Building of Informatics, Technology and Science (BITS)
Volume 4, No 3, Desember 2022 Page: 1438−1446
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v4i3.2614

Implementation of Neural Machine Translation for English-Sundanese Language using Long Short Term Memory (LSTM)

Teguh Ikhlas Ramadhan[1,*], Nur Ghaniaviyanto Ramadhan[2], Agus Supriatman[1]

[1]Fakultas Teknik, Teknik Informatika, Universitas Perjuangan, Tasikmalaya, Indonesia
[2]Fakultas Informatika, Rekayasa Perangkat Lunak, Institut Teknologi Terkom, Purwokerto, Indonesia
Email: [1,*]Teguhikhlas@unper.ac.id, [2]Ghani@ittelkom-pwt.ac.id, [3]Agussupriatman@unper.ac.id
Email Penulis Korespondensi: Teguhikhlas@unper.ac.id
Submitted: **02/12/2022**; Accepted: **30/12/2022**; Published**: 30/12/2022**

Abstract−In this modern era, machine translation has been used all over the world for solving humankind's problems such as it deals with language. There are many purposes for using machine translation such as learning another language, communicating, finding a certain or better word to use, and even writing something in a book, or another article Machine translation also used by people who want to translate their native language into their foreign language. The international language being used is the English language. The input of it is a word or a sentence from the source language and it will be translated into another language. Several methods have been conducted to do the machine translation task such as the statistical approach and the neural approach In terms of Sundanese machine translation, there are several methods or several approaches that other researchers have conducted. However the study about Sundanese machine translation, none of the research conducted the English into Sundanese language.Whereas In 5 years before there are approximately 2,67 million average from all over the world to come to West Java and approximately 12163 people who end up stays in West Java region. In this study using the encoder and decoder LSTM architecture achieve a good result regarding building a model for machine translation task. The performance of this model has achieved 0.99 accuracies in both training and testing as well as less than 0.1 loss value to both training and testing data. This model also achieves more than 0.8 average BLEU score for both training and testing data.

Keywords: Neural Machine Translation; Recurrent Neural Network; Encoder Decoder Long Short-Term Memory (LTSM)

1. INTRODUCTION

West Java is one of the most populous provinces in Indonesia. There are approximately 49 million people who live in West Java in the data statistics on 2022 by the West Java government [1]. The people in West Java are using the Sundanese language to communicate.

In this modern era, machine translation has been used all over the world for solving humankind's problems such as it deals with language. Machine translation is almost used by people who want to translate their native language into their foreign language. The international language being used is the English language. Many tourists come to West Java which is the origin of the Sundanese language. In 5 years before there is approximately 2,67 million [2] average from all over the world came to West Java and approximately 12163 people who end up staying in the West Java region. That is the reason why this study builds a machine translation model from English to Sundanese which is still rarely found in the current study.

Machine translation is the task to translate a source language to another language. The input of it is a word or a sentence from the source language and it will be translated into another language. There are many purposes for using machine translation such as learning another language, communicating, finding a certain or better word to use, and even writing something in a book or another article. Several methods have been conducted to do the machine translation task such as the statistical approach and the neural approach. The statistical approach mostly uses the word-to-word or sentences to perform each word translation based on features and statistics of a word for a certain language. Neural-based approaches are using deep learning methods to perform machine translation. Most of the model is a Recurrent Neural Network modified into a Long Short Term Memory model (LSTM) or Gated Recurrent Unit (GRU). There are two types of neural machine translation approaches based on the form of the data input. The first is word-based neural machine translation, and the second is a character-based machine translation. The neural-based machine translation is improved by some attention mechanism which is the based mechanism for the transformer model. This study will be focused on neural machine translation with a simple LSTM method since it was good enough to perform machine learning tasks from another language also.

In terms of Sundanese machine translation, there are several methods or several approaches that other researchers have conducted. For example, Suryani et. al. use a phrase-based statistical approach [3] with the help of PoS Tag information regarding the Sundanese word. This statistical approach also can be enhanced with some monolingual corpus methods like in the research of Darwis et. al. [4]. There is some research that instead of using a statistical approach used the neural machine translation or using deep learning approach which is the state of the current machine translation task according to Yang et. al. survey [5]. Yustiana et. al. [6] using the Recurrent Neural Network based to do the machine translation from Indonesian to Sundanese language. The result that the study achieves a great result with GRU and attention-based model.

