ICIC Express Letters
Volume **17**, Number **9**, September **2023**

ICIC International ©2023 ISSN 1881-803X
pp. 979–987

OUT-OF-VOCABULARY HANDLING IN UNSTRUCTURED DATA USING MODIFIED SOUNDEX PHONETIC RULE AND SIMILARITY ALGORITHMS

Afiyati Afiyati[1,2,*], Azhari Azhari[1] and Anny Kartika Sari[1]

[1]Department of Computer Science and Electronics
Universitas Gadjah Mada
Sekip Utara Bulaksumur, Yogyakarta 55281, Indonesia
{ arisn; a_kartikasari }@ugm.ac.id

[2]Faculty of Computer Science
University Mercu Buana
South Meruya, Jakarta 11650, Indonesia
[*]Corresponding author: afiyati.reno@mercubuana.ac.id

Received September 2022; accepted December 2022

Abstract. *The biggest challenge in analyzing unstructured data, such as data from social media, is the presence of out-of-vocabulary (OOV) words, which are words that are not listed in the standard dictionary. They can decrease the accuracy of text classification tasks. In this study, the modification of the Soundex phonetic algorithm is proposed to handle the problem by adding the last character to the Soundex code to accelerate the discovery of the most similar word to replace each OOV word. Four similarity algorithms, namely sequence matcher, Levenshtein distance, Jaro similarity, and Jaro-Winkler similarity, are used to find the best replacement for the OOV words. The method is applied to Indonesian tweet data, in which 73% of them are OOV words; hence, the Indonesian phonetic rule is applied to the modified Soundex algorithm. It was found that the best correlation value of 1.00 was obtained between the sequence matcher and Jaro-Winkler algorithms. The similarity values of the modified Soundex with the Indonesian phonetic rule for Jaro and Jaro-Winkler were higher (0.92) than the sequence matcher (0.75). The normalization time using the proposed method was faster than the original Soundex algorithm. The results of OOV normalization were proven to increase the accuracy of the sarcasm detection task.*

Keywords: Out-of-vocabulary, Phonetic rule, Similarity measures, Soundex algorithm, Unstructured data

1. **Introduction.** The unstructured words in social media consist of misspelled words, abbreviations, slang words, and other forms of informal language that are not listed in the standard dictionary. Words that do not exist in the standard dictionary are called out-of-vocabulary (OOV) words. Several factors cause the OOV words to become unstructured data: increasing users' foreign language skills, mixing sentences with regional languages, and writing informal words and abbreviations. The OOV words were deleted in previous research because it often causes uncounted problems in language models, and the system cannot provide output during decoding [1]. The OOV words can lead to inconsistencies in the database and provide different research results that use text as the main object. The production of clean data has become a significant challenge in the normalization process to produce clean data, particularly in sentiment analysis research [2]. The effective handling of normalization has resulted in accurate sentiment identification of users on Twitter [3]. The normalization of OOV words directly affects the performance of many NLP systems (i.e., machine translation, sentiment analysis, and intelligent question-answering).

DOI: 10.24507/icicel.17.09.979

The OOV problems in Indonesian tweet text have been handled using the skip-gram technique and similarity probability [4]. However, skip-gram only maps OOV words to a dictionary and replaces them with the most probable adjacent words. Unfortunately, skip-gram is unsuitable for fast-growing data because it will always require dictionary improvement. A phonetic algorithm is used to transform a non-standard micro text into a standard micro text and proved to increase the normalization performance for English micro text pre-processing [5]. One of the methods used to normalize OOV is Soundex. The Soundex phonetic algorithm in Hindi has been successfully implemented to retrieve the exact match of a misspelled word, which is one of the OOV categories [6]. Phonetic code is formed from the key value of each phoneme in a language. Table 1 shows that the phonetic code for English letters has different key values from Indonesian letters.

