U-Tapis: A Hybrid Approach to Melting Word Error Detection and Correction with Damerau-Levenshtein Distance and RoBERTa

Prudence Tendy[a)] and Marlinda Vasty Overbeek[b)]

Author Affiliations
Department of Engineering and Informatics, Universitas Multimedia Nusantara, Tangerang, Indonesia

Author Emails
[a)]Corresponding author: prudence.tendy@student.umn.ac.id
[b)]marlinda.vasty@umn.ac.id

Abstract. In the current digital era, the demand for rapid news delivery increases the risk of linguistic errors, including inaccuracies in the usage of melting words. This study introduces the U-Tapis application, a platform designed to detect and correct such errors using the Damerau-Levenshtein Distance algorithm and the RoBERTa model. The system achieved an average recommendation accuracy of 92.84%, with performance ranging from 91.30% to 95.45% across 3000 news articles. Despite its effectiveness, the system faces limitations, such as the static nature of its dataset, which does not update dynamically with new entries in the Indonesian Language Dictionary, and its tendency to flag all words with "me-" and "pe-" prefixes, regardless of context. These challenges highlight opportunities for future enhancements to improve the platform's adaptability and precision.

INTRODUCTION

Machine learning, a rapidly evolving subset of artificial intelligence (AI), has been widely applied to solve diverse problems in domains such as business, robotics, mathematics, financial analysis, and natural language processing (NLP) [1], [2]. Among these, NLP focuses on enabling machines to understand and process human language, facilitating effective and efficient user interaction [3], [4]. NLP is instrumental in building computational models that process textual and spoken information [5], [6], making it particularly valuable in fields like journalism [7]. In the fast-paced digital era, journalism demands real-time access to news, requiring journalistic teams to produce content under tight deadlines. This urgency often results in limited time for thorough editing and proofreading, leading to frequent linguistic errors [8], [9].

Common language errors in mass media include spelling mistakes, syntactic inconsistencies, and semantic inaccuracies [10]. Adhering to proper linguistic conventions is essential to maintaining the quality and fostering the growth of the Indonesian language in mass media [11], [12]. One common error found in news writing is related to "peluluhan kata" (melting words). "Peluluhan", a subset of morphophonemics, occurs when morphemes (words or syllables) meet, resulting in phoneme changes to facilitate easier pronunciation [13]. The correct linguistic rule for "peluluhan" in root words with consonant clusters (double consonants) is that the root word does not undergo softening if the prefix is "me-" or "pe-", whereas root words with single consonants do soften [14].

To address these errors, a computational tool called U-Tapis has been developed. U-Tapis is an application designed to detect and correct linguistic rule errors in Indonesian, particularly assisting journalists [15], [16]. Previous research on detecting and correcting melting words errors employed the Jaccard Similarity algorithm, yielding an F1-Score of 66.6% [17]. Another study utilized a combination of the Damerau-Levenshtein Distance algorithm and BERT but did not effectively address melting word errors in root words with double consonants [18].

This study aims to improve U-Tapis by enhancing its capacity to identify and correct melting word errors, focusing on both single and double consonant root word cases using NLP. This approach has proven effective in correcting misspelled words in documents and assisting typists by providing helpful tools [19]. The proposed approach integrates the Damerau-Levenshtein Distance algorithm and the RoBERTa model. The Damerau-Levenshtein Distance algorithm was chosen for its ability to compute the distance between target and source strings, considering adjacent symbol transpositions, making it suitable for building spell-checking systems [20]. It compares detected words against a

dictionary of correct forms and uses the computed distance as the basis for correction. Meanwhile, the RoBERTa model was selected for its superior performance compared to BERT, as RoBERTa is trained on a larger corpus and achieves accuracy in the range of 85–86%, whereas BERT achieves accuracy in the range of 83–85% [21].

By developing this enhanced machine learning model, which will be integrated into U-Tapis, this study seeks to assist journalistic teams in improving the efficiency and accuracy of their editorial processes. It specifically aims to address the challenges of melting word errors in Indonesian journalism, contributing significantly to the fields of NLP and computational linguistics through the creation of an advanced language correction system.

LITERATURE REVIEW

The development of this research is grounded in an extensive review of relevant literature, which serves as both a foundation and a point of comparison. By examining various scholarly works, key theories and algorithmic approaches have been identified and adapted to enhance the framework and methodology of this research. An overview of these literature reviews is presented below.

Melting Word

Melting Word or "Peluluhan Kata" is a subset of morphophonemics, which refers to the meeting of one morpheme with another morpheme (word or syllable), causing changes in letters or phonemes [13]. The rule for melting word for words starting with "k", "p", "s", and "t" is divided based on the consonants in the root word, namely single and double consonants. If the root word starts with a single consonant, the word with the prefixes "me-" and "pe-" will undergo alternation. Meanwhile, the correct linguistic rule for melting word with consonant clusters (double consonant letters) is that root words with the prefixes "me-" and "pe-" will not undergo alternation [14]. Further details can be seen in Table 1 below.