Regarding the study about Sundanese machine translation, none of the research conducted the English into the Sundanese language. The purpose of this research is to build English Sundanese machine translation with a neural approach or deep learning approach because that is the state of the art of the current method in the machine translation

Building of Informatics, Technology and Science (BITS)
Volume 4, No 3, Desember 2022 Page: 1438−1446
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v4i3.2614

task. Yustiana et. al. [6] has achieved good result in research before so this study will be applied and expected a good result as well.

2. RESEARCH METHODOLOGY

2.1 Research Stages

Figure 1. Data preparation is to get the data and convert text into numerical so the model can process it.

Building the model is preparation for how the model can process the text and the model can train so the model can solve the translation problem. Evaluation is to evaluate the result that has been yielded by the model. This study will be going through the data preparation, building the model, and evaluation of the model see Figure 1.

2.1.1 Data Preparation

The data for this study are 20000 sentences total data from both languages which are English as a source language and Sundanese as a target language. Table I is an example of sentences from both languages. There are several counts of words in each sentence, from just one word to more than 20 words. This text data must be going through the data preprocessing [7],[8] because the model cannot read words however it must be represented as a vector to process it.

Table 1. An example of the dataset is a sentence from the source language (English) and target language (Sundanese). The number of words contained in every sentence is various.

English	Sundanese
a Royal teacher	Jadi guru Royal
Happy Birthday!	Wilujeng tepang taun!
significant	penting
jungle	leuweung
Can I have a room again	Dupi abi tiasa nampi
next month	kamar deui bulan payun ?

First, the sentences must be converted into lowercase, and then the tokenization method. Tokenization is the task to convert sentences into words [9]. Table 2 is the example of tokenization which is to convert all of the sentences into words and then the words will be used to make the vocabulary. Vocabulary will later be useful as the reference to encoding or vectorizing the input.

Table 2. Tokenization example for both target language and source language dataset. Tokenization is a process to change sentences into tokens. The token will be used later to build the vocabulary

English		Sundanese	
Sentence	Token	Sentence	Token
I go to school	I, go, to, school	Abdi berangkat ka sakola	Abdi, berangkat, ka, sakola

The vectorize method for this study is using the simple integer encoding method or ordinal encoding method [10]. With integer encoding, the words in the vocab are labeled with the integer from one to the size of the vocab itself (Table 3). Then the word that has already been labelled with the integer number is used as a reference to make the sequence encoder.

The purpose of the encoding is to make each sentence from the data to be converted into the vector. For example, in Table

Table 3. Integer encoding example in target and source tokens. Every token is change to different integer number.

English		Sundanese	
Word	Integer	Word	Integer
i	1	anu	1
go	2	itu	2
to	3	ka	3
school	4	kamari	4

Table 4 each of every sentence is converted into the vector with using integer encoding as a reference to convert each word in the sentence. The length of a vector is different between source data and target data, therefore using neural method it will make the input more flexible.

Building of Informatics, Technology and Science (BITS)
Volume 4, No 3, Desember 2022 Page: 1438−1446
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v4i3.2614

Table 4. Encode sentences into integer encoding sequences. This is the final process of changing every sentence to a vector so the method can process it to make a translation model.

English		Sundanese	
Sentences	Encoding	Sentences	Encoding
I go to school	[1, 2, 3, 4, 0, …]	Abdi bade ka sakola	[24, 12, 17, 5, 0, …]
I run	[1, 80, 0, ….]	Abdi lumpat	[1, 10, ….]

After the data has been vectorized it must be split into a training set and a testing set. The training set is used to train the model, and the test set is used to test the model and make the evaluation of whether using this neural model is already a good or bad model. Table 5 show the portion and the total number of data used for training, validation, and testing data. The purpose of validation data is to validate the model while in the training process. It will be simulated as a test set however in the training process [5].

Table 5. Data is split into training, validation, and testing. Training data will be used to train the model, validation data will be used to validate the data before using the testing data to prevent overfitting. The testing data will be used to test and evaluate the model.

	Training	Validation	Testing
Portion	0,8	0,1	0,1
Total	16000	2000	2000

2.1.2 Building the Model

This study will be using the modified Recurrent Neural Network as a model which is the Long Short Term Memory (LSTM) [11], [12]. LSTM model is good to capture the information of a sequence since its capability to store some of the memory so it will solve the vanishing gradient problem which occurs in plain Recurrent Neural Network [13]. The input of the model will be vectorised English sentence and the output will also be the vectorised Sundanese sentence (Figure 2).

