International Journal of Informatics and Communication Technology (IJ-ICT)
Vol. 15, No. 3, September 2026, pp. 1340~1351
ISSN: 2252-8776, DOI: 10.11591/ijict.v15i3.pp1340-1351

Evaluation of text correction using a combination of Levenshtein distance and Trie algorithm

Cynthia Natalie, Abba Suganda Girsang
Department of Computer Science, BINUS Graduate Program Master of Computer Science,
Bina Nusantara University, West Jakarta, Indonesia

Article Info

Article history:

Received Jul 10, 2024
Revised Jun 24, 2026
Accepted Jul 5, 2026

Keywords:

ChatGPT
Google translate
Levenshtein distance
Text correction
Trie algorithm

ABSTRACT

Nowadays, technology is advancing with various applications, especially in text processing, such as news recommendations, sentiment analysis, automatic scoring, and language translation. In some cases, spelling errors often occur when inputting text for the translation process, necessitating text correction methods to display suggestions as a result. Therefore, the problem statement raised is about how to improve the accuracy of text correction and evaluate the translation quality at the word level after correcting input text in the context of translation from Indonesian to English. This research aims to develop and evaluate the combination of Levenshtein distance algorithm and Trie to correct input text and evaluate the translation quality at the word level after correcting text. There are various text correction methods, such as Hamming distance, Levenshtein distance, Damerau-Levenshtein distance, and N-Gram. Among several text correction methods. Levenshtein distance algorithm is commonly used to calculate the distance between texts and can be enhanced by using a Trie for more efficient computation in evaluating text correction. This research method resulted in an accuracy of 82.25% with an F1 score of 84.39%, where the developed text correction model produced a good translation using BLEU score with an increase of 1.55% after text correction.

Corresponding Author:

Abba Suganda Girsang
Department of Computer Science, BINUS Graduate Program Master of Computer Science
Bina Nusantara University
Street K. H. Syahdan 9, Kemanggisan, Palmerah, West Jakarta, Indonesia
Email: agirsang@binus.edu

1. INTRODUCTION

Foreign language translation is the interpretation of text from the native language, resulting in a text that conveys a similar message [1]. With the advancement of technology, foreign language translation can now be learned and performed using applications, one of which is Google Translate [2]. In some cases, spelling errors often occur when inputting text for the translation process, necessitating text correction methods to display suggestions as a result of input text correction to help users obtain optimal translation results.

Various methods have been proposed to handle text correction problems [3]-[5], such as N-Gram and Levenshtein distance [6], which can provide good results, but these methods often display too many suggestions. Meanwhile, the research on Levenshtein distance and Cosine Similarity has not provided appropriate suggestions due to the lack of word grouping. Edit Distance and BERT-based mask language model methods also need to be developed for text correction based on existing contexts [7]. Therefore, the problem statement raised is about how to improve the accuracy of text correction and evaluate the translation quality at the word level after correcting input text in the context of translation from Indonesian to English.

Previous research conducted by Youness Chaabi and his colleagues, applying Damerau-Levenshtein algorithm and N-gram has handled spelling errors in Amazigh language, however, this combination corrects only spelling errors [6]. In contrast, Viny Christanti M and her team's study successfully handled spelling errors using Trie and Damerau-Levenshtein distance Bigram. However, this approach requires substantial memory usage for word correction [8]. Moreover, Romila Aziz and team has implemented Damerau-Levenshtein distance Trigram in correcting context errors, still this research can be enhanced to detect and correct multiple contextual errors in a single sentence [9]. Other research related to the combination of Levenshtein distance and Rabin-Karp algorithms conducted by Andre Hasudungan Lubis and team, succeeded in showing that the combination of the two algorithms performs good accuracy and its application can be improved based on certain parameters, such as N-Gram, Base and Modulo [10]. Similarly, Aaqila Dhiyaanisafa Goenawan applied Trie and depth-first search algorithms to implement search suggestions, yet this research's limitation lies in its inability to display frequently user-inputted word recommendations [11]. Another study by Kavita T. Patil and colleagues corrected Marathi words using combinations of Levenshtein distance and Cosine Similarity. However, word suggestions in Marathi did not yield precise results due to ungrouped words in the vocabulary set [12].

Based on previous research, several issues such as excessive and ungrouped word correction results, imprecise word suggestions, and high memory usage require further evaluation. Enhancing Levenshtein distance and Trie algorithm could lead to more efficient text correction evaluations [13]. Trie data structures can be stored for text processing, reducing memory usage in grouping texts based on similar prefixes, combined with Levenshtein distance algorithm to calculate the distance between texts. This combination offers an excellent balance between speed and accuracy. The application of Trie allows for fast and efficient similar prefix searches [14], while Levenshtein distance algorithm provides high accuracy in detecting and correcting spelling errors and is not affected by the length of the text. The contributions of our work include the following items:

- Introduce the combination of Levenshtein distance and Trie algorithm for text correction purpose.
- Evaluate the combination of Levenshtein distance and Trie algorithm using Confusion Matrix, AUC-ROC Curve and BLEU Score measurement.

This paper is structured as follows: 1. Introduction, 2. The Proposed Method 3. Method, 4. Results and Discussion, and 5. Conclusion. Section 1 introduces the proposed approach. Section 2 describes related work done in this area. Section 3 describes research method. Section 4 describes the results of the proposed approach followed by discussion, and section 5 concludes the paper.