Table 1. The phonetic code rules

English alphabet	**Indonesian alphabet**	**Key value**
a, e, i, o, u, h, w, y	a, e, i, o, u	0
b, f, p, v	w, y	1
c, g, j, k, q, r, s, x, z	f, h, q, s, v, x, z	2
d, t	b, c, d, g, j, k, p, t	3
l	l	4
m, n	m, n	5
r	r	6

This study proposes a method for handling the out-of-vocabulary words in unstructured data using modified Soundex phonetic rule and similarity algorithms. The OOV words are encoded using the Soundex phonetic code rule. The last character is added to speed up the search process. The code is then compared with the Soundex codes of the words from the standard dictionary and the similarity value of the codes is calculated using sequence matcher, Levenshtein distance, Jaro similarity, and Jaro-Winkler similarity algorithms. The Jaro-Winkler similarity is used in this study because previously it is proven to reduce the runtime process effectively when applied to measuring the number of matched characters to ensure the similarity of two strings using a certain threshold [7]. The main difference between this study and the previous research is the use of modified Soundex algorithm with Indonesian phonetic rules and the combination of distance and similarity algorithms. This is proven to increase the number of normalized OOV words.

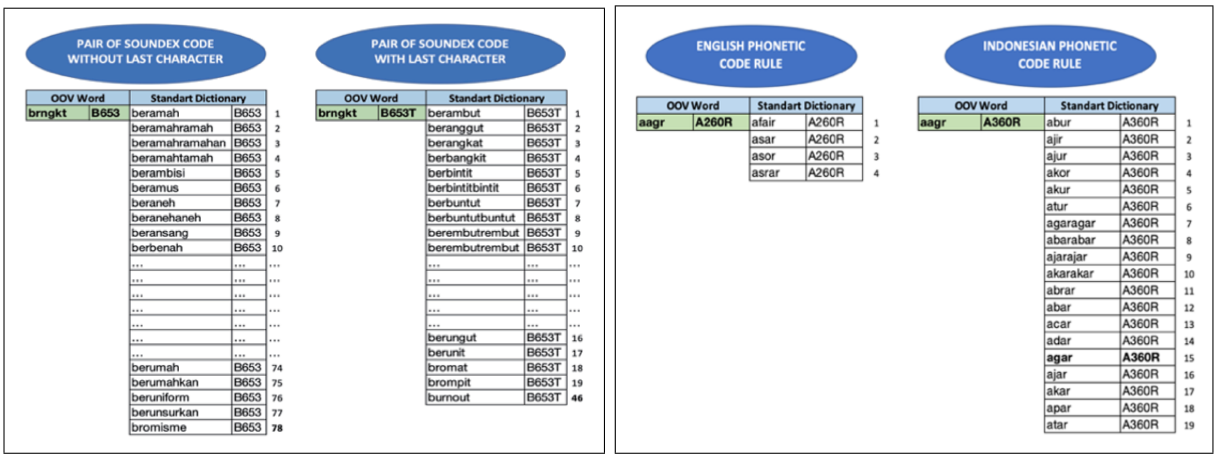
The remainder of this paper is organized as follows. Section 2 describes related work on out-of-vocabulary handling. Section 3 explains the proposed approach and how the algorithms are used. Section 4 presents the experimental results and discussion. Section 5 concludes this paper.

2. **Related Work.**

The out-of-vocabulary words significantly degrade the performance of the language model used for linguistic analysis and affect the detection of cyberbullying tweets [8]. Combined and mixed mapping approaches using the BERT model with pre-trained multilingual data have been used to address out-of-vocabulary (OOV) problems in part-of-speech tagging, named entity recognition, and machine translation quality [9]. The OOV problem of pre-trained word vectors is also solved by adding character-level embedding to the word-embedding layer and improving the normalization of the Twitter and movie review datasets. The experimental results showed that the accuracy of the model with the Twitter dataset reached only 66% [10]. However, the study does not mention the percentage of OOV words in the dataset. The Levenshtein distance algorithm is used to compare the misspelled words with the correct spelling words in the dictionary and is trained using global vectors (Glove) as character representations that generate vectors to obtain better suggestion lists of misspelled words [11]. The combination of Levenshtein distance results with a set of vowel rules is used to correct word errors in Chinese English learners and showed a precision score of 0.86. Automatic scoring for spelling performance produced a correlation between Sequence Matcher and Jaro-Winkler of 0.96 [12].