The RNN architecture of the neural networks model is using many-to-many architecture [14],[15]. Figure 3 shows the common RNN architecture for different inputs and outputs. The architecture follows different tasks. For example, in the sentiment analysis task which the input is sequence and the output is just one label the architecture will be using the many-to-one and so on.

Since neural machine translation inputs sentences with more than one length vector as well as the output. So, it will be

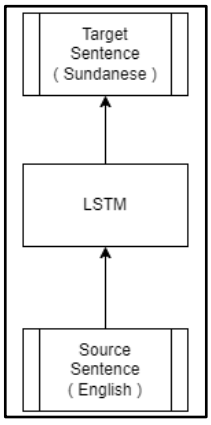

Figure 2. Model input is the source sentence (English) that will going through or processed with LSTM which is the neural-based approach method, and the output is the target sentence (Sundanese)

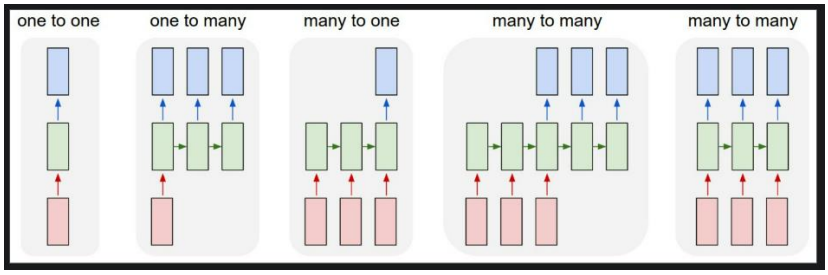

Figure 3. Common RNN architecture is conducted in various types of machine learning tasks. This study will be using many-to-many architecture because the input source sentences (one or more tokens) and the output is also a sentence (one or more tokens)

Building of Informatics, Technology and Science (BITS)
Volume 4, No 3, Desember 2022 Page: 1438−1446
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v4i3.2614

The using the many-to-many architecture with different vector sizes in input and output. The details of the input and output can be seen in Figure 4 with the word in the sentence example.

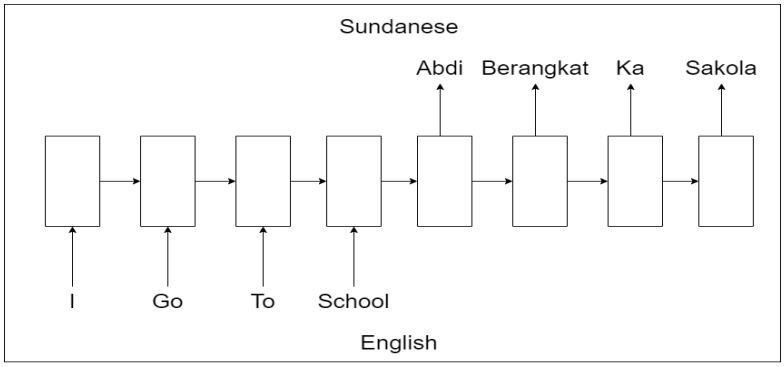

Figure 4. Model input is the sentences that contain one or many words or tokens in the English language, and then map or processed to the output which is Sundanese words or tokens. The number of words in input and output can be the difference because the translation is not always as simple as a word to word translation.

This model will be compiled with the system specification as shown in Table 6. The hardware specification is mostly using the latest hardware and the operating system as well. For software, specifications are using a python programming language in the Anaconda Jupyter Notebook application with Tensorflow Keras deep learning model.

2.1.3 Evaluation

The purpose of the evaluation is to know if the model that we have created is already good or not good enough. There are two approaches to the evaluation of the machine translation model which are the automatic approach and the manual approach. The manual approach is simply using human experts in the language field to evaluate the result of the model [16]. The automatic approach is using a

Table 6. System specification to compile the model later. Many deep learning tasks require a high specification to process a lot of data. This study will be processing 16000 of data in form of encoding sentences.

Processor	1 lth Gen Intel® Corel(TM) I7-1165G7 @2.8GHz 2.80 GHz
RAM	16.0 GB
Storage	512 GB SSD
Programming Language	Windows 11 Home Single Language
Software	Anaconda Jupyter Notebook
Model Library	Tensorflow Keras

Certain calculation based on the model result for example using the Bilingual Evaluation Understudy (BLEU) score [17]. This study will be using the automatic approach which is using the BLEU score to evaluate the machine translation model.

