# A Deep Learning Approach for Word Segmentation in Javanese Letter Manuscript Transliteration

Muhammad Nevin
*Institut Teknologi Sepuluh Nopember*
*Department of Informatics*
Surabaya, Indonesia
6025231085@student.its,ac.id

I Kadek Agus Ariesta Putra
*Institut Teknologi Sepuluh Nopember*
*Department of Informatics*
Surabaya, Indonesia
6025231078@student.its.ac.id

Dwinanda Bagoes Ansori
*Institut Teknologi Sepuluh Nopember*
*Department of Informatics*
Surabaya, Indonesia
6023231086@student.its.ac.id

Riyanarto Sarno
*Institut Teknologi Sepuluh Nopember*
*Department of Informatics*
Surabaya, Indonesia
riyanarto@if.its.ac.id

Agus Haryono
*Institut Teknologi Sepuluh Nopember*
*Department of Informatics*
Surabaya, Indonesia
7025231020@student.its.ac.id

2024 2nd International Conference on Technology Innovation and Its Applications (ICTIIA) | 979-8-3503-5161-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICTIIA61827.2024.10761167

***Abstract*—Traditionally written using the Javanese Letter or "Aksara Jawa" script, the Javanese language encompasses a rich corpus of manuscripts that record diverse subjects such as history, culture, and traditional practices. With over 121,668 Javanese manuscript titles identified, only a fraction has been transliterated into alphabetical writing, underscoring the urgent need for efficient language preservation methods. This study evaluates deep learning models for word segmentation in Javanese manuscript transliteration. Results from experiments conducted on an unseen dataset reveal that the Bidirectional Long Short-Term Memory (BiLSTM) model outperforms other architectures consistently across all metrics. With an accuracy of 98.62% and superior scores in f1-score, precision, and recall, the BiLSTM model demonstrates robustness in capturing the intricate linguistic patterns and textual structures inherent in Javanese manuscripts.**

***Index Terms*—BiLSTM, deep learning, javanese manuscript, natural language processing, word segmentation**

## I. Introduction

The Javanese Letter, or "Aksara Jawa," is used to write the Javanese language within the Javanese tribe in Indonesia. Over time, numerous manuscripts have been written in Javanese script, covering a wide range of subjects such as history, politics, government, law, customs, folklore, traditional medicine, culture, worship rituals, manners, ancestral genealogy, poetry, spells, and traditional technologies. National records indicate at least 121,668 titles of Javanese script manuscript documents, with 2,495 of them transliterated into alphabetical writing. Therefore, it is crucial to preserve and advance the language [1], [2].

Overcoming the challenges in Javanese script recognition, a complex task due to the unique characteristics of the script, requires an innovative artificial intelligence approach [3]–[6]. One of the subfields of AI, object detection, is crucial in the image recognition of special objects like Javanese script letters. Recent research has made significant strides, developing algorithms that can detect letters other than those of the alphabet with a higher level of accuracy than previous methods [7]–[9]. These transliteration are illustrated in Fig. 1. However, the main challenge persists the detected letters often appear as a set without separators, hindering the understanding of the sentence as a whole. This underscores the need for further development in the word segmentation and separation algorithm to ensure the correct interpretation of the text's overall meaning.

To tackle these challenges, the specialized subfield of artificial intelligence, Natural Language Processing (NLP), assumes a pivotal role. NLP harnesses computational techniques to interpret human language, encompassing tasks such as language translation, word segmentation, and more [10]–[14]. Recent research has introduced numerous word segmentation methods, both with and without dictionaries. A dictionary-based word segmentation method has been proposed in the context of Javanese manuscript manuscript transliteration. This method matches syllables against a dictionary to verify word correctness, suggesting alternative words if a syllable match is poor [15]–[19]. However, this approach relies heavily on the dictionary's completeness and struggles with inflexible text handling.

