Proceedings of SIGUL2022 @LREC2022, pages 122–128
Marseille, 24-25 June 2022

A Neural Network Approach to Create Minangkabau-Indonesia Bilingual Dictionary

Kartika Resiandi, Yohei Murakami, Arbi Haza Nasution
Ritsumeikan University, Ritsumeikan University, Universitas Islam Riau
1-1-1 Noji-higashi, Kusatsu, Shiga, Japan, 1-1-1 Noji-higashi, Kusatsu, Shiga, Japan,
Jl. Kaharuddin Nst 113 Pekanbaru, Riau, Indonesia
gr0502ee@ed.ritsumei.ac.jp, yohei@fc.ritsumei.ac.jp, arbi@eng.uir.ac.id

Abstract

Indonesia has many varieties of ethnic languages, and most come from the same language family, namely Austronesian languages. Coming from that same language family, the words in Indonesian ethnic languages are very similar. However, there is research stating that Indonesian ethnic languages are endangered. Thus, to prevent that, we proposed to create a bilingual dictionary between ethnic languages using a neural network approach to extract transformation rules using character level embedding and the Bi-LSTM method in a sequence-to-sequence model. The model has an encoder and decoder. The encoder functions read the input sequence, character by character, generate context, then extract a summary of the input. The decoder will produce an output sequence where every character in each time-step and the next character that comes out are affected by the previous character. The current case for experiment translation focuses on Minangkabau and Indonesian languages with 13,761-word pairs. For evaluating the model's performance, 5-Fold Cross-Validation is used. The character level seq2seq method (Bi-LSTM as encoder and LSTM as decoder) with an average precision of 83.55% outperforms the sentence piece byte pair encoding (vocab size of 32) with an average precision of 79.93%.

Keywords: Indonesian ethnic language, character level, Bi-LSTM, sequence to sequence model

1. Introduction

Indonesia's riches extend beyond natural resources such as minerals, vegetation, and fauna. Furthermore, the archipelago's culture is highly diversified, and so does a variety of ethnic languages in Indonesia.

The Austronesian language family includes Indonesian, derived from the Malay language. Since prehistoric times, Indonesian ethnic languages have developed, resulting in a different language for each ethnic group in Indonesia (Paauw, 2009). Belong to the same language family and based on the similarity matrix by utilizing the ASJP database (Nasution et al., 2019), most of Indonesian ethnic languages are closely related and similar.

Currently, the phenomenon of ethnic language extinction in Indonesia has become a problem that grabs the attention of scholars, especially linguists. The Summer Institute of Linguistic states that the local languages are endangered and may cease to be spoken in Indonesia. Therefore, we started the Indonesia Language Sphere project that aims at comprehensively creating bilingual dictionaries between the ethnic languages using a neural network approach and crowdsourcing approach, in order to conserve local languages on the verge of extinction (Murakami, 2019). As an expected result, the vocabulary of the ethnic language will expand, more people will learn it, and if there are no more speakers in the future, the language will become extinct.

The current translation experiment case focuses on Minangkabau and Indonesian languages since most of the nationalist writers who contributed to the early development of Indonesian were of Minangkabau ethnicity. Minangkabau language (closely linked to Malay) significantly influenced Indonesian in its formative years (Nasution et al., 2019). Between two languages, we presume they have several phonetic transformation rules. For example, there appears to be a rule in Indonesian and Minangkabau that the last phoneme "a" in Indonesian tends to turn "o" in Minangkabau. Although this rule isn't always valid, it can help predict a rough translation as a preliminary translation.

This study predicts the translation using character level embedding and the Bi-LSTM approach, compared to the sentence piece method using the sequence-to-sequence model.

2. Bilingual Dictionary Induction

Creating a bilingual dictionary is the first crucial step in enriching low-resource languages. Especially for the closely related ones, it has been shown that the constraint-based approach helps induce bilingual lexicons from two bilingual dictionaries via the pivot language (Nasution et al., 2016; Nasution et al., 2017a). However, implementing the constraint-based approach on a large scale to create multiple bilingual dictionaries is still challenging in determining the constraint-based approach's execution order to reduce the total cost. Plan optimization using the Markov decision process is crucial in composing the order of creation of bilingual dictionaries considering the methods and their costs (Nasution et al., 2017b; Nasution et al., 2021).

Heyman et al. (2018) have proposed a method to make bilingual lexical induction as a binary classification

task in the biomedical domain for English to Dutch. They create a classifier that predicts whether a pair of words is a translation using character and word level, LSTM method. This study reveals that character-level representations successfully induce bilingual lexicons in the biomedical domain.

Zhang et al. (2016) presented a character-level sequence-to-sequence learning approach proposed in this study. RNN is the encoder-decoder technique used to generate character-level sequence representation for the task of English-to-Chinese.

3. A Neural Network Approach

We would like to extract transformation rules or patterns from the Minangkabau to Indonesia language. The first approach is using character level one hot embedding where words will be separated as characters, and each vector has the same length size adjusted by total characters. Then, sequence to sequence (seq2seq) model, which has two RNN encoders and decoders is utilized. Bi-LSTM as encoder and LSTM as decoder processes are being used in this research. The Bi-LSTM encoder processes the word in the source language (Minangkabau) character by character and produces a representation of the input words. The LSTM decoder takes the output of the encoder as an input and produces a character by character in the target language (Indonesia). Similarly to the first method, the second method employs a sequence to sequence model. The distinction is in the input words, which are tokenized using SentencePiece with byte pair encoding for input to the encoder and decoder in a sequence to sequence model. The tokenization is splitting the words into chunck of characters.

