UNIVERSITAS BUMIGORA

Volume 23 No. 2

MATRIK

Jurnal : Manajemen, Teknik Informatika dan Rekayasa Komputer

e-ISSN 2476-9843
p-ISSN 1858-4144

UNIVERSITAS BUMIGORA UBG 1997

MATRIK

Jurnal : Manajemen, Teknik Informatika dan Rekayasa Komputer
e-issn 2476-9843 | p-issn 1858-4144

Editorial Team

journal.universitasbumigora.ac.id/index.php/matrik/

DOI: https://doi.org/10.30812/matrik.v23i2

PUBLISHED: 2024-03-30

ARTICLES

Matrik: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer
Vol. 23, No. 2, March 2024, pp. 453∼464
ISSN: 2476-9843, accredited by Kemenristekdikti, Decree No: 200/M/KPT/2020
DOI: 10.30812/matrik.v23i2.3465

Implementation of Neural Machine Translation in Translating from Indonesian to Sasak Language

Helna Wardhana, I Made Yadi Dharma, Khairan Marzuki, Ibjan Syarif Hidayatullah
Universitas Bumigora, Mataram, Indonesia

Article Info

Article history:

Received October 12, 2023
Revised March 22, 2024
Accepted March 28, 2024

Keywords:

Corpus Source Collection
Dataset
Neural Machine Translation
Sasak Regional Language

ABSTRACT

Language translation is part of Natural Language Processing, also known as Machine Translation, which helps the process of learning foreign and regional languages using translation technology in sentence form. In Lombok, there are still people who are not very fluent in Indonesian because Indonesian is generally only used at formal events. This research aimed to develop a translation model from Indonesian to Sasak. The method used was the Neural Machine Translation with the Recurrent Neural Network - Long Short Term Memory architecture and the Word2Vec Embedding with a sentence translation system. The dataset used was a parallel corpus from the Tatoeba Project and other open sources, divided into 80% training and 20% validation data. The result of this research was the application of Neural Machine Translation with the Recurrent Neural Network - Long Short Term Memory algorithm, which could produce a model with an accuracy of 99.6% in training data and 71.9% in test data. The highest ROUGE evaluation metric result obtained on the model was 88%. This research contributed to providing a translation model from Indonesian to Sasak for the local community to facilitate communication and preserve regional language culture.

Corresponding Author:

Helna Wardhana, +62 817-463-573
Faculty of Engineering, Computer Science Study Program
Universitas Bumigora, Mataram, Indonesia
Email: helna.wardhana@universitasbumigora.ac.id.

How to Cite:
H. Wardhana, I. M. Yadi Dharma, K. Marzuki, and I. Syarif Hidayatullah, "Implementation of Neural Machine Translation in Translating from Indonesian to Sasak Language", *MATRIK: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer*, Vol. 23, No. 2, pp. 453-464, Mar, 2024.

1. INTRODUCTION

Indonesia is an archipelagic country consisting of various tribes, cultures, religions, and customs. The unitary state of the Republic of Indonesia has one language used to communicate in public life, namely Indonesian, apart from the large diversity of ethnicities, cultures, religions, and customs. Language also influences language as a communication medium in different regions to adapt to the surrounding environment. Regional languages are part of the diversity of language forms in Indonesia, one of which is Sasak, used by tribes on the island of Lombok. Sasak is a well-known language on the island of Lombok because it is known as the mother tongue, making it the language of communication for the majority of the island. In remote areas of Lombok, there are still people who are not very fluent in Indonesian because Indonesian is generally only used at formal events. People still use Sasak in everyday life, so their Indonesian is not fluent. Therefore, language translation is needed using a method that can translate quite accurately [1]. The methods that can be used are Natural Language Processing (NLP) or Machine Translation (MT) [2]. Machine translation is a development in the world of technology that assists the process of learning foreign and regional languages with translation in the form of sentences [3].