The OOV problems in Indonesian tweet texts have been examined using skip-gram and similarity probability. Recognizing the OOV words reduces the time required to identify the most similar words in a standard dictionary. Accuracy-weighted ensemble (AWE) and binary relevance (BR) models to learn OOV words have been implemented for the multi-label classification of Indonesian news articles, thereby reducing the processing time [13]. Text normalization with Soundex code and deep convolutional character-level embedding neural networks have been used to solve the problems of processing noisy sentences for sentiment detection of the Twitter dataset, handling small memory space in word-level embedded learning, and conducting accurate sentiment analysis of unstructured data. The sub-word tokenization is more robust to the OOV problem than other tokenization and achieves the highest accuracy of 81% [14]. However, the OOV rate used in those studies was only 26% of all datasets. The problem of OOV in Indonesian-language tweets has been addressed using the skip-gram technique and similarity probability [15]. However, those studies only map OOV words to the word dictionary and replace them with words most likely to be close together. Jaro-Winkler algorithm and Levenshtein distance as a character-based similarity are also used for measuring the similarity of requirements documents [16]. The normalization of OOV words directly affects NLP systems such as machine translation, sentiment analysis, and intelligent question-answering systems and showed increased performance for non-standard words [17]. The OOV words are removed from the dataset because they can cause problems in language modeling and provide unexpected output during the process [18,19]. However, the number of OOV words in those studies is only small and cannot be applied to a dataset with a very large number of OOV words. From the research mentioned above and the large number of OOV words in the dataset, OOV handling is still a problem to be solved and becomes the focus of this research.

3. **Proposed Method.** People often shorten words and phrases and make them abbreviations on social media to reduce the number of characters or to type them quicker [20]. Most abbreviations are written by removing vowels, leaving the first and last letters intact. Therefore, we decided to use the last letter of the OOV word to find a replacement word. Our experiment showed that using the last character in the Soundex code, the most appropriate substitute word was obtained. The novelty of this study lies in developing a phonetic Soundex code model that follows the rules of the same language as the dataset and the addition of the last character to the Soundex code. The addition of one character was intended to ensure the similarity of the target word to the OOV word. Each word in the OOV and standard dictionaries was coded using the modified Soundex algorithm and then grouped according to their codes. The code generated through the original Soundex algorithm (steps 1-6), as an example for the word "aagr" is coded as A360. With the modified algorithm (step 6 as the added step), the word "aagr" is coded as A360R. According to our experiments, the number of Soundex codes with the last character added is less than the Soundex codes without the last character; therefore, it will speed up the process of mapping OOV words to standard dictionaries in the pre-processing stage of large amounts of data. The acceleration of pre-processing is evidenced by the reduced number of Soundex code pair groups, as shown in Figure 1. For the OOV word "BRNGKT", using the phonetic rule code of Indonesian, found 78 words that returned from the standard dictionary using a Soundex code without the last character; conversely, only 46 pairs of words were returned when including the last character.



PAIR OF SOUNDEX CODE WITHOUT LAST CHARACTER

OOV Word		Standart Dictionary		
brngkt	B653	beramah	B653	1
		beramahramah	B653	2
		beramahramahan	B653	3
		beramahtamah	B653	4
		berambisi	B653	5
		beramus	B653	6
		beraneh	B653	7
		beranehaneh	B653	8
		beransang	B653	9
		berbenah	B653	10
		...	...	...
		...	...	...
		...	...	...
		...	...	...
		...	...	...
		...	...	...
		...	...	...
		berumah	B653	74
		berumahkan	B653	75
		beruniform	B653	76
		berunsurkan	B653	77
		bromisme	B653	78

PAIR OF SOUNDEX CODE WITH LAST CHARACTER

OOV Word		Standart Dictionary		
brngkt	B653T	berambut	B653T	1
		beranggut	B653T	2
		berangkat	B653T	3
		berbangkit	B653T	4
		berbintit	B653T	5
		berbintitbintit	B653T	6
		berbuntut	B653T	7
		berbuntutbuntut	B653T	8
		berembutrembut	B653T	9
		berembutrembut	B653T	10
		...	...	...
		...	...	...
		...	...	...
		...	...	...
		...	...	...
		berungut	B653T	16
		berunit	B653T	17
		bromat	B653T	18
		brompit	B653T	19
		burnout	B653T	46

ENGLISH PHONETIC CODE RULE

OOV Word		Standart Dictionary		
aagr	A260R	afair	A260R	1
		asar	A260R	2
		asor	A260R	3
		asrar	A260R	4