2. THE PROPOSED METHOD

Various researches have been conducted in the context of text correction using combinations of the Levenshtein distance Algorithm. Research on the combination of Damerau Levenshtein distance and N-Gram was conducted by Youness Chaabi and his colleagues [6] to identify spelling errors in Amazigh corpus language. The N-Gram method was used to measured for random spelling errors from the dictionary. The results showed the achieved F-measure value ranged from 86.62% to 98.74%.

Other research on the combination of edit distance algorithms was conducted by Viny Christanti and team [8] by combining Damerau-Levenshtein distance with Trie in the context of spell checking. The study showed a word correction accuracy of 84.62% and a sentence correction accuracy of 50%, with a processing time per sentence of 18.89 ms. Furthermore, the research on the combination of Levenshtein distance and Rabin-Karp algorithms was conducted by Andre Hasudungan Lubis and team [10] to improve the accuracy of document similarity. The Rabin-Karp algorithm is a search algorithm that looks for substring patterns in text using hashing, typically used to match text with various patterns. In this study, Levenshtein distance replaced the hash calculation in the Rabin-Karp algorithm by calculating the number of hashes with the same value in both documents. By adding the Levenshtein distance algorithm, the distance calculation between the two documents resulted in better accuracy. The study showed that the combination of these two algorithms produced better accuracy, which could be further improved based on certain parameters, such as N-Gram, Base, and Modulo.

Another research combining enhanced N-gram and Levenshtein distance algorithm conducted by Salah Al-Hagree and his team showed similarity mean of 0.90 and F-measure mean of 0.91 in name matching [15]. A similar study on the autocomplete text feature in an Indonesian dictionary conducted by Goenawan and team [11] was also implemented with the Trie and depth-first search algorithm. This combination applied to shorten the user's typing time, where this feature would display a list of words that the user might mean without having to type the word completely. The results included implementing search suggestions based on node location in the testing.

Similar study was conducted on language models (LMs) by applying a similar algorithm, byte pair encoding (BPE), to evaluate the model by comparing the implementation of BPE tokenization and Unigram LM tokenization [16]. The experimental results showed that the adjustment model trained with Unigram LM tokenization performed better than the adjustment model trained with BPE tokenization. This study showed that although the BPE algorithm was quite useful in translating OOV words in the dataset, most OOV words were still incorrectly translated [17].

Research combining the Edit Distance algorithm was again conducted by Fatemeh Tohidian [7] with a BERT-based masked language model for spell correction. BERT is a transformer-based architecture initially trained on a masked language model to find misspelled word candidates and select the best candidate based on Edit Distance calculations. The study showed that the combination of masked language model and Edit Distance resulted in a recall of 81.21% in spell correction obtained from prediction ranking and combining character-level criteria (edit distance) with BERT prediction scores. In the same year, a comparative study was conducted comparing the use of the minimum edit distance algorithm with Cosine Similarity to check and correct spelling errors in Marathi. The minimum edit distance is a character-based approach, while Cosine similarity is a similarity approach based on certain terms. The study showed the best accuracy of 85.88% and 86.76% with the use of minimum edit distance and Cosine similarity [12].

The latest research on the Levenshtein distance algorithm combined with several string matching algorithms, such as the Gestalt and SweetCoat-2D algorithms, was conducted to find comparison patterns of the two strings in the Bangla language. The Gestalt algorithm determines text data patterns using machine learning and can predict text based on these patterns using the SweetCoat-2D model. SweetCoat-2D is built on a two-dimensional model, where the position of a point can be determined using x and y-axis values. The results of this study showed that the application of spelling suggestions was successfully improved by 92.404%, with accuracy calculated based on correct spelling suggestions in the Bangla language [18]. Hereafter in the same year, the research using augmented Damerau-Levenshtein distance successfully developed by Nurzhan Mukazhanov and team, resulted best accuracy of 92.8% in Kazakh text [19]. Summary of the literature review for combination of Levenshtein distance and Trie presented in Table 1.

Table 1. Related work of combination of Levenshtein distance and Trie algorithm

Year	Methods	Language	Dataset	Result
2018 [8]	Trie and damerau-Levenshtein distance bigram	Indonesian	Kompas News	Accuracy 84.62 Time 18.89 ms
2018 [10]	Levenshtein distance and rabin karp	English	Documents	10% of document similarity
2019 [15]	N-gram and Levenshtein distance	English, Portugese, and Arabic	English, Portugese, and Arabic names	Similarity mean 0.90 F-Measure mean 0.91
2020 [16]	Byte pair encoding	English-Japanese	Wikipedia	Model trained with Unigram LM performed better
2022 [17]	Byte pair encoding	German-English, Russian-English, and Romanian-English	WMT2020, WMT2016, WMT2017	BPE algorithm was quite useful in translating OOV words in the dataset
2022 [6]	Damerau Levenshtein distance and N-gram	Amazigh	Amazigh corpus	F-measure ranged from 86.62% to 98.74%
2022 [11]	Trie and depth-first search	Indonesian	Indonesian Dictionary	Implementing search suggestions based on node location in the testing
2023 [7]	BERT-based masked language model and edit distance	English	Neuspell dataset	Recall 81.21%
2023 [12]	Minimum edit distance and cosine similarity	Marathi	Marathi documents	MED accuracy 85.88% Cosine accuracy 86.76%
2023 [18]	Levenshtein edit distance and string-matching algorithm	Bengali	Bengali documents	Accuracy 92.404%
2023 [19]	Damerau Levenshtein distance	Kazakh	Kazakh texts	Accuracy 92.8 %.

3. METHOD

This section described the combination of Levenshtein distance and Trie algorithm as the method. This combination method used to evaluating text corrections to obtain appropriate translation results. Figure 1 illustrated the following steps of the research method.