Several papers discuss language translation using the Neural Network algorithm method, namely simple Recurrent Neural Network (RNN) and bidirectional RNN, Long Short-Term Memory, such as Cultural research [2], then translating non-formal sentences into formal sentences using a machine translation approach [4], applications of advanced natural language processing for clinical pharmacology [5]. Another research is the implementation of the long short-term memory (LSTM) algorithm to detect hate speech in the 2019 presidential election case [6], while the research we propose is entitled Implementation of Neural Machine Translation in Translation from Indonesian to Sasak Language using NMT with the RNN-LSTM and Word2vec algorithms. Implementing language translation from Indonesian to regional languages is usually carried out using a statistics-based approach [2, 7, 8]. The research results show that the implementation of translation using a statistical approach is still word for word, not sentence, which creates new problems in determining grammar and word order [9]. Using lexical rules to search for basic words in a language consists of a rule process using a morphological approach to word prefixes, suffixes, and insertions so that the basic words can be found. The use of NMT in developing translation models is new and has not been widely developed, especially in regional language translation, which uses artificial neural networks [8]. Research related to Neural Machine Translation was carried out in an experiment using 3000 data, and 20 hours were spent conducting the training process [2]. Research on the application of the Indonesian dictionary to the Sasak language has been carried out to find vocabulary to make it easier for students and tourists in Indonesia to understand the Sasak language [9]. The implementation of the Natural Language Processing (NLP) algorithm and Cosine Similarity is also used to assess essay exams using a comparison of Document Similarity automatically, used to check the level of similarity (Cosine Similarity) using two vectors to measure document similarity with results ranging from 0 to 1 so that get the expected result [10]. So, NLP can also be used to compare the Indonesian and Sasak languages.

Machine Translation has many approaches in facilitating the machine translation design process, including Neural Machine Translation (NMT) [3, 4, 7, 11–14]. NMT is a new approach to translation applications that uses a Recurrent Neural Network (RNN) encoder and decoder architecture to define the translation context and shape modeling [6, 15]. In applying RNN, various model improvements will be made in dealing with translations that experience long-term dependence. Long Short-Term Memory Network (LSTM) is a special type of RNN that is relatively complex but capable of remembering information and learning long-term dependencies. LSTM was implemented to detect hate speech related to the 2019 Presidential Election [6, 15–17]. With the similarity of research in searching for words or sentences in a language, the LSTM method can also be used for this researcher. LSTM also uses data or information from outside the normal flow of the recurrent network in the gate cell. Based on the research title several researchers have carried out, it still has not shown good results in building NMT models from Indonesian to Regional Languages. Therefore, this study aims to implement NMT with the RNN-LSTM architecture and use Embedding Word2vec [8, 10, 18] to build a model for Indonesian to Sasak translation. This research results in a translation model from Indonesian to the Sasak regional language, which can translate Indonesian sentences into Sasak language sentences. This research contributes to providing a translation model from Indonesian to Sasak for the local community or tourists to facilitate communication and preserve regional language culture.

2. RESEARCH METHOD

This research is qualitative. The research method used is Neural Machine Translation (NMT), which applies the RNN-LSTM algorithm and Embedded Word2Vec. The dataset used is a corpus dataset. The steps/stages in this research are:

2.1. Dataset

In this study, the dataset that will be used as input into the neural network architecture is a text dataset in the form of a parallel corpus. The parallel corpus is a pair of sentences in the source and target languages (Indonesian-Sasak). Making the corpus in the target language is done manually using the 2017 edition of the Big Sasak Language Dictionary as a standardization of the meaning of each vocabulary. The dataset in the source language used by the researcher was obtained from the Tatoeba Project and other open sources with a total of 1000 data. The dataset in this study is divided into 2: 1) Training Data and 2) Data Validation. Meanwhile, after the data is collected, it will enter the preprocessing stage.

2.2. Preprocessing

At the preprocessing stage, several methods and techniques are used in data processing to clean the data before it is used for training and testing the model. Figure 1 shows the stages in carrying out preprocessing while the first stage is 1) Cleaning, which is carried out to remove unwanted characters from the dataset. The cleaning process in this study was carried out in Microsoft Excel. 2) Lowercase: This study applies lowercase to all corpus datasets so that all datasets used in the next stage use lowercase letters. 4) Tokenization: this stage is carried out to separate sentences into individual words that will be used as input for the model. 5) Padding: the data used is in the form of sentences from the source language and target language, which do not always have the same sentence length, so the use of padding aims to fill in blank values in the data so that they have the same length. The pad_sequences function from the keras. The preprocessing sequence library is used to pad sequence data.

