Mathematical Sciences and Informatics Journal
Vol. 1, No 2, Nov. 2020, pp. 34-42
http://www.mijuitmjournal.com
DOI : 10.24191/mij.v1i2.14183

An Application of Malay Short-Form Word Conversion Using Levenshtein Distance

Azilawati Azizan
Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Perak Branch Tapah Campus, Perak, Malaysia
azila899@uitm.edu.my

NurAine Saidin
Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Perak Branch Tapah Campus, Perak, Malaysia
2017412258@isiswa.uitm.edu.my

Nurkhairizan Khairudin
Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Perak Branch Tapah Campus, Perak, Malaysia
nurkh098@uitm.edu.my

Rohana Ismail
Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Kampus Besut, 22200 Besut Terengganu, Malaysia
rohana@unisza.edu.my

Article Info

Article history:

Received July 01, 2020
Revised Aug. 10, 2020
Accepted Sept. 30, 2020

Keywords:

Malay short form word
Noisy text normalization
Levenshtein Distance
Rule-based

ABSTRACT

Formerly, short-form word was widely used in the field of journalism. However, nowadays, short-form word has been widely used by many people, especially in online communication. These short-form words trigger problems in the field of data mining, especially those involving online text processing. It leads to inaccurate result of text mining activities. On the other hand, only few works have investigated on Malay short-form word identification and conversion. Therefore, this work aims to develop an application that can identify and convert Malay short-form words into its' full word. In order to develop this application, the short-form rules need to be carefully examined. The formal rules from Dewan Bahasa & Pustaka (DBP) are used as the primary reference for generating the short form word identification algorithm. While for the conversion algorithm, Levenshtein Distance (LD) is used to measure the similarity. The rule-based technique is also used as a complement to LD technique. As a result, 70.27% of the Malay short-form words have been correctly converted into their full words. The conversion rate is quite promising, and this work can be further strengthened by incorporating more rules into the algorithm.

Corresponding Author:
Azilawati Azizan,
Faculty of Computer and Mathematical Sciences,
Universiti Teknologi MARA, Perak Branch, Tapah Campus
Perak, Malaysia.
email: azila899@uitm.edu.my

1. Introduction

A short-form word or formally known as abbreviation is a shorter version of word by eliminating certain characters. Short-form exists in many languages, and it exists in the Malay

PENERBIT PRESS
UNIVERSITI TEKNOLOGI MARA

language as well. The following is an example of a comment in the Malay language taken from the official Facebook page of Ministry of Health Malaysia:

"Hampir 1 juta dh yg mati akibat covid dlm masa +-10 bln, tp syukurlh msia masih trkawal…semoga wabak ni cpt berlalu…tlg la weh..jaga sop kita"

The above message contains short-form words (abbreviation and acronym), incorrect spell words, symbols, improper casing, and punctuation. There is also other improper word structure of a short form found in online messages created by Malaysian such as mixed language words, code-mixing [1], phonetic spelling [2], and homophone words [3]. Sometimes, when the creation of a short-form is too creative than the usual standard, the machine cannot process the words, but even humans are also confused with the actual meaning of the short forms used.

Previously, the purpose of using short-form was to save space and speed up the writing process, which is very common in the journalism industry. However, nowadays, in the digital communication era, short-form is widely used by almost all levels of people. This is because they want to minimize keystroke (especially those using mobile devices to generate the message) and to speed up the communication process. The issue is that the use of these short-form words has a significant impact on society, such as the youngsters who also tend to use it in their formal writing [4]. This situation is not good for their language development. In fact, using short-form can degrade the original value of the language [3].

Short-form words, however, bring greater problems in the data mining field. In text processing activity, the short form word is categorized as an out-of-vocabulary (OOV) terms. It is a kind of noisy text. It cannot be processed accurately. Findings from [5] report that, on average in an online Malay text message containing 60 words, as many as 15 words were made up of noisy text. It means, as much as 25% of the words found in a text message are meaningless. If this kind of noisy text (short form) is omitted, most likely some important information in a message will be lost. Therefore, it becomes necessary to fix or normalize the short-form word so that it can be processed accurately.