BLEU score was calculated using n-gram matching between the result of the model (candidate) against the references. BLEU score value lies between 0 and 1 where 0 is not related at all and 1 is most related between all the references. This study just uses one as a reference and will be evaluated training data and testing data using 1-grams, 1-2 grams, 1-3 grams, and 1-4 grams.

$$BP = \begin{cases} 1, & c > r \\ exp\left(1 - \frac{r}{c}\right), & c \leq r \end{cases} \quad (1)$$

$$BLEU = BP.exp(\sum_{n=1}^{N} w_n log p_n) \quad (2)$$

In addition to using the BLEU score as the evaluation, this study also will be showing the model loss and accuracy for both training and validation data [18]. The purpose is to track the performance while in the training process.

3. RESULT AND DISCUSSION

3.1 Application Implementation

3.1.1 Data Preparation

The total data that will be used is 20000 (and its translation) parallel sentences which are 18000 for training data as well validation data and 2000 for testing data (Figure 5).

Building of Informatics, Technology and Science (BITS)
Volume 4, No 3, Desember 2022 Page: 1438−1446
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v4i3.2614

```
20000 "parallel sentences" will be loaded (original sentence + its translation)
18000 "parallel sentences" will be used to train the model
2000 "parallel sentences" will be used to test the model
```

Figure 5. The total number of parallel sentences which are 20000 sentences in total and 18000 will be used as training and validating data. 2000 sentences will be used as testing data

The length of a sentence (the count of the words that contains in the sentences) varies. Figure 6 is an example of random sentences from the source language (English) and target language (Sundanese) with various numbers of words between the sentences. Figure 7 is the sentences that already do the preprocessing which is lowercasing and deleting some unused symbols.

After the data goes through the tokenization process it yields 3149 and 3251 vocabulary sizes for the source and target

	inggris	sunda
1741	a Royal teacher	jadi guru Royal
10549	Happy birthday!	Wilujeung tepang taun!
8785	By spite about do of do allow blush.	Ngalangkung perkawis eusina ngijinkeun blush.
19238	The Tortoise meanwhile kept going slowly but s...	Samentara Tortoise tetep lalaunan tapi terus-t...
8094	I am sorry I have nothing better to offer you	Hapunten kuring teu ngagaduhan nanaon anu lang...
10439	"If you wait, there'll be a table for you free...	Upami anjeun ngantosan, bakal aya méja anjeun ...
6928	significant	penting
9285	jungle	leuweung
13808	Can I have a room again next month?	Dupi abdi tiasa nampi kamar deui bulan payun?
1844	How can the painter make the king's portrait b...	Kumaha pelukis ngajantenkeun gambar raja anu é...

Figure 6. The example of 10 first raw data is already shuffled. The raw data must be processed before making the data clean.

	0	1
0	a royal teacher	jadi guru royal
1	happy birthday	wilujeung tepang taun
2	by spite about do of do allow blush	ngalangkung perkawis eusina ngijinkeun blush
3	the tortoise meanwhile kept going slowly but s...	samentara tortoise tetep lalaunan tapi terus t...
4	i am sorry i have nothing better to offer you	hapunten kuring teu ngagaduhan nanaon anu lang...
5	if you wait there ll be a table for you free i...	upami anjeun ngantosan bakal aya méja anjeun g...
6	significant	penting
7	jungle	leuweung
8	can i have a room again next month	dupi abdi tiasa nampi kamar deui bulan payun
9	how can the painter make the king's portrait b...	kumaha pelukis ngajantenkeun gambar raja anu é...

Figure 7. Data after preprocessing is just to remove the unused symbols and lowercasing all of the words. Because making a vocabulary is case-sensitive.

Language respectively (Table 7). Afterward, it yields the maximum 79 and 66 vector sizes for the source and target language respectively. In addition, the input vector size will be 79 and the output vector size will be 66 with the 0 padding in between if the sentences did not match with 79 lengths of a word as shown in Table 8.

Table 7. Vocabulary and maximum vector in both languages. Vocabulary is a set of a token and max vector length is the sum of tokens that contains in each of every encoded sentence.