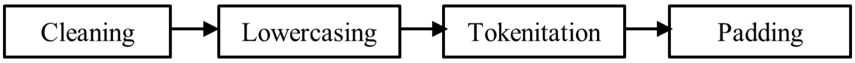

Figure 1. Preprocessing Stage

2.3. Design Analysis

At this stage, design analysis is carried out to make a translation model from Indonesian to Sasak with several RNN-LSTM architectural designs such as Flowcharts, Network Architectures, Training Flowcharts, and Testing, as follows. Figure 2 shows the research method stage.

The model flow chart above describes the general steps in creating a model. After the whole process has been carried out, the final output is an evaluation of the model from the test results. Data collection, analysis, and preprocessing methods were discussed in the Needs Analysis section of the previous sub-chapter. This section will explain how to embed the word to the end.

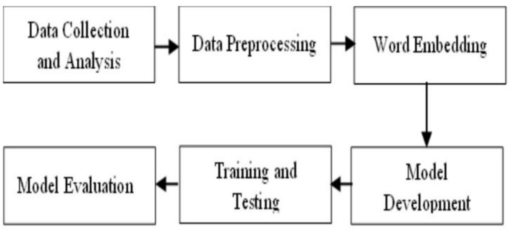

Figure 2. Research Method Stage

1. Word Embedding

In this process, word embedding uses the Word2Vec embedding model built using the Deeplearning4J framework. The Word2Vec model formation method stages start from Load & Vectorize Sentences, then Building the Model, and then Fitting Word2Vec, as depicted in Figure 3.

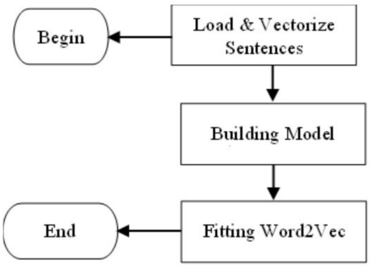

Figure 3. Word2Vec Model

2. Model Building

In the model creation section, some steps were carried out before the formation of the NMT translation model. The NMT model built is shown in the following Figure 4.

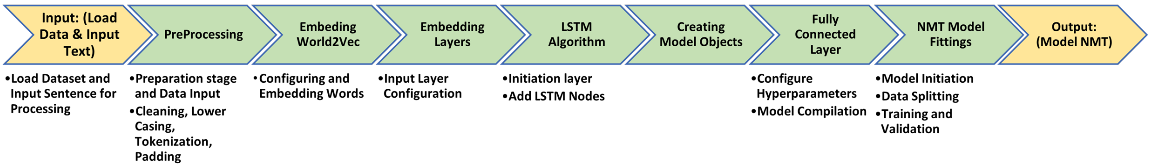

Figure 4. Neural Machine Translation (NMT) Models

Figure 4 is the NMT translation model with the LSTM architecture. The translation model was built using Google tools, namely Google Colab and the use of IntelliJ IDEA in building the Word2Vec embedding. The working stages of the model in translating input sentences can be seen in the following figure.

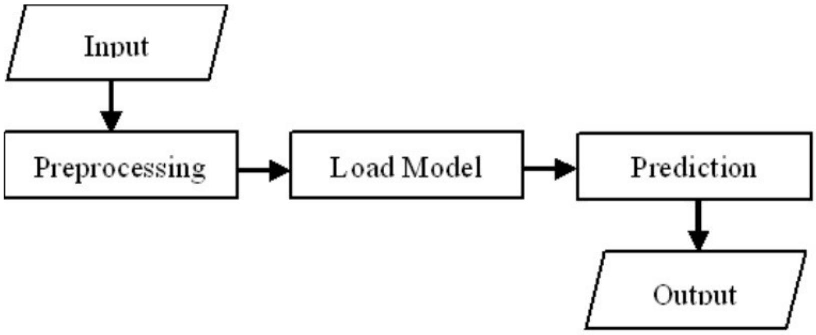

Figure 5. Workflow Models

Figure 5 shows the workflow model for getting prediction results, starting from Preprocessing, Loading the Model, and getting the Prediction results.

3. Training and Testing

The Training and Testing Process stages start with Model Initialization, determining the epoch value, in this case, the epoch value = 0, and then determining the num_epoch value = 100. The next stage of the testing process in repetition is carried out (epoch <= num_epoch). As long as the epoch value is smaller than the num_epoch run, the testing model will consist of train_loader, test, and loader. Then, the increment epoch value (epoch++). If num_epoch is greater than epoch, then the testing process is stopped. The stages of the training and testing process are depicted in Figure 6.