3.1. Data collection

In the data collection section, data used in this research includes the Indonesian language dictionary as a comparison to the input text using the kbbi-python library and 200 input text data collected from Indonesian news with various spelling errors [20]. The spelling errors were generated by a machine previously processed in research conducted in 2020, categorized based on the text length: short text, medium text, and long text. Total amount of input text data used is 200 contains of 57.098 words, consisting of 100 texts without spelling errors and 100 texts with various types of spelling errors, specifically lexical errors, with 30% training and 70% testing data, amounting to 140 training data and 60 testing data

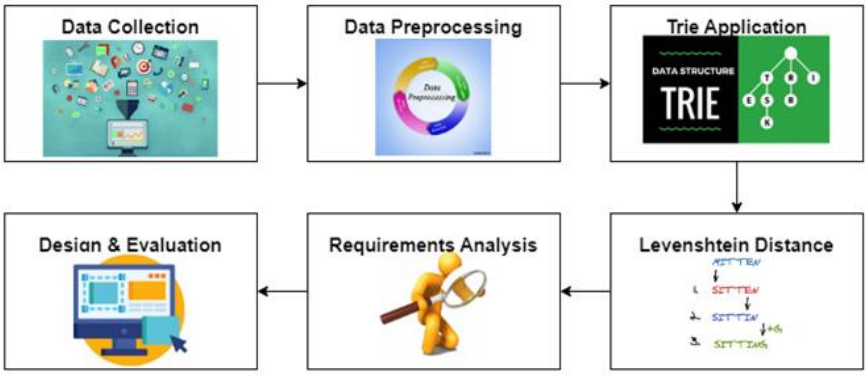

Figure 1. Research method

3.2. Data preprocessing

Before processing data using the combination of Levenshtein distance and Trie algorithm, the text data must go through preprocessing steps. Preprocessing used to remove data inconsistencies in the following steps, such as parsing, case folding, and tokenizing. In this context, the preprocessing steps do not correct the text data [21]. The preprocessing steps illustrated in Figure 2 [22].

Figure 2. Data preprocessing

3.1. TRIE APPLICATION

In Algorithm 1 Trie data structure can be applied by creating a class to search for text with similar prefixes [23]. The purpose is for grouping several texts that have similarities with an accuracy of more than 50% to proceed to the next step [24].

Algorithm 1. Trie data structure algorithm

```
class TrieNode:
    def __init__(self):
        self.children = {}
        self.is_end_of_word = False
        self.frequency = 0

class Trie:
    def __init__(self):
        self.root = TrieNode()

    def insert(self, word):
        node = self.root
        for char in word:
            if char not in node.children:
                node.children[char] = TrieNode()
            node = node.children[char]
        node.is_end_of_word = True
```

```
        node.frequency += 1
    def search(self, word):
        node = self.root
        for char in word:
            if char not in node.children:
                return False
            node = node.children[char]
        return node.is_end_of_word
```

Applying Trie data structures speed up searching time.based on similar prefixes. Trie data structures grouped all words with the same prefix stored in the list of texts. For example, Trie data structures applied as shown in Table 2.

As shown in Table 2, each text data, whether with or without spelling errors, had grouped similar text data to the input text in purpose to find texts with the same prefix. Trie data structure searched in Indonesian language dictionary using the word of 'ayan' based on two prefixes to find all words with the same 'ay' prefix. The same prefixes stored in the list of texts before move through next step.

Table 2. Trie data structure application

Text	Spelling error text	Right text	Trie data structure
ayan (*False*)	ayan (*False*)	ayam (*True*)	ayah | ayam | ayan | ayom
bali (*True*)		bali (*True*)	bali | balik | balita
betutu (*True*)		betutu (*True*)	betul | betutu
dimasa (*True*)		dimasa (*True*)	dimasa | dimasak | dimasuki

3.3. Levenshtein distance

The calculation of the Levenshtein distance algorithm is performed using string operations such as insertion, deletion, and substitution to compute the character distance [15]. Levenshtein distance algorithm calculated between the input text and the list of texts to find minimum character distance and display the corrected text based on the minimum character distance [25]. The distance is determined by the (1), (2), and (3) function below.

$$lev_{a,b}(i,j) = \begin{cases} 0 \\ i \\ j \\ min \begin{cases} lev_{a,b}(i-1,j)+1 & (1) \\ lev_{a,b}(i,j-1)+1 & (2) \\ lev_{a,b}(i-1,j-1)+1\,[a_i \neq b_j] & (3) \end{cases} \end{cases}$$

The function of the Levenshtein distance algorithm in (1), (2), and (3) implemented on two example strings (string A: ayan; string B: ayam). All characters in both strings compared each other to find either the character is the same or not by using Levenshtein distance calculation. Afterwards, a calculation table can be generated as shown in Table 3.