This project aims to develop an application that can identify short-form words and translate them into its' correct word. Firstly, we observed the short form words that are commonly used by Malaysian on the social media platform. Then we thoroughly studied the guidelines for creating the Malay short-form words. After that, we constructed an algorithm that automatically identifies and replaces the short-form words. Finally, this project manages to develop an application that can convert the Malay short-form word into their correct word using Levenshtein Distance and rule-based technique.

This paper is arranged as follows. Section 2 presents several pieces of research related to the normalization of Malay short form or abbreviation. Section 3 explains about the Levenshtein Distance string metric. Then, section 4 lists out several rules and guidelines in Malay short forms. While section 5 clarifies the methodology used to develop this project. Lastly, section 6 and 7 discuss the results and concludes the findings.

2. Related Works

Research done by Samsudin [6] has been cited by many researchers who research Malay OOV or Malay noisy text. There were two (2) major contributions made by them; first is the common noisy text list, and second is the creation of the artificial abbreviation list. The common noisy text list is generated based on the noisy text that occurred more than two times in their corpus. A total of 10,550 common noisy words were listed. On the other hand, the artificial abbreviation list was created by adopting the DBP rules and observing the noisy text pattern from the common noisy word list. The list is comprised of 80,000 noisy artificial words, which is used to improve the normalization process. They managed to get to 76% of correctly identified noisy words by using both the common noisy list and the abbreviation list. Besides, they also highlighted the rules they used to manipulate the characters of Malay OOV words. Those rules have drawn the attention of many scholars. We gained a lot of knowledge on Malay noisy words from this publication.

Omar [7], attempted to build a Malay abbreviation corpus by using social media data, and they named it as Malay Social Media Corpus (MSMC). They have extracted and normalized one (1) million user-generated content (post) from Twitter and Facebook. All the content was filtered with multilayer pattern matching and statistical machine translation. They attempted to identify the

abbreviation words by removing all the matched in-vocabulary (IV) terms. They assumed the remaining (OOV) words are the abbreviation words. Then, all the identified abbreviation words (OOV) were linked to IV words semi-automatically. It means that their corpus contains one million social media posts with all the words (normal and abbreviation words) linked to IV words. So indirectly, the corpus can be used to translate any abbreviation if the abbreviation pattern is the same as the pattern in the MSMC. Other than that, they also published eight (8) common abbreviation patterns being used in the user-generated content from Twitter and Facebook. We took into consideration the abbreviation patterns figured out by them. Besides that, we also pay attention to their way of identifying the OOV (abbreviation) words.

A research work proposed by Roza [8] has brought us an idea of using LD and rule manipulation to convert the short-form words. They conclude that LD alone is not enough to normalize the OOV words, and it creates a benchmark that can be further improved. Their testing result shows that with additional rules adaptation, the results have increased significantly. For the character manipulation rules, they have closely followed the rules published by [5]. Besides, they also proposed new rules to normalize the OOV. They suggest adding a stemming process before the OOV can be normalized with other rules. Besides that, they managed to compile a list of the most frequent OOV words that appear in their dataset. They also agreed that the Malay language's colloquial terms will continue to evolve and change over time, which makes the research in this area always relevant.

2.1 Levenshtein Distance

Levenshtein Distance (LD) is a very common string metric used in spelling correction and plagiarism application [9]. It is often referred to as edit distance. The word 'distance' implies the number of changes required to alter a word [10]. This technique is used to measure the amount of edit needed to correct a word [11]. LD formula is as in (1).

$$lev_{a,b}(i,j) = \begin{cases} \max(i,j) \quad if \min(i,j) = 0, \\ \min \begin{cases} lev_{a,b}(i-1,j) + 1 \\ lev_{a,b}(i,j-1) + 1 \\ lev_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} \quad otherwise \end{cases} \tag{1}$$

In simple words, LD is a number showing how different two strings are. The higher the number, the more different the two strings are [12], [13]. The LD algorithm involves addition, deletion, and character substitution into a word.