	English	Sundanese
Vocabulary size	3149	3251
Max vector length	79	66

Building of Informatics, Technology and Science (BITS)
Volume 4, No 3, Desember 2022 Page: 1438−1446
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v4i3.2614

Table.8 Encode sentenses into integer encoding sequences. The length of vector in every sentences follows the max vector length of every languages. For example In the source sentence will be encoded in 79 length of vector. So if the sentences just contains one word the rest of vector will be filled as zero.

eng]		Sundanese	
Sentences	Encoding	Sentences	Encoding
they	[41, …,0, 0]	Aranjeunna	[21,..,0, 0]
	[4, 1298, 20, …,0]	Abdi tiasa nyisiran rambut	[3, 10, 1235, …,0]

3.1.2 Building The Model

The architecture implementation model is shown in Figure 8. The first layer is the input layer to read the input which is the source language vector with a size of 66 and then goes to the Embedding layer to embed the vector so the LSTM layer later can be processed. The first LSTM layer is used as an encoder and then the output will be processed with the RepeatVector layer. The purpose of the RepeatVector layer is to work as a bridge from the first LSTM layer to the second LSTM layer as the encoder.

Afterward, the LSTM decoder will go to the Dense layer with the time-distributed form with softmax activation (equation 3) function to distribute the probabilities to each class which is the target vocabulary. The model summary can be shown in Figure 9.

$$\sigma(z_i) = \frac{e^{zi}}{\sum_{j=1}^{K} e^{zj}} \quad for\ i = 1,2,\ldots,K \tag{3}$$

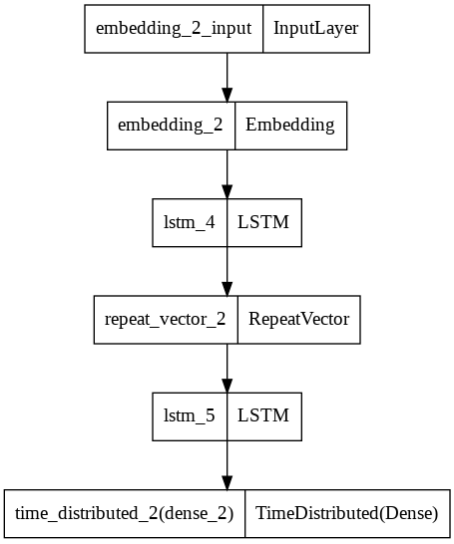

Figure 8. The model architecture will first going through input layer and then embedding layer to embed the model into LSTM. The output of LSTM will be processed in RepeatVector Layer before it will process to LSTM again. Finally, the output will be Dense time distributed layer.

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_1 (Embedding)     (None, 79, 256)           806144

 lstm_2 (LSTM)               (None, 256)               525312

 repeat_vector_1 (RepeatVect  (None, 66, 256)          0
 or)

 lstm_3 (LSTM)               (None, 66, 256)           525312

 time_distributed_1 (TimeDis  (None, 66, 3251)         835507
 tributed)