On the other hand, several studies have explored using a deep learning approach for word segmentation. Some notable examples of deep learning approaches for word segmentation include DeepCut, ThaiLMCut, and BiLSTM for Thai words, CNN-BiLSTM-CRF for Tibetan words, LSTM for Japanese words, and ANN for Dzongkha Word [20]–[25]. These research achievements have shown better performance than the previous state-of-the-art methods. Using a deep learning approach, word segmentation can be performed more effectively and efficiently, especially when processing large and complex data. The advantages of using a deep learning approach in word segmentation include learning complex patterns, making more accurate predictions, and processing huge and complex

979-8-3503-5161-3/24/$31.00 ©2024 IEEE

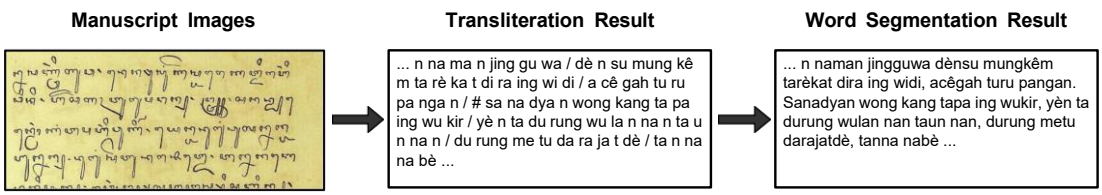


Fig. 1. The transliteration process of the Javanese letter manuscript image along with word segmentation

data. However, this approach has never been applied to Javanese script segmentation, making it a novel and potentially impactful addition to the field.

This paper proposes word segmentation for Javanese letters using a deep learning approach resulting from Javanese manuscript transliteration. The main contribution of this paper is to propose a deep learning model for word segmentation from Javanese characters. These models incorporate multiple attentions to estimate the characters' relationships in character-based word segmentation. The deep learning models used include BiLSTM, SimpleRNN, GRU, and CNN.

## II. Literature Review

Word segmentation in natural language processing helps identify word boundaries within text. There are two primary approaches: dictionary-based and machine learning-based [26]. Dictionary-based methods utilize predefined dictionaries and rules to detect word boundaries, suitable for languages with clear morphological structures but less effective for those with complex morphology or many homophones [27]. Conversely, machine learning methods, especially deep learning, learn from large datasets to identify word patterns and boundaries, proving more effective for languages with intricate structures [18], [23], [28], [29].

For languages like Javanese, which feature complex morphology and numerous homophones, a hybrid approach combining both methods is beneficial [8], [9]. This involves using dictionary-based strategies to pinpoint familiar word patterns and machine learning techniques to adapt to and learn the language's unique traits. This integrated approach ensures a more robust segmentation by leveraging the strengths of both methodologies. In addition, rule-based and deep learning techniques also play significant roles [11], [21], [30], [31]. Rule-based methods apply specific linguistic rules derived from the grammar and syntax of the language. These rules often involve recognizing phonological and morphological patterns, such as prefixes and suffixes, which can indicate boundaries in languages like Javanese. This method ensures that common structural patterns within a language are adhered to, but it may lack flexibility in handling exceptions or novel usage not covered by the rules.

In the study by Widiarti, a rule-based approach is employed for word segmentation, starting with the development of a word dictionary [7]. This method relies heavily on establishing databases that detail Javanese script characteristics and a probability table that facilitates the accurate grouping and identification of words. When tested on a specific Javanese manuscript (catalog number SB.141), this model achieved a transliteration success rate ranging between 69.20% and 87.29% with a 95% confidence level. The precision of syllable grouping is highly dependent on the thoroughness and accuracy of the dictionary. Should the dictionary be incomplete or contain inaccuracies, it may result in incorrectly grouping syllables into words. Furthermore, applying probabilistic models to predict syllable sequences might lead to errors if the probability tables lack detail or fail to accurately reflect the Javanese script's complexities.

Deep learning methods, particularly those employing neural networks, are trained on extensive datasets that allow the models to learn complex and subtle patterns in text [9]. These models operate by understanding context and varying text patterns, making them highly effective for languages with non-standardized word boundaries or extensive slang. For Javanese, deep learning methods can adapt to and accurately predict word boundaries despite linguistic complexity and variability.