Table 3. Levenshtein distance algorithm calculation

Position	Character 1	Character 2	Distances	Formula	Result
D(1,1)	a	a	There isn't any	D(1-1,1-1)	0
D(1,2)	a	y	There is	D(1,2-1)+1	1
D(1,3)	a	a	There isn't any	D(1-1,3-1)	2
D(1,4)	a	m	There is	D(1,4-1)+1	3
D(2,1)	y	a	There is	D(2,1-1)+1	1
D(2,2)	y	y	There isn't any	D(2-1,2-1)	0
D(2,3)	y	a	There is	D(2,3-1)+1	1
D(2,4)	y	m	There is	D(2,4-1)+1	2
D(3,1)	a	a	There isn't any	D(3-1,1-1)	1
D(3,2)	a	y	There is	D(3,2-1)+1	2
D(3,3)	a	a	There isn't any	D(3-1,3-1)	0
D(3,4)	a	m	There is	D(3,4-1)+1	1
D(4,1)	n	a	There is	D(4,1-1)+1	2
D(4,2)	n	y	There is	D(4,2-1)+1	2
D(4,3)	n	a	There is	D(4,3-1)+1	1
D(4,4)	n	m	There is	D(4,4-1)+1	1

Based on the results of the Levenshtein distance algorithm calculation in Table 3, the matrix can be filled in according to character position. Each columns defined ‘character 1’ as the text with spelling error and each rows defined ‘character 2’ as the word from the list of texts generated from Trie data structure. Final matrix generated as shown in Table 4. Based on Table 4, it can be seen that D(4, 4) = 1, showing that there is one character difference out of the four characters compared. The word similarity score calculation result is 75%. With this score, the combination algorithm had displayed suggestions in the text correction recommendation table for user to choose one as the correct text correction [26]. Thus, accurate text correction resulted in a good and correct text translation.

Table 4. Levenshtein distance algorithm result

	-	a	y	a	n
-	0	1	2	3	4
a	1	0	1	2	3
y	2	1	0	1	2
a	3	1	2	0	1
m	4	2	2	1	1

3.4. Levenshtein distance and Trie algorithm

Implementation of Levenshtein distance and Trie algorithms is performed using Python programming language. This implementation required an input text in purpose to be corrected and generated translated text as the output. The text correction flowchart can be seen in Figure 3. Based on Figure 3, text correction is carried out by inputting text in the context of language translation. Spelling errors had detected in the input text and if spelling error is detected, Trie data structure search based on similar prefixes [14]. Furthermore, Trie search results stored in a list to be calculated using Levenshtein distance algorithm.

This experiment collected dataset in Indonesian language and splitted into 30% training data and 70% testing data as mentioned in sub-section 3.1 [20]. Dataset had to be processed in preprocessing stage to get cleansed dataset and can be properly processed by the system. At this stage, the preprocessing process would not directly correct spelling error marked by red in Table 5 and only clean the text data into normalized text data. This stage is conducted in several steps, such as parsing, case folding, and tokenizing. Text preprocessing results are also shown in Table 5.

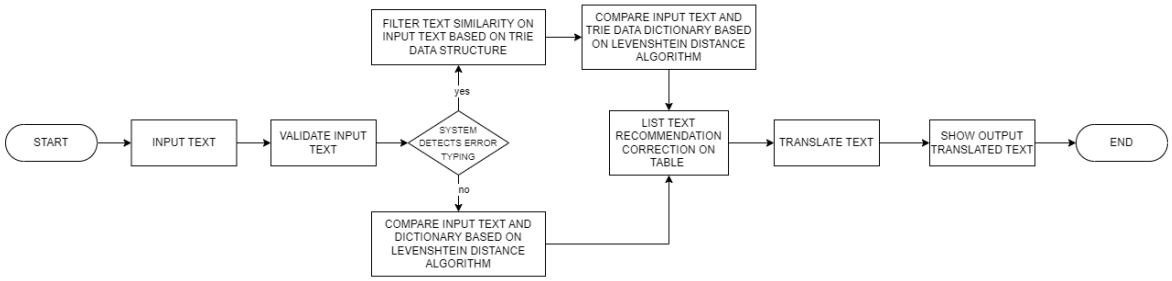

Figure 3. Process flowchart

Table 5. Preprocessing result

Text	Parsing	Case Folding	Tokenizing
Josh Earnest menyatakan tidak mungkin Amerika Serikat mengerahkan pasukan lagi ke Irak. *Josh Earnest stted that it is not possible for the United States to deploy troops to Iraq again.*	Josh Earnest menyatakan tidak mungkin Amerika Serikat mengerahkan pasukan lagi ke Irak. *Josh Earnest stted that it is not possible for the United States to deploy troops to Iraq again.*	josh earnest menyatakan tidak mungkin amerika serikat mengerahkan pasukan lagi ke irak *josh earnest stted that it is not possible for the united states to deploy troops to iraq again.*	[‘josh’, ‘earnest’, ‘menyatakan’, ‘tidak’, ‘mungkin’, ‘amerika’, ‘serikat’, ‘mengerahkan’, ‘pasukan’, ‘lagi’, ‘ke’, ‘irak’] [*‘josh’, ‘earnest’, ’stted’, ‘that’, ‘it’, ‘is’, ‘not’, ‘possible’, ‘for’, ‘the’, ‘united’, ‘states’, ‘to’, ‘deploy’, ‘troops’, ‘to’, ‘iraq’, ‘again’*]

Trie data structure is applied to search for text data with similar prefixes in the dictionary used as a comparison for text correction purposes [24]. Levenshtein distance algorithm calculated to find the smallest distance between the original text and the list of texts from the dictionary used after searching for text data

with similar prefixes using Trie data structure to find the smallest distance in the purpose of text correction. Levenshtein distance calculation resulted as shown in Table 6. Levenshtein distance algorithm used to calculate the smallest distance between Indonesian texts as shown in Table 6. After the calculation results obtained, the distance between texts sorted from the smallest distance obtained to the largest distance in the suggestion list displayed. The top correction result is displayed as the best matches text correction to the input text as in Table 7. Based on Table 7, it can be seen that suggestions sorted based on Levenshtein distance calculation distance as the result of text correction.. It shown that first suggestion had the shortest distance using the combination of Levenshtein distance and Trie algorithm. First suggestion of the top suggestions selected as the result of text correction, translated and displayed as translated text.