INDONESIAN PHONETIC CODE RULE

OOV Word		Standart Dictionary		
aagr	A360R	abur	A360R	1
		ajir	A360R	2
		ajur	A360R	3
		akor	A360R	4
		akur	A360R	5
		atur	A360R	6
		agaragar	A360R	7
		abarabar	A360R	8
		ajarajar	A360R	9
		akarakar	A360R	10
		abrar	A360R	11
		abar	A360R	12
		acar	A360R	13
		adar	A360R	14
		agar	**A360R**	15
		ajar	A360R	16
		akar	A360R	17
		apar	A360R	18
		atar	A360R	19

FIGURE 1. Soundex phonetic code English and Indonesian – With and without the last character

Using Indonesian phonetic rules with the addition of the last character indicates an increase in the number of words mapped into words in the standard dictionary. For example, the word "aagr" is coded by the English and Indonesian phonetic rules with the addition of the last character. The Soundex code using English phonetic rules produces 4-word pairs that match the Soundex code from the standard dictionary. However, from the 4-word pairs, no words were found that had the closest similarity with the word "aagr". Meanwhile, the Soundex code using Indonesian phonetic rules found 19-word pairs with the same Soundex code as the standard Indonesian dictionary and had the closest similar word, "agar". Based on our preliminary experiment, some words were not found when English language phonetic codes were used for Indonesian language text but were found when using Indonesian phonetic codes. Our proposed method, as shown in Figure 2, consists of several stages: pre-processing, the OOV detection to identify the words that are considered the OOV words, coding the words using the modification of the Soundex rule code, and adding one last character to the code.

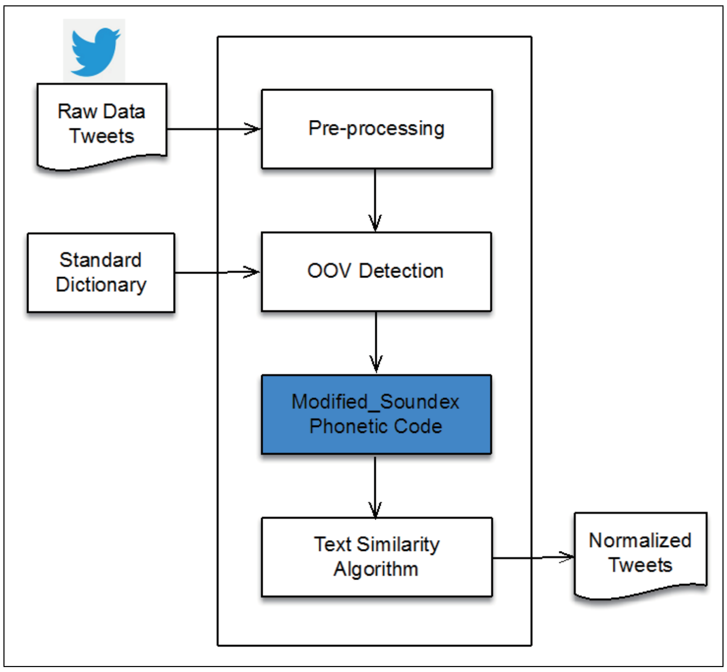


FIGURE 2. The text normalization flow process

3.1. **Pre-processing.** The dataset used in this study contained 50,000 lines with 92,244 words of tweets collected directly from the API of Twitter from the year 2019 to 2021. The unnecessary symbols such as hyperlinks, mentions, special characters, numbers, and the same words were removed from the dataset, and found 73% percentage of OOV words.

3.2. **OOV detection.** The OOV word detection was performed by comparing all the OOV words with a standard dictionary. The words which were not listed in the standard dictionary are flagged as OOV words and used in the following process.

3.3. **Modified Soundex phonetic code.** This study proposed a Soundex code that was generated using the Soundex algorithm with Indonesian phonetic rules and adding the last character to ensure the appropriate word to replace the OOV words. The code will then be used to measure the distance and similarity between two codes using sequence matcher (SM), Levenshtein distance (LD), Jaro similarity (JS), and Jaro-Winkler similarity (JWS).

3.3.1. *Levenshtein distance algorithm.* The distance between the two strings x and z was then calculated, as in (1).

$$d(x,z) = \sum_{t=1}^{T} w_t d_t(x,z) \tag{1}$$

In the formula, w_t is the weight of distance d_t and T is the total distance. However, the Levenshtein distance can return more than one similar word; therefore, we must improve the process with other algorithms to obtain the most appropriate substitute.

