Available online at website: https://jurnal.iaii.or.id/index.php/RESTI

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 9 No. 5 (2025) 1210 - 1221 e-ISSN: 2580-0760

Empowering Low-Resource Languages: Javanese Machine Translation

Danang Arbian Sulistyo[1], Aji Prasetya Wibawa[2], Wayan Firdaus Mahmudy[3], Fadhli Almu'iini Ahda[4], Andrew Nafalski[5]
[1,4]Information Technology, Technology and Design Faculty, Institut Teknologi dan Bisnis Asia, Malang, Indonesia
[2]Electrical Engineering and Computer Science, State University of Malang, Indonesia
[3]Computer Science Department, Brawijaya University, Malang, Indonesia
[5]Electrical Engineering, University of South Australia, Adelaide, Australia

[1]danangarbian@gmail.com, [2]aji.prasetya.ft@um.ac.id, [3]wayanfm@ub.ac.id, [4]fadhlial@asia.ac.id, [5]andrew.nafalski@unisa.edu.au

Abstract

This study addresses the critical need to preserve and revitalize the Javanese language, which despite its widespread popularity, faces challenges as a low-resource language in Indonesia. The decline in Javanese proficiency among younger generations poses a significant threat to the language's cultural significance and heritage. To address this issue, this study introduces an innovative approach to machine translation, focusing on the development of a robust Indonesian-Javanese translation system. Utilizing advanced neural machine translation (NMT) techniques, including Long Short-Term Memory (LSTM) networks, the proposed system aims to bridge the linguistic gap between Indonesian and Javanese. Special attention was given to the unique linguistic characteristics and challenges of Javanese, with the goal of achieving exceptional translation accuracy and fluency. Through extensive experimentation and evaluation, this study aims to demonstrate the effectiveness of the translation system in facilitating cross-cultural communication and language preservation efforts within the Javanese-speaking community. By emphasizing the significance of Javanese as a widely spoken yet under-resourced language, this study underscores the importance of innovative technological solutions in safeguarding linguistic diversity and cultural heritage. Through its contributions, the research seeks to address the pressing need for language preservation and revitalization, particularly in the context of low-resource languages like Javanese.

Keywords: javanese translation; NMT; linguistic heritage; low-resource language

How to Cite: D. A. Sulistyo, Aji Prasetya Wibawa, Wayan Firdaus Mahmudy, Fadhli Almu'iini Ahda, and Andrew Nafalski, "Empowering Low-Resource Languages: Javanese Machine Translation", *J. RESTI (Rekayasa Sist. Teknol. Inf.)* , vol. 9, no. 5, pp. 1210 - 1221, Oct. 2025. *Permalink/*DOI: https://doi.org/10.29207/resti.v9i5.6887

Received: July 3, 2025
Accepted: October 8, 2025
Available Online: October 26, 2025

This is an open-access article under the CC BY 4.0 License
Published by Ikatan Ahli Informatika Indonesia

1. Introduction

This work seeks to further the field of machine translation by creating a resilient Indonesian Javanese translation system to address these issues. The suggested system aims to address the linguistic disparity between Indonesian and Javanese by employing advanced neural machine translation (NMT) methods [1], [2], specifically Long Short-Term Memory (LSTM) networks. Through the utilization of artificial neural networks, the system aims to attain enhanced levels of translation precision and fluency, hence enabling smooth and uninterrupted communication between individuals proficient in these two languages.

The main aim of this study is to create a machine translation system for Indonesian Javanese that is specifically designed to address the distinct linguistic difficulties presented by Javanese, a widely spoken language with limited resources. Based on data from UNESCO, BPS, and studies by Sulistyo et al [3] - [5], the number of active Javanese speakers under 30 has declined by over 30% in the past decade, with daily use becoming rare even in native regions such as Yogyakarta and Central Java. Sulistyo et al. highlight challenges in low-resource language processing,

including sparse corpora and limited NLP tools. This underscores a research gap in developing robust Indonesian-Javanese translation systems that effectively handle sociolinguistic and resource limitations. The primary objective of this study is to: (1). The objective is to develop a resilient NMT framework that is specifically tailored to the linguistic attributes of Indonesian and Javanese languages. The primary aim is to improve the precision and fluency of translations. (2). To effectively capture long-distance interdependence and contextual nuances inherent in both languages, it is recommended to incorporate LSTM architecture into the encoder-decoder RNN framework. (3). Assess the efficacy of the proposed translation system by conducting thorough experimentation and comparative analysis, evaluating its performance in comparison to established translation models and standards. (4). Examine the possible uses and consequences of the created translation system in promoting intercultural communication, safeguarding languages, and supporting educational efforts in Javanese-speaking communities.

Unlike previous research which often overlooked the hierarchical nature of Javanese language levels (e.g., ngoko, krama), our approach considers these sociolinguistic features in both preprocessing and evaluation. Our approach enhances NMT development by leveraging aligned religious corpora for cultural consistency, incorporating sentence structure harmonization in preprocessing, and conducting dual-direction evaluation (Indo-Javanese and Javanese-Indo) with BLEU score and error interpretation.

2. Methods

This study utilizes a systematic procedure, beginning with the gathering of data, followed by the processing of text, modeling of data, training, and evaluation of output. The sequential advancement of these stages is depicted in Figure 1. The initial step involves gathering corpora from the Bible [6], followed by the complexities of text processing. This culminates in the creation of data sets in the necessary formats. Subsequently, the critical stages of modeling and training are undertaken, employing Neural Machine Translation (NMT) techniques. The BLEU evaluation metric [7], [8] is used to evaluate the precision of translation results.

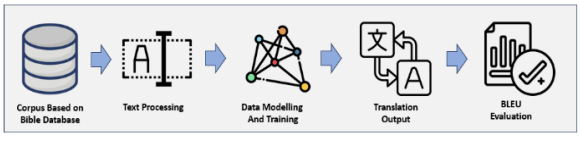


Figure 1. Research block diagram.

2.1 Dataset

This study utilizes a dataset derived from the New Testament Bible, which includes a diverse range of languages. The dataset consists of 7,900 sentences in both Indonesian and Javanese languages, resulting in a large collection of over 15,000 pairs of sentences. However, using only Bible texts introduces domain specificity, limiting generalizability to casual or contemporary discourse. Although the dataset comprises 7,900 sentence pairs, it was validated for balance in sentence length, complexity, and syntactic diversity, making it viable for low-resource LSTM training scenarios. Further studies should incorporate additional sources such as online forums, news media, and everyday dialogues to enhance corpus diversity. A Web Scraper [9] was utilized to pull data from online Bible repositories to curate this corpus. The first table provides a thorough summary of the dataset's contents, highlighting differences among English versions and enhancing our understanding of linguistic changes between translations.