A prevalent methodology in word segmentation leverages the capacities of Bidirectional Long Short-Term Memory (BiLSTM) networks, enhancing the processing of sequential data by capturing contextual information from both preceding and subsequent text elements. This technique significantly enriches the contextual understanding necessary for accurate segmentation. Current research illustrates the effectiveness of BiLSTM in textual data and multi-modal settings where audio cues are integrated, demonstrating versatility across various data forms. Additionally, integrating multiple classifiers for segmentation further optimizes performance by combining different machine-learning strategies to refine accuracy and adaptability, as explored in the work by scholars at Brandeis University [20]–[22].

In light of the existing works, several deep learning models are experimented with for segmenting Javanese letters to separate the characters from the Javanese manuscript transliteration result.". The deep learning models used include BiLSTM, SimpleRNN, GRU, and CNN. The objective is to achieve the best results in terms of accuracy with a deep learning approach.

## III. Method

In this paper, a method was proposed for segmenting Javanese script characters into meaningful sentences based on the transliteration of Javanese script manuscripts. The stages include preparing data input, creating a character dictionary and preparing datasets, encoding data, developing a BiLSTM

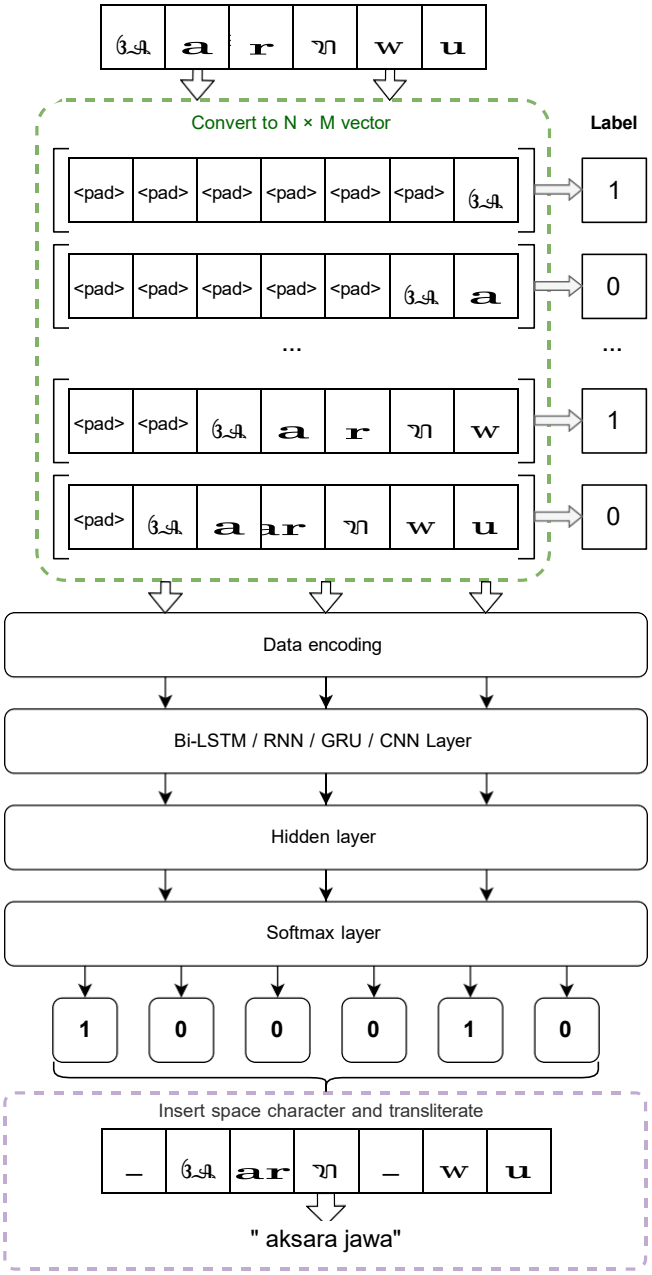


Fig. 2. Word segmentation model

model, adding hidden layers, and integrating a softmax layer for word segmentation classification.

### A. Data Preparation

The initial step in the method involves preparing the data input by processing Unicode Javanese characters from the transliteration of Javanese script manuscripts. In this stage, the Javanese characters are presented without separators, which makes identifying individual words or sentences challenging. The successful processing and segmentation of these characters are crucial for the next steps of the method, which aims to segment the Javanese script characters into sentences.

