JIKO (Jurnal Informatika dan Komputer) Accredited KEMENDIKBUD RISTEK, No.105/E/KPT/2022
Vol. 6, No. 3, December 2023, pp. 214-220 p-ISSN: 2614-8897
DOI: 10.33387/jiko.v6i3.7072 e-ISSN: 2656-1948

COMPARISON OF SPELL CORRECTION IN BAHASA INDONESIA: PETER NORVIG, LSTM, AND N-GRAM

Anggasta TA Kusuma[1], Chanifah I Ratnasari*[2]

[1,2] Program Studi Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia, Indonesia
*Email: [1]19523126@students.uii.ac.id, [2]chanifah.indah@uii.ac.id

(Received: 3 December 2023, Revised: 15 December 2023, Accepted: 20 December 2023)

Abstract

This study conducts a comprehensive comparison of spell-checking methods in Bahasa Indonesia, specifically focusing on three approaches: Peter Norvig's method, Long Short-Term Memory (LSTM), and N-gram. The primary metric for evaluation is the accuracy in correcting spelling errors. Notably, Peter Norvig's method outperforms the others, with N-gram following closely, and LSTM trailing behind. The study draws valuable insights that contribute to the enhancement of spelling correction accuracy in the Bahasa Indonesia language. To carry out the evaluation, the research employs SPECIL data (Spell Error Corpus for Indonesian Language), which includes documents with various error types such as insertion, deletion, transposition, and substitution. The testing dataset consists of 150 words, aligning with the 150-word corpus references from the 'Leipzig Corpora Collection' used for Peter Norvig's and N-gram methods. Peter Norvig's method stands out as the most robust, achieving an impressive accuracy rate of 89%. The N-gram method follows closely with a 75% accuracy rate, showcasing its effectiveness. Meanwhile, LSTM, while still providing reasonable accuracy at 74%, trails behind the other two approaches. It's noteworthy that the LSTM method utilizes a reference dataset from SPECIL, comprising 150 data points and specifically focusing on insertion errors for the test data. This research provides valuable insights for researchers, developers, and language technology enthusiasts seeking to refine spell-checking techniques for the Bahasa Indonesia language. By leveraging diverse error types and a standardized testing dataset, the study aims to contribute to the continual improvement of spell-checking tools.

Keywords: *Spelling Correction, Peter Norvig, LSTM, N-gram, Spell Check*

**Corresponding Author: Chanifah I Ratnasari*

1. INTRODUCTION

In 2023, 82% of all internet data is comprised of text, as reported by the Cisco Visual Networking Index. This underscores the dominance of text as the predominant data type on the internet. Textual data can be found in various forms, such as news articles, blogs, social media posts, and official documents. To illustrate the magnitude of textual data on the internet, as of January 2022, Google processed more than 5.6 billion searches per day, each search encompassing text in various languages and topics [1]. The openness of information on the internet has made the compilation and management of textual data an increasingly complex task. This data varies in quality, accuracy, and relevance, necessitating careful processing. In an academic context, precise and typo-free writing is of paramount importance [2]. Research, assignments, papers, and scientific reports require the ability to compose clear, cohesive, and accurate text. Writing errors can diminish credibility, disrupt comprehension, interpretation of text, and the impact of the writing. This can lead to confusion, affect the validity of the writing, and undermine the impression of professionalism. Hence, error-free writing is essential.

Typographical errors in documents are clearly produced by a variety of factors, including unintentional errors, mechanical faults, hand or finger slips, and the proximity of letters on the keyboard [3]. The system that can help detect errors and provide suggestions for the correct words is the spelling correction or spelling suggestion system. This system's function is to detect errors and provide alternative word recommendations [4]. Spelling correction includes two types of checking: real-word spell checking and non-word error spell checking. While real-word spell checking focuses on processing words that remedy flaws in the phrase, non-word error spell checking deals with misspelled words caused by typographical errors [2].