3.3.2. *Sequence matcher algorithm.* The sequence matcher algorithm computes the double number of sequence matching characters divided by the number of characters in the two strings, as in (2).

$$D_{ro} = \frac{2K_m}{|x| + |z|} \tag{2}$$

K_m denotes the same number of letters in both words, divided by $|x|$ and $|z|$, which indicates the number of characters present in the two words. The similarity metric can take a value between zero and one ($0 \leq D_{ro} \geq 1$). When the D_{ro} value is 1, it indicates that the two strings match entirely, and if the value is 0, it indicates that no letter matches. For example, the letters in the word "aagr" and "agar", which have the highest similarity under a phonetic rule using the last character, have a value of 0.75.

The Soundex code is generated according to the following algorithm.
1) Uppercase all the letters.
2) Get the first letter.
3) Remove all other vowels.
4) Consecutive (double) consonants are treated as single consonants.
5) Replace the second consonants with key values, as in Table 1.
6) Retain only up to the first three key values.
7) A zero is added to the last digit if the resulting key value is less than three.
8) The last character was added to the code.

3.3.3. *Jaro similarity algorithm.* The Jaro similarity algorithm measures the similarity between two strings [21]. The Jaro distance value ranges from 0 to 1, where one means strings are equal, and zero means no similarity exists between strings, as in (3). The Jaro algorithm is similar, albeit based on the number and order of familiar characters between the two strings.

$$d_j = \begin{cases} 0, & \text{if } m = 0 \\ \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right), & \text{for } m! = 0 \end{cases} \tag{3}$$

In the formula, $|s_1|$ and $|s_2|$ are the lengths of the strings s_1 and s_2, respectively, m is the number of matching characters, and t is half the number of transpositions.

3.3.4. *Jaro-Winkler similarity algorithm.* The Jaro-Winkler similarity algorithm is similar to the Jaro similarity. Both differ when the prefixes of the two strings match. Jaro-Winkler similarity uses a prefix scale 'p', giving a more accurate answer when the strings have a common prefix up to a defined maximum length l, as in (4).

$$sim_w = sim_j + lp(1 - sim_j) \tag{4}$$

sim_j is the Jaro similarity for strings s_1 and s_2, the l is the length of the common prefix at the start of the string up to a maximum of four characters, and p is a constant scaling factor for how much it is adjusted upward to have the same common prefixes. The p must not exceed 0.25 (i.e., 1/4, where 4 is the maximum length of the prefix considered); otherwise, the similarity can be greater than 1. The standard value for this constant in Winkler's study was $p = 0.1$.

4. Results and Discussion.

The correlation values between SM, LD, JS, and JWS between the English phonetic rule without the last character (ENG-SDX-NOLC) and the Indonesian phonetic rule with an additional character (INDO-SDX-LC) are shown in Table 2. The highest correlation values between SM and JS or between SM and JWS for the Indonesian language with an additional last character (INDO-SDX-LC) indicate that these combined algorithms could be used as models to find the most OOV words similar to the standard dictionary.

Table 2. The correlation value for the word "aagr"

Tasks		SM	LD	JS	JWS
SM	ENG-SDX-NOLC	1.00	0.77	0.77	0.81
	INDO-SDX-LC	1.00	0.72	**0.96**	**1.00**
LD	ENG-SDX-NOLC	–	1.00	0.78	0.78
	INDO-SDX-LC	–	1.00	0.72	0.72
JS	ENG-SDX-NOLC	–	–	1.00	0.96
	INDO-SDX-LC	–	–	1.00	0.96
JWS	ENG-SDX-NOLC	–	–	–	1.00
	INDO-SDX-LC	–	–	–	1.00

The word "aagr" from the OOV dataset has 15 possible substitutes from the standard dictionary, each of which is listed in Table 3. With the phonetic language rule with the inclusion of the last character, the LD value returned the highest similarity value of 0.75 for a few words and gave a minimum edit distance of 2 for several words. Meanwhile, the JS and JWS algorithms returned the highest similarity value of 0.92 in exactly one word, "agar". The highest correlation between SM and JS returns a value of 0.96 for Indonesian Soundex with an additional last character (INDO-SDX-LC), indicating that the proposed model improved the performance. Likewise, the highest correlation between SM and JWS returns a value of 1.00 for Indonesian Soundex's additional last character (INDO-SDX-LC), indicating that the combination of the algorithms could be used as models to replace the OOV.