Table 6. Levenshtein distance calculation results

Position	Character 1	Character 2	Distances	Formula	Result
d[1] [1]	j	j	There isn't any	d[1-1] [1-1]	0
d[1] [2]	j	o	There is	d[1] [2-1] + 1	1
d[1] [3]	j	s	There is	d[1] [3-1] + 1	2
d[1] [4]	j	h	There is	d[1] [4-1] + 1	3
d[1] [5]	j		There is	d[1] [5-1] + 1	4
d[1] [6]	j	e	There is	d[1] [6-1] + 1	5
d[1] [7]	j	a	There is	d[1] [7-1] + 1	6
d[1] [8]	j	r	There is	d[1] [8-1] + 1	7
d[1] [9]	j	n	There is	d[1] [9-1] + 1	8
d[1] [10]	j	e	There is	d[1] [10-1] + 1	9
d[1] [11]	j	s	There is	d[1] [11-1] + 1	10
d[1] [12]	j	t	There is	d[1] [12-1] + 1	11
d[1] [13]	j		There is	d[1] [13-1] + 1	12
d[1] [14]	j	m	There is	d[1] [14-1] + 1	13
d[1] [15]	j	e	There is	d[1] [15-1] + 1	14
d[1] [16]	j	n	There is	d[1] [16-1] + 1	15
d[1] [17]	j	y	There is	d[1] [17-1] + 1	16
d[1] [18]	j	a	There is	d[1] [18-1] + 1	17
d[1] [19]	j	t	There is	d[1] [19-1] + 1	18
d[1] [20]	j	a	There is	d[1] [20-1] + 1	19
d[1] [21]	j	k	There is	d[1] [21-1] + 1	20
d[1] [22]	j	a	There is	d[1] [22-1] + 1	21
d[1] [23]	j	n	There is	d[1] [23-1] + 1	22
…	…	…	…	….	…

Table 7. Sorted suggestions text

Rank	Original Text	Suggestion Text based on Dictionary	Distances
1st	josh earnest menytakan tidak mungkin amerika serikat mengerahkan pasukan lagi ke irak *josh earnest stted that it is not possible for the united states to deploy troops to iraq again.*	josh earnest menyatakan tidak mungkin amerika serikat mengerahkan pasukan lagi ke irak *josh earnest stated that it is not possible for the united states to deploy troops to iraq again.*	1 (suggested)
2nd	josh earnest menytakan tidak mungkin amerika serikat mengerahkan pasukan lagi ke irak *josh earnest stted that it is not possible for the united states to deploy troops to iraq again.*	serikat pekerja jakarta international school jis mendatangi komisi kepolisian nasional kompolnas untuk melayangkan pengaduan…… *the jakarta international school (jis) workers union visited the national police commission (kompolnas) to file a complaint ……*	1541
3rd	josh earnest menytakan tidak mungkin amerika serikat mengerahkan pasukan lagi ke irak *josh earnest stted that it is not possible for the united states to deploy troops to iraq again.*	serikat pekerja jakarta international school jis mendatangi komisi kepolisian nasional kompolnas untuk mengadukan keberatan…… *the jakarta international school (jis) workers union visited the national police commission (kompolnas) to lodge a complaint ……*	2097

4. RESULTS AND DISCUSSION

This research conducted in order to evaluating the combination of Levenshtein distance and Trie algorithm in the context of text correction. Previous research have proposed various methods to handle text correction problems, however, there's limitation of the previous research, such as excessive word correction results caused of ungrouped word, imprecise word suggestions, and high memory usage. We found that enhancing Levenshtein distance and Trie algorithm could lead to more efficient text correction. Trie data

structures reduced memory usage in grouping based on similar prefixes, combined with Levenshtein distance to calculate distance. As mentioned in sub-section 3.1, dataset used in consisting of 100 texts without spelling errors and 100 texts with various types of spelling errors. After the experiment done, evaluation measured by calculating performance evaluation metrics.

5. CONFUSION MATRIX EVALUATION

Evaluation measured by calculating accuracy, precision, recall and F-measure as the performance evaluation metrics. Precision is defined as the number of correctly detected spelling errors compared to all detected spelling errors. Recall is defined as the number of correctly detected spelling errors divided by the total number of corrected text in the dataset. The experimental results shown in Table 8. The performance evaluation metrics have to be compared to other algorithm combinations as benchmarking. Evaluation results of the combination of Levenshtein distance and Trie algorithm using the Confusion Matrix are shown in Table 9.

Table 8. Experimental results using the combination of Levenshtein distance and Trie

Evaluation	Result
Total errors in dataset	400
Detected spelling errors and corrected (TP)	192
Detected spelling errors and not corrected (FN)	8
Not detected spelling errors and corrected (FP)	63
Not detected spelling errors and not corrected (TN)	137
Recall rate	192/(192+8) = 0.9600
Precision rate	192/(192+63) = 0.7529
F1 Score rate	2(0.9600)(0.7529)/(0.9600+0.7529) = 0.8439

Table 9. Confusion matrix evaluation

Evaluation	LD and Trie (Proposed)	LD and N-Gram [15]	LD and Rabin Karp [10]
Accuracy	0.8225	0.7500	0.2825
Recall	0.9600	0.5000	0.5050
Precision	0.7529	1.0000	0.3495
F1 Score	0.8439	0.6667	0.4131

Based on Table 9, the evaluation was conducted to measure the performance of the combination of Levenshtein distance algorithm and Trie in text correction by calculating accuracy, recall, precision, and F1 score. This evaluation compared with the combination of other algorithm combinations, such as the combination of Levenshtein distance and N-Gram, and the combination of Levenshtein distance and Rabin-Karp algorithm. From the result illustrated in Table 9, we found that the combination of Levenshtein distance and Trie performs better than the other algorithm combinations.