2.2 Malay Short-form Rules

As per our observation, Malaysian commonly tend to remove the vowels when creating short forms such as '*mkn'* for '*makan'* (eating), '*sy'* for '*saya'* (I) and '*bsr'* for '*besar'* (big). Some tend to remove the syllable such as '*ni'* for '*ini*' (this), '*gi*' for '*pergi*' (go) and 'tu' for 'itu' (that). And some even try to create a shorter version of short-form such as '*y*' for '*ya*' (yes), '*k*' for '*ok*', '*x*' for '*tidak*' (no) and '*g*' for '*pergi*'.

Apart from the observation, we also delved into several works done by [5], [14], [15], [16] and [17], to get more in-depth about this subject. Almost all of them referred to the same basic guidelines produced by Dewan Bahasa & Putaka (DBP). DBP is an institution set up by the government of Malaysia to preserve 'Bahasa Melayu' (Malay language). DBP has published more than 15 guidelines for creating short-form word and acronym [18], but we only chose eight guidelines as our primary reference to suit our project scope, which did not cover the acronym type short form. The referred guidelines are as follows:

i. Remove all vowel
ii. Remove vowel and consonant simultaneously
iii. Remove based on syllable
iv. Use one character from the word
v. Combination of character and number
vi. Use the first character from a closed compound word

vii. Use the first and last character of the word
viii. Use 2 to 3 number of characters from a word

Those guidelines were used to create the rule-based algorithm which complements the LD technique.

3. Methodology

The development of this application is based on the prototyping model. This model suggests six (6) phases of operation: planning, analysis, design, building a prototype, user evaluation, refined prototype, and final product. We chose this model because it focuses on producing the end product by refining it until it achieves the project requirement. Figure 1 illustrates the activity involved during each phase of the prototyping model.

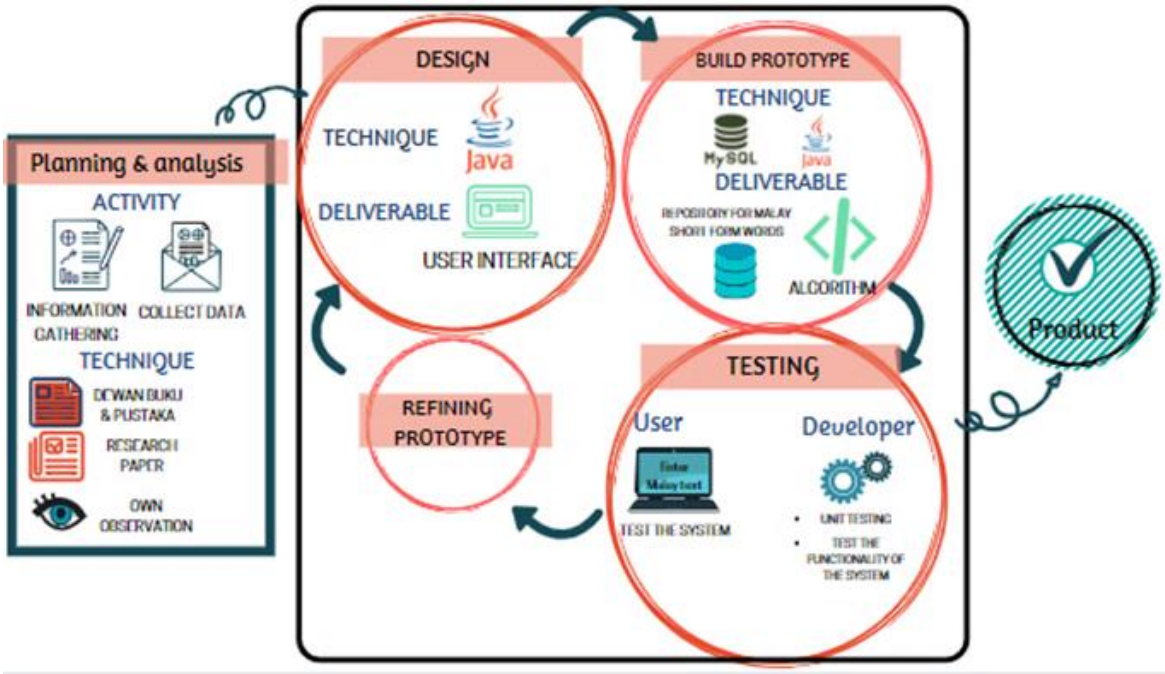

Figure 1. Development methodology-prototyping model

System architecture, as in Figure 2 provides an overview of the processes that take place in the proposed application. The first block on the right is the back-end process, which occurs before the application run. At this stage, the rules for short-form were created. It is created from three (3) main sources. The first source is from our own observation on the social media atmosphere, and the remaining is from the DBP guidelines and research publications. Then we triangulate these three (3) sources to acquire the rules, patterns, and concepts that are commonly used in Malay short-form words. This information is used in the next stage of development.