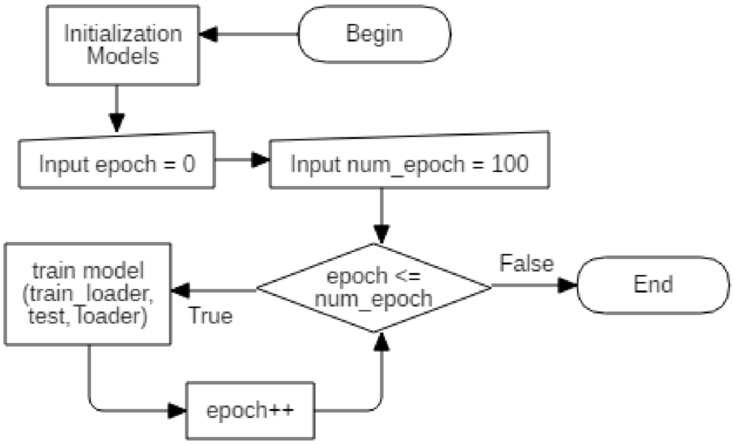

Figure 6. FlowChart of Training in Each Epoch

The NMT model is used to test the data at the initialization stage. The flow diagram of the iteration process for each training and testing batch can be depicted in Figure 7. Starting from determining the size of the data group (batch_size), repeated testing of the train and test values, and then getting feedforward and validation of loss and validation accuracy values for all batch sizes fulfilled.

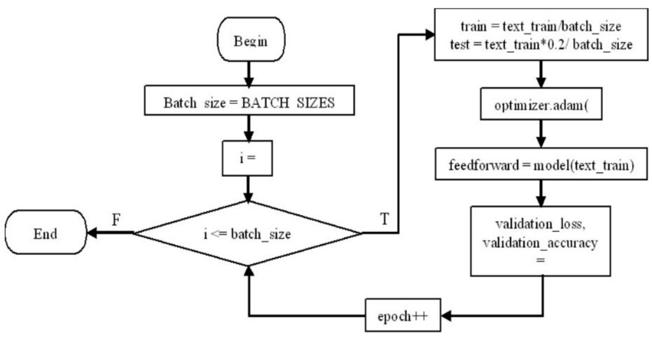

Figure 7. Flowchart Iteration in Each Batch

After the modeling process is complete, the next process is to train the model. The training process can be seen in the flowchart image above. The number of iterations during training is the batch_size value, i.e., the number of training sections divided by batch_size. The first process in training and iterating the target data is divided into batches with the size batch_size. The model is trained using the optimizer and a predefined loss function for 100 epochs (training iterations). The model performs feedforward and backpropagation for each epoch in each batch and updates the model parameters based on the calculated gradients. Finally, outputting metrics to measure model performance in each epoch, validation loss, and validation accuracy will be calculated using validation data retrieved from validation_split percent training data.

4. Model Testing

After all design stages have been completed, proceed to the testing phase, namely making predictions using a 20% random validation test dataset to see the model's success in translating into the target language to be implemented.

5. Evaluation

At the evaluation stage, researchers will evaluate the performance of the translation model developed using the ROUGE quantitative analysis method. Evaluation is done by calculating each trial's precision, recall, and F1-score on the new data.

3. RESULT AND ANALYSIS

The test results use predictions from all experiments to assess the model's ability to translate from the source language to the target language.

3.1. Datasets

Datasets consist of raw data that has been prepared but has not been converted into a format that will be used as a reference. In this case study, the data used is 1000 text data in the form of a parallel corpus of Indonesian as Source language sentences and Sasak as Target language sentences; examples of raw data are used in Table 1.

Table 1. Parallel Corpus Dataset

No	Source language sentences	Target language sentences
1	aku masih mencoba selama ini.	aku masi nyobaq selaĕq ne
2	acara	acare
3	acara apa itu	acare ape tie
4	acara hari senin besok	acare jelo senĕn lĕmaq
5	Acara itu dilaksanakan.	acare ino tegawĕq
6	acara saya besok senin	acare aku lmaq senĕn
7	Acara seminar ditutup dengan doa	acare seminar tetutup isiq doa
8	Acara yang bagus.	acare siq solah

3.2. Preprocessing Results

At this stage, the text data is processed into a matrix form, with the value of each word in the sentence being an integer, representing the source sentence and the target sentence. The stages of data processing results are as follows.