5.1. AUC-ROC evaluation

The evaluation results can be visualized using threshold settings with the AUC ROC curve. Therefore, calculations were performed using the values generated from the confusion matrix to calculate true positive rate (TPR) and false positive rate (FPR). The calculation results are shown in Table 10.

Table 10. TPR-FPR calculation

Threshold	TPR	FPR
0.1	0.010204	0.349057
0.2	0.033835	0.529851
0.3	0.070248	0.651899
0.4	0.107477	0.736559
0.5	0.176768	0.816832
0.6	0.244186	0.868421
0.7	0.335616	0.909449
0.8	0.46875	0.955882
0.9	0.663265	0.976821
1.0	1	1

Based on Table 10, the AUC ROC curve can be created using the calculated true positive rate (TPR) and false positive rate (FPR). The higher TPR value is, it can be seen that the combination of Levenshtein distance and Trie algorithm provide better results than the other combination in the context of text correction. AUC-ROC Curve are shown in Figure 4.

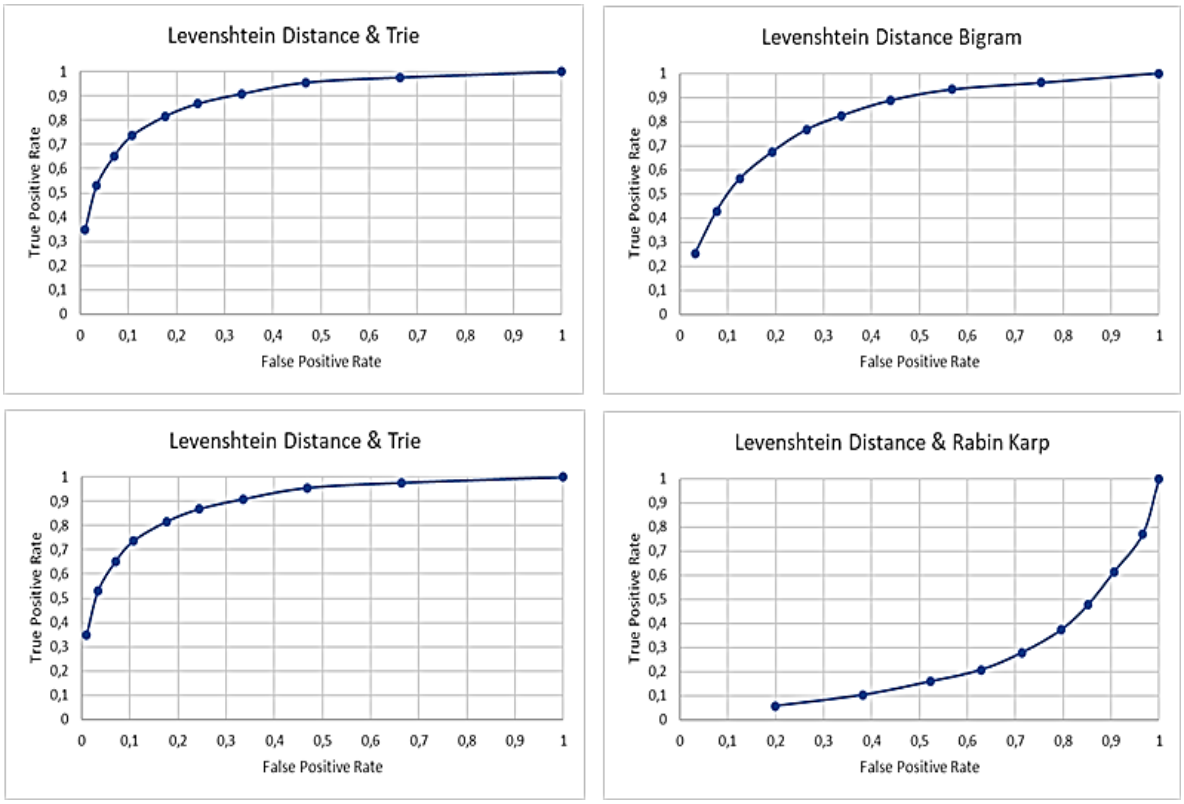

Figure 4. AUC-ROC Curve

5.2. Bleu score evaluation

Evaluation results can be enhanced to measure the quality of translation results after text correction by calculating the bilingual evaluation understudy score (BLEU) using Unigram Precision. Good translation quality can be proven by BLEU score approaching to 1. As shown in Table 11, BLEU score increased by 0.0155 or about 1.55% after text correction was performed.

Table 11. BLEU score evaluation

Evalution	BLEU score
BLEU Score before text correction	0.6602
BLEU Score after text correction	0.6757

6. CONCLUSION

Based on the results and discussions that have been conducted, the research successfully developed and evaluated text correction using a combination of Levenshtein distance algorithm to measure the distance between texts, using the application of Trie data structure in text search based on the prefix of the input text, trained using 70% of the training data and tested with 30% of the testing data from 200 text data scraped. Combination of Levenshtein distance algorithm and Trie achieved an accuracy rate of 82.25% with an F1 score of 84.39%. This text correction model performed good translations using BLEU score with an increase of 1.55% after text correction was performed.