=================================================================
Total params: 2,692,275
Trainable params: 2,692,275
Non-trainable params: 0
```

Figure 9. The embedding layer will take 79 input sizes which is the maximum vector in the source language. The final layer which Dense time-distributed layer yield 66 output size which is the maximum vector size for the target language.

Building of Informatics, Technology and Science (BITS)
Volume 4, No 3, Desember 2022 Page: 1438−1446
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v4i3.2614

This model parameter can be shown in Table 9. The optimizer will be using Adam [19] since this optimizer is good in most of the deep learning performances in the recent studies [20]. The loss function is using categorical cross-entropy (equation 4) to compute the loss for determining the output which are the words from the target vocabulary. The validation split is 0.1 which means 10% data will be used as the validation data from the data training. The epochs are 200 but using the Early Stopping callbacks to prevent the overfitting problem [21] occurred in the models with the patience of two steps.

$$-\sum_{c=1}^{M} y_{o,c} log(P_{o,c}) \quad (4)$$

Table 9. Model parameters for the model architecture are Adam optimizer, loss function, epochs number, validation split, and a callback function.

Parameter	Value
Optimizer	Adam
Loss	Categorical crossentropy
Epochs	200
Validation Split	0.1
Callback	Early stopping

3.1.4 Training Process

The input is 79 vector size with each number meaning the tokens of the word in an English sentence. This input will go to the input layer and then the Embedding layer to embed the input and then processed by LSTM. The LSTM layer captures the information in the sequence of the word in a sentence. Thereafter output of the LSTM is processed by a time-distributed layer to map or to match 66 input size vectors of the output which are the tokens in Sundanese Sentence. If the model is wrong guessing it will update the weight until the weight is good enough to fit with the test sentences. Adam optimizer approach will find the best weight from epoch to epoch. If between 2 epochs occur the indication of overfitting. The callback will force the training process to stop. The overfitting occurred if the accuracy or loss for training data is bigger than validation data with a certain threshold. This process is repeated until 200 epochs or the callback function is triggered.

3.1.3 Evaluation

The training process takes approximately 6 hours with 70 epochs total because of the Early Stopping trigger. The accuracy and the loss as shown in Figure 10 0.996 and 0.016 for the data training. The average BLEU score for training data is 0.863 (Figure 11) and for testing, data is 0.806 (Figure 12).

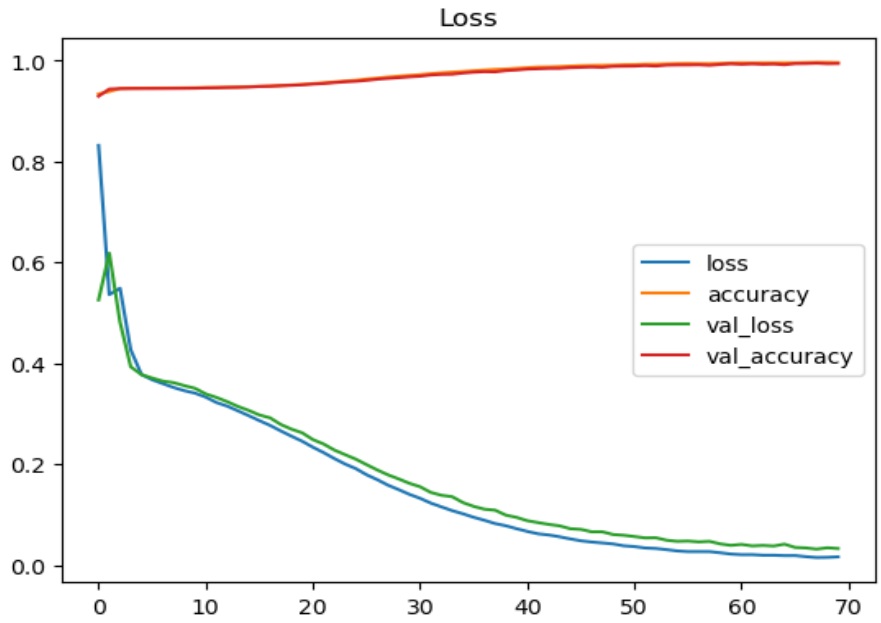

Figure 10. Accuracy and loss value from epoch to epoch. In this experiment, the epoch must be 200 but since the model uses an early stopping callback function, the epoch stops at 70 because the last 2 epoch indicates the model encounter an overfitting problem. it

Building of Informatics, Technology and Science (BITS)
Volume 4, No 3, Desember 2022 Page: 1438−1446
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v4i3.2614

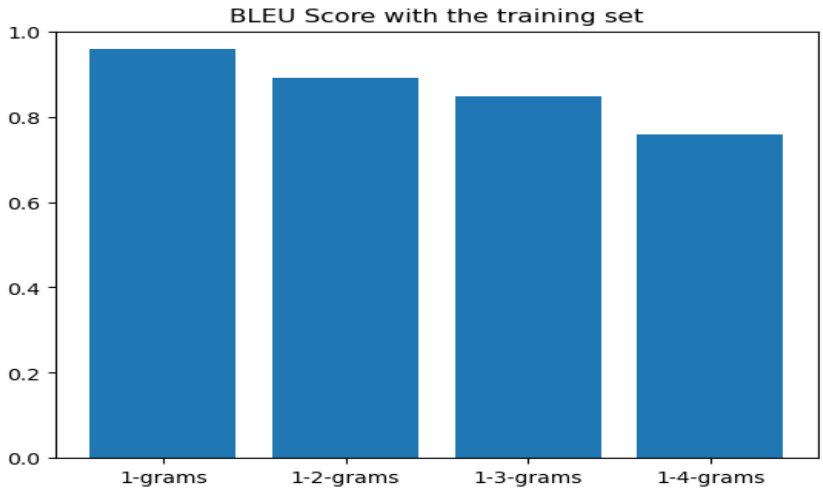