TABLE 1. Melting Word Rules

Consonant	K	P	S	T	Melt?
Single	Meng-	Mem-	Meny-	Men-	Yes
Double	Meng-	Mem-	Men-	Men-	No

The prefixes "me-" and "pe-" in these words will change based on the root word starting with certain phonemes. For example, with the starting phoneme "k", it will change to "meng-", with "p" it becomes "mem-", with "s" it becomes "meny-", and with "t" it becomes "men-". There is an exception for root words with consonant clusters starting with the letter "s", where the prefix "meny-" changes to "men-" for easier pronunciation. For example, the root word "sponsor", which has the consonant cluster "sp" and starts with the letter "s", will have the correct form as "mensponsori" (meaning "to sponsor"), not "meny-sponsori" [13], [22].

Natural Language Processing (NLP)

Natural Language Processing (NLP) is one area of computer science that studies how to build interactions between computers and humans that are easy to understand and efficient, with machines receiving syntax in the form of human language and processing it to then provide responses to users for performing various tasks. NLP is becoming increasingly important because it can help build models that take input in the form of speech, text, or both, and manipulate it according to algorithms within the computer [3]. The foundation of NLP lies in several disciplines, ranging from computer science and information science, linguistics, mathematics, artificial intelligence and robotics, psychology, and so on. NLP can be implemented in various fields, such as machine translation, natural language text processing and summarization, user interfaces, speech recognition, and so on [23].

Text Preprocessing

Text preprocessing is an important step before building an NLP model, which involves processing text data to reduce the vocabulary size by removing unnecessary parts or noise. By performing text preprocessing, the text data size is reduced, and the machine learning algorithms that will be used become more effective and efficient because the data is clean [24]. There are several main steps in text preprocessing as explained in the following points [25].

- Case Folding: is the step of converting uppercase letters in the text to lowercase.
- Tokenizing: is the step of breaking the text into sentences or words, depending on the analysis needs, which are then referred to as "tokens".
- Stop-Word Removal: is the step of removing words that are not important or do not carry much meaning, such as "and", "is", and so on, which do not contribute much to the overall understanding of the text content. This step helps reduce the text dimensions, thus improving the efficiency of the model.
- Stemming: is the step of mapping and breaking down words into their root forms, such as removing affixes to make the text simpler and reduce the variation of words with the same basic meaning.

Damerau-Levenshtein Distance

Damerau-Levenshtein Distance is an algorithm that can be used to calculate the distance between a target string and a source string, resulting in an edit distance between the two strings [26]. Damerau-Levenshtein Distance is also defined as the minimum number of substitutions, insertion, deletion, and transposition of adjacent characters required to transform one string into another. This algorithm can be implemented to detect spelling errors, data mining, clustering, virus detection in software, and so on [27]. The algorithm can be defined using the following formula [28].

$$f_{a,b}(i,j) = \min \begin{cases} 0 \; if \; i = j = 0, \\ f_{a,b}(i-1,j) + 1 \; if \; i > 0, \\ f_{a,b}(i,j-1) + 1 \; if \; j > 0, \\ f_{a,b}(i-1,j-1) + 1 \; if \; i,j > 0 \; and \; a_i \; and \; b_j \; are \; not \; equal \\ f_{a,b}(i-2,j-2) + 1 \; if \; i,j > 1 \; and \; a_{i-1} = b_j \; and \; a_i = b_{j-1} \end{cases} \quad (1)$$

The minimum distance between two strings, namely *a* and *b*, can be defined using Formula 1, with *i* and *j* representing the length of the prefix of string *a* and *b* based on editing operations such as deletion, insertion, substitution, and transposition. Formula 1 shows the minimum distance required to change the prefix of length *i* in string *a* to the prefix of length *j* in string *b*, performed recursively. When both strings are empty (*i* and *j* are zero), the distance is zero. In the deletion step, if $i > 0$, the distance is determined by deleting the last character of the prefix, and in the insertion step, if $j > 0$, the distance is determined by adding a character to string *b*. In the substitution operation, if $i, j > 0$ and the last character of both prefixes are different, the distance is determined by replacing the character in *a* with the character in *b* or vice versa, increasing the distance by one. If a_i and b_j are the same, the resulting distance is the same as the distance from the previous prefix. In the transposition operation, if $i, j > 1$ and character a_{i-1} is equal to b_j and a_i is equal to b_{j-1}, the minimum distance is obtained by swapping the two characters.