The correlation distribution of the SM, LD, JS, and JWS values using Indonesian phonetic rule and with additional characters in the Soundex code, is shown in Figure 3. The correlation between the lowest value of the LD and the higher value of the SM, JS, and JWS, which are closer to one, indicates a greater relative similarity between the two strings.

Table 3. The similarity value for word "aagr"

OOV word	Standard word	SM	LD	JS	JWS
aagr	abur	0.50	2	0.67	0.70
aagr	ajir	0.50	2	0.67	0.70
aagr	ajur	0.50	2	0.67	0.70
aagr	akor	0.50	2	0.67	0.70
aagr	akur	0.50	2	0.67	0.70
aagr	atur	0.50	2	0.67	0.70
aagr	abrar	0.66	3	0.63	0.67
aagr	abar	0.75	2	0.83	0.85
aagr	acar	0.75	2	0.83	0.85
aagr	adar	0.75	2	0.83	0.85
aagr	**agar**	0.75	2	**0.92**	**0.92**
aagr	ajar	0.75	2	0.83	0.85
aagr	akar	0.75	2	0.83	0.85
aagr	apar	0.75	2	0.83	0.85
aagr	atar	0.75	2	0.83	0.85

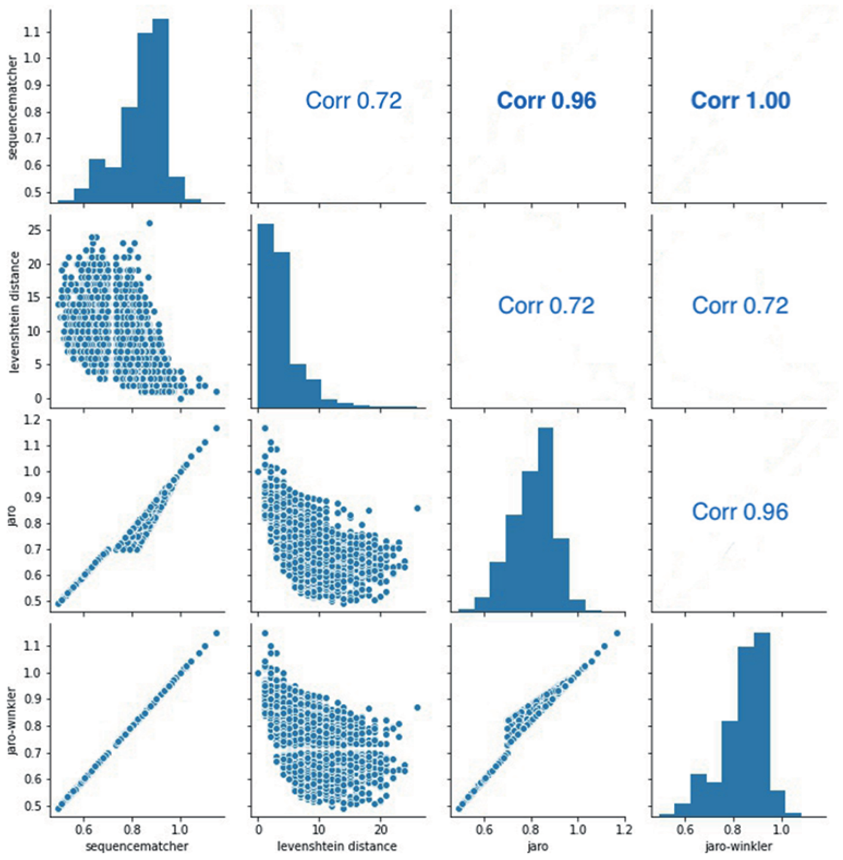


Figure 3. The correlation distribution between all the algorithms

The direction of the straight-line shows the relationship between the two variables. If the two variables move in the same direction, they have a positive correlation. A positive value indicates a positive association, and a negative value indicates a negative association. The normalization performance obtained using the Soundex phonetic algorithms is listed in Table 4. Column #time shows the duration of the process for the algorithms to find the candidate replacement for the OOV words, and column #cand the number

