        a match, apply the rule to the word to get a stem word.
Step-4: Check the stem word against the dictionary; perform any necessary
        recoding and recheck the dictionary.
Step-5: If the stem word appears in the dictionary, then this stem word is the
        root of the word and go to Step-1. Otherwise go to Step-2.
```

Fig. 2. Othman algorithm [23].

```
Step-1: Find the next word till the last word.
Step-2: Check the word in the dictionary; if the word appears in dictionary, it
        is the root word and return to Step 1.
Step 3: Get the next rule; if no further rules are available, the word are root
        word and return to Step 1.
Step 4: Apply the rule to the word to get a stem word.
Step 5: Check the dictionary and recode for prefix spelling exceptions.
Step 6: If the stem word appears in the dictionary, it is root word and proceed
        to Step 1; otherwise go to Step 7.
Step 7: Examine the stem from Step 4 for spelling variations in the dictionary.
Step 8: If the word stem appears in the dictionary, it is root word and proceed
        to Step 1; otherwise go to Step 9.
Step 9: Check the dictionary and recode for suffix spelling exceptions.
Step 10: If the stem word appears in the dictionary, it is root word and
        proceed to Step 1; otherwise go to Step 3.
```

Fig. 3. Ahmad algorithm [23].

C. Performance Evaluation

After a series of experiments is run, the stemming result is collected for measuring the algorithm's performance. The performance value that has been considered in this study is the analysis of stemming errors, word stemming accuracy measurement, and processing speed. The analysis of stemming errors involves the examination of various types of errors, such as over-stemming, under-stemming, unstemmed words, and spelling exception errors. These errors have a negative impact on the efficacy of stemming algorithms. Over-stemming occurs when a larger portion of a word is cut off than is necessary, resulting in the incorrect reduction of an inappropriate stem word. As an example, the word “memakan” (eating) needs to be stemmed into “makan” (eat), but then the result gives only “mak” (mother) after stemming. Under-stemming, meanwhile, happens when the smaller portion of the word is stemmed into the inappropriate stem word. For instance, the word “pengadu” (informer) should be stemmed into “adu” (inform), but the result gives “gadu” (trigonostemon longifolius). For unstemmed words, there are no changes in the derivative word before stemming or after stemming. Spelling exceptions occur when a word is chopped off from the accurate affixes, resulting in the formation of different root words. For example, the word “pengawal” (guard) is stemmed into “gawal” (confuse) instead of “kawal” (control). The other category for stemming errors refers to results other than these four types.

Meanwhile, for measuring the stemming accuracy, The equation below depicts the formula. Correctly stemmed words are calculated by conducting a comparison with the manually stemmed data and the root word as listed in the Malaysian dictionary, Kamus Dewan Edisi Keempat.

$$Accuracy = \frac{Total\ of\ correctly\ stemmed\ word}{Total\ of\ correctly\ and\ incorrectly\ stemmed\ words} x\ 100\%$$

In regard to processing speed, the duration is measured subsequent to the completion of the stemming process. The experiment was conducted utilising the Java Programming language within the NetBeans Integrated Development Environment, Version 12.0. The stemmer programme is executed on a desktop computer equipped with an Intel(R) Core(TM) i7-10700 CPU @ 2.90GHz 2.90 GHz processor and 16 GB of DDR3 memory.

IV. Result and Discussion

This section discusses the results of the experiments and the evaluation of the stemming process.

A. Analysis of the Stemming Errors

Despite the ongoing encouragement for the development of the Malay stemmer, there are still several challenges that need to be addressed. The analysis of stemming errors involves the examination of various types of errors, such as over-stemming, under-stemming, unstemmed words, and incorrect stemming.

Table III provides a summary of the total number of stop words that were eliminated from the education dataset and the housebreaking crime dataset. The implementation of the stop word removal process resulted in the elimination of 1,555 insignificant words in total from a corpus consisting of 100 course summaries. In the context of analysing 100 housebreaking crime reports, there are 864 frequently occurring words with no significant meaning that were eliminated. The stop word removal process results in a total word count of 3,965 for the course summaries dataset, while the housebreaking crime dataset retains 2,486 words.