Table 1. Dataset contents in English, Indonesian and Javanese languages

English Verse	Indonesian Verse	Javanese Verse
Now after the Sabbath, at dawn on the first day of the week, Mary Magdalene and the other Mary went to look at the tomb.	Setelah hari Sabat lewat, menjelang menyingsingnya fajar pada hari pertama minggu itu, pergilah Maria Magdalena dan Maria yang lain, menengok kubur itu.	Bareng dina Sabat wis kliwat, ing wayah bangun ésuk, ngarepaké dina kapisan ing minggu iku, Maria Magdaléna karo Maria sijiné padha lunga niliki pasaréan.
Suddenly there was a severe earthquake, for an angel of the Lord descending from heaven came and rolled away the stone and sat on it.	Maka terjadilah gempa bumi yang hebat sebab seorang malaikat Tuhan turun dari langit dan datang ke batu itu dan menggulingkannya lalu duduk di atasnya.	Lah tumuli ana lindhu gedhé, awit ana malaékating Pangéran tumedhak saka langit lan nyelak ngglundhungaké watuné, nuli dilenggahi.
His appearance was like lightning, and his clothes were white as snow.	Wajahnya bagaikan kilat dan pakaiannya putih bagaikan salju.	Pasuryané katon kaya kilat, pangagemané putih kaya salju.
The guards were shaken and became like dead men because they were so afraid of him.	Dan penjaga-penjaga itu gentar ketakutan dan menjadi seperti orang-orang mati.	Awit saka wediné marang malaékat, sing padha jaga gumeter wedi banget nganti kaya wong mati.
But the angel said to the women, “Do not be afraid; I know that you are looking for Jesus, who was crucified.	Akan tetapi malaikat itu berkata kepada perempuan-perempuan itu: "Janganlah kamu takut; sebab aku tahu kamu mencari Yesus yang disalibkan itu.	Ananging malaékat mau ngandika marang wong-wong wadon iku, "Kowé aja padha wedi, sabab aku ngerti, kowé padha nggolèki Yésus kang disalib iku.
He is not here, for he has been raised, just as he said. Come and see the place where he was lying.	Ia tidak ada di sini, sebab Ia telah bangkit, sama seperti yang telah dikatakan-Nya. Mari, lihatlah tempat Ia berbaring.	Panjenengané ora ana ing kéné, awit wis wungu, kaya kang wus kapangandikakaké. Padha mrénéa, delengen papan panggonané sumaré.

Before its application, various processes were necessary to develop the dataset. A human verification was performed after the scraping procedure to extract data from the online Bible platform [10], [11], as there were differences in translation places between Indonesian and Javanese [12].

Table 2. Differences in translation location between Indonesian and Javanese

English Verse	Indonesian Verse	Javanese Verse
Then they returned and prepared aromatic spices and perfumes	Dan setelah pulang, mereka menyediakan rempah-rempah dan minyak mur.	Saulihe banjur padha nyawisake anggi-anggi lan lenga jebad. Dene ing dina Sabat padha leren, netepi ing angger-angger.
On the Sabbath they rested according to the commandment	Dan pada hari Sabat mereka beristirahat menurut hukum Taurat,	

Selected verses within the Javanese language exhibited notable differences. Certain Javanese verses have been seen to combine many sentences into a single unit, as depicted in Table 2.



Figure 2. Indonesian-Javanese Dataset

After manual verification, the dataset was prepared for further processing, as shown in Figure 2, which illustrates the composition of the dataset used in this investigation. The dataset consists of two distinct files, mainly an Excel file (.xlsx) and a text file (.txt). The translations from Indonesian and Javanese are organized in separate columns inside the Excel file, with tab separators used to facilitate their transfer to the text file (.txt).

2.2 Neural Machine Translation

The integration of recurrent neural network (RNN) encoders with language model decoder RNNs is central to enhancing Machine Translation (MT) systems. Studies like Y. Dai's work on syntactic representations in pre-trained Transformers for MT improvement [13], U. Farooq et al.'s crowdsourcing-based framework for parallel corpora development [14], and S. K. Mondal et al.'s evaluation of MT systems [15] have contributed significantly to this field. However, these studies mainly focus on well-resourced languages, leaving a gap in addressing the unique challenges posed by low-resource languages like Javanese. This research aims to fill that gap by incorporating language-specific complexities into the Indonesian-Javanese translation model. In contrast, the decoder recurrent neural network (RNN) employs the encoded output generated by the encoder to perform a word-by-word translation of the input vocabulary [16]. Figure 3 illustrates the symbiotic interaction between the encoder and decoder models, showcasing the essential synergy between them. The encoder-decoder architecture, which is an architectural paradigm, serves as the foundation for machine translation systems, introducing a new era of linguistic synthesis and comprehension.

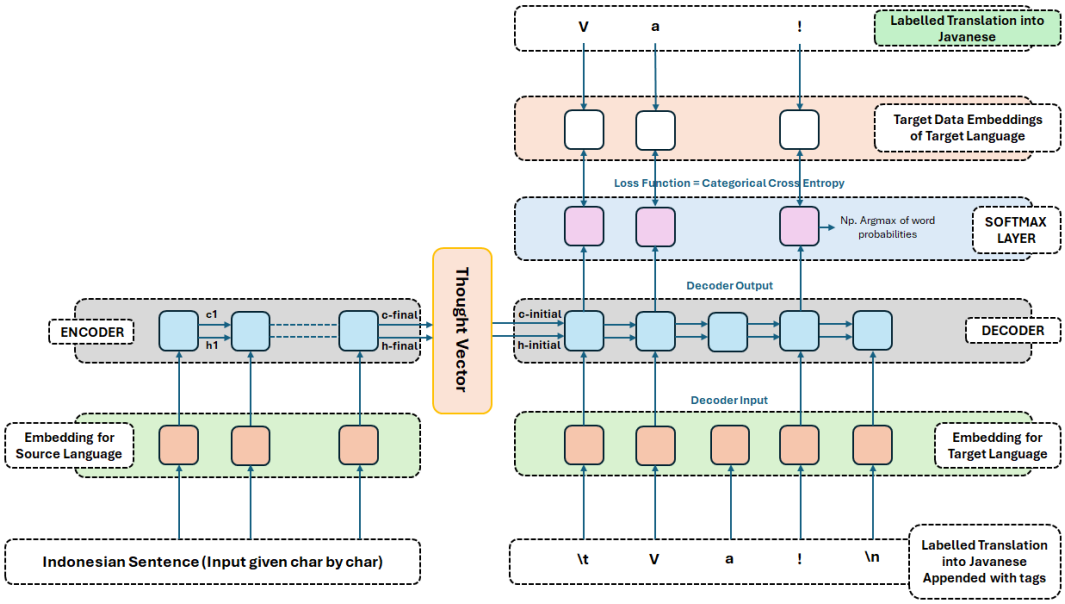


Figure 3. Encoder-Decoder model architecture in NMT.

For this translation task, we explicitly selected Long Short-Term Memory (LSTM) networks over alternative architectures such as Transformer models, based on several key considerations relevant to our research context, particularly for a low-resource language like Javanese. Firstly, LSTMs are inherently adept at capturing long-range dependencies within sequential data, a critical capability for accurate language translation where contextual information spans across sentences. This makes them a robust choice for processing complex linguistic structures. Secondly, from a practical standpoint, LSTM architectures are generally less computationally intensive and possess a more established ecosystem of tools and research at the time of this study compared to the then-emerging Transformer models. This facilitates more efficient training and experimentation, especially given potentially limited computational resources. Thirdly, for datasets of the size available for low-resource languages (approximately 7,900 sentence pairs), LSTMs often provide a strong balance between learning capacity and mitigating the risk of overfitting, which can be more pronounced with the higher parameter count of Transformer models on smaller datasets. While Transformer models have since emerged as state-of-the-art for large-scale NMT, our choice of LSTM provides a solid and interpretable baseline for the Indonesian-Javanese language pair, laying foundational work for future explorations with more complex architectures as data resources for Javanese expand.