Figure 11. BLEU for training data. Calculated in 1, 1-2, 1-3, and 1-4 grams. The 1-grams is the highest result and the rest keep decreasing. It is because it calculated for all of the grams for 1 to the rest.

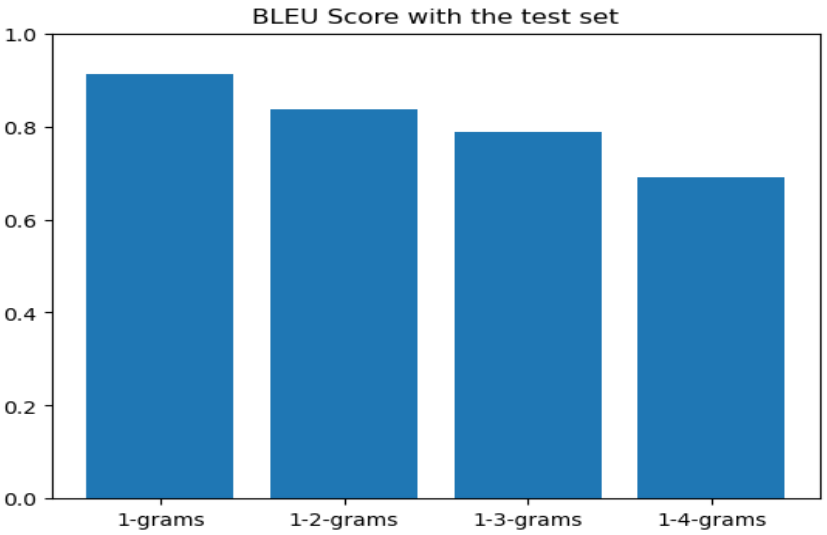

Figure 12. The BLUE score results are not far from the training data which is a good indication. Calculated in 1, 1-2, 1-3, and 1-4 grams. The 1-grams is the highest result and the rest keep decreasing. It is because it calculated for all of the grams for 1 to the rest.

Shown in Figure 8. Despite the training, the process is so long this model achieves a good result regarding the translation with the loss accuracy as well as the BLEU score evaluation. For the training loss and accuracy, it gets 0.016 and 0.996 respectively. For the testing loss and accuracy, it gets 0.0403 and 0.993 respectively. For the training set with 1-gram , 1-2 grams, 1-3 grams, and 1-4 are 0.957, 0.891, 0.846, and 0.759 respectively with a 0.863 average. For the test set with 1-gram , 1-2 grams, 1-3 grams, and 1-4 are 0.913, 0.835, 0.787, and 0.690 respectively with a 0.806 average. Table X show 5 sentence translation example in the test set. The sentence mostly translated well same the target language.

Table 10. Translation example on testing data. The English and Sundanese column is the true value of the translation. The predict column is the translation result which is predicted by the model. For these five examples, the result is approximately the same. The difference is just in the third data

English	Sundanese	Predict
horse	kuda	kuda
It was nice meeting you	Éta saé pendak sareng anjeun	Éta saé pendak sareng anjeun
Said that he has to wait for	nyarios yén anjeunna ´ kedah ngantosan	Nyarios yén kedah kedah ngantosan
He said he didn't want to lire a car	Cenah annjeuna henteu hoyong nyéwa mobil	Cenah anjeunna henteu hoyong nyéwa mobil
I like sports for example football	Abdi resep olahraga sapertos maén bal	Abdi resep olahraga sapertos maén bal

Building of Informatics, Technology and Science (BITS)
Volume 4, No 3, Desember 2022 Page: 1438−1446
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v4i3.2614

4. CONCLUSION

The encoder and decoder LSTM architecture are enough to make the machine translation model in the English - Sundanese language case. The performance of this model is good enough with 0.99 accuracies in both training and testing as well as less than 0.1 loss in both training and testing. This model also achieves a 0.8 average BLEU score for both training and testing data. In the next study regarding English - Sundanese machine translation can try to add the sequence to sequence model or attention mechanism to the model since that model mostly solves the LSTM problem architecture for machine translation

ACKNOWLEDGMENT

We thank Dicky totti Juniferdyaz, Diki Nurul Rivai , Marsela Arsya Sakinah, and Sri Agustina Dewi for maintenance of the English-Sundanese datasets and editing some of this manuscript.

REFERENCES

[1] B. J. Barat, “https://jabar.bps.go.id/indicator/12/731/1/jumlahpenduduk-hasil-proyeksi-interim-di-provinsi-jawa-barat-menurutkabupaten-kota-dan-jenis-kelamin.html. Accessed: 2022-11-19.” 2022.
[2] O. data jawa Barat, “https://opendata.jabarprov.go.id/id/dataset/jumlahwisatawan-berdasarkan-kategori-di-jawa-barat. Accessed: 2022-11-19.” 2022.