**Algorithm 1:** Converting to $N \times M$ vector

**Data:** $s$ as string of characters; $M$ as the lookback value; $N$ as the length of the string
**Result:** $X$,$Y$ as $N \times M$ vector and its label

```
X,Y ← [];
d ← load character dictionary;
s = '|' + s; /* add a delimiter '|' at the beginning */
data ← [d['<pad>']] × M;
for ( i = 1; i < N; i = i + 1 ) {
    current = s[i];
    before = s[i − 1];
    if current == '|' then
        continue;
    end
    data = data[1 : M] + [d[current]];
    if before == '|' then
        label ← 1;
    else
        label ← 0;
    end
    X[i] ← data; Y[i] ← label;
}
```

When conducting word segmentation, segmenting characters before and after a word is often necessary. The order of character information in the future is also very important when processing character-based inputs. Therefore, the Javanese script string is divided into $N \times M$ vectors for dataset creation. Here, $N$ is the length of the string of characters, and $M$ is the "lookback" value. This value is key to dividing the string of characters into labeled vectors. The size of the $N$ value can vary, the smaller the $N$ value, the faster the training process is because of the smaller vector size. These vectors are illustrated in Fig. 2 with dashed green square lines, and the procedure for processing them is detailed in Algorithm 1. For each vector, the label will be given a '1' value if the last character marks the beginning of a word and a '0' value if the character is part of a word.

Following the dataset creation phase, the subsequent step involves data encoding through one-hot encoding. This encoding procedure necessitates the establishment of a character dictionary, wherein each character is mapped to a corresponding index. Within the Javanese script, a sum total of 56 characters is identified, supplemented by special characters including '<pad>' to represent blank values and '<unk>' for unknown characters. The transformation of characters into vectors, culminating in the encoding process, lays the foundation for what will be construed as an input sequence within the designed neural network.

### B. Word Segmentation Model

The encoded data input is then processed through a Bi-directional Long Short-Term Memory (BiLSTM) layer. This layer can analyze input sequences in forward and backward

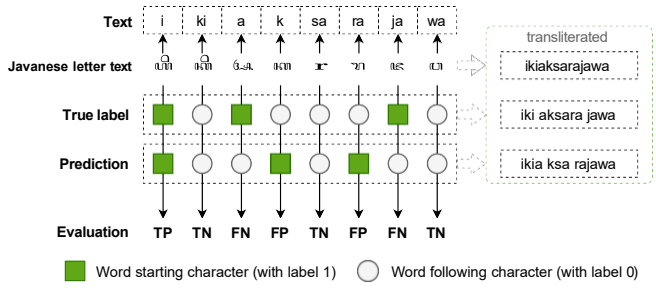


Fig. 3. Segmentation evaluation metrics

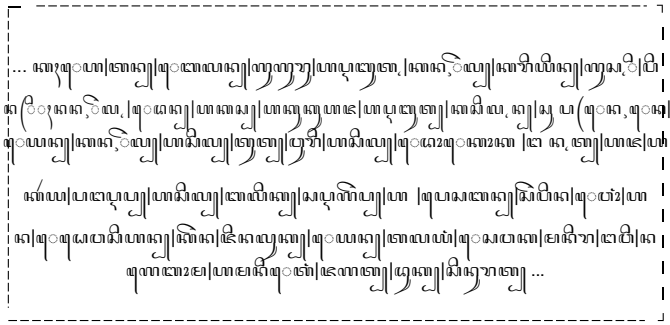

Fig. 4. Dataset format

directions, capturing contextual information from both the previous and next areas in the sequence. This two-way process completes the network to make informed decisions about word boundaries based on a comprehensive understanding of context. The output from the BiLSTM layer is directed through one or more dense hidden layers, further refining the features extracted by the BiLSTM. These layers are important for learning complex patterns in the data and are crucial for improving the predictive accuracy of the network.

The architecture concludes with a softmax layer, which classifies each input character into one of two categories: '1' indicating the beginning of a new word and '0' denoting the continuation of the current word. This classification relies on the probability distribution over potential outcomes generated by the network, enabling precise word segmentation. The result from the softmax layer is a sequence of binary labels for each character in the input text, effectively delineating word boundaries within the sequence.