The operating dynamics of the encoder-decoder machine translation (MT) system are illustrated in Figure 4, providing a clear depiction of the translation process from Javanese to Indonesian. The encoding components are designed to analyze every lexical unit of the Javanese phrase, including the punctuation mark "\t" at the start and the punctuation mark "\n" at the conclusion. It is worth mentioning that in this particular instance, the original source sentence is not presented, a tactic that is substantiated by research that suggests improved translation results.

In order to explore the field of Machine Translation (MT) using artificial neural networks, it is crucial to incorporate RNN encoders with language model decoder RNNs [17]. The rigorous examination performed by recurrent neural network (RNN) encoders on every individual word inside the input sentence enables the generation of intricate representations that are crucial for the processing of sentences in the original language [18]. Following this, the decoder recurrent neural network (RNN) employs the encoded output generated by the encoder to successively translate the input lexicon. Figure 3 illustrates the symbiotic interaction between the encoder and decoder models, showcasing the essential synergy between them. The encoder-decoder architecture , which is an architectural paradigm, serves as the foundation for machine translation systems , introducing a new era of linguistic synthesis and comprehension To provide a clearer understanding, let's detail the operational dynamics of our encoder-decoder system, encompassing both translation directions, as depicted in Figures 3 and 4.

Translation Process: Indonesian to Javanese (Figure 3) begins with the Encoder Phase begins by taking an Indonesian source sentence, processing it character by character. Each character is first transformed into a numerical representation through the 'Embedding for Source Language' layer, and these embeddings are then fed sequentially into the 'ENCODER', an LSTM network. This encoder processes the entire input sequence, compressing its semantic meaning into a fixed size 'Thought Vector'. This vector, represented by the LSTM's final hidden state (h_{final}) and cell state (c_{final}), effectively summarizes the complete Indonesian sentence. Subsequently, in the Decoder Phase, this 'Thought Vector' initializes the 'DECODER' (another LSTM network). The decoder begins by receiving a special 'start-of-sequence' token (e.g., \t) as its initial input. Based on its internal state (initialized by the 'Thought Vector') and the current input, the decoder generates an output, which is then passed through a 'SOFTMAX LAYER' to produce a probability distribution over the entire Javanese vocabulary.

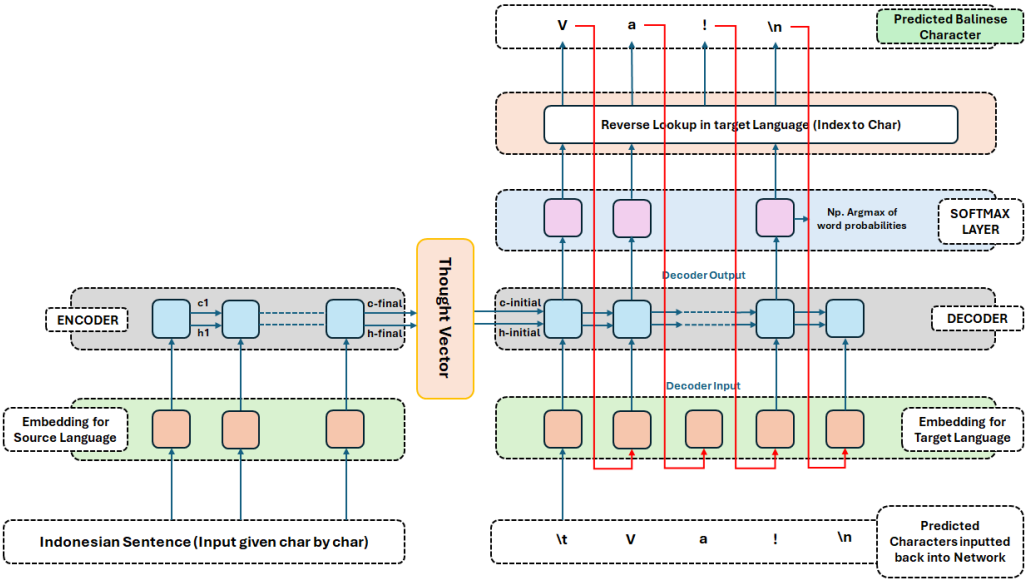


Figure 4. Example of translation using the encoder-decoder model architecture

The Javanese character with the highest probability is selected as the first predicted character of the translation. Crucially, this newly predicted character is fed back as the 'Decoder Input' for the next time step, enabling an auto-regressive process where the decoder generates one Javanese character at a time, with each prediction informed by the 'Thought Vector' and all previously generated characters. The translation concludes when the decoder predicts an 'end-of-sequence' token (e.g., \n).

Our NMT Model 2, responsible for Javanese-to-Indonesian translation, is detailed in Figure 4 and employs an encoder-decoder architecture. In the Encoder Phase (Javanese Input), the Javanese source sentence, marked with starting,) and ending (\n) tokens, is processed. The 'Embedding for Source Language' converts these characters into numerical vectors, which are then fed into the 'ENCODER' (an LSTM network). This encoder processes the sequence of Javanese characters, producing a 'Thought Vector' (consisting of (c_{final} and h_{final}) that encapsulates the sentence's meaning. Subsequently, in the Decoder Phase (Indonesian Output), this 'Thought Vector' initializes the 'DECODER's internal states ($c_{initial}$ and $h_{initial}$). The decoder begins by receiving a 'start-of-sequence' token (\t) as its initial input. Based on its current state and input, the decoder computes an output, which passes through a 'SOFTMAX LAYER' to predict the next Indonesian character. A 'Reverse Lookup in target Language (Index to Char)' then converts this predicted index back into its character form. The critical aspect, highlighted by the red arrows in Figure 4, is the feedback mechanism: the recently predicted character is immediately fed back as the input for the next time step. This iterative loop ensures that the decoder constructs the Indonesian translation character by character, leveraging the contextual understanding from the 'Thought Vector' and all previously generated characters. The translation is finalized when the decoder predicts the 'end-of-sequence' token (\n), signalling the completion of the Indonesian sentence.

2.3. Evaluation

The assessment of translated texts is strongly dependent on the thorough analysis of the degree of alignment between the translated sentences and their corresponding reference sentences. This process is made easier by employing the Bilingual Evaluation Index (BLEU) metric. The metric in question is a highly effective tool that has been specifically developed to assess the precision and consistency of machine-generated texts when translated from the original language to the desired language[19]. BLEU is a computational framework that allows evaluators to quantify the level of similarity between machine-generated translations and reference translations statistical accuracy measure that includes a short penalty constant (BP) [20].

$$BP_{BLUE} = \begin{cases} 1, & if\ c > r \\ e^{1\frac{r}{c}}, & if\ c \leq r \end{cases} \quad (1)$$

$$P_n = \frac{\sum C \in corpus\ n-gram \in C\ \sum count\ clip^{(n-gram)}}{\sum C \in corpus\ n-gram \in C\ \sum count_{(n-gram)}} \quad (2)$$

$$BLEU = BP_{BLEU}.e^{\sum_{n-1}^{N} W_n \log p_n} \quad (3)$$

Attaining a good BLEU score requires more than simply aligning the sentence lengths of the translated text with those of the reference text [21]. Additionally, it involves conforming to the intrinsic structural organization and word order of the original text. The mathematical foundations of the BLEU metric are explained by Formulas 1 to 3 [22], [23], providing a deeper understanding of the complex relationship between precision, the penalty for brevity [24], and the

alignment of reference translations inside the evaluation framework [25].