TABLE III. A Summary of the Total Number of Eliminated Words After Stop Word Removal

Textual Dataset	Number of Removed Words	Total Number of Remaining Words
Course Summaries	1555	3965
Housebreaking Crime Reports	864	2486

Meanwhile, a comparative analysis of stemming errors generated by the Othman and Ahmad algorithms is presented in Table IV. Experiments are conducted in six series, with Set A containing 121 rules for the Othman algorithm, Set B containing 432 rules, and Set C containing 561 rules for the Ahmad algorithm. By using a collection of course summaries as the dataset in the education domain, the Othman algorithm correctly stems 1,582 words and identifies 1,643 root words. The Ahmad algorithm with Set B rules correctly stems 1,529 words, while the Ahmad algorithm with Set C rules correctly stems 1,526 words. In contrast to the Othman algorithm, which manages to identify only 1,643 root words, the Ahmad algorithm detects 1,959 root words for both sets. The Ahmad algorithm yields a higher number of under-stemmed words within the range of 168 to 174 in comparison to the Othman algorithm, which yields 130 under-stemmed words. In contrast, the Othman algorithm demonstrates a higher frequency of over-stemming errors, with 366 over-stemmed words, in comparison to the Ahmad algorithm, with 45 and 54 over-stemmed words for Set B and Set C, respectively. Meanwhile, the number of spelling exception errors generated by the Ahmad algorithm is greater in comparison to the Othman algorithm. Irrelevant words refer to root words that are not listed in the dictionary.

On the other hand, for the housebreaking crime dataset of the crime domain, the Othman and Ahmad algorithms for Set B and Set C achieve within the range of 736 to 737 words, where the difference between the two algorithms is not significantly different. However, with regard to root word identification, the Ahmad algorithm successfully detects 1,531 root words for Set B and Set C, while the Othman algorithm only detects 1,273 root words. In the context of under-stemming errors, the Ahmad algorithm yields a total of 113 under-stemmed words, which is higher than the count of 110 under-stemmed words produced by the Othman algorithm. In the meantime, the Othman algorithm also produced 285 over-stemmed words, which is higher than the Ahmad algorithm, which produced within the range of 21 to 22 over-stemmed words for both Set B and Set C. Meanwhile, the spelling exceptions generated by the Othman and Ahmad algorithms are within the range of 14 to 16 words.

TABLE IV. SUMMARY OF STEMMING ERRORS PRODUCED BY OTHMAN AND AHMAD ALGORITHMS FOR 100 COURSE SUMMARIES AND 100 HOUSEBREAKING CRIME REPORTS

Algorithms	**Othman Algorithm (SET A)**				**Ahmad Algorithm (Set B)**				**Ahmad Algorithm (Set C)**			
Dataset Domain	***Education***		***Crime***		***Education***		***Crime***		***Education***		***Crime***	
	#	**%**	**#**	**%**	**#**	**%**	**#**	**%**	**#**	**%**	**#**	**%**
Correct Stemming	1582	39.90%	736	29.61%	1529	38.56%	736	29.61%	1526	38.49%	737	29.65%
No stemming, root word	1643	41.44%	1273	51.21%	1959	49.41%	1531	61.58%	1959	49.08%	1531	61.58%
Under-stemming	130	3.28%	110	4.42%	174	4.39%	113	4.55%	168	4.24%	113	4.55%
Over-stemming	366	9.23%	285	11.46%	45	1.13%	22	0.88%	54	1.36%	21	0.84%
Spelling exceptions	5	0.13%	14	0.56%	19	0.48%	16	0.64%	19	0.48%	16	0.64%
Irrelevant Words	239	6.03%	68	2.74%	239	6.03%	68	2.74%	239	6.03%	68	2.74%

When these two distinct dataset domains are compared, it is discovered that the Othman algorithm outperforms the Ahmad algorithm in correctly stemming 39.66% of the words in the education dataset. In contrast, the Ahmad algorithm has a higher success rate of 29.65% in correctly stemming words from the crime dataset compared to the Othman algorithm. The aforementioned contrast finding indicates that the efficacy of the stemming process is influenced not only by the variety of morphological rules, but also by the extent of vocabulary richness present in the datasets, which has the potential to impact the accuracy of stemming. Meanwhile, the Ahmad algorithm demonstrates superior performance in root word identification for the education and crime datasets, with respective efficiencies of 49.08% to 49.41% and 61.58%. This observation aligns with the technique employed by Ahmad's algorithm, wherein the search process is conducted on the dictionary prior to the application of the stemming rules.