The second block of the system architecture exhibits the main process in the development. It involves several operations. The first operation is the pre-processing procedure, which includes case-folding and tokenization. It converts all the letters to lower case and separates all the submitted words or sentences into tokens. Then the tokens undergo identification and segregation process. It is done with the aid of the short-form information (rules and patterns) that we have obtained earlier.

Later, the short-form word is classified by counting the number of vowel in a string (token). If the number of vowel is found equal to or less than one, then it is labelled as a short form candidate. Next, the candidate is checked with Malay single-syllable word (‘*kata tunggal satu suku kata*’) list. If the candidate is not in the list of Malay single-syllable words, it is classified as a short-form. Only then, the LD algorithm is used to find the nearest word to convert the short-form word into its full words. This algorithm calculates the number of edits needed between two strings. A

string match with the lowest LD value is chosen as the best word to replace the short form candidate. In addition to the LD method, the short form candidate will go through the rule-based operation. The rule is based on the guidelines that have been published by the DBP. Finally, the application displays the full word of the short-form that has been entered by the user.

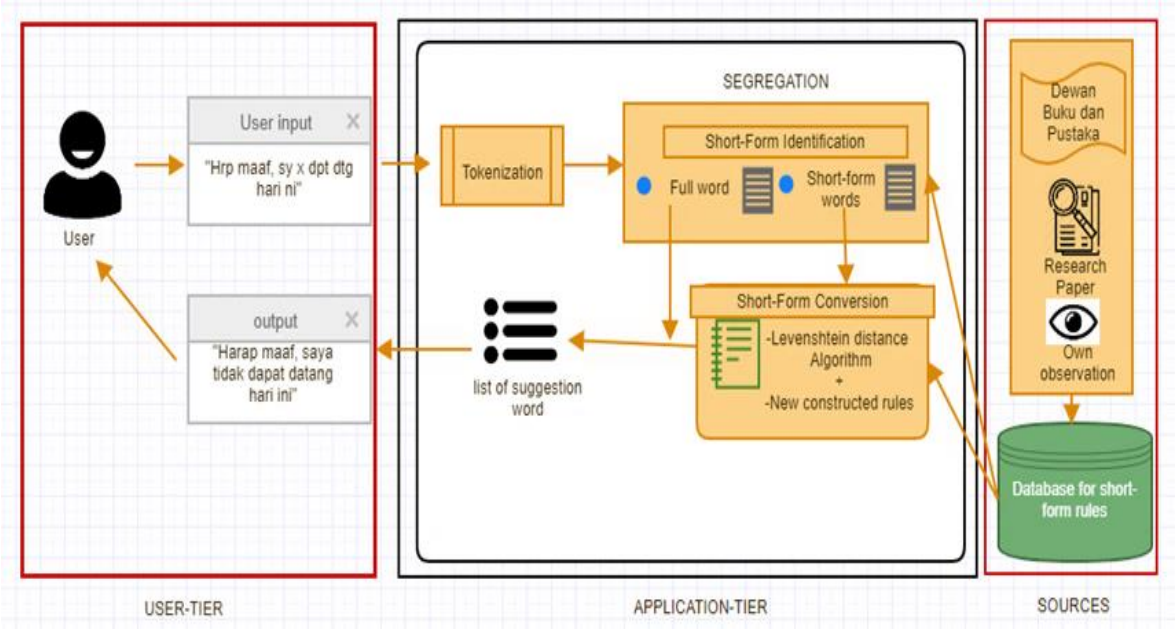

Figure 2. System architecture

The user interface of our application is as in Figure 3. The interface has an input area, convert ("*TUKAR*") button, and output area. The input area is where the user enters the text/sentences, while the output area is where the result of the conversion is shown.