The BLEU measure provides a complete methodology for assessing machine-translated texts, encompassing not only the precision of the translations but also their coherence and adherence to the structure and semantics of the original language [26]. By means of this intricate assessment procedure, BLEU offers significant perspectives on the caliber of machine-generated translations [27], empowering researchers and practitioners to create well-informed judgments concerning the efficacy of translation systems and areas that might be enhanced.

3. Results and Discussions

The dataset utilized in this research has undergone thorough preprocessing to ensure its hygiene and appropriateness for analytical endeavors [28], [29].

3.1 Dataset Cleaning

The dataset was subjected to a number of text preprocessing methods before being used. These operations included normalization, tokenization, conversion of all words to lowercase, removal of punctuation, and eradication of numerical values within sentences [30], [31]. Table 3 presents the pseudocode that outlines the text processing operations. The implementation of thorough preprocessing techniques is crucial in order to reduce noise and normalize textual inputs [28], [32], hence improving the strength and dependability of following analytical tasks.

Table 3. Pseudocode for Text Processing

```
clean sentences
    prepare regex for char filtering.
    prepare translation table for removing punctuation.
        normalize unicode characters.
        tokenize on white space.
        convert to lowercase.
        remove punctuation from each token.
        remove non-printable chars form each token.
        remove tokens with numbers in them.
    store as string
```

Table 4 presents a comprehensive overview of the rigorous text processing processes implemented in this investigation. These rigorous techniques are carefully implemented on datasets for both the Indonesian and Javanese languages, promoting thorough analysis. The technique of normalization involves the careful utilization of Unicode characters to accurately normalize string vectors. This process transmutes the vectors using Canonical Decomposition Normalization, which promotes precise alignment [33].

Table 4. Text Processing for Javanese

No	Process	Input	Output
1	Normalize Unicode characters	Pilatus ndangu manèh, "Yèn mangkono apa kang kudu daktindakaké tumrap Yésus kang kasebut Kristus iku?" Kabèh padha matur, "Tiyang punika kasaliba!"	Pilatus ndangu manèh, Yèn mangkono apa kang kudu daktindakaké tumrap Yésus kang kasebut Kristus iku? Kabèh padha matur, Tiyang punika kasaliba!
2	Tokenize on white space	Pilatus ndangu manèh, Yèn mangkono apa kang kudu daktindakaké tumrap Yésus kang kasebut Kristus iku? Kabèh padha matur, Tiyang punika kasaliba!	Pilatus ndangu manèh, Yèn mangkono apa kang kudu daktindakaké tumrap Yésus kang kasebut Kristus iku? Kabèh padha matur, Tiyang punika kasaliba!
3	Convert to lowercase	Pilatus ndangu manèh, Yèn mangkono apa kang kudu daktindakaké tumrap Yésus kang kasebut Kristus iku? Kabèh padha matur, Tiyang punika kasaliba!	pilatus ndangu manèh, yèn mangkono apa kang kudu daktindakaké tumrap yésus kang kasebut kristus iku? kabèh padha matur, tiyang punika kasaliba!
4	Remove punctuation from each token.	pilatus ndangu manèh, yèn mangkono apa kang kudu daktindakaké tumrap yésus kang kasebut kristus iku? kabèh padha matur, tiyang punika kasaliba!	pilatus ndangu manèh yèn mangkono apa kang kudu daktindakaké tumrap yésus kang kasebut kristus iku kabèh padha matur tiyang punika kasaliba
5	Remove non-printable chars from each token	pilatus ndangu manèh yèn mangkono apa kang kudu daktindakaké tumrap yésus kang kasebut kristus iku kabèh padha matur tiyang punika kasaliba	pilatus ndangu manèh yèn mangkono apa kang kudu daktindakaké tumrap yésus kang kasebut kristus iku kabèh padha matur tiyang punika kasaliba
6	Remove tokens with numbers in them	pilatus ndangu manèh yèn mangkono apa kang kudu daktindakaké tumrap yésus kang kasebut kristus iku kabèh padha matur tiyang punika kasaliba	pilatus ndangu manèh yèn mangkono apa kang kudu daktindakaké tumrap yésus kang kasebut kristus iku kabèh padha matur tiyang punika kasaliba

Tokenization, an essential component of text analysis, systematically breaks down unprocessed textual information into smaller, more manageable entities known as tokens, therefore effectively enhancing understanding [34], [35]. The tokenization approach employed in this work is whitespace tokenization [36], [37], which involves the careful segmentation of phrases into relevant terminology.

The method of uppercase folding is accomplished with precise precision by converting to lowercase, ensuring meticulous homogeneity across the corpus. The primary aim of lowercase folding is to systematically convert all alphabetical characters in the document [38], [39], ranging from 'a' to 'z', into their lowercase counterparts, so assuring stringent uniformity. In addition, the painstaking procedure of noise removal effectively

removes non-printable letters and punctuation marks, such as [!,-./:;ó?@_"#$%&'()*.'|, so systematically enhancing the dataset for rigorous analysis [40]

3.2 Training and Testing Process

This study comprehensively examines the effectiveness of neural machine translation (NMT) utilizing Long Short-Term Memory (LSTM) method within the encoder-decoder framework across various input data scenarios. The pre-existing dataset is initially partitioned into training and testing data subsets. Subsequently, both training and testing phases are conducted on all available datasets, followed by the preparation of tokenizers for Javanese and Indonesian languages derived from the aforementioned dataset. The pseudocode employed to define the NMT model utilized in training and testing datasets is presented in Table 5.

Table 5. Pseudocode for training and testing datasets to define the model.

```
load the clean dataset.
set a tokenizer for Javanese and Indonesian
defined max sentence length.
define the NMT model.
        prepare training data.
        prepare validation data.
        compile the model.
summarize the defined model.
save
```

Based on the defined NMT model outcomes, two distinct models are generated for training: Model 1 facilitates translation from Indonesian to Javanese, whereas Model 2 serves the converse translation process, i.e., from Javanese to Indonesian. To achieve optimal performance, each model is rigorously evaluated through five repeated trials spanning 1000 epochs. Comparative analysis of the results is then facilitated by assessing BLEU scores, offering a quantitative metric for translation quality.