Robustly Optimized BERT Pretraining Approach (RoBERTa)

Robustly Optimized BERT Pretraining Approach (RoBERTa) is an extension of the pretrained Bidirectional Encoder Representations from Transformer (BERT) model, designed for sequence-to-sequence tasks with long-range dependencies. RoBERTa employs the "self-attention" mechanism to capture dependencies between input tokens, allowing the model to evaluate token significance in the context of the entire sequence. The self-attention and feed-forward layers work together to generate and refine contextual representations, ultimately producing the final output [29]. As illustrated in Fig. 1, the architecture consists of several Transformer blocks stacked together, with each block applying self-attention and feed-forward operations to the input. The input tokens are first processed using the RoBERTaFastTokenizer which utilizes byte-level Byte Pair Encoding for tokenization, resulting in a smaller vocabulary and lower computational resources compared to BERT, which uses character-level Byte Pair Encoding [30]. The processed tokens are then passed through the Transformer blocks to produce contextualized embeddings, which are further refined through additional layers and combined with metadata features to generate the final output.

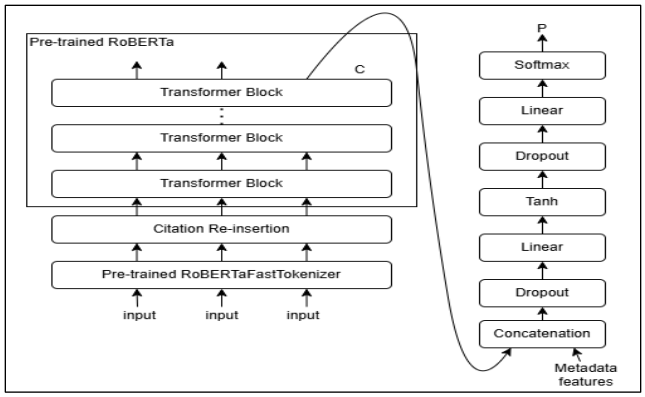


FIGURE 1. RoBERTa Model Architecture

METHOD

Figure 2 illustrates the research flow, starting with data collection, where relevant textual data is gathered. The next step involves text preprocessing to clean and standardize the data before feeding it into the model. Following this, the model is developed to detect and correct melting word errors. Finally, the model undergoes testing and evaluation to assess its accuracy and overall performance.

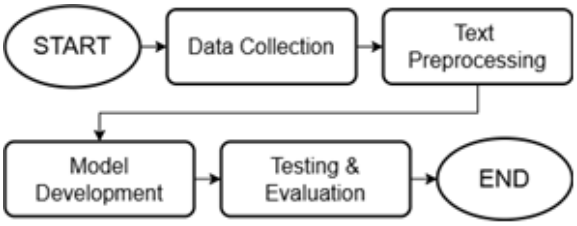


FIGURE 2. Research Method Flowchart

Data Collection

The data collection stage is carried out by gathering all correct forms of melting words through crawling the Kamus Besar Bahasa Indonesia (The Indonesian Language Dictionary), both words with single and double consonants. The collected words must start with the prefix *me-* and *pe-*. The data will then be stored in the *.xlsx* format for use in the model. Additionally, news articles are also collected as a testing dataset for the system. This dataset, in *.pdf* and *.docx* formats, was collected by students from the Journalism department and stored in a Google Drive folder accessible to the U-Tapis research team. In this study, 3000 news articles are used as the testing dataset to evaluate the system's detection and correction capabilities.

Text Preprocessing

Preprocessing is applied to the input data from news articles to ensure consistency and cleanliness. As shown in Fig. 3, this stage involves several steps, including removing control characters, stripping HTML tags, converting Unicode characters to ASCII, and applying case folding. Additionally, numerals, extra spaces, and punctuation are removed to enhance text uniformity. Finally, the processed text is tokenized, preparing it for further analysis.

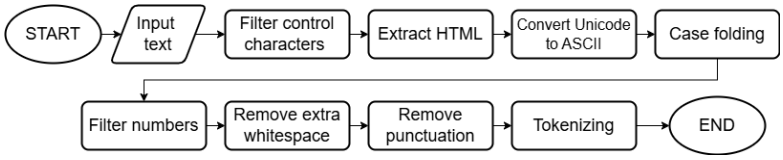


FIGURE 3. Text Preprocessing Flowchart

Model Development

In the model development stage, the Damerau-Levenshtein Distance (DLD) algorithm detects misspellings of melting words in the preprocessed news text, while the RoBERTa model predicts synonyms for those incorrect words. Figure 4 illustrates the overall model development flow, starting with the input of preprocessed text. The system first checks whether the root word contains a double consonant, directing it to either single or double consonant detection accordingly. The results are then combined and processed through the RoBERTa model to generate final predictions before producing the output.