1. Cleaning and Lowercasing

The Cleaning and Lowercasing process is carried out to clean the data from unwanted characters in the dataset and change all text to lowercase. Table 2 is an example of data cleaning and letter reduction.

Table 2. The Results of The Cleaning & Lowercasing Process

Before	After
Acara apa itu	acare ape tie
Acara hari senin besok!	acare jelo senĕn lĕmaq
Acara itu dilaksanakan.	acare ino tegawĕq
Acara saya besok senin	acare aku lĕmaq senĕn
Acara seminar ditutup dengan doa	acare seminar tetutup isiq doa
Acara yang bagus.	acare siq solah
Acaranya bagus	acaren solah
Ada apa	araq ape

2. Splitting Data

Table 3 is an example of data that has not undergone a data separation process, so data in Indonesian and Sasak are still written side by side and have not been given parameters.

Table 3. Before Splitting

No.	Indonesian Language Sentences	Sasak Language Sentences
1	dia sangat terkenal	iye paling terkenal
2	dia sedang mencuci mobil	ie kenyengke bisoq montor
3	dia sedikit berkarat	ie sekediq bekarat
4	dia segera berjalan pergi	ya aru lalo lampak
5	dia segera pergi	ya aruan lalo
6	dia seharusnya datang	ya mule datang
7	dia selalu bercanda	ya girang bejorak
8	dia sendiri mencobanya	ya mesak nyobaqan ne
9	dia seorang ahli jantung	iye dengan ceket leq sedoq
10	dia seorang mekar yang terlambat	ie tahu mekar saq telat
11	dia sepertinya mengenal kita	ie kenal ite jage
12	dia sepertinya sakit	ie sakit rue ne
13	dia sepertinya sakit	ie sakit jage
14	dia sering datang terlambat	ie girang dateng telat
15	dia seukuran saya	ie seukuran aku
16	dia siap bekerja	ie siep begawean

Table 4 is an example of data that has undergone a separation process so that data in Indonesian and Sasak have been grouped in their respective columns and given parameters.

Table 4. After Splitting

Input Sentence	Output Sentence	Output Sentence Input
aku masih mencoba selama ini.	aku masi nyobaq seleaq ne $< eos >$	$< eos >$ aku masi nyobaq seleaq ne
Acara	acare $< eos >$	$< eos >$ acare
Acara	acare $< eos >$	$< eos >$ acare
acara apa itu	acare ape tie $< eos >$	$< eos >$ acare ape tie
acara hari senin besok	acare jelon senen lemaq $< eos >$	$< eos >$ acare jelon senen lemaq
acara itu dilaksanakan.	acare ino tegaweq $< eos >$	$< eos >$ acare ino tegaweq
acara saya besok senin	acare aku lemaq senen $< eos >$	$< eos >$ acare aku lemaq senen
acara seminar ditutup dengan doa	acare seminar tetutup isiq doa $< eos >$	$< sos >$ acare seminar tutup isiq doa
acara yang bagus	acare siq solah $< eos >$	acare siq solah $< eos >$
acaranya bagus	acaren solah $< eos >$	$< sos >$ acaren solah

3. Tokenization and Padding Results

Table 5 is an example of the tokenization process of changing a sentence into a set of tokens, where the token is the smallest part of the string that a machine can process. In this case, each word in the sentence will be converted into a unique number representing that word.

Table 5. Tokenization Results

Sentence Before Tokenization (text)	Sentence Before Tokenization (text)
aku masih mencoba selama ini.	[6.7. 8. 9. 10]
Acara	[1]
Acara	[1]
acara apa itu	[1. 11. 2]
acara hari senin besok	[1. 12. 3. 4]

The padding stage is carried out to achieve the same and consistent data size in matrix form, then batch data is processed efficiently, and important information is stored. The following is an example of data on the padding process starting from before and after the application of padding, shown in Table 6.

Table 6. Padding Results

Sentence Before Tokenization (text)	Before Padding	After Padding
aku masih mencoba selama ini.	[6. 7. 8. 9. 10]	[6. 7. 8. 9. 10]
Acara	[1]	[0. 0. 0. 0. 1]
Acara	[1]	[0. 0. 0. 0. 1]
acara apa itu	[1. 11. 2]	[0. 0. 1. 11. 2]
acara hari senin besok	[1. 12. 3. 4]	[1. 12. 3. 4]

3.3. Results of Training and Testing

The results of training and testing in this study are displayed in several trials to see the level of accuracy of the evaluation metrics that are applied according to Hyperparameter tuning, which is done by configuring each experiment in the model-building process to find the right combination of Hyperparameter values to give the best results in training in Table 7.