In the context of under-stemming errors, the Ahmad algorithm for Set B produces higher under-stemmed words, which is around 4.39% for the education domain and 4.55% for the crime domain. These percentages represent the ratio of under-stemmed words to the overall count of words remaining subsequent to the elimination of stop words. This is the result of employing a dictionary lookup function, which verifies the existence of the root word in the stemmed word even after it has been stemmed by a single rule. In the situation where the stemmed word is present in the dictionary, the evaluation against subsequent stemming rules stops, and the word is deemed a root word. Meanwhile, the Othman algorithm produces a higher frequency of over-stemming errors, which is around 9.46% for the education dataset and 11.46% for the crime domain. This demonstrates that the diversity of morphological rules is crucial for reducing over-stemming errors in the Malay language, which has a complex morphological structure due to the presence of affixes.

B. Stemming Acuracy

The accuracy of stemming performance is assessed by computing the ratio of correctly stemmed words to the sum of correctly and incorrectly stemmed words. Correctly stemmed words are recognised as the word that has been truncated to the appropriate root word and the word that has been correctly identified as the root word. On the other hand, words that have been incorrectly stemmed are classified as either under-stemmed, over-stemmed, or words with spelling exceptions.

Table V and Fig. 4 present a comparative analysis of the performance accuracy achieved by the Othman and Ahmad algorithms in the domains of education and crime. The Ahmad algorithm with Set B rules, when applied to the education dataset consisting of 100 course summaries, demonstrates the highest level of accuracy in stemming performance, achieving a rate of 93.61%. This accuracy rate is only marginally different from the Ahmad algorithm with Set C rules, which achieves a rate of 93.53%. In contrast, the Othman algorithm yields the least accurate results, approximately 86.04%.

In the context of a crime dataset consisting of 100 housebreaking crime reports, the Ahmad algorithm, when employing Set C rules, achieves the highest level of accuracy in stemming performance, with a rate of 93.80%. However, the difference in stemming accuracy achieved by implementing Set B rules is minimal, with a mere 0.04% variation. Meanwhile, the Othman algorithm exhibits a comparatively lower level of accuracy, measuring at 83.09%.

TABLE V. COMPARATIVE ANALYSIS OF ACCURACY PERFORMANCE FOR OTHMAN ALGORITHM AND AHMAD ALGORITHM

Algorithms	**Education Domain**					**Crime Domain**				
	Correctly stemmed words		***Incorrectly stemmed words***	***Correctly stemmed words + Incorrectly stemmed words***	***Accuracy (%)***	***Correctly stemmed words***		***Incorrectly stemmed words***	***Correctly stemmed words + Incorrectly stemmed words***	***Accuracy (%)***
	Correct stemmed word	***Correct root word identified***				***Correct stemmed word***	***Correct root word identified***			
Othman Algorithm (SET A)	1572	1633	520	3725	86.04%	736	1273	409	2418	83.09%
Ahmad Algorithm (SET B)	1529	1959	238	3726	93.61%	736	1531	151	2418	93.76%
Ahmad Algorithm (SET C)	1526	1959	241	3726	93.53%	737	1531	150	2418	93.80%

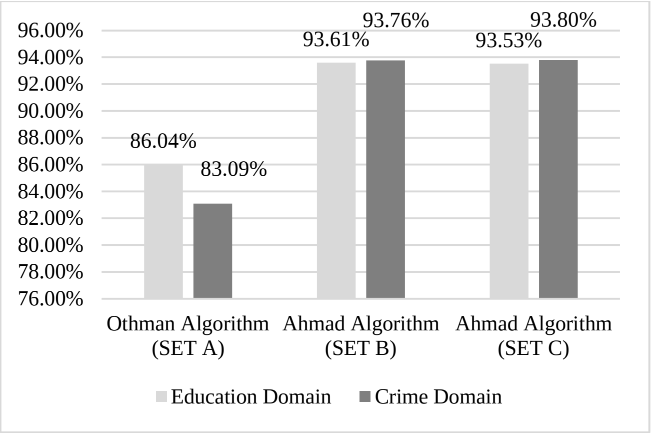

Fig. 4. Stemming accuracy for othman algorithm and ahmad algorithm.

The Ahmad algorithm consistently exhibits superior accuracy in stemming performance when compared to the Othman algorithm, irrespective of the nature of the datasets used. The findings show that the variants of morphological rules and the dictionary lookup approach are significantly contributing to the overall stemming accuracy.