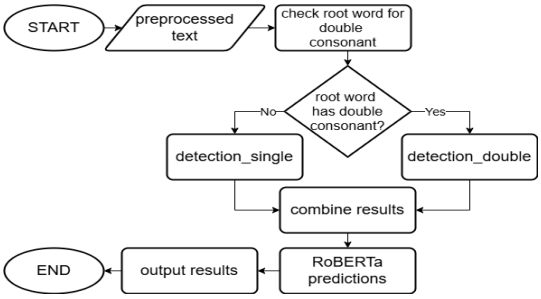


FIGURE 4. Overall Model Development Flowchart

In both the detect_single and detect_double functions, the Damerau-Levenshtein Distance (DLD) between the words in the news text and the correct word set in the Excel dataset is first compared, as shown in Fig. 5. If the DLD is 0, the word will be added to the correct word list. However, if the DLD is not 0, the word will be added to the incorrect word list. The difference between the detect_single and detect_double functions lies in the distance comparison used as the evaluation criterion. In the detect_single function, a DLD greater than 0 and less than or equal to 1 is used as the rule to add words from the dataset to the recommendation list for correction. In the detect_double function, the rule is based on a DLD greater than 0 and less than or equal to 2. The resulting words from both functions are then returned as the correct word list, incorrect word list, and recommendation word list.

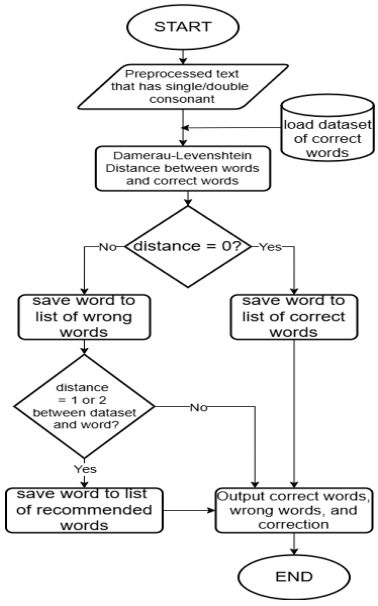


FIGURE 5. Flowchart of Word Detection Using DLD

To build the synonym prediction mechanism, the pre-trained transformer model “cahya/roberta-base-indonesian-1.5G” from HuggingFace is utilized [31]. Incorrect words in the list of wrong words are masked using the “<mask>” format before being input into the model. The preprocessed text and the masked words serve as the basis for the model to predict synonyms as word recommendations. These prediction results are appended to the detailed output, so the

final output of the model comprises the detected incorrect words, their corrections, and the corresponding word recommendations. This process is illustrated in Fig. 6.

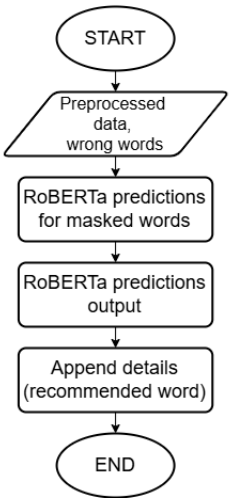


FIGURE 6. Flowchart of Word Prediction Using RoBERTa

Testing and Evaluation

This stage is conducted using 3000 news articles with various scenarios, including 100, 150, 200, 300, 350, 400, 450, and 500 news articles from TribunNews. Once the testing results are obtained, the confusion matrix will be determined, which can then be used to calculate accuracy, precision, f1-score, and recall. Additionally, the recommendation accuracy of the system is calculated to evaluate its effectiveness in suggesting corrections. The evaluation results will help identify potential weaknesses in the model and provide insights for further improvements.

RESULT AND ANALYSIS

This section presents the findings of the study, highlighting the performance evaluation and practical implementation of the proposed system. The "Testing and Evaluation" subsection discusses the accuracy and effectiveness of the hybrid approach, analyzing key performance metrics such as precision, recall, F1-score, and recommendation accuracy based on extensive testing with news articles. The "Website Interface Implementation" subsection showcases the integration of the error detection and correction model into a user-friendly web platform, emphasizing usability and real-time processing capabilities. Lastly, the "Limitations and Ethical Implications" subsection explores the challenges faced, including potential computational constraints and ethical concerns related to automated text correction, such as biases in language models and responsible use in journalism. These analyses provide insights into the system's strengths, areas for improvement, and its broader implications in NLP and computational linguistics.

Testing and Evaluation

In this section, the system was tested using 3000 articles with a testing scheme involving 100, 150, 200, 300, 350, 400, 450, 500, and 550 news articles. The tests were conducted to evaluate the detection and correction results of the system, which employs Damerau-Levenshtein Distance and RoBERTa. Additionally, the recommendation accuracy was calculated by dividing the total number of correct recommendations by the total number of incorrect melting words.

$$Recommendation\ Accuracy = \frac{total\ number\ of\ correct\ recommendations}{total\ number\ of\ incorrect\ melting\ words} \quad (2)$$

- 100 News Test Case: Table 2 presents the confusion matrix for 100 tested news articles. Based on the testing results, it was found that there were 579 words classified as True Positive (TP), 22 words as True Negative (TN), 24 words as False Negative (FN), 3 words as False Positive (FP), and the accuracy of recommendation is 95.45%.

TABLE 2. 100 Test Case Confusion Matrix

	Actual Value	
Predicted Value	**0**	**1**
0	579	3
1	24	22

- 150 News Test Case: Table 3 presents the confusion matrix for 150 tested news articles. Based on the testing results, it was found that there were 777 words classified as True Positive (TP), 23 words as True Negative (TN), 32 words as False Negative (FN), 2 words as False Positive (FP), and the accuracy of recommendation is 91.30%.