Table 4. The result of pairing algorithms

Variable	**#time**	**#cand**	**#corr**	**#P**	**#R**	**#F1**
IND-SDX-LC	213'	**42.06**	**10.88**	**0.26**	**0.16**	**0.20**
IND-SDX-NOLC	299'	60.88	9.28	0.15	0.14	0.15
ENG-SDX-LC	239'	42.15	9.78	0.16	0.15	0.15
ENG-SDX-NOLC	297'	61.36	9.41	0.16	0.14	0.15

of normalization candidates returned for each OOV using the algorithms. Column #corr indicates the number of OOV words the algorithm could provide the correct answer. The precision value (#P) is given as the ratio of correct candidates (*corr*) to the total number of candidates retrieved from the shortest distance between the OOV and the standard dictionary (*stand*). The recall (#R) was calculated as the number of correct candidates over the total number of OOV words. The F-score value (#F1) is given as the aggregation of precision and recall values.

OOV words were replaced with the most appropriate words from the standard dictionary using four text similarity algorithms. The highest probability of OOV word substitution can be obtained using JS and JWS. The method of finding substitute words for OOV data was then tested on one of the NLP tasks using a logistic regression and BERT model. Sarcasm detection increased the accuracy by 3% with normalized OOV words, as shown in Table 5. The accuracy value after OOV words were normalized only increased slightly due to the small number of words in the OOV dictionary, which was only 9,162 words, compared to the number of OOV words that must be corrected, 67,377 words.

Table 5. The evaluations result

	Model	**Acc**	**Pre**	**Rec**	**F1**
Before OOV normalized	Logistic regression	0.71	0.70	0.67	0.68
	BERT	0.75	0.75	0.72	0.72
After OOV normalized	Logistic regression	0.73	0.73	0.69	0.70
	BERT	0.76	0.76	0.73	0.74

5. **Conclusion.** Using the same phonetic rules of the language and adding the last character of the OOV word to the Soundex code has proven to solve the unstructured data problem. The combination of SM with JS and JWS values indicates the most similar association, represented by correlation numbers 0.96 of 1.00. At the same time, the combination of the LD, JS, and JWS cannot provide the best word suggestions to replace the OOV words, indicating that the correlation value is only 0.72. The JS and JWS produce a 0.92 value higher than the sequence matcher, which has a value of 0.75, using the Indonesian phonetic rule with the last character. The search process found alternative words from the standard dictionary more than 50 min faster by using the Soundex phonetic code with the last character rather than without the last character. The number of words corrected using the Soundex code algorithm for Indonesian was only around 16%. However, the increase in accuracy showed that the normalization of OOV words could improve the accuracy of detecting sarcasm sentences. This research can be developed using other languages with phonetic rules for every language and proven that the model using the Indonesian phonetic rule with the last character could find 9,162 appropriate substitute words for OOV. In contrast, the English phonetic rule without the last character can only find 635 alternative substitute words for OOV. The use of better similarity algorithms, such as cosine similarity, to find more possible substitutes for OOV words can be explored for future work.

REFERENCES

[1] R. Schifanella, P. De Juan, J. Tetreault and L. Cao, Detecting sarcasm in multimodal social platforms, *Proc. of 2016 ACM Multimed. Conf.*, no.8, pp.1136-1145, DOI: 10.1145/2964284.2964321, 2016.

[2] E. Egorova and L. Burget, Out-of-vocabulary word recovery using FST-based subword unit clustering in a hybrid ASR system, *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, AB, Canada, pp.5919-5923, DOI: 10.1109/ICASSP.2018.8462221, 2018.

[3] M. Arora and V. Kansal, Character level embedding with deep convolutional neural network for text normalization of unstructured data for Twitter sentiment analysis, *Soc. Netw. Anal. Min.*, vol.9, no.1, DOI: 10.1007/s13278-019-0557-y, 2019.

[4] F. Muhammad and A. Purwarianti, Handling out of vocabulary in supervised event extraction on Indonesian tweets: Using word representation, word list, word context and document level features, *Proc. of 2016 Int. Conf. Data Softw. Eng. (ICoDSE2016)*, DOI: 10.1109/ICODSE.2016.7936113, 2017.