### C. Evaluation

For the evaluation, a test set that the model has not encountered during training will be used to ensure an unbiased performance measurement. [32]. The evaluation metrics will include accuracy, precision, recall, and the F1-score. An overview of the evaluation method is illustrated in Fig. 3. Furthermore, a comparative analysis of the evaluation metrics obtained from four pre-existing deep learning models is planned. This comparison will help discern the relative effectiveness and efficiency of the proposed method. By doing so, the aim is to understand where the model stands in relation to established methodologies and identify areas where improvements can be made.

TABLE I
DATASET PROPORTION

| Data split | | Proportion |
|---|---|---|
| Train | 44,489,707 | 62.18% |
| Test | 16,999,584 | 23.76% |
| Validation | 10,064,278 | 14.07% |

## IV. EXPERIMENTAL RESULT

In this section, the dataset used and how it was prepared will be described. Furthermore, the results achieved in the experiment will be explained. However, a direct comparison of this experiment with the related word segmentation for Javanese letter research using a rule-based approach achieved in [7] is not possible, as it uses a different dataset that is not accessible.

### A. Datasets

For the experiment, data from public datasets comprising Indonesian language articles [33] were used to train the model. These articles were translated into Javanese and then further translated into Javanese letters. Subsequently, the translated text was split into individual words to create a corpus for word-level training. The dataset format can be seen in Fig. 4, which displays a text of Javanese letters that have been separated by character '|'. The dataset includes 89.3 MB of data with 71,553,569 lines.

### B. Experiment Result

After conducting experiments with an epoch value of 5, accuracy and loss metrics were obtained on training and validation data which can be seen in Fig. 5-8. The graphs in Fig. 5 and Fig 7. show the training and validation accuracy of various models used in this experiment, including BiLSTM, RNN, GRU, and CNN. From the training accuracy graph, it can be seen that the BiLSTM model achieved a higher accuracy rate compared to the other models. All models showed a stable increase in accuracy values. The validation accuracy graph confirms this superiority, where the BiLSTM model achieved a validation accuracy of about 98.86% at the final epoch. Meanwhile, the RNN, GRU, and CNN models each achieved validation accuracies of about 97.71%, 98.55%, and 93.87% respectively. This performance confirms that the BiLSTM model is not only superior in learning training data but also has better character pattern recognition capabilities compared to other models.

The graphs in Fig. 6 and Fig. 8 show the training and validation loss of the three models during the training process. A lower loss value indicates better performance in terms of predictions on data not seen during training. On the training loss graph, all models show a decrease in loss that tends to flatten as the epoch increases. The BiLSTM model has the lowest loss value among all models. At the end of the