Figure 5 illustrates the significant methodological steps undertaken in this research. The tokenization function segments Indonesian and Javanese language texts, enabling determination of vocabulary size and maximum sequence length. This preprocessing stage culminates in the transformation of textual elements into indices through the encoding sequence function. Indonesian texts have a maximum sequence length of 51 characters, while Javanese texts have a maximum length of 52 characters. Following this preparatory stage, the training data is partitioned into several distinct sets: trainX (Javanese) and trainY (Indonesian), and vice versa for alternative translation direction.
The one-hot encoding methodology is used as a crucial part of the modeling process to include every word in the lexicon [3] - [5], [41]. Moreover, to improve the performance of the model, supplementary layers are included. Specifically, Model 1 incorporates an embedding layer that combines the length of Javanese sentences as input and the vocabulary size of the Indonesian language as output. Conversely, Model 2 employs the reverse configuration. The activation functions incorporate the fundamental principles of Softmax, while the implementation of the Dropout layer, with a probability of 0.2, serves as a protective measure against overfitting. This ensures the completion of the 128-layer architecture during the training phase. The selection of hyperparameters for our NMT model, including embedding size, number of layers, and dropout rate, was guided by established practices in neural network design for sequence-to-sequence tasks and empirical considerations for low-resource language settings.

```
Idn Vocabulary Size: 2542
Idn Max Length: 51
Jav Vocabulary Size: 3292
Jav Max Length: 52
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 52, 128)           421376

 lstm (LSTM)                 (None, 128)               131584

 dropout (Dropout)           (None, 128)               0

 repeat_vector (RepeatVecto  (None, 51, 128)           0
 r)

 lstm_1 (LSTM)               (None, 51, 128)           131584

 dropout_1 (Dropout)         (None, 51, 128)           0

 time_distributed (TimeDist  (None, 51, 2542)          327918
 ributed)

=================================================================
Total params: 1012462 (3.86 MB)
Trainable params: 1012462 (3.86 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 5. Sequential Model for Javanese - Indonesian and vice versa Machine Translation

Hyperparameter selection was meticulously chosen to optimize performance while robustly addressing potential overfitting, especially given our extensive 1000-epoch training regime. We utilized an embedding size of 128 (as shown in Figure 5) to ensure a balanced representation of vocabulary without excessive complexity or parameters that could lead to overfitting on a limited dataset. The model employs single LSTM layers for both the encoder and decoder, a configuration known for its stability and efficiency with medium-sized datasets, thus preventing over-parameterization. A Dropout rate of 0.2 was critically integrated into the architecture (as shown in Table 6) to serve as a primary regularization strategy, randomly deactivating neurons to enhance generalization.

Table 6. Hyperparameter in NMT Model.

Parameter	Value
Embedding Size	128
LSTM Layers	1 (Encoder), 1 (Decoder)
Dropout Rate	0.2
Epochs	1000
Optimizer	Adam
Softmax Activation	Yes
Batch Size	128

The Adam optimizer was chosen for its adaptive learning rate capabilities, which provide robust and efficient convergence for deep learning models. The Softmax activation function was incorporated in the output layer, enabling the model to produce a

probability distribution over the target vocabulary, which is essential for predicting the next character in the sequence. Finally, a batch size of 128 was selected to strike an optimal balance between training stability, computational efficiency, and effective gradient estimation for our dataset. These combined choices ensured that despite the high number of epochs, the models were effectively trained to learn generalized patterns, as confirmed by the consistent decline or plateauing of the validation loss (Figures 6 and 7), thereby mitigating overfitting.

The assessment of Model 1's efficacy in translating Indonesian into Javanese is undertaken through a sequence of five experiments, each spanning 1000 epochs. Figure 5 portrays the outcomes of the trial involving the Indonesian-Javanese translation model, with evaluation conducted utilizing the robust and effective BLEU metric. This quantitative instrument offers an objective assessment, measuring the fidelity of the model's output concerning the desired phraseology's length. The employed language exhibits clarity, conciseness, and precision, characterized by a formal register and judicious word selection.

The coherence of sentences and paragraphs ensures a logical progression of information, delineating causal relationships among statements. The text maintains adherence to conventional structural norms and formatting features, complemented by consistent citation styles and footnotes. Noteworthy, the text remains devoid of grammatical, spelling, or punctuation errors.

A comprehensive summary of the evaluation outcomes, derived from the application of the BLEU metric across the five experimental trials, is presented in Table 7. The quantitative evaluation of translation accuracy involves juxtaposing the translated outputs of candidate texts against one or more reference translations, employing the BLEU metric for assessment

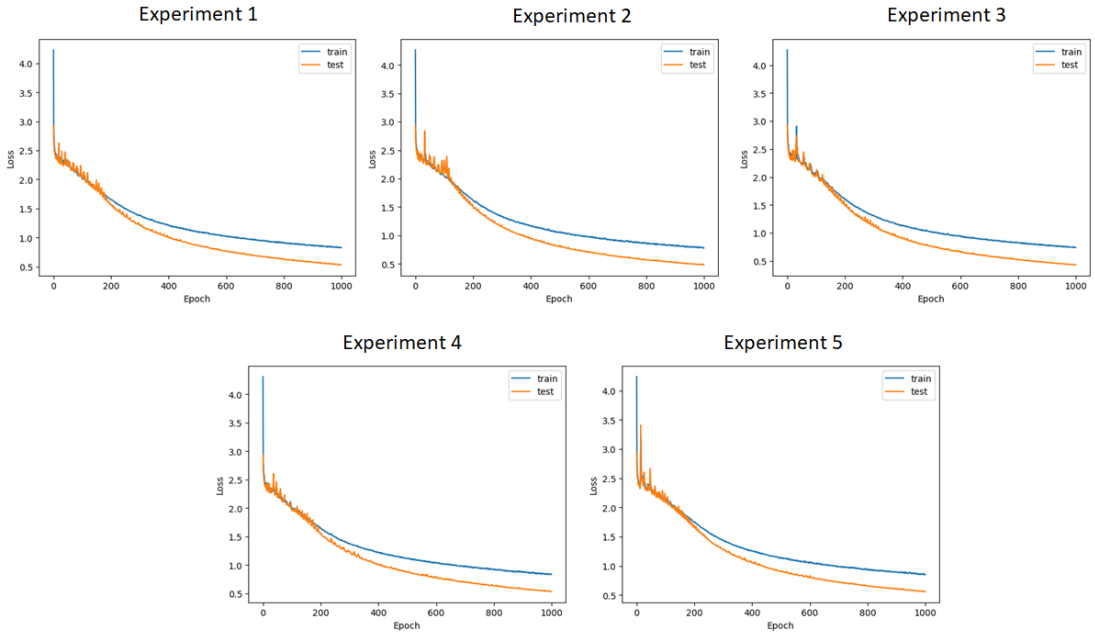


Figure 6. Indonesian-Javanese graphic model result

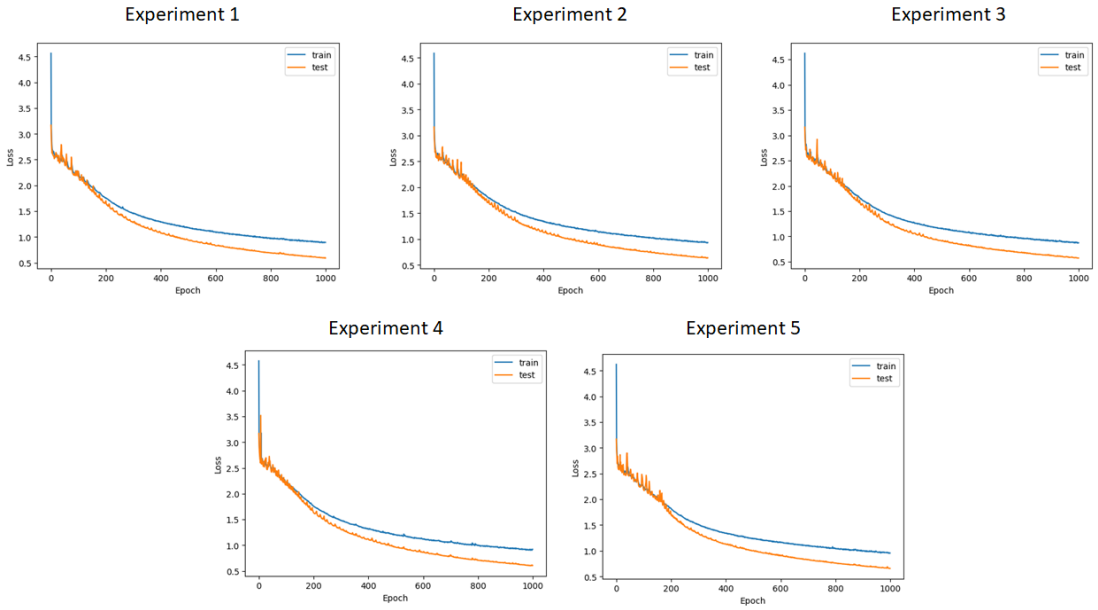


Figure 7. Javanese-Indonesian graphic model result

Table 7. Evaluating result, the Model-1 with BLEU Score

Metric	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5	Average
BLE	0.674	0.702	0.730	0.676	0.669	0.690
U-1	655	566	311	663	249	6888
BLE	0.503	0.543	0.584	0.502	0.505	0.528
U-2	078	852	831	839	596	0392
BLE	0.421	0.467	0.511	0.420	0.430	0.450
U-3	808	124	951	845	788	5032
BLE	0.299	0.345	0.394	0.298	0.312	0.330
U-4	668	188	121	393	802	0344

To fortify the reliability and fidelity of our translations, we recalibrated the dataset, concentrating solely on the trainX and testX subsets. The aggregated findings unveil an average BLEU-1 score of 0.6906888, signifying a notable level of unigram precision. Additionally, the commendable BLEU-2 score of 0.5280392 underscores substantial bigram coherence. The model's adeptness in capturing trigram and quadrigram structures within the translated texts is evidenced by elevated BLEU-3 and BLEU-4 scores, respectively, standing at 0.4505032 and 0.3300344.