Several spelling-check studies focus more on typographical errors as the source of word errors [5]. In relation to earlier research, the study conducted by [6] determined the common type of spelling error by utilizing Levenshtein distance and N-gram. This study used 4,453 misspelling datasets in English gathered by Wikipedia contributors. This dataset gives the right words for each misspelled word token and addresses typographical issues in Wikipedia articles. In the evaluation stage of this research, recall calculations were processed using the correct words to achieve the research objectives. The output findings reveal that the Levenshtein distance has a greater recall value than the N-gram, with 79% and 65%, respectively. Another similar study was conducted by [7], examining spelling correction using Peter Norvig and N-gram. According to the findings of this study, the Peter Norvig approach is incapable of correcting spelling problems, such as sentences with two misspellings in a single word. There are also a few sentences that include personal names. As a result, the terms containing those surnames are considered spelling mistakes because they are not included in the KBBI dictionary word list. Using 55 texts as test material, the spelling correction accuracy value is 69.09%. Another study was undertaken by [8], which used the LSTM (Long Short-Term Memory) approach to perform a spell check. There are 12,961 unique words and 100,000 words in the tiny data set used to train and test sizes. For the massive data set, 80% of the total data set is used for training and 20% for testing. The reasoning behind evaluating both small and large data sets is that some applications, such as query correction, require just terms from dictionaries with a limited vocabulary. The LSTM approach has 73.77% accuracy and a processing time of 0.328 seconds per word.

In this study, we will compare numerous spelling correction methods, including Peter Norvig, N-gram, and LSTM. The approach itself is used because it is flexible and has parallels with other methods. Three techniques Peter Norvig, n-gram, and LSTM are capable of handling big datasets. The capacity to maximize spelling correction accuracy improves with dataset size. Presenting the findings, we show how our suggested method works more accurately than previous techniques, particularly when managing intricate linguistic structures that are exclusive to the Indonesian language. Beyond the particular techniques employed, our study adds value by shedding light on the difficulties associated with correcting spelling in Indonesian and laying the groundwork for further studies in the area of natural language processing. Our results can be used by other scholars working in this field to improve their approaches to spell checking and correction specifically for the Indonesian language, which will ultimately help to advance NLP applications in the area.

2. RESEARCH METHOD

In this study, a model will be built for checking and correcting spelling in Indonesian utilizing the Peter Norvig, N-gram, and LSTM approaches. This study intends to improve spelling correction accuracy through a systematic series of processes. Figure 1 depicts the stages of this study. The first stages involve developing a corpus and collecting test data, followed by data preprocessing techniques such as tokenization and case folding. Following that, several spelling correction methods, such as those proposed by Peter Norvig, N-gram, and LSTM, will be built and applied to the test data. Each method is based on its own set of principles. Performance evaluation is conducted by comparing the accuracy of each method, and the one yielding the best results is identified. This research focuses on the development and evaluation of a spelling correction model with the goal of reaching the highest level of correctness.

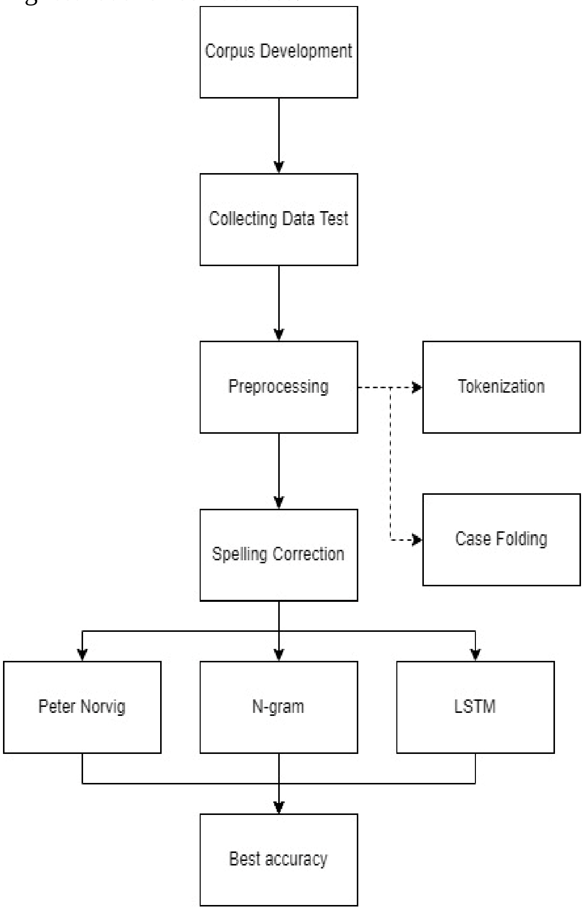

Figure 1. Research methodology

2.1 Corpus Development

The corpus dataset used for spelling correction comparison is obtained from the “Worschatz Leipzig” website, with reference data of 10,000 words in the format of a text file (.txt). This website offers services in a growing number of languages under the name

Leipzig Corpora Collection. The site provides the most extensive publicly available text resources in many languages. The selection of this dataset is based on the completeness and comprehensiveness of the word sources it contains in the Indonesian language. The reason for choosing this corpus is to validate sentences or words that are correct in the KBBI.