TABLE 3. 150 Test Case Confusion Matrix

	Actual Value	
Predicted Value	**0**	**1**
0	777	2
1	32	23

- 200 News Test Case: Table 4 presents the confusion matrix for 200 tested news articles. Based on the testing results, it was found that there were 1061 words classified as True Positive (TP), 42 words as True Negative (TN), 36 words as False Negative (FN), 2 words as False Positive (FP), and the accuracy of recommendation is 92.86%.

TABLE 4. 200 Test Case Confusion Matrix

	Actual Value	
Predicted Value	**0**	**1**
0	1061	2
1	36	42

- 300 News Test Case: Table 5 presents the confusion matrix for 300 tested news articles. Based on the testing results, it was found that there were 1642 words classified as True Positive (TP), 75 words as True Negative (TN), 75 words as False Negative (FN), 6 words as False Positive (FP), and the accuracy of recommendation is 93.33%.

TABLE 5. 300 Test Case Confusion Matrix

	Actual Value	
Predicted Value	**0**	**1**
0	1642	6
1	75	75

- 350 News Test Case: Table 6 presents the confusion matrix for 350 tested news articles. Based on the testing results, it was found that there were 1835 words classified as True Positive (TP), 54 words as True Negative (TN), 54 words as False Negative (FN), 3 words as False Positive (FP), and the accuracy of recommendation is 92.59%.

TABLE 6. 350 Test Case Confusion Matrix

	Actual Value	
Predicted Value	**0**	**1**
0	1835	3
1	54	54

- 400 News Test Case: Table 7 presents the confusion matrix for 400 tested news articles. Based on the testing results, it was found that there were 2223 words classified as True Positive (TP), 108 words as True Negative (TN), 114 words as False Negative (FN), 10 words as False Positive (FP), and the accuracy of recommendation is 93.52%.

TABLE 7. 400 Test Case Confusion Matrix

	Actual Value	
Predicted Value	**0**	**1**
0	2223	10
1	114	108

- 450 News Test Case: Table 8 presents the confusion matrix for 450 tested news articles. Based on the testing results, it was found that there were 2656 words classified as True Positive (TP), 108 words as True Negative (TN), 84 words as False Negative (FN), 6 words as False Positive (FP), and the accuracy of recommendation is 92.59%.

TABLE 8. 450 Test Case Confusion Matrix

	Actual Value	
Predicted Value	**0**	**1**
0	2656	6
1	84	108

- 500 News Test Case: Table 9 presents the confusion matrix for 500 tested news articles. Based on the testing results, it was found that there were 2774 words classified as True Positive (TP), 111 words as True Negative (TN), 118 words as False Negative (FN), 8 words as False Positive (FP), and the accuracy of recommendation is 92.79%.

TABLE 9. 500 Test Case Confusion Matrix

	Actual Value	
Predicted Value	**0**	**1**
0	2774	8
1	118	111

- 550 News Test Case: Table 10 presents the confusion matrix for 550 tested news articles. Based on the testing results, it was found that there were 2935 words classified as True Positive (TP), 124 words as True Negative (TN), 120 words as False Negative (FN), 8 words as False Positive (FP), and the accuracy of recommendation is 91.13%.

TABLE 10. 550 Test Case Confusion Matrix

	Actual Value	
Predicted Value	**0**	**1**
0	2935	8
1	120	124

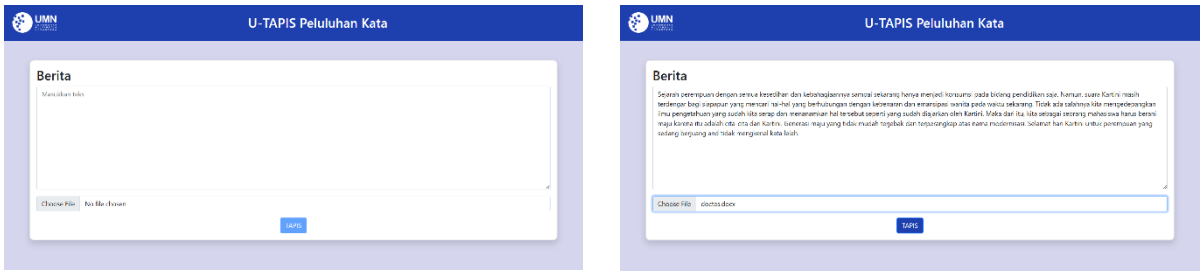
The performance metrics from the various test cases can be seen in Table 11 below. The proposed system achieved an average recommendation accuracy of 92.84% across multiple test cases in identifying and correcting melting word errors, outperforming the combination of BERT and Damerau-Levenshtein Distance as the baseline, which achieved 89% recommendation accuracy. These results were obtained from a dataset of 3000 Indonesian news articles, focusing on both single and double consonant root word cases. For instance, the system corrected "mensejahterakan" to "menyejahterakan" (meaning "to prosper") and "menkritik" to "mengkritik" (meaning "to criticize"), demonstrating its effectiveness in identifying and correcting melting word errors.