As previously outlined, this study entails the creation of two models: Model 1 for Indonesian to Javanese translation and Model 2 for Javanese to Indonesian translation. The purpose of developing these two models is to ascertain the optimal Neural Machine Translation (NMT) model for Indonesian-Javanese language translation. Methodologically, the same approach was employed for Model 2, involving equivalent processes of training and testing. A series of identical experiments, comprising five iterations, each spanning 1000 epochs, yielded outcomes as depicted in Figure 7.

The results of the five experimental iterations are presented in Table 8, demonstrating the evaluation outcomes for Model 2, focusing on Javanese to Indonesian translation using the BLEU metric. These findings underscore the intricate nature of machine translation, wherein the interplay between lexical complexity and translation accuracy necessitates careful consideration. A comprehensive analysis of linguistic characteristics is imperative for determining optimal performance metrics.

Table 8. Evaluating Result, the Model-2 with BLEU Score

Metric	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5	Average
BLE	0.652	0.631	0.659	0.650	0.620	0.6429
U-1	275	519	988	610	322	428
BLE	0.476	0.443	0.485	0.471	0.434	0.4621
U-2	860	099	506	063	223	502
BLE	0.395	0.356	0.402	0.386	0.348	0.3778
U-3	966	586	196	210	540	996
BLE	0.276	0.237	0.281	0.264	0.232	0.2586
U-4	831	488	721	977	219	472

When comparing the evaluation results presented in Tables 6 and 7, detailing the performance metrics of Model 1 and Model 2 respectively, several noteworthy observations arise. Firstly, it is evident that Model 1 consistently outperforms Model 2 across all BLEU score metrics, as evidenced by the higher scores recorded in Table 6 compared to Table 7. This disparity is particularly pronounced in BLEU-1, wherein Model 1 attains an average score of 0.6906888, surpassing the average score of 0.6429428 achieved by Model 2. Moreover, this trend persists across all n-gram precision levels, with Model 1 consistently exhibiting superior performance to Model 2 for BLEU-2, BLEU-3, and BLEU-4 metrics. For instance, in the case of BLEU-2, Model 1 achieves an average score of 0.5280392, whereas Model 2 attains a lower average score of 0.4621502.

The performance disparity between these two models is attributed to various factors, including differing lexical and syntactic complexities inherent in the translation task. Model 1 generates more accurate translations, benefiting from the relatively simple syntactic structure of the Indonesian language. Conversely, Model 2 grapples with the complexities of the Javanese language, thereby yielding lower overall BLEU scores. Figure 8 illustrates translation errors in Model 2, wherein the word "duka" in Javanese, which should be translated as "marah (Angry)" is rendered as "tidak tahu (Unaware)," consequently affecting the overall meaning of the translation output. Whereas in the second example in Figure 8, the error occurred because the word "ora ganep" which should be interpreted as "sebagian saja (Mostly)" was translated into "ganjil (odd)" so that the translation results were not appropriate.

The translation results in model 1 and model 2 show that considering the linguistic and structural aspects of the language is very important in designing the machine translation model. Special parameters need to be used to improve the model and translation results of the machine translation.



Figure 8. Examples of errors in the translation process

Table 9 presents a comprehensive overview of the key factors that are crucial for assessing machine translation systems. These parameters have a substantial influence on the efficacy of translation, especially in languages with limited resources. When examining languages with limited resources, such as indigenous or less frequently spoken languages, many obstacles arise that intersect with the parameters specified in the table.

For example, when it comes to selecting datasets, languages with limited resources frequently do not have extensive digital text collections, which makes it difficult to find appropriate datasets for training machine translation. Hence, the scarcity of varied datasets can limit the portrayal of linguistic settings beyond a limited range, impeding the capacity to comprehend the complexities of the language. In low-resource language environments, translation operations may face challenges. The absence or inadequate documentation of official translation protocols for many languages poses a significant challenge in maintaining faithfulness to the source text. In addition, the limited availability of proficient translators may lead to translations that exhibit deficiencies in terms of precision or cultural awareness, thereby intensifying concerns regarding the quality of translations.

Table 9. Parameter for evaluating machine translation.

Parameter	Strength	Weakness
Dataset Collection	Utilizing Bible text ensures a rich dataset for analysis	Limited diversity beyond religious texts may restrict representation
Translation procedures	Adhering to official protocols ensures fidelity to the original text	Potential introduction of biases from the original text
Text Processing	Meticulous preprocessing ensures clean data for robust analysis	Requires significant computational resources and time
Mode Training and Testing	Comprehensive training enhances model performance	Risk of overfitting due to extensive training
Evaluation Metrics	BLEU scores provide quantitative measures of translation quality	May overlook nuanced aspects of translation quality
Comparative Analysis	Comparison between models aids in understanding relative performance	Interpretation may require expertise in computational linguistics
Error Analysis	Identification of errors facilitates model refinement	Subjectivity in error interpretation may lead to verified judgments
Visualizations	Visual representation offers insights into model behavior and performance	Interpretation may vary based on individual perspective

Text processing presents distinct difficulties in language situations with limited resources. Thorough data preprocessing is essential for accurate and dependable analysis, but the absence of standardized linguistic resources for languages with limited resources might make this process more complex. The task of assuring data purity and suitability for analysis becomes increasingly complex in the absence of sophisticated text processing tools or linguistic resources. In addition, the process of training and testing models in linguistic environments with limited resources may encounter limitations as a result of the restricted accessibility of annotated training data and computational resources. Working with sparse data