2.2 Collecting Test Data

This research's data was gathered utilizing a method. To have a more thorough grasp of the data being processed, data was gathered and examined [9]. The test data used to compare several approaches, such as Peter Norvig, N-gram, and LSTM, was gathered from the Kaggle website named SPECIL (Spell Error Corpus for Indonesian Language). This corpus can be used by practitioners and academics to identify and correct spelling mistakes in the Indonesian language. This study's data has a total of 21,500 entries.

Table 1. List of correct words and misspelled words in the corpus

Number	Correct sentence	Incorrect sentence	Error Type
1	Perkenalkan nama kalian.	Perknalkan nama kalian.	Insertion
2	Buatlah kartu nama kalian	Buatlah kartu nma kalian	Insertion
3	Amati gambar ini	Amati gambar inti	Deletion
4	Ikuti petunjuk guru	Ikuti petunjuk gurut	Deletion
5	Aku melihat dengan mataku	Aku melihat denagn mataku	Transposition
6	Aku mencium dengan hidungku	Aku mencium denagn hidungku	Transposition
7	Nyanyikan lagu di bawah ini	Nyanyikan lagu di barah ini	Substitution
8	Bacalah suku kata berikut ini	Bacalah suku kata bevikut ini	Substitution

In this research, as shown above in Table 1, we utilize data categories such as insertion, deletion, transposition, and substitution. Each category signifies specific types of errors encountered in the text; the explanation of each category is as follows:

a. Deletion
An algorithm is used to eliminate characters from incorrect words. Before adding the created term to the list of ideas, the algorithm eliminates one character from the word and verifies that it is accurate. For every letter in the word, the procedure is repeated.

b. Insertion
This method fixes typos in words that have a missing character. The basic idea behind this method is to put a letter from the alphabet in the spot where the error happened and then verify that the resultant word is accurate before adding it to the list of potential words.

c. Substitution
The substitution algorithm takes a word and substitutes one letter for another in the alphabet, then tests to see whether the resultant word makes sense before adding it to the list of suggestions.

d. Transposition
The algorithm changes a single letter in a word by inserting it in every other location. Before adding the newly formed word to the recommendation list, it verifies that it is accurate each time. The procedure is carried again once more for each letter in the word [10].

$$\begin{aligned} D[i][j] &= D[i-1][j]+1 \\ D[i][j] &= D[i][j-1]+1 \\ D[i][j] &= D[i-1][j-1]+\delta(a_i,b_j) \\ D[i][j] &= D[i-2][j-2]+1 \end{aligned} \tag{1}$$

This formula shows how to calculate the edit distance between two strings, $D[i][j]$ referring to the edit distance algorithm's dynamic matrix cells, i represents the row of the matrix, and j represents the column of the matrix. The symbol $\delta(ai,bj)$ represents the delta function or a function that measures the similarity or difference between two characters [3].

2.3 Pre-processing

Before text data undergoes data processing, there are several preprocessing steps to obtain keywords [11]. Case folding and tokenization are two of the preprocessing stages [12]. Case folding is the process of transforming a word's characters into their most basic form. Changing the composition, which includes capital and lowercase letters, to a uniform form first makes it easier to correct misspelled writing.

Table 2. Examples of case-folding words

Raw Text	Case Folding
Tulisan Kamu Sulit diBaca	tulisan kamu sulit dibaca
Kita Harus Membaca Buku	kita harus membaca buku
Lapar Sekali	lapar sekali

In table 2, retain consistency in the letter forms; this typically entails changing all of the characters to lowercase [13]. Sometimes writing faults cause a composition that includes capital letters or similar characters to lack coherence [14].

Tokenization is the process of tokenizing a sentence, paragraph, or text by dividing it up into individual words or smaller sections. Especially for agglutinative languages, it is an essential step in building a highly accurate spelling error detection model [15].