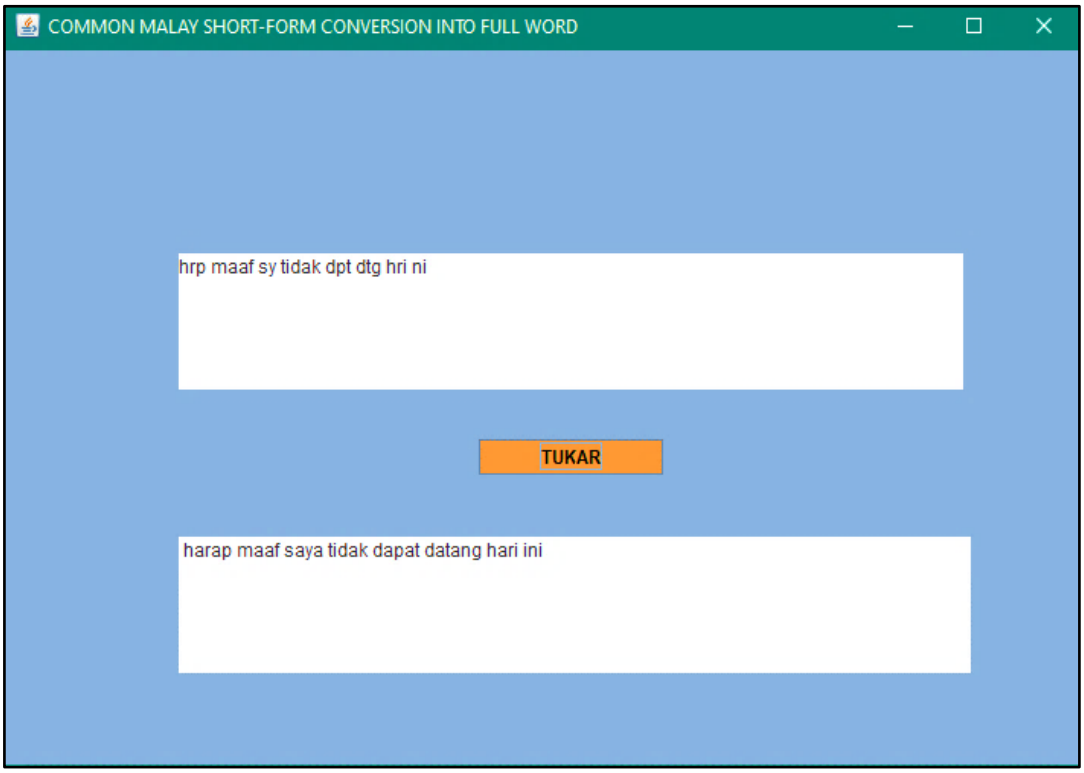

Figure 3. The user interface of the application

4. Results and Discussion

The application is built entirely in the Java Eclipse environment, and the algorithms are constructed using Java language. Once the development completes, the first module (tokenization, identification, and segregation algorithm) is tested by submitting several sentences containing a mixture of short-form and non-short-form word to the application. Sample of the output in Figure 4 shows that the application managed to identify and segregate the short form word from the non-short form word.

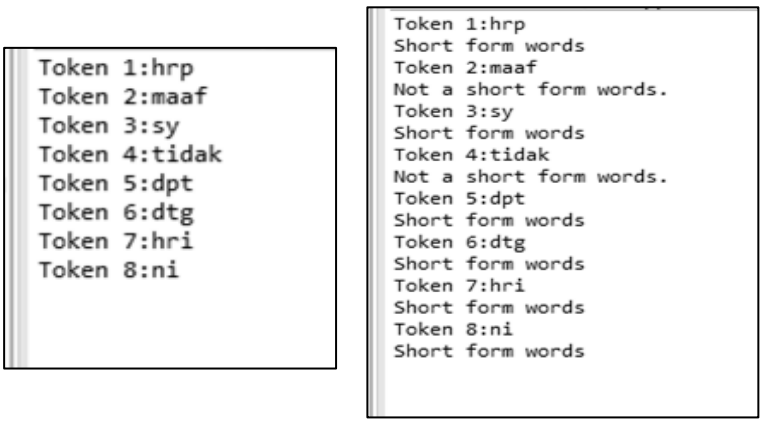

Figure 4. The output of short-form identification and segregation module

Next, the second module is tested: the short form conversion module (LD and rule-based algorithm). We have prepared 111 of short-form words for the testing. All the words were Table 1 shows the sample of short-form word used.