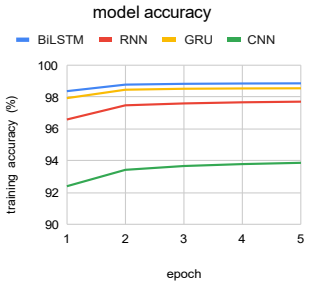


Fig. 5. Model training accuracy

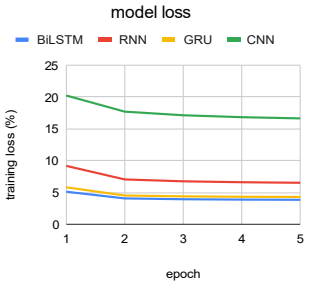


Fig. 6. Model training loss

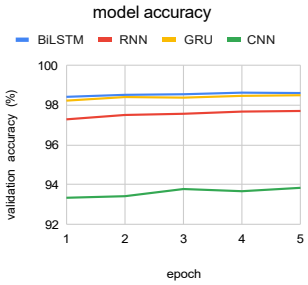


Fig. 7. Model validation accuracy

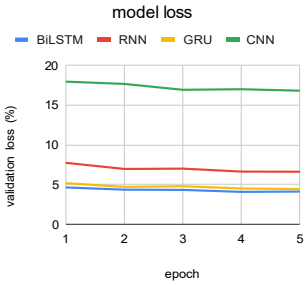


Fig. 8. Model validation loss

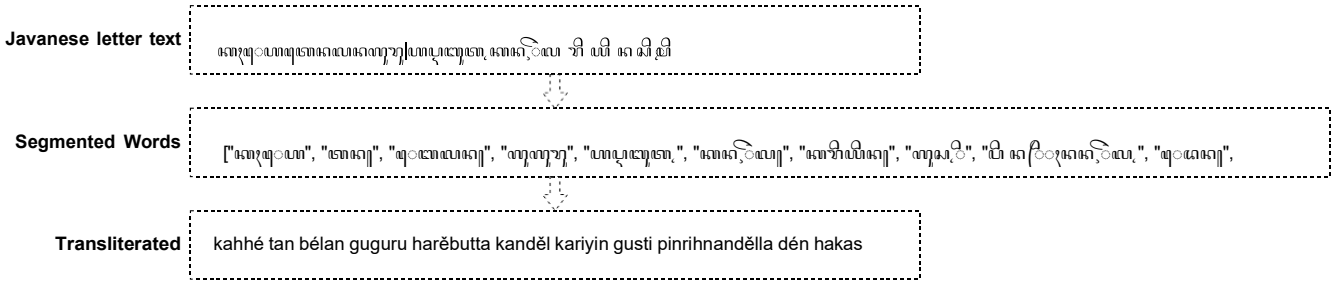


Fig. 9. The word segmentation process using the method

TABLE II
EVALUATION OF DEEP LEARNING MODEL PERFORMANCE ON UNSEEN DATASET

| Evaluation Metrics | BiLSTM (%) | RNN (%) | GRU (%) | CNN (%) |
|---|---|---|---|---|
| Accuracy | **98.62** | 97.68 | 98.50 | 93.80 |
| Precision | **96.51** | 94.15 | 96.30 | 88.50 |
| Recall | **96.00** | 93.03 | 95.10 | 75.75 |
| F1 | **96.23** | 93.58 | 95.70 | 81.65 |

training, the BiLSTM model had a training loss value of about 3.87%, while the RNN, GRU, and CNN were around 6.57%, 4.32%, and 16.65%, respectively. The validation loss graph again shows the superiority of the BiLSTM model with a lower validation loss than the other models, which is about 4.13%. This shows that the BiLSTM model not only achieves high accuracy but also converges faster minimizes prediction errors during training and has better generalization capabilities on new data.

Overall, the analysis from these four graphs shows that the BiLSTM model offers better performance compared to RNN, GRU, and CNN in the task of word segmentation in the transliteration of Javanese script manuscripts. BiLSTM achieved higher training and validation accuracy, as well as lower training and validation loss, indicating superior ability in learning and generalizing patterns from complex data. These findings support the use of BiLSTM as an effective method to improve accuracy and efficiency in the digitization and preservation process of these valuable manuscript texts.

Table II summarizes the evaluation results of deep learning models' performance on unseen datasets. The BiLSTM model achieved the highest scores across all metrics, including accuracy, f1-score, precision, and recall. With an accuracy of 98.62%, an f1-score of 96.23%, a precision of 96.51%, and a recall of approximately 96%, BiLSTM consistently outperforms other models. This suggests that using the BiLSTM architecture for word segmentation tasks in the transliteration of Javanese script manuscripts yields the best results.

Fig. 9 is an example of the word segmentation process using the proposed method. Javanese manuscripts, using this method, produce a word array that is then transliterated. This shows that the proposed model has successfully performed the word segmentation process accurately.

## V. CONCLUSION

In the contemporary era, the precision of word segmentation in Javanese letter manuscript transliteration presents a crucial need to enhance the accuracy and efficiency of digital text processing. In addressing the challenges of manuscript transliteration, this work proposes an exploration of a deep learning approach for word segmentation in Javanese letter manuscripts. This significantly improves the accuracy of detecting word boundaries. The results of the experiment show that the BiLSTM method is the best model in performing the process of segmenting Javanese manuscript words, achieving higher accuracy and robust performance in recognizing complex word formations. The implications of these findings are very significant for the field of transliteration and digitization of ancient manuscripts. By using the proven effective deep learning approach, the process of transliterating Javanese

manuscripts can be done more accurately and efficiently. This not only helps in the preservation of ancient Javanese literature but also facilitates more in-depth linguistic and cultural studies.