Table 3. Examples of tokenization words

Tokenization	Example
Tokenizer built around words	Indonesia: [“i”, “n”, “d”, “o”, “n”, “e”, “s”, “i", “a”]
Tokenizer according to consonants	guru akan membimbing kalian: [“guru”, “akan”, “membimbing”, “kalian”]

In Table 3, the process described involves breaking down the input text into segments, or tokens. This considers the sequence of the tokenized text while also removing certain characters, such as punctuation. The results of this tokenization procedure are individual words [15].

2.4 Spelling Correction

Spelling checkers are computer-based programs designed to identify and correct word mistakes. Users can employ spelling checkers to detect errors in words. The spelling checker searches the manuscript for all kinds of errors, flags them, notifies the writer of the faults, and provides ideas for fixing them [16]. Spelling correction is one tool that may fix spelling mistakes. Errors might happen because there are too many or too few characters, or because certain characters are inappropriate [17].

Typographical errors in the text result in a string with more, fewer, or different characters than the text that corresponds to the vocabulary. Three primary steps are often involved in spelling error detection and correction: lexicon preparation, candidate creation, and string correction, depending on the intended term and context [18].

2.5 Peter Norvig Method

The Peter Norvig technique forecasts the likelihood of a relationship between the typographical word and the words in the corpus using probability. The method will look for word candidates that are close to the actual word using candidate models such as splits, deletion, transpostion, substitution, and insertion. Peter Norvig In order to find the proper words and match the words in the corpus based on likelihood, Spelling Corrector will search for a word's character combination. At the splits step, the typo word will be divided into the left and right words [19]. Peter Norvig The most comparable spelling correction c for the word w may be chosen using Spelling Corrector's word corrector feature. Since probability is just suggested, none of the word possibilities are 100% chosen. Following the equation, the formula looks for the correction c among all potential candidate corrections that maximizes the likelihood that c is the targeted adjustment with the original word w.

$$correction(w) = argmax_{c\epsilon candidates}P(c|w) \quad (2)$$

Based on the Bayes Theorem, this equals:

$$correction(w) = argmax_{c\epsilon candidates}\frac{P(c)P(w|c)}{P(w)} \quad (3)$$

We may delete it and write P(w) as follows since it is the same for every potential candidate c:

$$correction(w) = argmax_{c\in candidates}P(c)P(w|c)$$
(4)

Based on the above equation, there are four parts: (1) $argmax$, is used to select the candidate whose sum of probabilities is greatest; (2) $c \in candidates$, shows the word "c" for the candidate in the candidate set; (3) $P(c)$, the likelihood that candidate c will show up in a corpus of documents; (4) $P(w|c)$, shows the likelihood that candidate c intended text is the word w.

2.6 N-gram Method

N-gram is a technique for locating misspelled words in large text volumes. A consecutive sequence of N items, such as words, characters, syllables, or phonemes, is known as an N-gram. For instance, bigram (2-gram) is a series of two words, like "apa kabar," "dunia lain," and "makan besar." N-gram frequencies are recorded in an n-dimensional matrix, which is used to perform a check. The system marks the word as misspelled if it discovers an uncommon or nonexistent n-gram; otherwise, it does not [20].

Rather than matching every word in a text to a dictionary, in this study, the N-gram is examined. Long sentences can have their probabilities calculated by breaking them up into smaller chunks and using the conditional probability rule to get the total probability. Word-gram-level similarity comprehension is used to find and correct misspelled words [21]. The formula for determining the probability of N-grams is as follows:

$$N - gram_k = X - (N - 1) \quad (5)$$

X is the word count of a sentence, and N is the quantity of N-grams. The formula for determining the probability bigram is as follows:

$$P(W_n|W_{n-N+1}^{n-1}) = \frac{C(w_{n-N+1}^{n-1}w_n)}{C(w_{n-N+1}^{n-1})} \quad (6)$$

P is the probabilities of N-gram, w is word, n is the index, and c is the frequency of words in a bigram.