poses challenges for comprehensive training, and the restricted variation in the training dataset may heighten the danger of overfitting.

When assessing machine translation systems for low-resource languages, it may be necessary to modify or enhance evaluation criteria, such as BLEU scores. Conventional measurements may not sufficiently encompass the distinct linguistic features or translation difficulties found in these languages, thereby requiring the creation of alternative assessment frameworks specifically designed for resource-limited environments.

In summary, effectively addressing the factors specified in the table within the framework of low-resource languages necessitates the implementation of inventive methodologies and customized remedies. Subsequent research and development endeavors should prioritize surmounting these obstacles to enhance the accessibility and caliber of machine translation for individuals who speak languages with limited resources

4. Conclusions

In conclusion, assessing machine translation systems for languages with limited resources requires careful consideration of datasets, translation methodologies, text processing methods, model training, evaluation metrics, comparison analysis, error analysis, and visual representations. The above qualities demonstrate the complexity and challenges of translating languages with little resources.

Future research in this subject must focus new solutions to these issues. Transfer learning, which uses pretraining on languages with extensive data to improve translation accuracy in languages with limited resources, has significant promise. Transfer learning may enhance translation accuracy and efficiency by fine-tuning models using smaller datasets from low-resource languages.

Expanding and diversifying databases for languages with low resources is also crucial. This may include collaborating with linguists, community groups, and language experts to collect and analyze texts. Resilient text processing methods created for languages with limited resources may also improve data quality and suitability for analysis.

Future model training and assessment research should focus on adaptive methods that can effectively use sparse data and limited computer resources. More research is needed to establish rating metrics that better reflect translation quality in languages with low resources. Comparative and error analysis are also essential for improving machine translation systems. Further research in these areas may lead to more efficient translation models and techniques.

By overcoming these challenges and using cutting-edge research methodologies, machine translation for languages with limited resources may improve accessibility and quality for speakers. This improves communication and information access and preserves and promotes global language variety.

Acknowledgements

The author declares no conflict of interest. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

[1] B. Klimova, M. Pikhart, A. D. Benites, C. Lehr, and C. Sanchez-Stockhammer, ‘Neural machine translation in foreign language teaching and learning: a systematic review’, *Education and Information Technologies*, vol. 28, no. 1, pp. 663–682, 2023.

[2] S. Ranathunga, E.-S. A. Lee, M. Prifti Skenduli, R. Shekhar, M. Alam, and R. Kaur, ‘Neural Machine Translation for Low-resource Languages: A Survey’, *ACM Comput. Surv.*, vol. 55, no. 11, pp. 1–37, Nov. 2023, doi: 10.1145/3567592.

[3] D. A. Sulistyo, D. D. Prasetya, F. A. Ahda, and A. P. Wibawa, ‘Pivoted Low Resource Multilingual Translation with NER Optimization’, *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, vol. 24, no. 5, pp. 1–16, May 2025, doi: 10.1145/3727876.

[4] D. A. Sulistyo, A. P. Wibawa, D. D. Prasetya, and F. A. Ahda, ‘An enhanced pivot-based neural machine translation for low-resource languages’, *International Journal of Advances in Intelligent Informatics*, vol. 11, no. 2, 2025, Accessed: June 30, 2025. [Online]. Available: http://www.ijain.org/index.php/IJAIN/article/view/2115

[5] D. A. Sulistyo, A. P. Wibawa, D. D. Prasetya, and F. A. Ahda, ‘Indonesian cross-linguistic named entity recognition’, *Research Methods in Applied Linguistics*, vol. 4, no. 3, p. 100236, Dec. 2025, doi: 10.1016/j.rmal.2025.100236.

[6] D. A. Sulistyo, A. P. Wibawa, D. D. Prasetya, and A. Nafalski, ‘Autogenerated Indonesian-Madurese Parallel Corpus Dataset Using Neural Machine Translation’, *Available at SSRN 4644430*, Accessed: Mar. 20, 2024. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4644430

[7] R. Bawden, B. Zhang, L. Yankovskaya, A. Tättar, and M. Post, ‘A Study in Improving BLEU Reference Coverage with Diverse Automatic Paraphrasing’, Oct. 08, 2020, *arXiv*: arXiv:2004.14989. Accessed: Mar. 07, 2024. [Online]. Available: http://arxiv.org/abs/2004.14989

[8] M. Evtikhiev, E. Bogomolov, Y. Sokolov, and T. Bryksin, ‘Out of the bleu: how should we assess quality of the code generation models?’, *Journal of Systems and Software*, vol. 203, p. 111741, 2023.

[9] W. A. Farias, D. M. Melo, L. M. dos Santos, Â. A. de Oliveira, R. L. Medeiros, and Y. K. Silva, ‘Web Scraping as a scientific tool for theoretical reference’, 2024, Accessed: Apr. 03, 2024. [Online]. Available: https://www.researchsquare.com/article/rs-3854342/latest

[10] A. Abodayeh, R. Hejazi, W. Najjar, L. Shihadeh, and R. Latif, ‘Web Scraping for Data Analytics: A BeautifulSoup Implementation’, in *2023 Sixth International Conference of Women in Data Science at Prince Sultan University (WiDS PSU)*, IEEE, 2023, pp. 65–69. Accessed: Apr. 03, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10145369/

[11] C. Biswas, R. Mallick, S. Paul, and D. Mukherjee, ‘Solution to web scraping’, in *2023 11th International Conference on Internet of Everything, Microwave Engineering, Communication and Networks (IEMECON)*, IEEE, 2023, pp. 1–5. Accessed: Apr. 03, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10092327/

[12] A. N. Handayani, H. Herwanto, K. L. Chandrika, and K. Arai, ‘Recognition of Handwritten Javanese Script using Backpropagation with Zoning Feature Extraction’, *Knowledge Engineering and Data Science*, 2021, doi: 10.17977/um018v4i22021p117-127.

[13] Y. Dai, 'Exploring Syntactic Representations in Pre-trained Transformers to Improve Neural Machine Translation by a Fusion of Neural Network Architectures', PhD Thesis, University of Leeds, 2023. Accessed: Apr. 03, 2024. [Online]. Available: https://etheses.whiterose.ac.uk/33962/

[14] U. Farooq, M. S. M. Rahim, N. S. Khan, S. Rasheed, and A. Abid, 'A crowdsourcing-based framework for the development and validation of machine readable parallel corpus for sign languages', *IEEE Access*, vol. 9, pp. 91788–91806, 2021.

[15] S. K. Mondal *et al.*, 'Enhancement of English-Bengali Machine Translation Leveraging Back-Translation', *Applied Sciences*, vol. 14, no. 15, p. 6848, 2024.