Table 1. Sample of the short-form word and their full word

No	Short Form	Full Word
1	ank	anak
2	ap	apa
3	awk	awak
4	abg	abang
5	adk	adik
6	bln	bulan
7	bwh	bawah
8	bli	beli
9	btl	betul
10	byk	banyak

They were all carefully tested, and 78 of them were converted successfully by the algorithm. It indicates that 70.27% of the short form words were correctly converted. It is illustrated in Figure 5. We compare our result to work done by [8], which also apply a similar approach to our work. They recorded only 16% of the OOV words were successfully normalized, compared to our work that has achieved better accuracy. We believe the percentage can be increased if more rules are introduced to the algorithm.

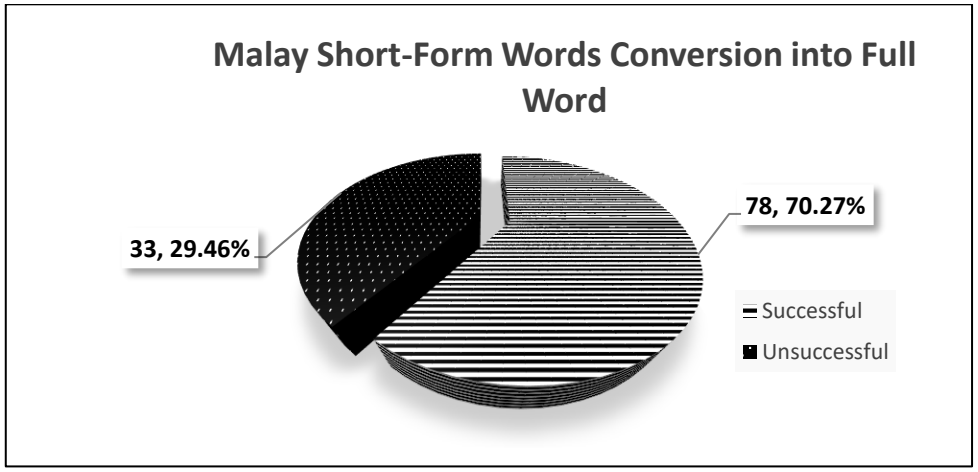

Figure 5 . Percentage of conversion results

The test result is also analyzed based on the category of rules aspect. Figure 6 shows the numbers of successful and unsuccessful conversion based on its category of rules. The conversion of short-form words from vowel removal, one character, and a combination of character and number categories, are among the most successful. In contrast, the short-form conversion from the first character of a closed compound word category is the most unsuccessful. An example of short-form from this category is 'tt,' which refers to '*tandatangan'* (signature). The word '*tandatangan*' is a Malay compound word comprised of the word '*tanda*' and '*tangan*'. So the first character from each word (the compound word) devoted the short form of 'tt'. The failure of the conversion may be due to inaccurate identification of the compound word, which also leads to inaccurate LD calculation. Apart from it, the LD algorithm is also ineffective to be used with a long string.