2.7 LSTM Method

In addition to solving the exploding and disappearing gradient issues that the fundamental RNN design experienced, the LSTM technique has gained favor in recent years due to its overall superior performance over the RNN architecture [22]. When detecting spelling errors, the model may assess the prior character or word components in addition to the subsequent ones because of the LSTM architecture's recurrent connections. By breaking words up into letters, an LSTM-based model and a character-based tokenizer were employed. When compared to previous seq2seq models [15]. A natural extension of feed-forward neural networks to sequences is the recurrent neural network (RNN). A conventional RNN iterates the following equation to compute a succession of outputs $(x_1, \ldots, x_T)$, given a sequence of inputs $(y_1, \ldots, y_T)$:

$$h_t = sigm(W^{hx}x_t + W^{hh}h_{t-1}) \qquad (7)$$
$$y_t = W^{yh}h_t$$

Word	Misspelled Word	N-gram	Peter Norvig	LSTM
Bunyi	Bunyti	√	√	×
Kemana	Keana	√	√	√
Ini	Init	√	√	×

Because of the vanishing gradient problem, RNNs have difficulty handling long-term dependency in the data. Recurrent neural networks with Long Short Term Memory (LSTM) are used to tackle this issue. The use of an encoder and decoder in LSTM simplifies the problem. The encoder, an LSTM working at the character level, processes the input sequence as a series of vectors. Each vector represents the meaning of characters in the sequence that has been read up to that point. On the other hand, the decoder is a character-level LSTM recurrent network with attention. It takes the final hidden state of the character-based LSTM encoder as its input [23].

3. RESULT AND DISCUSSION

The spelling correction model is developed using Python programming language version 3.10.2 with Visual Studio Code (VSC) software, utilizing the Jupyter Notebook extension. We conduct a comparison analysis test of several spelling checking methods using SPECIL data (Spell Error Corpus for Indonesian Language), which includes 4 documents for insertion, deletion, transposition, and substitution errors. The testing dataset comprises a total of 150 words. Meanwhile, the Peter Norvig and N-gram methods use 150 words as well, with the corpus reference from the 'Leipzig Corpora Collection'. Based on the experiment, Peter Norvig and N-gram calculated the probabilities of words entered into the system and identified the highest percentage for the most favorable probability of being considered as a candidate. The n-gram technique itself makes use of Bigrams, which are made up of two-word tokens, and unigrams, which are made up of single-word tokens. Norvig and n-gram cannot find all the suggestions present in the corpus, and they are unable to provide suggestions for misspelled words. The LSTM method involves a comprehensive set of steps to significantly enhance correction accuracy. Initially, text data undergoes tokenization and pre-processing to ensure readiness for subsequent stages. Following validation and fine-tuning, the model is tested using independent test data to ensure robust performance beyond the training sample. The primary advantage of LSTM lies in its capacity to provide accurate and contextual spelling corrections. Optimizing and successfully implementing the tested model can enhance spelling correction quality across various application contexts and text environments. The three methods, including LSTM, share the common limitation of requiring complete data for effective training. Specifically, LSTM's higher computational efficiency comes with the need for a more substantial and dense dataset. The choice among these methods should balance model complexity, data completeness, and computational resources for optimal performance.

Table 4. Examples of spelling correction results using three models

Table 4 explains that the study's findings demonstrate a highly notable distinction between the LSTM (Long Short Term Memory), N-gram, and Peter Norvig algorithms. Experiments conducted on spelling correction yielded quite satisfactory results. For the implementation of the LSTM method, the data utilized is referenced from SPECIL (Spell Error Corpus for Indonesian Language) with a total of 150 data points. In the dataset used, only data labeled as 'insertion' is employed. However, it's worth noting that the sentences within the .csv file are randomly selected, which has resulted in a lower accuracy for the LSTM compared to the accuracy of the other two methods.

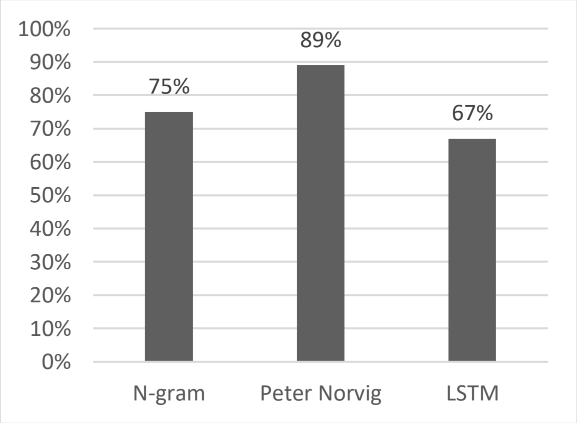

Figure 2. The result of the comparative analysis of spelling correction methods

Figure 2 illustrates the accuracy achieved in each experiment. Peter Norvig achieved 89% accuracy with a calculation speed of 35 words per second, correcting 387 words, while 5% word remained unknown. Meanwhile, N-gram achieved the second-best result, obtaining 75% accuracy with a calculation speed of 21 seconds per word, correcting 150 words, and leaving 11% word unknown. On the other hand, the LSTM algorithm achieved a model accuracy of 67% using 1000 epochs.