[16] G. Rafiq, M. Rafiq, and G. S. Choi, 'Video description: A comprehensive survey of deep learning approaches', *Artif Intell Rev*, vol. 56, no. 11, pp. 13293–13372, Nov. 2023, doi: 10.1007/s10462-023-10414-6.

[17] S. Das, A. Tariq, T. Santos, S. S. Kantareddy, and I. Banerjee, 'Recurrent Neural Networks (RNNs): Architectures, Training Tricks, and Introduction to Influential Research', in *Machine Learning for Brain Disorders*, vol. 197, O. Colliot, Ed., in Neuromethods, vol. 197. , New York, NY: Springer US, 2023, pp. 117–138. doi: 10.1007/978-1-0716-3195-9_4.

[18] A. Baliyan, A. Batra, and S. P. Singh, 'Multilingual Sentiment Analysis using RNN-LSTM and Neural Machine Translation', in *2021 8th International Conference on Computing for Sustainable Global Development (INDIACom)*, Mar. 2021, pp. 710–713.

[19] S. Lee *et al.*, 'A survey on evaluation metrics for machine translation', *Mathematics*, vol. 11, no. 4, p. 1006, 2023.

[20] S. Yang, D. Chen, S. Li, and W. Wang, 'Carbon price forecasting based on modified ensemble empirical mode decomposition and long short-term memory optimized by improved whale optimization algorithm', *Science of the Total Environment*, vol. 716, p. 137117, 2020.

[21] D. Bowman *et al.*, 'The Mathematical Foundations of Epistemic Network Analysis', in *Advances in Quantitative Ethnography*, vol. 1312, A. R. Ruis and S. B. Lee, Eds, in Communications in Computer and Information Science, vol. 1312. , Cham: Springer International Publishing, 2021, pp. 91–105. doi: 10.1007/978-3-030-67788-6_7.

[22] B. Li *et al.*, 'ODE Transformer: An Ordinary Differential Equation-Inspired Model for Neural Machine Translation', Apr. 06, 2021, *arXiv*: arXiv:2104.02308. Accessed: Apr. 03, 2024. [Online]. Available: http://arxiv.org/abs/2104.02308

[23] T. Liu, Q. Fang, W. Ding, H. Li, Z. Wu, and Z. Liu, 'Mathematical word problem generation from commonsense knowledge graph and equations', in *Proceedings of the 2021 conference on empirical methods in natural language processing*, 2021, pp. 4225–4240. Accessed: Apr. 03, 2024. [Online]. Available: https://aclanthology.org/2021.emnlp-main.348/

[24] P. V. Abhiram *et al.*, 'Enhanced Auto Language Prediction with Dictionary Capsule -- A Novel Approach', Mar. 09, 2024, *arXiv*: arXiv:2403.05982. Accessed: Apr. 04, 2024. [Online]. Available: http://arxiv.org/abs/2403.05982

[25] P. Garg, J. Villasenor, and V. Foggo, 'Fairness metrics: A comparative analysis', in *2020 IEEE international conference on big data (Big Data)*, IEEE, 2020, pp. 3662–3666. Accessed: Apr. 03, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9378025/

[26] A. Wang and K. Cho, 'BERT has a Mouth, and It Must Speak: BERT as a Markov Random Field Language Model', Apr. 09, 2019, *arXiv*: arXiv:1902.04094. Accessed: Apr. 03, 2024. [Online]. Available: http://arxiv.org/abs/1902.04094

[27] K. Kaur and S. Chauhan, 'Original Research Article A comparative analysis of lexical-based automatic evaluation metrics for different Indic language pairs', *Journal of Autonomous Intelligence*, vol. 7, no. 4, 2024, Accessed: Apr. 03, 2024. [Online]. Available: http://jai.front-sci.com/index.php/jai/article/view/1393

[28] C. P. Chai, 'Comparison of text preprocessing methods', *Natural Language Engineering*, vol. 29, no. 3, pp. 509–553, 2023.

[29] S. V. Mashtalir and O. V. Nikolenko, 'Data preprocessing and tokenization techniques for technical Ukrainian texts', *Applied Aspects of Information Technology*, vol. 3, no. 6, pp. 318–326, 2023.

[30] I. Gupta, 'Punctuation Restoration as Post-processing Step for Swedish Language Automatic Speech Recognition'. 2023. Accessed: Apr. 03, 2024. [Online]. Available: https://www.diva-portal.org/smash/record.jsf?pid=diva2:1807710

[31] A. Kathuria, A. Gupta, and R. K. Singla, 'A Review of Tools and Techniques for Preprocessing of Textual Data', in *Computational Methods and Data Engineering*, vol. 1227, V. Singh, V. K. Asari, S. Kumar, and R. B. Patel, Eds, in Advances in Intelligent Systems and Computing, vol. 1227. , Singapore: Springer Singapore, 2021, pp. 407–422. doi: 10.1007/978-981-15-6876-3_31.

[32] U. Naseem, I. Razzak, and P. W. Eklund, 'A survey of pre-processing techniques to improve short-text quality: a case study on hate speech detection on twitter', *Multimedia Tools and Applications*, vol. 80, pp. 35239–35266, 2021.

[33] L. Huang, J. Qin, Y. Zhou, F. Zhu, L. Liu, and L. Shao, 'Normalization techniques in training dnns: Methodology, analysis and application', *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 8, pp. 10173–10196, 2023.

[34] S. Supriyono, A. P. Wibawa, S. Suyono, and F. Kurniawan, 'Analyzing Audience Sentiments in Digital Comedy: A Study of YouTube Comments Using LSTM Models', *Journal of Applied Data Sciences*, vol. 5, no. 4, pp. 1877–1889, 2024.

[35] C. Toraman, E. H. Yilmaz, F. Şahinuç, and O. Ozcelik, 'Impact of Tokenization on Language Models: An Analysis for Turkish', *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, vol. 22, no. 4, pp. 1–21, Apr. 2023, doi: 10.1145/3578707.

[36] F. A. Ahda, A. P. Wibawa, D. D. Prasetya, and D. A. Sulistyo, 'Comparison of Adam Optimization and RMS prop in Minangkabau-Indonesian Bidirectional Translation with Neural Machine Translation', *JOIV: International Journal on Informatics Visualization*, vol. 8, no. 1, pp. 231–238, 2024.

[37] T. L. M. Suryanto, A. P. Wibawa, H. Hariyono, and A. Nafalski, 'Comparative performance of transformer models for cultural heritage in NLP tasks', *Advance Sustainable Science Engineering and Technology*, vol. 7, no. 1, pp. 02501015–02501015, 2025.

[38] A. F. Del Carpio and L. B. Angarita, 'Trends in software engineering processes using deep learning: a systematic literature review', in *2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, IEEE, 2020, pp. 445–454. Accessed: Apr. 03, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9226311/

[39] A. R. Lubis, Y. Y. Lase, D. A. Rahman, and D. Witarsyah, 'Improving Spell Checker Performance for Bahasa Indonesia Using Text Preprocessing Techniques with Deep Learning Models.', *Ingénierie des Systèmes d'Information*, vol. 28, no. 5, 2023, Accessed: Apr. 03, 2024. https://doi.org/10.18280/isi.280522

[40] K. Dedes, A. B. P. Utama, A. P. Wibawa, A. Afandi, A. N. Handayani, and L. Hernandez, 'Neural Machine Translation of Spanish-English Food Recipes Using LSTM', *undefined*, 2022, Accessed: Sept. 12, 2022. [Online]. Available: https://www.semanticscholar.org/paper/Neural-Machine-Translation-of-Spanish-English-Food-Dedes-Utama/eafa6d3f83ec945b30ad4ea75925e2799a75eaf3

[41] D. A. Sulistyo, A. P. Wibawa, D. D. Prasetya, and F. A. Ahda, 'LSTM-Based Machine Translation for Madurese-Indonesian', *Journal of Applied Data Sciences*, vol. 4, no. 3, pp. 189–199, 2023.