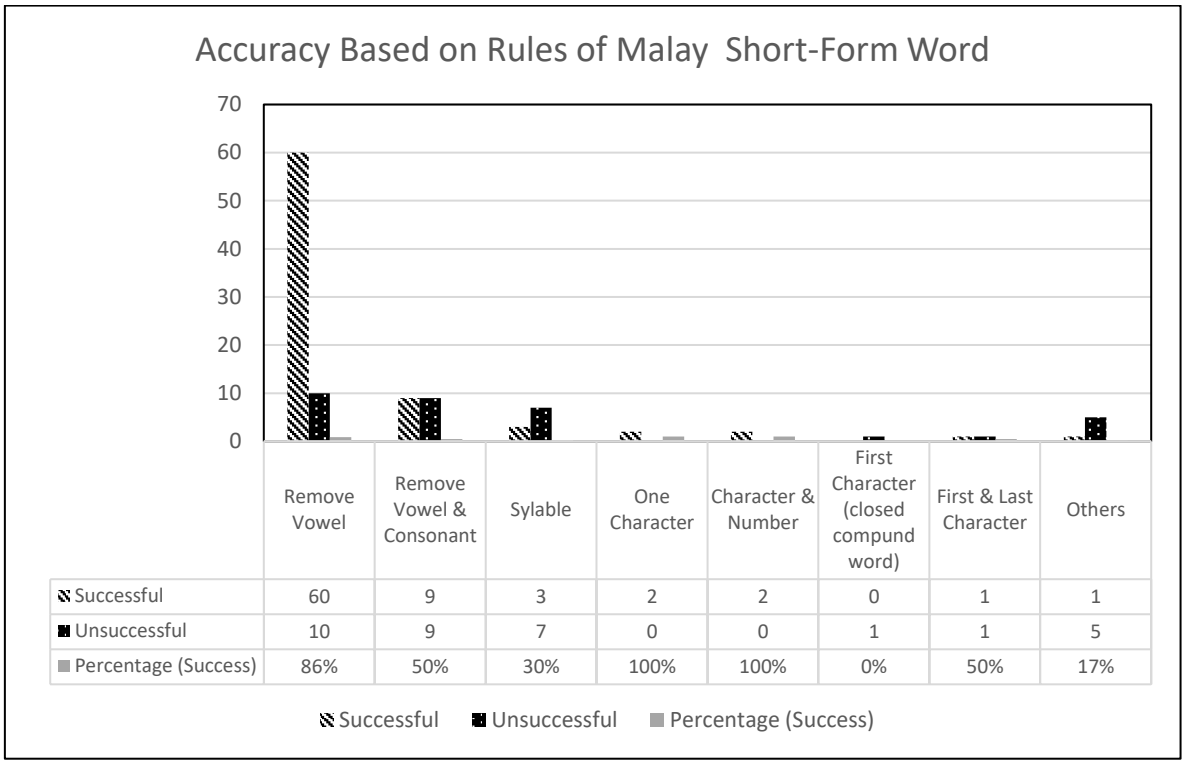

Figure 6. Conversion result based on types of short-form rules

In total, 29.46% of the short form words were not successfully converted. It indicates that further research needs to be done. It has been our main concern to add more rules and adopt other technique that may increase the conversion rate in future.

5. Conclusion

This work aims to develop an algorithm that can convert Malay short-form words into its' full word. Before constructing the algorithm, there were eight (8) rules have been identified for creating the Malay short-forms words which are removing vowel from the words, removing vowel and consonant simultaneously, short-form words that used syllable, use one character from the word, combination of character and number, using the first character from closed compound words, and by using the last character of the word. These rules were used as the principal guidance for constructing the identification and conversion algorithm. As for the basis of the development, Levenshtein Distance was used to calculate and measure the similarity of the strings. In addition, several rules were created and added to the algorithm as a supplementary to the Levenshtein Distance technique. Finally, 70.27% of the short-form words were converted correctly. However, this work has some limitation, such as the number of test data can be further increased to produce richer results. Moreover, the LD algorithm itself has its drawback as it is ineffective when dealing with a long string (word). Although this application is a prototype version, it managed to garner positive results based on the proposed algorithm that applied both LD and rules-based technique.

Acknowledgments

The authors would like to thank Universiti Teknologi MARA (UiTM), Perak branch, Tapah campus for providing the opportunity, support and facilities to carry out this project successfully.

References

[1] N. I. B. Ahmad Bukhari, A. F. Anuar, K. M. Khazin, and T. M. F. Bin Tengku Abdul Aziz, "English-Malay Code-Mixing Innovation in Facebook among Malaysian University Students," *Res. World – J. Arts Sci. Commer.*, no. Cmc, pp. 01-10, 2015.