Going forward, the aim is to enhance this research by integrating more sophisticated deep learning frameworks and expanding the dataset to refine the model's capabilities in handling various manuscript styles and complexities.

## ACKNOWLEDGMENT

This research was funded by Institut Teknologi Sepuluh Nopember (ITS) under Penelitian Dana Departemen Program, Penelitian Keilmuan, Penelitian Flagship, Penelitian Kolaborasi Pusat and under the Project Scheme of the Publication Writing and Intellectual Property Rights (IPR) Incentive (Penulisan Publikasi dan Hak Kekayaan Intelektual/PPHKI) Program 2024.

## REFERENCES

[1] A. Perdana, "Ragam langgam aksara jawa dari manuskrip hingga buku cetak," *Manuskripta*, vol. 10, p. 1, 08 2020.

[2] E. Nurhayati, *Dunia Manuskrip Jawa: Teori, Metode, dan Aplikasinya dalam Praktik Pernaskahan Jawa*, 2018.

[3] A. Haryono, R. Sarno, and K. Sungkono, "Stock price forecasting in indonesia stock exchange using deep learning: a comparative study," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 14, pp. 861–869, 02 2024.

[4] L. Farokhah, R. Sarno, and C. Fatichah, "Cross-subject channel selection using modified relief and simplified cnn-based deep learning for eeg-based emotion recognition," *IEEE Access*, vol. 11, pp. 110 136–110 150, 2023.

[5] M. Malikhah, R. Sarno, S. Inoue, M. S. Ardani, D. Purbawa, S. Sabila, K. Sungkono, C. Fatichah, D. Sunaryono, A. Bakhtiar, Libriansyah, C. Prakoeswa, D. Tinduh, and Y. Hernaningsih, "Detection of infectious respiratory disease through sweat from axillary using an e-nose with stacked deep neural network," *IEEE Access*, vol. 10, pp. 1–1, 01 2022.

[6] S. Sabila, R. Sarno, K. Triyana, and K. Hayashi, "Deep learning in a sensor array system based on the distribution of volatile compounds from meat cuts using gc–ms analysis," *Sensing and Bio-Sensing Research*, vol. 29, p. 100371, 07 2020.

[7] A. R. Widiarti, R. Pulungan, A. Harjoko, Marsono, and S. Hartati, "A proposed model for javanese manuscript images transliteration," vol. 1098. Institute of Physics Publishing, 10 2018.

[8] A. W. Mahastama, "Model berbasis aturan untuk transliterasi bahasa jawa dengan aksara latin ke aksara jawa," *Jurnal Buana Informatika*, vol. 13, pp. 146–154, 2022.

[9] F. Ilham and N. Rochmawati, "Transliterasi aksara jawa tulisan tangan ke tulisan latin menggunakan cnn," *Journal of Informatics and Computer Science (JINACS)*, 2020.

[10] K.-H. Chang, "Natural language processing: Recent development and applications," *Applied Sciences*, vol. 13, p. 11395, 10 2023.

[11] B. An and C. Long, "Ancient tibetan word segmentation based on deep learning," in *2021 International Conference on Asian Language Processing (IALP)*, 2021, pp. 292–297.

[12] P. Limkonchotiwat, W. Phatthiyaphaibun, R. Sarwar, E. Chuangsuwanich, and S. Nutanong, "Domain adaptation of thai word segmentation models using stacked ensemble," B. Webber, T. Cohn, Y. He, and Y. Liu, Eds. Association for Computational Linguistics, 11 2020, pp. 3841–3847. [Online]. Available: https://aclanthology.org/2020.emnlp-main.315

[13] M. Maryamah, A. Arifin, R. Sarno, R. Indraswari, and R. Sholikah, "Pseudo-relevance feedback combining statistical and semantic term extraction for searching arabic documents," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 5, pp. 238–246, 2021, publisher Copyright: © 2021. All Rights Reserved.