This research evaluated text correction using combination of Levenshtein distance and Trie algorithm. The combination used dictionary of Indonesian language to find and correct spelling error focused

in lexical error. Suggestion for further research using the combination of Levenshtein distance and Trie can be conducted by applying this combination to multilanguage texts with semantic level, including by adding more training data and test data in terms of increasing the accuracy value and F1 Score.

ACKNOWLEDGEMENTS

Author would like to thank Bina Nusantara University for its support and resources provided throughout this research. Author also acknowledge Prof. Abba Suganda Girsang for his valuable suggestions and guidance that helped improve the quality of this research. All individuals acknowledged have provided consent to be mentioned in this section.

FUNDING INFORMATION

This research is supported by the funding from Directorate of Research, Technology, and Community Service of Ministry of Education, Culture, Research, and Technology of the Republic of Indonesia.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Cynthia Natalie	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓			
Abba Suganda Girsang										✓		✓	✓	✓

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

INFORMED CONSENT

Informed consent was not applicable to this research because no human participants were directly involved. This research utilized secondary textual data obtained from publicly available sources, including an Indonesian language dictionary accessed through the kbbi-python library and Indonesian news text samples.

ETHICAL APPROVAL

Ethical approval was not required for this research because it did not involve human participants, animals, or sensitive personal data. This research was conducted using secondary textual data, including an Indonesian language dictionary accessed through the kbbi-python library and Indonesian news text samples for text mining and spelling-correction evaluation.

DATA AVAILABILITY

The data supporting the findings of this research consist of an Indonesian language dictionary accessed through the kbbi-python library and 200 Indonesian news text samples containing both correct and misspelled words, randomly sampled from the Indonesia News Dataset available on Kaggle (https://www.kaggle.com/datasets/azizainunnajib/indonesia-news).

REFERENCES

[1] D. K. Izmayanti, "Project based learning in Indonesian-Japanese translation course," *Prosiding MINASAN*, vol. 4, pp. 52–63, 2023.

[2] S. C. Tsai, "Using google translate in EFL drafts: a preliminary investigation," *Computer Assisted Language Learning*, vol. 32, no. 5–6, pp. 510–526, Jul. 2019, doi: 10.1080/09588221.2018.1527361.

[3] B. F. Dhini, A. S. Girsang, U. U. Sufandi, and H. Kurniawati, "Automatic essay scoring for discussion forum in online learning based on semantic and keyword similarities," *Asian Association of Open Universities Journal*, vol. 18, no. 3, pp. 262–278, Dec. 2023, doi: 10.1108/AAOUJ-02-2023-0027.

[4] G. Z. Nabiilah, S. Y. Prasetyo, Z. N. Izdihar, and A. S. Girsang, "BERT base model for toxic comment analysis on Indonesian social media," in *Procedia Computer Science*, Elsevier B.V., 2022, pp. 714–721, doi: 10.1016/j.procs.2022.12.188.

[5] B. Juarto and A. S. Girsang, "Neural collaborative with sentence BERT for news recommender system," *International Journal on Informatics Visualization*, vol. 5, no. 4, pp. 448–455, 2021, [Online]. Available: www.joiv.org/index.php/joiv

[6] Y. Chaabi and F. A. Allah, "Amazigh spell checker using damerau-levenshtein algorithm and N-gram," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 8, pp. 6116–6124, 2022.

[7] F. Tohidian, A. Kashiri, and F. Lotfi, "BEDSpell: spelling error correction using BERT-based masked language model and edit distance," in *International Conference on Service-Oriented Computing*, Springer, 2022, pp. 3–14, doi: 10.1007/978-3-031-26507-5_1.

[8] M. V. Christanti and D. S. Rudy, "Fast and accurate spelling correction using Trie and damerau-Levenshtein distance bigram," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 16, no. 2, pp. 827–833, 2018, doi: 10.12928/telkomnika.v16i2.6890.

[9] R. Aziz *et al*., "Real word spelling error detection and correction for urdu language," *IEEE Access*, pp. 1–1, Sep. 2023, doi: 10.1109/access.2023.3312730.

[10] A. H. Lubis, A. Ikhwan, and P. L. E. Kan, "Combination of Levenshtein distance and rabin-karp to improve the accuracy of document equivalence level," *International Journal of Engineering and Technology*, vol. 7, no. 2.27, pp. 17–21, 2018.

[11] A. D. Goenawan, A. Ammar, M. P. Pulungan, and D. Komalasari, "Autocomplete Indonesian dictionary with Trie and depth-first search algorithm," *Jurnal Ilmiah Teknik Informatika dan Komunikasi*, vol. 2, no. 1, pp. 99–103, 2022.

[12] K. T. Patil, R. P. Bhavsar, and B. V Pawar, "Contrastive study of minimum edit distance and cosine similarity measures in the context of word suggestions for misspelled Marathi words," *Multimedia Tools and Applications*, vol. 82, no. 10, pp. 15573–15591, 2023.

[13] M. N. Samsuri, A. Yuliawati, and I. Alfina, "A comparison of distributed, PAM, and Trie data structure dictionaries in automatic spelling correction for indonesian formal text," in *2022 5th International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 525–530, doi: 10.1109/ISRITI56927.2022.10053062.

[14] A. P. Saikia "Enhancing expertise identification in community question answering systems (CQA) using a hybrid approach of TRIE and semantic matching algorithms," *Annals of Multidisciplinary Research, Innovation and Technology (AMRIT)*, 2023.