TABLE 11. Performance Metrics for Test Cases

Test Case	Accuracy	Precision	Recall	F1-Score
100 News	95.7%	99.48%	96.02%	97.72%
150 News	95.92%	99.74%	96.04%	97.85%
200 News	96.67%	99.81%	96.72%	98.24%
300 News	95.49%	99.64%	95.63%	97.59%
350 News	97.07%	99.84%	97.14%	98.47%
400 News	94.95%	99.55%	95.12%	97.29%
450 News	96.85%	99.77%	96.93%	98.33%
500 News	95.82%	99.71%	95.92%	97.78%
550 News	95.98%	99.73%	96.07%	97.87%

Website Interface Implementation

Figure 7 displays the initial interface of the program and how it appears after a user uploads a *.docx* file. The page includes a text area, a file input, and a button labeled "Tapis." Users can enter news text either by typing directly into the text area or by uploading a .docx file, which automatically populates the text area with its content. Once the text is entered, they can press the "Tapis" button to check for melting words.

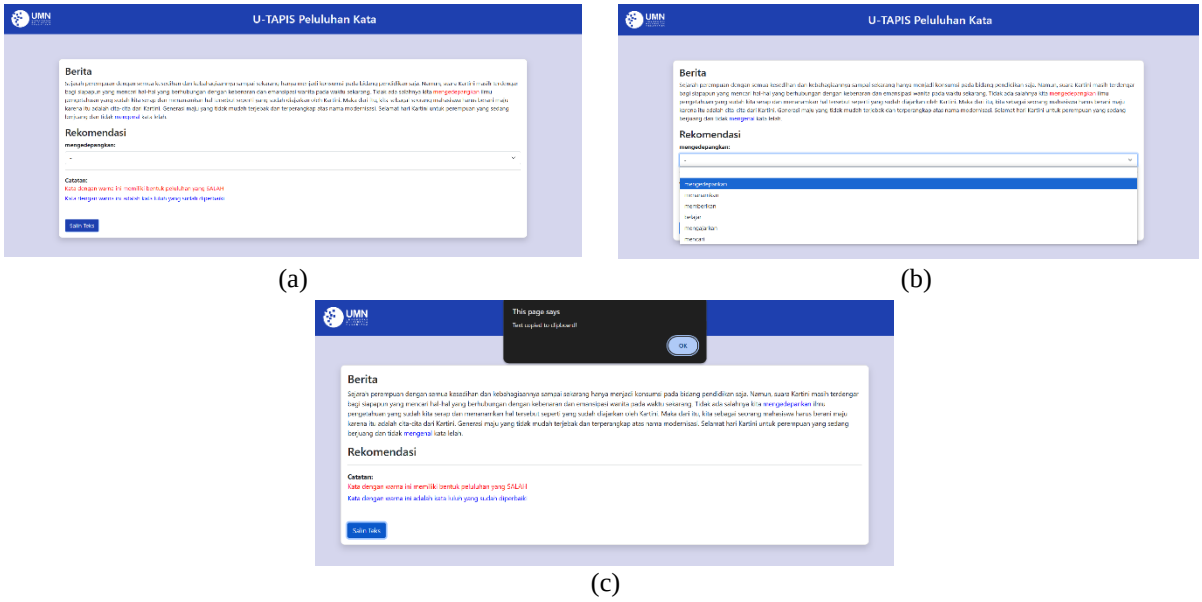


(a) (b)

FIGURE 7. Main Page

After the user presses the "Tapis" button, the system will process the news and display the news text along with the detection results, as shown in Fig. 8. The red color indicates that the detected word has an incorrect melting word form according to the Kamus Besar Bahasa Indonesia (The Indonesian Language Dictionary) and the blue color indicates that the previously incorrect word has been replaced with the system's corrected version. If a red-colored word, which

indicates an incorrect melting word form, is clicked, the user can press the dropdown menu to view a list of suggested corrections and recommendations for the word. This feature allows users to evaluate the provided suggestions, reducing the risk of blindly accepting corrections without ensuring their accuracy and relevance. The user can select one of the correction and recommendation options provided by the system. After the user selects one of the correction options, the incorrect word will automatically be replaced with the selected option. The corrected text can be copied by clicking the "Salin Teks" (Copy Text) button. Once the text is successfully copied, the system displays an alert confirming the completion of the process.