[14] M. Maryamah, A. Z. Arifin, R. Sarno, and A. M. Hasan, "Adapting google translate using dictionary and word embedding for arabic-indonesian cross-lingual information retrieval," in *2020 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS)*, 2021, pp. 205–209.

[15] P. Chormai, P. Prasertsom, J. Cheevaprawatdomrong, and A. Rutherford, "Syllable-based neural thai word segmentation," D. Scott, N. Bel, and C. Zong, Eds. International Committee on Computational Linguistics, 12 2020, pp. 4619–4637. [Online]. Available: https://aclanthology.org/2020.coling-main.407

[16] T. Chay-intr, H. Kamigaito, K. Funakoshi, and M. Okumura, "Latte: Lattice attentive encoding for character-based word segmentation," , vol. 30, pp. 456–488, 2023.

[17] S. Choi, T. Kim, J. Seol, and S. goo Lee, "A syllable-based technique for word embeddings of korean words," 8 2017.

[18] Y. Chen, K. Lim, and J. Park, "Korean named entity recognition based on language-specific features," 5 2023. [Online]. Available: http://arxiv.org/abs/2305.06330 http://dx.doi.org/10.1017/S1351324923000311

[19] T. Chay-Intr, H. Kamigaito, and M. Okumura, "Character-based thai word segmentation with multiple attentions." Incoma Ltd, 2021, pp. 264–273.

[20] L. Wang, H. Yang, X. Xing, and Y. Yan, "Tibetan word segmentation method based on cnn-bilstm-crf model," 2019, pp. 319–324.

[21] Y. Jamtsho and P. Muneesawang, "Dzongkha word segmentation using deep learning," 2020, pp. 1–5.

[22] T. Lapjaturapit, K. Viriyayudhakom, and T. Theeramunkong, "Multi-candidate word segmentation using bi-directional lstm neural networks," 2018, pp. 1–6.

[23] S. Seeha, I. Bilan, L. M. Sanchez, J. Huber, M. Matuschek, and H. Schütze, "Thailmcut: Unsupervised pretraining for thai word segmentation," pp. 11–16, 2020. [Online]. Available: https://github.com/meanna/ThaiLMCUT

[24] Y. Kitagawa and M. Komachi, "Long short-term memory for japanese word segmentation," 09 2017.

[25] R. Kittinaradorn, K. Chaovavanich, T. Achakulvisut, K. Srithaworn, P. Chormai, C. Kaewkasi, T. Ruangrong, and K. Oparad, "DeepCut: A Thai word tokenization library using Deep Neural Network." Sep. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.3457707

[26] C. Haruechaiyasak, S. Kongyoung, and M. Dailey, "A comparative study on thai word segmentation approaches," in *2008 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, vol. 1, 2008, pp. 125–128.

[27] D. Tanaya and M. Adriani, "Dictionary-based word segmentation for javanese," in *Workshop on Spoken Language Technologies for Under-resourced Languages*, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:207427196

[28] S. N. Khan, K. Khan, W. Khan, A. Khan, F. Subhan, A. U. Khan, and B. Ullah, "Urdu word segmentation using machine learning approaches," 2018. [Online]. Available: www.ijacsa.thesai.org

[29] X. Huang and S. E. Robertson, "Applying machine learning to text segmentation for information retrieval," pp. 333–362, 2003.

[30] C. Zhuoma, R. Cai, Z. San, Q. Guan, and M. Zhuo, "Cross-domain tibetan word segmentation based on deep learning," *Journal of Physics: Conference Series*, vol. 1631, p. 012025, 09 2020.

[31] P. Chormai, P. Prasertsom, and A. T. Rutherford, "Attacut: A fast and accurate neural thai word segmenter," *ArXiv*, vol. abs/1911.07056, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:208138985

[32] K. Sirts and K. Peekman, "Evaluating sentence segmentation and word tokenization systems on estonian web texts," vol. 328. IOS Press BV, 9 2020, pp. 174–181.

[33] "Indonesian oscar corpus," https://huggingface.co/datasets/oscar-corpus/OSCAR-2201/tree/main/compressed/id_meta, accessed: 2024-05-01.