Table 7. Training Results

Test	Hyparameter Values					Training Accuracy (%)	Testing Accuracy (%)
	epoch	learning rate	optimizer	dropout	dataset		
1	10	0.01	*Adam*	-	600	77.50%	68.40%
2	30	0.1	*RMSprop*	0.2	600	77.60%	72.50%
3	150	0.01	*RMSprop*	-	1000	99.60%	71.90%
4	100	0.001	*RMSprop*	0.5	1000	73.70%	67.50%
5	20	0.01	*Adam*	0.5	1000	99.30%	71.90%
6	100	0.01	*RMSprop*	0.1	1000	99.60%	71%

3.4. Test Result

Based on the results of the training that has been done in Table 7, with a combination of hyperparameter tuning settings, the best results are taken with an accuracy above 70%, namely experiment 3, experiment 5, and experiment 6 as automatic evaluation samples with new data using the ROUGE evaluation metric. The final test results of the samples taken to evaluate the performance of the model in terms of recall (r), precision (p), and f1-score (f) in Table 8.

Table 8. Test Result

	Translation results								
Test 3	**Input**			**Actual**			**Translate**		
	dia ingin pulang ke rumah			ie mĕlĕ ulĕq ojok balĕ			iĕ mĕlĕ lalo ojok peken		
	Evaluation Result								
	Rouge 1			Rouge 2			Rouge-L		
	r	p	f	r	p	f	r	p	f
	1	0.8	0.88888	1	0.75	0.85714	1	0.8	0.88888
	Translation Results								
Test 5	**Input**			**Actual**			**Translate**		
	dia ingin pulang ke rumah			ie mĕlĕ ulĕq ojok balĕ			ye lalo jok balĕ nu		
	Evaluation Result								
	Rouge 1			Rouge 2			Rouge-L		
	r	p	f	r	p	f	r	p	f
	1	0.75	0. 85714	1	0.67	0.79999	1	0.75	0. 85714
	Translation results								
Test 6	**Input**			**Actual**			**Translate**		
	apa kamu suka dia			ape side demen ie			ape side demen nulis		
	Evaluation Result								
	Rouge 1			Rouge 2			Rouge-L		
	r	p	f	r	p	f	r	p	f
	1	0.5	0.66666	0	0	0	1	0.5	0.66666

3.5. Model Evaluation

Testing the translation model above is carried out using input sentences outside the dataset to determine the model's performance and ensure that the model can produce good and quality translations in various data conditions. Table 9 provides varying results in the three experiments, so a detailed evaluation can be found. A comparative model analysis process was carried out on the ability to translate sentences in their complete form as follows.

Experiment 3 in Table 9 has the best evaluation results, with the highest F1 score of 88% of all ROUGE evaluation metrics measured. Findings from the results of experiment 5, the NMT test results show a slight decrease in performance (Table 8), namely in ROUGE-1 results with a decrease in precision level of 5% and F1-Score of 3% and in ROUGE-2 with a decrease in precision of 8% and f1 -score of 6%, while for ROUGE-L with a decrease in precision of 5% and f1-score of 3%. Meanwhile, Experiment 6 showed a significant decrease in precision and F1 score in ROUGE-1 with a decrease in precision of 3% and f1 score of 22%, while in ROUGE-2, there was the largest decrease.