4. CONCLUSION

In conclusion, the assessment of Indonesian spell-checking methods reveals that Peter Norvig's approach emerges as the most effective, boasting an impressive accuracy rate of 89%. Subsequently, the N-gram method secures the second position with 75% accuracy, while LSTM lags slightly behind at 74%. These findings underscore the significance of exploring diverse techniques in the realm of spell checking, with Norvig's method standing out as a frontrunner in enhancing accuracy for the Indonesian language.

Researchers and developers can leverage these insights to make informed decisions about the advancement of spell-checking technologies tailored to the intricacies of the Indonesian language. The use of a reference dataset of 10,000 words provides a solid foundation for testing. Although Norvig performs the best, both n-gram and LSTM also contribute significantly. The SPICEL test dataset of 21,500 words demonstrates the robustness of the three methods against a larger dataset. This research provides important insights into the suitability and effectiveness of spell-checking methods in the context of the Indonesian language.

5. REFERENCE

[1] J. Li, W. Xiao, and C. Zhang, “Data security crisis in universities: identification of key factors affecting data breach incidents,” *Humanit Soc Sci Commun*, vol. 10, no. 1, Dec. 2023, doi: 10.1057/s41599-023-01757-0.

[2] A. I. Fahma, I. Cholissodin, and R. S. Perdana, “Identifikasi Kesalahan Penulisan Kata (Typographical Error) pada Dokumen Berbahasa Indonesia Menggunakan Metode N-gram dan Levenshtein Distance,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 1, pp. 53–62, Jan. 2018, doi: http://j-ptiik.ub.ac.id.

[3] J. Jatminto and I. K. D. Nuryana, “Implementasi Spelling Checker dengan Algoritma Levenshtein distance pada Ensiklopedia IT (Information Technology) berbasis website,” *Jurnal Ilmiah Inovasi Teknologi Informasi*, vol. 1, no. 1, May 2016.

[4] Mutammimah, H. Sujaini, and R. D. Nyoto, “Analisis Perbandingan Metode Spelling Corrector Peter Norvig dan Spelling Checker BK-Trees pada Kata Berbahasa Indonesia,” *Jurnal Sistem dan Teknologi Informasi*, vol. 5, no. 1, pp. 12–16, 2017.

[5] M. S. Simanjuntak, H. Sujaini, and N. Safriadi, “Spelling Corrector Bahasa Indonesia dengan Kombinasi Metode Peter Norvig dan N-Gram,” *Jurnal Edukasi dan Penelitian Informatika*, vol. 4, no. 1, p. 17, Jun. 2018, doi: 10.26418/jp.v4i1.24075.

[6] M. Hardiyanti, “Identifying The Common Type of Spelling Error by Leveraging Levenshtein Distance and N-gram,” *Scientific Journal of Informatics*, vol. 8, no. 1, 2021, doi: 10.15294/sji.v8i1.xxxxx.

[7] R. Martin, D. S. Naga, and V. C. Mawardi, “Penggunaan Spelling Correction Dengan Metode Peter Norvig dan N-gram,” *Jurnal Ilmu Komputer dan Sistem Informasi*, vol. 9, no. 1, pp. 175–180, 2021, doi: https://doi.org/10.24912/jiksi.v9i1.11591.

[8] T. Soisoonthorn, H. Unger, and M. Maliyaem, “Spelling Check: A New Cognition-Inspired Sequence Learning Memory,” *Journal of Advances in Information Technology*, vol. 14, no. 3, pp. 399–410, 2023, doi: 10.12720/jait.14.3.399-410.