[3] A. A. Suryani, D. H. Widyantoro, A. Purwarianti, and Y. Sudaryat, “Experiment on a phrase-based statistical machine translation using PoS Tag information for Sundanese into Indonesian,” in 2015 International Conference on Information Technology Systems and Innovation (ICITSI), 2015, pp. 1–6.
[4] R. Darwis, H. Sujaini, and R. D. Nyoto, “Peningkatan Mesin Penerjemah Statistik dengan Menambah Kuantitas Korpus Monolingual (Studi Kasus: Bahasa Indonesia-Sunda),” JUSTIN (Jurnal Sist. dan Teknol. Informasi), vol. 7, no. 1, pp. 27–32, 2019.
[5] S. Yang, Y. Wang, and X. Chu, “A survey of deep learning techniques for neural machine translation,” arXiv Prepr. arXiv2002.07526, 2020.
[6] Y. Fauziyah, R. Ilyas, and F. Kasyidi, “MESIN PENTERJEMAH BAHASA INDONESIA-BAHASA SUNDA MENGGUNAKAN RECURRENT NEURAL NETWORKS,” J. Teknoinfo, vol. 16, no. 2, pp. 313–322, 2022.
[7] Hickman, Louis, et al. "Text preprocessing for text mining in organizational research: Review and recommendations." Organizational Research Methods, vol. 25, no. 1, pp. 114-146, 2022.
[8] N. G. Ramadhan, “Indonesian Online News Topics Classification using Word2Vec and K-Nearest Neighbor,” J. RESTI (Rekayasa Sist. Dan Teknol. Informasi), vol. 5, no. 6, pp. 1083–1089, 2021.
[9] S. Kannan et al., “Preprocessing techniques for text mining,” Int. J. Comput. Sci. \& Commun. Networks, vol. 5, no. 1, pp. 7–16, 2014.
[10] K. Potdar, T. S. Pardawala, and C. D. Pai, “A comparative study of categorical variable encoding techniques for neural network classifiers,” Int. J. Comput. Appl., vol. 175, no. 4, pp. 7–9, 2017.
[11] Su, Chao, et al. "Neural machine translation with Gumbel tree-LSTM based encoder." Journal of Visual Communication and Image Representation, vol. 71, pp. 102811, 2020.
[12] Ramadhan, Nur Ghaniaviyanto, Nia Annisa Ferani Tanjung, and Faisal Dharma Adhinata“Implementation of LSTM-RNN for Bitcoin Prediction,” Indones. J. Comput., vol. 6, no. 3, pp. 17–24, 2021.
[13] Y. Hu, A. Huber, J. Anumula, and S.-C. Liu, “Overcoming the vanishing gradient problem in plain recurrent networks,” arXiv Prepr. arXiv1801.06105, 2018.
[14] Tanti, Marc, Albert Gatt, and Kenneth P. Camilleri. "What is the role of recurrent neural networks (rnns) in an image caption generator?." arXiv preprint arXiv:1708.02043, 2017.
[15] A. Vaswani et al., “Attention is all you need,” Adv. Neural Inf. Process. Syst., vol. 30, 2017.
[16] M. Freitag, G. Foster, D. Grangier, V. Ratnakar, Q. Tan, and W. Macherey, “Experts, errors, and context: A large-scale study of human evaluation for machine translation,” Trans. Assoc. Comput. Linguist., vol. 9, pp. 1460–1474, 2021.
[17] Wieting, John, et al. "Beyond BLEU: training neural machine translation with semantic similarity." arXiv preprint arXiv:1909.06694, 2019.
[18] Neubig, Graham. "Neural machine translation and sequence-to-sequence models: A tutorial." arXiv preprint arXiv:1703.01619, 2017.
[19] Zhang, Jiacheng, et al. "Thumt: An open source toolkit for neural machine translation." arXiv preprint arXiv:1706.06415, 2017.
[20] S. H. Haji and A. M. Abdulazeez, “Comparison of optimization techniques based on gradient descent algorithm: A review,” PalArch’s J. Archaeol. Egypt/Egyptology, vol. 18, no. 4, pp. 2715–2743, 2021.
[21] Barone, Antonio Valerio Miceli, et al. "Regularization techniques for fine-tuning in neural machine translation." arXiv preprint arXiv:1707.09920, 2017.