V. CONCLUSION

The purpose of this study is to perform a comparative analysis of the stemming performance of the Ahmad and Othman algorithms, two prominent and pioneering Malay stemming algorithms, on housebreaking crime reports. The Ahmad algorithm has the greatest stemming accuracy rate, indicating that it is extremely unreliable for producing stem words across all dataset domains. The insignificant difference in stemming accuracy when Set B, which consists of 432 rules, and Set C that contains 561 rules, are implemented in the Ahmad algorithm indicates that the 432 rules are sufficiently significant to yield favourable outcomes, whether for stemming the dataset in the domain of education or crime. This is evident from the stemming accuracy results, which indicate a difference of approximately 0.08% and 0.04% between these two sets of rules applied to the education data and crime data set, respectively. Meanwhile, the dataset comprising textual records of housebreaking crime reports exhibits a substantial presence of root words, thereby requiring less effort for stemming operations. The performance of stemming accuracy is significantly impacted by this factor, as the quantity of root words present in the dataset directly correlates with the number of correctly stemmed words. There is a decreased probability of stemming errors occurring when fewer stemming operations are necessary. In addition, a restricted vocabulary range, and a notable prevalence of word repetition in the textual dataset also contribute to the stemming result. Henceforth, a thorough assessment of the functionalities of these two algorithms may be extended to encompass additional textual datasets that are more vocabulary-rich.

ACKNOWLEDGMENT

This research is supported by Ministry of Education Malaysia, under Fundamental Research Grant Scheme (FRGS) with vote number 59676 (Reference Code: FRGS/1/2021/ICT03/UMT/01/1).

REFERENCES

[1] Ittoo, L. M. Nguyen, and A. Van Den Bosch, “Text analytics in industry: Challenges, desiderata and trends,” Computers in Industry, vol. 78. Elsevier B.V., pp. 96–107, May 01, 2016. doi: 10.1016/j.compind.2015.12.001.

[2] S. J. Barnes, M. Diaz, and M. Arnaboldi, “Understanding panic buying during COVID-19: A text analytics approach,” Expert Syst. Appl., vol. 169, no. November 2020, p. 114360, 2021, doi: 10.1016/j.eswa.2020.114360.

[3] I. Feinerer, K. Hornik, and D. Meyer, “Text mining infrastructure in R,” J. Stat. Softw., vol. 25, no. 5, pp. 1–54, 2008, doi: 10.18637/jss.v025.i05.

[4] P. Carracedo, R. Puertas, and L. Marti, “Research lines on the impact of the COVID-19 pandemic on business. A text mining analysis,” J. Bus.

Res., vol. 132, no. September 2020, pp. 586–593, 2021, doi: 10.1016/j.jbusres.2020.11.043.

[5] S. Hassan, H. Mubarak, A. Abdelali, and K. Darwish, "ASAD: Arabic social media analytics and understanding," EACL 2021 - 16th Conf. Eur. Chapter Assoc. Comput. Linguist. Proc. Syst. Demonstr., pp. 113–118, 2021, doi: 10.18653/v1/2021.eacl-demos.14.

[6] V. Y. Radygin, D. Y. Kupriyanov, R. A. Bessonov, M. N. Ivanov, and I. V. Osliakova, "Application of text mining technologies in Russian language for solving the problems of primary financial monitoring," Procedia Comput. Sci., vol. 190, no. 2019, pp. 678–683, 2021, doi: 10.1016/j.procs.2021.06.078.

[7] S. V. Praveen, R. Ittamalla, and G. Deepak, "Analyzing the attitude of Indian citizens towards COVID-19 vaccine – A text analytics study," Diabetes Metab. Syndr. Clin. Res. Rev., vol. 15, no. 2, pp. 595–599, 2021, doi: 10.1016/j.dsx.2021.02.031.

[8] N. Zhang, Q. Jia, K. Yin, L. Dong, F. Gao, and N. Hua, "Conceptualized Representation Learning for Chinese Biomedical Text Mining," no. 1, pp. 1–4, 2020, [Online]. Available: http://arxiv.org/abs/2008.10813

[9] W. Chansanam and K. Tuamsuk, "Thai twitter sentiment analysis: Performance monitoring of politics in Thailand using text mining techniques," Int. J. Innov. Creat. Chang., vol. 11, no. 12, pp. 436–452, 2020.

[10] A. Rizk and A. Elragal, "Data science: developing theoretical contributions in information systems via text analytics," J. Big Data, vol. 7, no. 1, 2020, doi: 10.1186/s40537-019-0280-6.