[15] S. Al-Hagree, M. Al-Sanabani, M. Hadwan, and M. A. Al-Hagery, "An improved n-gram distance for names matching," in *2019 1st International Conference of Intelligent Computing and Engineering: Toward Intelligent Solutions for Developing and Empowering our Societies, ICOICE 2019*, Institute of Electrical and Electronics Engineers Inc., Dec. 2019, doi: 10.1109/ICOICE48418.2019.9035154.

[16] K. Bostrom and G. Durrett, "Byte pair encoding is suboptimal for language model pretraining," *In Findings of the Association for Computational Linguistics: EMNLP 2020*, Apr. 2020, [Online]. Available: http://arxiv.org/abs/2004.03720.

[17] A. Araabi, C. Monz, and V. Niculae, "How effective is byte pair encoding for out-of-vocabulary words in neural machine translation?," *In Proceedings of the 15th biennial conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, Aug. 2022, [Online]. Available: http://arxiv.org/abs/2208.05225.

[18] M. M. Hasan, D. D. Mallick, T. Khan, M. D. M. Alam, M. H. K. Mehedi, and A. A. Rasel, "SweetCoat-2D: two-dimensional bangla spelling correction and suggestion using levenshtein edit distance and string matching algorithm," in *2023 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2023, pp. 1–8.

[19] N. Mukazhanov, Z. Alibiyeva, A. Yerimbetova, A. Kassymova, and N. Alibiyeva, "Development of an augmented Damerau-Levenshtein method for correcting spelling errors in Kazakh texts," *Eastern-European Journal of Enterprise Technologies*, vol. 5, no. 2(125), pp. 23–33, 2023, doi: 10.15587/1729-4061.2023.289187.

[20] Transmedia Group, "CNN Indonesia," Accessed: Jan. 06, 2024. [Online]. Available: https://www.cnnindonesia.com/

[21] P. Santoso, P. Yuliawati, R. Shalahuddin, and A. P. Wibawa, "Damerau Levenshtein distance for Indonesian spelling correction," *Jurnal Informatika*, vol. 13, no. 2, p. 11, Jul. 2019, doi: 10.26555/jifo.v13i2.a15698.

[22] A. A. Al-Shalabi, G. Al-Gaphari, S. Al-Hagree, F. Alqasemi, and A. Info, "Investigating the impact of utilizing the K-nearest neighbor and Levenshtein distance algorithms for arabic sentiment analysis on mobile applications," *Sana'a University Journal of Applied Sciences and Technology*, vol. 1, no. 2, pp. 175–188, 2023, [Online]. Available: https://journals.su.edu.ye/index.php/jast

[23] K. K. Maurya, M. S. Desarkar, M. Gupta, and P. Agrawal, "Trie-NLG: Trie context augmentation to improve personalized query auto-completion for short and unseen prefixes," *Data Mining and Knowledge Discovery*, Jul. 2023, [Online]. Available: http://arxiv.org/abs/2307.15455.

[24] R. Yeasin, E. Sharmistha, and C. Tista, "An efficient word lookup system by using improved Trie algorithm," *arXiv preprint arXiv:1911.01763*., 2019, doi: 10.48550/arXiv.1911.01763.

[25] A. Naziri and H. Zeinali, "A comprehensive approach to misspelling correction with BERT and Levenshtein distance," arXiv preprint arXiv:2407.17383, [Online]. Available: https://taaghche.com

[26] V. C. Mawardi, F. Augusfian, J. Pragantha, and S. Bressan, "Spelling correction application with damerau-Levenshtein distance to help teachers examine typographical error in exam test scripts," in *E3S Web of Conferences*, EDP Sciences, Sep. 2020. doi: 10.1051/e3sconf/202018800027.

BIOGRAPHIES OF AUTHORS

Cynthia Natalie enrolled as a student majoring in computer science at Bina Nusantara University. Cynthia is building upon the foundational knowledge acquired during her undergraduate studies, where she earned a bachelor's degree in computer science from Tarumanagara University. Her professional journey includes pivotal roles such as Human Resources Information System and Data Analyst at PT Bina Karya Prima from 2023 to 2024. Furthermore, Cynthia gaining hands-on experience as Lecturer's Assistant at the Faculty of Computer Science on Tarumanagara University for 6 months and embarked on internships to deepen her practical understanding as RPA Developer at PT IDStar Cipta Teknologi from 2022 to 2023. For further information, she can be contacted at email: cynthia.natalie@binus.ac.id.

Abba Suganda Girsang is presently employed as a lecturer in the Master of Information Technology program at Bina Nusantara University in Jakarta. He completed his Ph.D. at the Institute of Computer and Communication Engineering within the Department of Electrical Engineering at National Cheng Kung University in Tainan, Taiwan, in 2014. His undergraduate studies were undertaken at the Department of Electrical Engineering at Gadjah Mada University (UGM) in Yogyakarta, Indonesia, graduating in 2000. Following this, he pursued his master's degree in the Department of Computer Science at the same institution from 2006 to 2008. Over the course of his career, he has amassed experience as a staff consultant programmer at Bethesda Hospital in Yogyakarta in 2001, as well as working as a web developer from 2002 to 2003. Subsequently, he transitioned to the Faculty of the Department of Informatics Engineering at Janabadra University, where he served as a lecturer from 2003 to 2015. Additionally, he has taught various subjects at different universities from 2006 to 2008. Abba Suganda Girsang's research interests include swarm intelligence, combinatorial optimization, and decision support systems. He can be contacted at email: agirsang@binus.edu.