Table 9. Evaluation Result

Evaluation	Results
Test 3	ROUGE-1: Evaluation results show recall of 1.0, precision of 0.8, and F1-score of 0.88888. ROUGE-2: Evaluation results show recall of 1.0, precision of 0.75, and F1-score of 0.85714. ROUGE-L: Evaluation results show recall of 1.0, precision of 0.8, and F1-score of 0.88888. ROUGE-L: measures the similarity between references and translation results using the Longest Common Subsequence (LCS), which considers word order. The evaluation results show a high level of similarity between the reference and the translation results.
Test 5	ROUGE-1: Evaluation results show recall of 1.0, precision of 0.75, and F1-score of 0.85714. Although recall remained perfect, precision and F1-score decreased slightly compared to Experiment 3, indicating some differences in the unigrams found between the reference and translated results. ROUGE-2: Evaluation results show recall of 1.0, precision of 0.67, and F1-score of 0.79999. In the case of bigrams, precision and F1-score decreased further compared to Experiment 3 in model testing. ROUGE-L: Evaluation results show recall of 1.0, precision of 0.75, and F1-score of 0.85714. The evaluation results show the same level of similarity as ROUGE-1, so it can be concluded that the model built in Experiment 5 is still less good than the model in Experiment 3.
Test 6	ROUGE-1: Evaluation results show recall of 1.0, precision of 0.5, and F1-score of 0.66666. Although recall remains perfect, precision and F1-score decrease drastically compared to previous experiments. ROUGE-2: Evaluation results show recall of 0, precision of 0, and F1-score of 0. The ROUGE-2 metric does not show any bigram similarities between the reference and the translation results. ROUGE-L: Evaluation results show recall of 1.0, precision of 0.5, and F1-score of 0.66666. The evaluation results show the same level of similarity as ROUGE-1 but with lower precision.

This research results in a translation model from Indonesian to the Sasak regional language. The previous model was an Indonesian-to-Sasak translation dictionary application, which was only able to translate word by word [9], while the model we propose is able to translate Indonesian sentences into Sasak sentences. As an initial researcher who carried out translation research from Indonesian to Sasak Regional Language, of course there were several obstacles and challenges found during this research, namely the lack of references to previous research in the same domain, difficulties in establishing or determining hyperparameters to find the best combination values. Both are used to get translation results with maximum accuracy. Apart from that, there are obstacles to using certain datasets, even though a good dataset complies with the Tatoeba Project standards.

4. CONCLUSION

Through the research that has been carried out, it can be concluded that the implementation of NMT to the Indonesian to Sasak language translation model can provide quite good results, especially in experiment 3, which is always constant based on the metric values produced. Experiment 3 had the best evaluation results, with the highest F1-score value of 88% of all ROUGE evaluation metrics measured. Meanwhile, experiment 5 showed a slight decrease in performance, namely in ROUGE-1 results with a decrease in the precision level of 5% and F1-Score of 3% and in ROUGE-2 with the precision of 8% and f1-score of 6%, while ROUGE-L with decreased precision by 5% and f1-score by 3%. Meanwhile, Experiment 6 showed a significant decrease in precision and F1-score in ROUGE-1 with a decrease in precision of 3% and f1-score of 22%, while ROUGE-2 had the largest decrease. Further research developments can be carried out in the form of adding a corpus dataset with a wider range of words, setting hyperparameter tuning to find more appropriate combinations, and applying morphological and syntactic rules to support grammatical translation.

5. ACKNOWLEDGEMENTS

We want to thank everyone who directly or indirectly helped us to finish this paper.

6. DECLARATIONS

AUTHOR CONTIBUTION

The first and second authors coordinated the researchers in analyzing the corpus dataset and developing the model. The third and fourth authors were responsible for building the model using the RNN-LSTM and Word2vec algorithms.

FUNDING STATEMENT

This study did not receive grants from anywhere, including the public, commercial, or non-profit sectors.

COMPETING INTEREST

The authors declare that they have no competing financial interests or personal relationships that could have influenced the research in this paper.

REFERENCES

[1] V. L. Kane, "Interpretation and machine translation towards google translate as a part of machine translation and teaching translation," *Applied Translation*, pp. 10–17, 2021, https://doi.org/10.51708/apptrans.v15n1.1337.

[2] I. G. A. Budaya, M. W. A. Kesiman, and I. M. G. Sunarya, "Perancangan Mesin Translasi berbasis Neural dari Bahasa Kawi ke dalam Bahasa Indonesia menggunakan Microframework Flask," *Jurnal Sistem dan Informatika (JSI)*, pp. 94–103, 2022.

[3] N. Rezaputra and Y. Denny, "Alih Bentuk Kalimat Non-Formal Menjadi Kalimat Formal Menggunakan Pendekatan Machine Translation," vol. 8, no. 1, pp. 379–390, 2022.

[4] N. Rezaputra and Y. D. Prabowo, "Alih Bentuk Kalimat Non-Formal Menjadi Kalimat Formal Menggunakan Pendekatan Machine Translation," *KALBISIANA Jurnal Sains, Bisnis dan ...*, vol. 8, no. 1, pp. 379–390, 2020.