(a) (b)

(c)

FIGURE 8. (a) Detection and Correction Result, (b) Correction and Recommendation Options, (c) Copy Text

Limitations and Ethical Implications

While the study made significant progress, several limitations were identified. The dataset containing the correct forms of melting words has not been dynamically updated, preventing the system from automatically incorporating new entries from the Kamus Besar Bahasa Indonesia (The Indonesian Language Dictionary). As a result, newly introduced words remain unrecognized by the system. Additionally, the system currently detects all words with the prefixes "me-" and "pe-", regardless of their relevance to melting words, as seen with words like "memang" (meaning "certainly" or "indeed") and "perempuan" (meaning "female"). Addressing these limitations offers opportunities for future research. Dynamic integration of the dictionary and improved prefix filtering mechanisms could enhance the system's accuracy and adaptability, contributing to the development of more robust tools for error detection and correction in Indonesian text processing.

Furthermore, the automation of language correction introduces potential ethical concerns. Users may blindly accept corrections without verifying their appropriateness, potentially introducing errors in specialized contexts. To mitigate these risks, the system includes manual oversight features, allowing users to select from multiple correction suggestions. It does not enforce corrections, enabling users to retain the original text if desired.

CONCLUSION

This research makes a valuable contribution to the fields of NLP and computational linguistics by addressing the challenges of melting word errors in journalism and providing practical advancements in automated language correction. The study successfully implemented the Damerau-Levenshtein Distance algorithm and RoBERTa for detecting and correcting melting word errors involving both single and double consonants, achieving satisfactory performance. Recommendation accuracy rates of 95.45%, 91.30%, 92.86%, 93.33%, 92.59%, 93.52%, 92.59%, 92.79%, and 91.13% were achieved across test cases involving 100, 150, 200, 300, 350, 400, 450, 500, and 550 news articles, respectively.The system achieved an average recommendation accuracy of 92.84%, outperforming the previous

approach (a combination of Damerau-Levenshtein Distance and BERT), which achieved an average recommendation accuracy of 89%.

While the study demonstrates promising results, certain limitations were identified. One limitation is the lack of dynamic updates to the dataset, which prevents the system from recognizing newly introduced words in the Kamus Besar Bahasa Indonesia (The Indonesian Language Dictionary). Additionally, the system tends to detect words with the prefixes “me-” and “pe-” regardless of their relevance to melting words. Addressing these issues, such as through dynamic dictionary integration and more refined prefix filtering, presents opportunities for further enhancing system accuracy and adaptability.

Moreover, the automation of language correction raises ethical considerations. Users may inadvertently accept corrections without evaluating their appropriateness, which could lead to errors in specialized contexts. To mitigate this risk, the system includes manual oversight features, giving users the option to choose from multiple suggestions and retain the original text if desired. Future research should focus on refining these aspects, contributing to the development of more robust and ethically responsible tools for error detection and correction in Indonesian text processing.

ACKNOWLEDGMENTS

The authors would like to express their gratitude to Universitas Multimedia Nusantara for their support of this research, as well as to TribunNews for partnering with us in this research.

REFERENCES

1. B. University, “4 alasan machine learning menjadi tren di tengah pandemi,” https://dcs.binus.ac.id/2022/05/14/4-alasan-machine-learning-menjadi-tren-di-tengah-pandemi/, May 2022, doctor of Computer Science.
2. A. Roihan, P. Sunarya, and A. Rafika, “Pemanfaatan machine learning dalam berbagai bidang: Review paper.ijcit (indonesian journal on computer and information technology), 5 (1), 75–82,” 2020.
3. S. C. Fanni, M. Febi, G. Aghakhanyan, and E. Neri, “Natural language processing,” in Introduction to Artificial Intelligence. Springer, 2023, pp. 87–99.
4. V. G. A. Siswanto and M. V. Overbeek, “Development of “kata terikat” detection and writing errors correction using rabin-karp and random forest algorithm,” AIP Conference Proceedings, vol. 3220, no. 1, p. 040004, 10 2024. [Online]. Available: https://doi.org/10.1063/5.0235496.
5. D. Khurana, A. Koli, K. Khatter, and S. Singh, “Natural language processing: state of the art, current trends and challenges,” Multimedia tools and applications, vol. 82, no. 3, pp. 3713–3744, 2023.
6. S. Pais, J. Cordeiro, and M. L. Jamil, “Nlp-based platform as a service: a brief review,” Journal of Big Data, vol. 9, no. 1, p. 54, 2022.
7. A. K. B. Saputra and M. V. Overbeek, “Harnessing long short-term memory algorithm for enhanced di-di word error detection and correction,” AIP Conference Proceedings, vol. 3220, no. 1, p. 040002, 10 2024. [Online]. Available: https://doi.org/10.1063/5.0235487
8. P. Hidayat, I. N. Sudiana, and A. A. S. Tantri, “Analisis kesalahan berbahasa pada penulisan berita detik finance dan detik news,” Jurnal Pendidikan Bahasa Dan Sastra Indonesia Undiksha, vol. 11, no. 3, pp. 318–326, 2021.
9. N. R. Dwitya and M. V. Overbeek, “Development of detection and correction of errors in spelling and compound words using long short-term memory,” AIP Conference Proceedings, vol. 3220, no. 1, p. 040005, 10 2024. [Online]. Available: https://doi.org/10.1063/5.0235850
10. A. M. D. Damayanti, F. Inayatillah et al., “Kesalahan frasa pada berita online surya. co. id 2023: Phrase mistakes in surya. co. id online news 2023,” Jurnal Bastrindo, vol. 4, no. 1, pp. 58–71, 2023.
11. R. Y. Nababan, “Wacana komunikasi ekspositif dalam youtube stefanie humena edisi “bahasa indonesia yang baik dan benar”,” Sitasi Ilmiah, vol. 1, no. 1, pp. 60–68, 2022.
12. S. Agan and E. Puspitoningrum, “Kosa kata bahasa asing dalam bahasa indonesia ragam jurnalistik,” Wacana: Jurnal Bahasa, Seni, dan Pengajaran, vol. 5, no. 2, pp. 63–74, 2022.
13. E. Hutapea, “Peluluhan kata dasar berawalan kpst halaman all,” KOMPAS.com, Jan. 2021. [Online]. Available: https://edukasi.kompas.com/read/2021/01/08/144019571/peluluhan-kata-dasar-berawalan-kpst
14. E. Kusumah, “Morfofonemik dalam proses afiksasi prefiks {men-} dan {pen-} yang menghadapi bentuk dasar berkluster,” Jurnal Membaca Bahasa dan Sastra Indonesia, vol. 8, no. 2, 2023.