[2] N. Samsudin, M. Puteh, A. Razak, and M. Zakree, "N_ormalization of Common

N_oisyTerms in Malaysian Online Media," *Proc. Knowl. Manag. Int. Conf.*, no. July, pp. 515–520, 2012.
[3] M. Mokhsin, A. A. Aziz, S. R. Hamidi, A. M. Lokman, and H. A. Halim, "Impact of using abbreviation and homophone words in social networking amongst Malaysian youth," *Adv. Sci. Lett.*, vol. 22, no. 5–6, pp. 1260–1264, 2016.
[4] R. Kasbon, N. A. Amran, E. M. Mazlan, and S. Mahamad, "Malay Language Sentence Checker," *World Appl. Sci. J.*, vol. 12, pp. 19–25, 2011.
[5] R. Alfred, S. B. Basri, J. H. Obit, and Z. I. B. A. Ismail, "Improved automatic spell checker for malay blog," *Adv. Sci. Lett.*, vol. 21, no. 10, pp. 3342–3345, 2015.
[6] N. Samsudin, M. Puteh, A. R. Hamdan, and M. Z. A. Nazri, "Normalization of noisy texts in Malaysian online reviews," *J. Inf. Commun. Technol.*, vol. 12, no. 1, pp. 147–159, 2013.
[7] N. Omar, A. F. Hamsani, N. A. S. Abdullah, and S. Z. Z. Abidin, "Construction of Malay abbreviation corpus based on social media data," *Journal of Engineering and Applied Sciences*, vol. 12, no. 3. pp. 468–474, 2017.
[8] R. A. Raja, S. Lay-Ki, and H. Su-Cheng, "Exploring Edit Distance for Normalising Out-of-Vocabulary Malay Words on Social Media," *MATEC Web Conf.*, vol. 255, p. 03001, 2019.
[9] N. H. Ariyani, Sutardi, and R. Ramadhan, "Aplikasi Pendeteksi Kemiripan Isi Teks Dokumen Menggunakan Metode Levenshtein Distance," *semanTIK*, vol. Vol 2, no. 1, pp. 279–286, 2016.
[10] V. Christanti Mawardi, N. Susanto, and D. Santun Naga, "Spelling Correction for Text Documents in Bahasa Indonesia Using Finite State Automata and Levinshtein Distance Method," *MATEC Web Conf.*, vol. 164, 2018.
[11] H. D. Tolentino *et al.*, "A UMLS-based spell checker for natural language processing in vaccine safety," *BMC Med. Inform. Decis. Mak.*, vol. 7, no. February, 2007.
[12] T. Anjali, T. R. Krishnaprasad, and P. Jayakumar, "A Novel Sentiment Classification of Product Reviews using Levenshtein Distance," pp. 507–511, 2020.
[13] D. K. Po, "Similarity Based Information Retrieval Using Levenshtein Distance Algorithm," *Int. J. Adv. Sci. Res. Eng.*, vol. 06, no. 04, pp. 06-10, 2020.
[14] N. Samsudin, A. Razak, M. Puteh, and M. Zakree, "Mining Opinion in Online Messages," *Int. J. Adv. Comput. Sci. Appl.*, vol. 4, no. 8, pp. 19–24, 2013.
[15] N. Samsudin, M. Puteh, A. R. Hamdan, and M. Z. A. Nazri, "Is artificial immune system suitable for opinion mining?," *Conf. Data Min. Optim.*, no. May 2016, pp. 131–136, 2012.
[16] M. F. R. Abu Bakar, N. Idris, L. Shuib, and N. Khamis, "Sentiment Analysis of Noisy Malay Text: State of Art, Challenges and Future Work," *IEEE Access*, vol. 8, pp. 24687–24696, 2020.
[17] S. N. A. N. Ariffin and S. Tiun, "Part-of-speech tagger for malay social media texts," *GEMA Online J. Lang. Stud.*, vol. 18, no. 4, pp. 124–142, 2018.
[18] P. Singkatan Khidmat Pesanan Ringkas Bahasa Melayu, "Khidmat Pesanan Kandungan.P65," *Dewan Bhs. Pustaka*, 2008.