[5] Y. Doval, M. Vilares and J. Vilares, On the performance of phonetic algorithms in microtext normalization, *Expert Syst. Appl.*, vol.113, pp.213-222, DOI: 10.1016/j.eswa.2018.07.016, 2018.

[6] V. Gautam, A. Pipal and M. Arora, SoundEx algorithm revisited for Indian language, *Int. Conf. Innov. Comput. Commun.*, vol.56, pp.47-55, DOI: 10.1007/978-981-13-2354-6_6, 2019.

[7] K. Dreßler and A. C. N. Ngomo, On the efficient execution of bounded Jaro-Winkler distances, *Semant. Web*, vol.8, no.2, pp.185-196, DOI: 10.3233/SW-150209, 2017.

[8] S. Bharti, A. K. Yadav, M. Kumar and D. Yadav, Cyberbullying detection from tweets using deep learning, *Kybernetes*, vol.51, no.9, pp.2695-2711, DOI: 10.1108/K-01-2021-0061, 2021.

[9] H. Wang, D. Yu, K. Sun, J. Chen and D. Yu, Improving pre-trained multilingual models with vocabulary expansion, *The 23rd Conf. Comput. Nat. Lang. Learn. Proc. Conf. (CoNLL2019)*, pp.316-327, DOI: 10.18653/v1/k19-1030, 2019.

[10] H. Xia, C. Ding and Y. Liu, Sentiment analysis model based on self-attention and character-level embedding, *IEEE Access*, vol.8, pp.184614-184620, DOI: 10.1109/ACCESS.2020.3029694, 2020.

[11] G. Huang, J. Chen and Z. Sun, A correction method of word spelling mistake for English text, *J. Phys. Conf. Ser.*, vol.1693, no.1, DOI: 10.1088/1742-6596/1693/1/012118, 2020.

[12] C. Themistocleous, K. Neophytou, B. Rapp and K. Tsapkini, A tool for automatic scoring of spelling performance, *Journal of Speech, Language, and Hearing Research*, vol.63, no.12. pp.4179-4192, DOI: 10.1044/2020_JSLHR-20-00177, 2020.

[13] D. G. Saputra and M. L. Khodray, An ensemble approach to handle out of vocabulary in multilabel document classification, *The 4th IGNITE Conf. 2016 Int. Conf. Adv. Informatics Concepts, Theory Appl. (ICAICTA2016)*, DOI: 10.1109/ICAICTA.2016.7803109, 2016.

[14] D. Cho, H. Lee and S. Kang, An empirical study of Korean sentence representation with various tokenizations, *Electronics*, vol.10, no.7, 845, DOI: 10.3390/electronics10070845, 2021.

[15] J. Zhao and X. Gui, Comparison research on text pre-processing methods on Twitter sentiment analysis, *IEEE Access*, vol.5, pp.2870-2879, DOI: 10.1109/ACCESS.2017.2672677, 2017.

[16] R. Delima, R. Wardoyo and K. Mustofa, CombineTF for requirements data similarity detection on AREM, *ICIC Express Letters*, vol.16, no.9, pp.913-921, DOI: 10.24507/icicel.16.09.913, 2022.

[17] Y. Tang, C. Tang and C. Zhu, Resolve out of vocabulary with long short-term memory networks for morphology, *Proc. of 2020 IEEE Int. Conf. Artif. Intell. Comput. Appl. (ICAICA2020)*, pp.115-119, DOI: 10.1109/ICAICA50127.2020.9182586, 2020.

[18] J. V. Lochter, R. M. Silva and T. A. Almeida, Multi-level out-of-vocabulary words handling approach, *Knowledge-Based Syst.*, vol.251, 108911, DOI: 10.1016/j.knosys.2022.108911, 2022.

[19] S. Liu, N. Tan, Y. Ge and N. Lukaè, Research on automatic question answering of generative knowledge graph based on pointer network, *Information*, vol.12, no.3, 136, DOI: 10.3390/info12030136, 2021.

[20] A. T. Wahyuni, *Abbreviations Used in Social Media Facebook Instagram*, Bachelor Thesis, University of Sumatera Utara, 2017.

[21] V. S. Vykhovanets, J. Du and S. A. Sakulin, An overview of phonetic encoding algorithms, *Autom. Remote Control*, vol.81, no.10, pp.1896-1910, DOI: 10.1134/S0005117920100082, 2020.