[5] J. C. Hsu, M. Wu, C. Kim, B. Vora, Y. T. Lien, A. Jindal, K. Yoshida, S. Kawakatsu, J. Gore, J. Y. Jin, C. Lu, B. Chen, and B. Wu, "Applications of Advanced Natural Language Processing for Clinical Pharmacology," *Clinical Pharmacology and Therapeutics*, vol. 0, no. 0, pp. 1–9, 2024, https://doi.org/10.1002/cpt.3161.

[6] A. S. Talita and A. Wiguna, "Implementasi Algoritma Long Short-Term Memory (LSTM) Untuk Mendeteksi Ujaran Kebencian (Hate Speech) Pada Kasus Pilpres 2019," *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 19, no. 1, pp. 37–44, 2019, https://doi.org/10.30812/matrik.v19i1.495.

[7] Z. Abidin, "Statistical Machine Translation Pada Bahasa Lampung Dialek Api Ke Bahasa Indonesia," vol. 4, pp. 519–528, 2020, https://doi.org/10.30865/mib.v4i3.2116.

[8] Y. Fauziyah, R. Ilyas, and F. Kasyidi, "Mesin Penterjemah Bahasa Indonesia-Bahasa Sunda Menggunakan Recurrent Neural Networks," *Jurnal Teknoinfo*, vol. 16, no. 2, p. 313, 2022, https://doi.org/10.33365/jti.v16i2.1930.

[9] L. E. Haris and A. S. Pardiansyah, "Aplikasi Android Kamus Bahasa Indonesia-Sasak," *Jurnal Manajemen Informatika dan Sistem Informasi*, vol. 1, no. 1, p. 1, 2018, https://doi.org/10.36595/misi.v1i1.10.

[10] D. O. Sihombing, "Implementasi Natural Language Processing (NLP) dan Algoritma Cosine Similarity dalam Penilaian Ujian Esai Otomatis," *Jurnal Sistem Komputer dan Informatika (JSON)*, vol. 4, no. 2, p. 396, 2022, https://doi.org/10.30865/json.v4i2.5374.

[11] S. Yang, Y. Wang, and X. Chu, "A Survey of Deep Learning Techniques for Neural Machine Translation."

[12] R. A. J. Dabre, C. Chu, and A. Kunchukuttan, "99 A Survey of Multilingual Neural Machine Translation," vol. 53, no. 5, 2020, https://doi.org/10.1145/3406095.

[13] Z. Abidin, "Penerapan Neural Machine Translation untuk Eksperimen Penerjemahan secara Otomatis pada Bahasa Lampung Indonesia," *Prosiding Seminar Nasional Metode Kuantitatif*, no. 978, pp. 53–68, 2017.

[14] A. Burhanuddin, A. L. Qosim, and R. Amaliya, “Phrase Based and Neural Network Translation for Text Transliteration from Arabic to Indonesia,” *Matics*, vol. 14, no. 1, pp. 13–17, 2022, https://doi.org/10.18860/mat.v14i1.13853.

[15] C. Avci, B. Tekinerdogan, and C. Catal, “Analyzing the performance of long short-term memory architectures for malware detection models,” *Concurrency and Computation: Practice and Experience*, vol. 35, no. 6, p. 1, 2023, https://doi.org/10.1002/cpe.7581.

[16] P. F. Muhammad, R. Kusumaningrum, and A. Wibowo, “ScienceDirect ScienceDirect Sentiment Analysis Using Word2vec And Long Short-Term Sentiment Analysis Using Word2vec And Long Short-Term Memory (LSTM) For Indonesian Hotel Reviews Memory (LSTM) For Indonesian Hotel Reviews,” *Procedia Computer Science*, vol. 179, no. 2020, pp. 728–735, 2021, https://doi.org/10.1016/j.procs.2021.01.061.

[17] A. Sherstinsky, “Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network,” *Physica D*, vol. 404, p. 132306, 2020, https://doi.org/10.1016/j.physd.2019.132306.

[18] T. Adewumi, F. Liwicki, and M. Liwicki, “Word2Vec : Optimal hyperparameters and their impact on natural language processing downstream tasks,” pp. 134–141, 2022.

[This page intentionally left blank.]