[9] A. Viamianni, R. Mulyana, and F. Dewi, “Cobit 2019 Information Securtiy Focus Area Implementation For Reinsurco Digital Transformation,” *JIKO (Jurnal Informatika dan Komputer)*, vol. 6, no. 2, pp. 106–115, Aug. 2023, doi: 10.33387/jiko.v6i2.6366.

[10] Y. Chaabi and F. A. Ataa, “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, Sep. 2022, doi: 10.1016/j.jksuci.2021.07.015.

[11] R. Saptono *et al.*, “Text Classification Using Naive Bayes Updateable Algorithm In SBMPTN Test Question,” *TELEMATIKA*, vol. 13, no. 02, pp. 123–133, 2016, doi: https://doi.org/10.31315/telematika.v13i2.1728.

[12] V. C. Mawardi, N. Susanto, and D. S. Naga, “Spelling Correction For Text Documents in Bahasa Indonesia Using Finite State Automata and Levinshtein Distance Method,” *International Conference on Electrical Systems, Technology and Information*, vol. 164, p. 1047, Apr. 2017, doi: https://doi.org/10.1051/matecconf/201816401047.

[13] M. E. Purbaya, D. P. Rakhmadani, Maliana Puspa Arum, and Luthfi Zian Nasifah, “Implementation of n-gram Methodology to Analyze Sentiment Reviews for Indonesian Chips Purchases in Shopee E-Marketplace,” *Rekayasa Sistem dan Teknologi Informasi*, vol. 7, no. 3, pp. 609–617, Jun. 2023, doi: 10.29207/resti.v7i3.4726.

[14] M. A. Rosid, A. S. Fitrani, I. R. I. Astutik, N. I. Mulloh, and H. A. Gozali, “Improving Text Preprocessing for Student Complaint Document Classification Using Sastrawi,” in *IOP Conf. Ser.: Mater. Sci. Eng.*, Institute of Physics Publishing, Jul. 2020. doi: 10.1088/1757-899X/874/1/012017.

[15] B. Aytan and C. O. Sakar, “Deep learning-based Turkish spelling error detection with a multi-class false positive reduction model,” *Turk. J. Elec. Eng. & Comp. Sci.*, vol. 31, no. 3, pp. 581–595, 2023, doi: 10.55730/1300-0632.4003.

[16] N. Hamidah, N. Yusliani, and D. Rodiah, “Spelling Checker using Algorithm Damerau Levenshtein Distance and Cosine Similarity,” 2020. [Online]. Available: http://sjia.ejournal.unsri.ac.id

[17] M. O. Braddley, M. Fachrurrozi, and N. Yusliani, “Pengoreksian Ejaan Kata Berbahasa Indonesia Menggunakan Algoritma Levensthein Distance,” *Prosiding Annual Research Seminar*, vol. 3, no. 1, pp. 1–5, 2017.

[18] E. Erwina, T. Tommy, and M. Mayasari, “Indonesian Spelling Error Detection and Type Identification Using Bigram Vector and

Minimum Edit Distance Based Probabilities," *SinkrOn*, vol. 6, no. 1, pp. 183–190, Nov. 2021, doi: 10.33395/sinkron.v6i1.11224.

[19] T. M. Fahrudin *et al.*, "A Rule-based Spelling Checker for Correcting Punctuation Errors in Indonesia Text using KEBI 1.0 Checker," in *International Seminar of Research Month 2021*, Galaxy Science, May 2022, pp. 1–8. doi: 10.11594/nstp.2022.2433.

[20] R. Kumar, M. Bala, and K. Sourabh, "A study of spell checking techniques for Indian Languages," *JK Research Journal in Mathematics and Computer Sciences*, no. 1, 2018.

[21] M. V. Christanti, Rudy, and D. S. Naga, "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, Apr. 2018, doi: 10.12928/TELKOMNIKA.v16i2.6890.

[22] M. AYDOĞAN and A. KARCİ, "Kelime Gömmelerini Kullanarak Türkçe Dili İçin Sözlük Metodu ile Yazım Düzeltme," *European Journal of Science and Technology*, pp. 57–63, Apr. 2020, doi: 10.31590/ejosat.araconf8.

[23] P. E. Ltrc, M. Chinnakotla, and R. Mamidi, "Automatic Spelling Correction for Resource-Scarce Languages using Deep Learning," Jul. 2018. [Online]. Available: https://github.com/PravallikaRao/SpellChecker