15. N. Mediyawati and S. Bintang, “Platform kecerdasan buatan sebagai media inovatif untuk meningkatkan keterampilan berkomunikasi: U-tapis,” in PROSIDING SEMINAR NASIONAL PROGRAM PASCASARJANA UNIVERSITAS PGRI PALEMBANG, 2021.
16. R. Sutomo and N. Mediyawati, “Utapis indonesian word error detection application: Design and development,” Indonesian Journal of Computer Science, vol. 13, no. 1, 2024.
17. N. E. Taslim, “Deteksi kesalahan eja kata luluh pada berita dengan algoritma jaccard similarity (studi kasus: Tribunnews),” 2023.
18. K. Gleneagles, M. V. Overbeek, N. Mediyawati, R. Sutomo, and S. Bintang, “U-tapis melting words: an artificial intelligence application for detecting melt word erros in indonesia online news,” International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), 2024 (Ongoing).
19. A. Revathi, M. Vimaladevi, and N. Arivazhagan, “Spelling correction using encoder-decoder and damerau-levenshtein distance,” in 2023 IEEE 5th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA), 2023, pp. 469–472.
20. Y. Chaabi and F. Ataa Allah, “Amazigh spell checker using damerau-levenshtein algorithm and n-gram,” Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 8, Part B, pp. 6116–6124, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1319157821001828.
21. P. Rajapaksha, R. Farahbakhsh, and N. Crespi, “Bert, xlnet or roberta: The best transfer learning model to detect clickbaits,” IEEE Access, vol. 9, pp. 154 704–154 716, 2021.
22. H. Elena, “Aturan kpst dan pengecualiannya, penulis harus paham,” https://jogja.idntimes.com/life/education/helmi-elena/aturan-kpst-dan-pengecualiannya-c1c2, 2023, iDN Times.
23. K. Chowdhary and K. Chowdhary, “Natural language processing,” Fundamentals of artificial intelligence, pp. 603–649, 2020.
24. A. Kurniasih and L. P. Manik, “On the role of text preprocessing in bert embedding-based dnns for classifying informal texts,” Neuron, vol. 1024, no. 512, p. 256, 2022.
25. 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 Conference Series: Materials Science and Engineering, vol. 874, no. 1. IOP Publishing, 2020, p. 012017.
26. B. V. Indriyono, “Kombinasi damerau levenshtein dan jaro-winkler distance untuk koreksi kata bahasa inggris,” Jurnal Teknik Informatika dan Sistem Informasi, vol. 6, no. 2, 2020.
27. C. Zhao and S. Sahni, “Linear space string correction algorithm using the damerau-levenshtein distance,” BMC bioinformatics, vol. 21, pp. 1–21, 2020.
28. GeeksforGeeks.org, “Damerau-levenshtein distance,” https://www.geeksforgeeks.org/damerau-levenshtein-distance/, Mar 2023, geeksforGeeks.
29. Z. Huang, C. Low, M. Teng, H. Zhang, D. Ho, M. Krass, and M. Grabmair, “Context-aware legal citation recommendation using deep learning,” 06 2021.
30. K. L. Tan, C. P. Lee, and K. M. Lim, “Roberta-gru: A hybrid deep learning model for enhanced sentiment analysis,” Applied Sciences, vol. 13, no. 6, p. 3915, 2023.
31. Cahya. *cahya/roberta-base-indonesian-1.5G*, Hugging Face, 2021. [Online]. Available: https://huggingface.co/cahya/roberta-base-indonesian-1.5G.