[11] M. L. Carnot, J. Bernardino, N. Laranjeiro, and H. G. Oliveira, "Applying text analytics for studying research trends in dependability," Entropy, vol. 22, no. 11, pp. 1–20, 2020, doi: 10.3390/e22111303.

[12] L. Hickman, S. Thapa, L. Tay, M. Cao, and P. Srinivasan, "Text preprocessing for text mining in organizational research: Review and recommendations," Organ. Res. Methods, vol. 25, no. 1, pp. 114–146, 2022.

[13] R. A. Sinoara, J. Antunes, and S. O. Rezende, "Text mining and semantics: a systematic mapping study," J. Brazilian Comput. Soc., vol. 23, no. 1, 2017, doi: 10.1186/s13173-017-0058-7.

[14] N. Wang, J. Zeng, M. Ye, and M. Chen, "Text mining and sustainable clusters from unstructured data in cloud computing," Cluster Comput., vol. 21, no. 1, pp. 779–788, 2017, doi: 10.1007/s10586-017-0909-1.

[15] J. Singh and V. Gupta, A systematic review of text stemming techniques, vol. 48, no. 2. Springer Netherlands, 2017. doi: 10.1007/s10462-016-9498-2.

[16] M. N. Kassim, M. A. Maarof, A. Zainal, and A. A. Wahab, "Word stemming challenges in Malay texts: A literature review," 2016 4th Int. Conf. Inf. Commun. Technol. ICoICT 2016, vol. 4, no. c, 2016, doi: 10.1109/ICoICT.2016.7571887.

[17] M. N. Kassim, S. H. M. Jali, M. A. Maarof, and A. Zainal, "Towards stemming error reduction for Malay texts," Lect. Notes Electr. Eng., vol. 481, pp. 13–23, 2019, doi: 10.1007/978-981-13-2622-6_2.

[18] N. I. and S. M. F. D. S. MUSTAFA, "Stemming For Term Conflation In Malay Texts," 2001.

[19] M. Yasukawa, H. T. Lim, and H. Yokoo, "Stemming malay text and its application in automatic text categorization," IEICE Trans. Inf. Syst., vol. E92-D, no. 12, pp. 2351–2359, 2009, doi: 10.1587/transinf.E92.D.2351.

[20] M. M. N. M. Kassim, M. M. A. M. Maarof, A. Zainal, and A. A. Wahab, "Enhanced Affixation Word Stemmer with Stemming Error Reducer to Solve Affxation Stemming Errors," J. Telecommun. Electron. Comput. Eng., vol. Vol 8, no. No. 3, pp. 37–41, 2016, [Online]. Available: http://journal.utem.edu.my/index.php/jtec/article/view/999

[21] M. Abdullah and F. Ahmad, "Rules frequency order stemmer for malay language," … Int. J. …, vol. 9, no. 2, pp. 433–438, 2009, [Online]. Available: http://paper.ijcsns.org/07_book/200902/20090258.pdf

[22] R. Alfred, L. C. Leong, C. K. On, and P. Anthony, "A Literature Review and Discussion of Malay Rule - Based Affix Elimination Algorithms," pp. 285–297, 2014, doi: 10.1007/978-94-007-7287-8.

[23] F. Ahmad, M. Yusoff, and T. M. T. Sembok, "Experiments with a stemming algorithm for Malay words," J. Am. Soc. Inf. Sci., vol. 47, no. 12, pp. 909–918, 1996, doi: 10.1002/(SICI)1097-4571(199612)47:12<909::AID-ASI4>3.0.CO;2-6.

[24] M. Sankupellay and S. Valliappan, "Malay-language stemmer," Sunw. Acad. J., vol. 3, pp. 147–153, 2006.

[25] M. A. Nazief, Bobby, "'Confix Stripping: Approach to Stemming Algorithm for Bahasa Indonesia.,'" Intern. Publ. Fac. Comput. Sci. Univ. Indones. Depok, Jakarta, 1996.

[26] M. A. Rosid, A. S. Fitrani, I. R. I. Astutik, N. I. Mulloh, and H. A. Gozali, "Improving Text Preprocessing for Student Complaint Document Classification Using Sastrawi," IOP Conf. Ser. Mater. Sci. Eng., vol. 874, no. 1, 2020, doi: 10.1088/1757-899X/874/1/012017.

[27] F. Z. Tala, "A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia," M.Sc. Thesis, Append. D, vol. pp, pp. 39–46, 2003.