The secondary data is obtained from Nasution et al. (2019) and Koto and Koto (2020) with a total of 13,761-word translation pairs. Pre-processing is completed by deleting duplicate word pairs and constructing an array of word pairs in the form of a data type dictionary given by Python. Because in this case, there are various word pairings of Minangkabau to Indonesian that have several meanings. A dictionary is made up of a set of key-value pairs. Each key-value pair corresponds to a certain value. The model's performance is evaluated using a 5-Fold Cross-Validation.

3.1. Long Short-Term Memory (LSTM)

The Long Short-Term Memory (LSTM) is an upgraded Recurrent Neural Network (RNN) that is used to overcome the problem of vanishing and exploding gradients (Hochreiter and Schmidhuber, 1997). LSTM addresses the problem of long-term RNN reliance, in which RNNs are unable to predict input data stored in long-term memory but can make more accurate predictions based on current information. The LSTM architecture can store large amounts of data for lengthy periods of time. They are applied to time-series data processing, forecasting, and categorization. Memory cells and gate units are the key components of the LSTM architecture. Forget gate, input gate, and output gate are the three types of gates in an LSTM. Figure 1 illustrates the structure of the LSTM model.

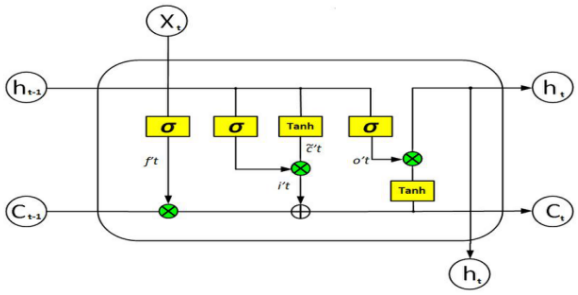

Figure 1: Unit structure of the LSTM

Cell memory tracks the dependencies between components in the input sequence. New values that enter the cell state are handled by the input gate. The LSTM unit utilizes a forget gate to select the value that remains in the cell state. The value in the cell state that remains will be sent to the output gate, where the LSTM activation function, also known as the logistic sigmoid function, will be used to start the calculation. The tanh and sigma symbols represent the types of activation functions employed in the neural network's training layers. Allowing information to flow through it unmodified, a sigmoid gate, which restricts how much information may pass through, is another essential feature of LSTM. The outputs of the sigmoid layer, which vary from zero to one, specify how much of each component should be permitted to pass. The equation that controls the LSTM flow is as follows:

$$f_t = \sigma(w_f \cdot [h_{t-1}, x_t] + b_f$$

$$i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i$$

$$C_t = \tanh(w_c \cdot [h_{t-1}, x_t] + b_c$$

$$C_t = f_t \times C_{t-1} + i_t \star C_t$$

$$o_t = \sigma(w_o \cdot [h_{t-1}, x_t] + b_o$$

$$h_t = o_t \times \tanh C_t$$

where

o_t	: at time t, output gate
i_t	: at time t, input gate
h_t	: output at time t
f_t	: forget gate, at time t
x_t	: input at time t
σ	: sigmoid function
C_t	: the state of the cell at time t
w_o, w_f, w_i, w_c	: weights that have been trained
b_c, b_i, b_f	: trained biases

3.2. Bidirectional Long Short-Term Memory (Bi-LSTM)

RNN has an advantage in the reliance between coding inputs. However, LSTM has an advantage in resolving RNN's long-term issues. Improvements are made with Bi- RNN because only one direction of previous contextual information can be used by LSTM and RNN (Schuster and Paliwal, 1997). As a result of the advantages of each technique, the LSTM form is kept in the cell memory, and Bi-RNN can process information from the previous and next contexts, resulting in Bi-LSTM (Schuster and Paliwal, 1997). Bi-LSTM can leverage contextual information and generate two separate sequences from the LSTM output vector. Each time step's output is a mixture of the two output vectors from both directions, as shown below, where ht is the forward or backward state (Yulita et al., 2017). Figure 2 depicts the combination of LSTM and Bi-RNN.

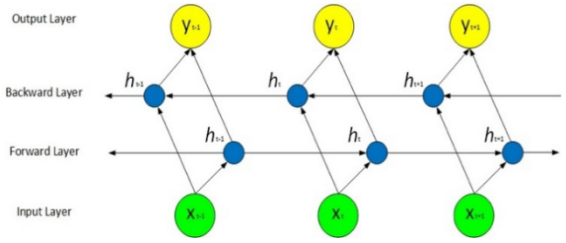

Figure 2: Bi-LSTM Architecture

3.3. Character Level Sequence to Sequence)

Figure 3 shows the Seq2Seq model considered in this study with a two-layered Bi-LSTM encoder and LSTM decoder. The encoder's functions are to character by character read the input sequence, build context, and extract a summary of the input. The decoder will provide an output sequence in which the previous character affects every character in each time step as well as the next character that emerges. The marker <eos> denotes the end of a sentence, and it will determine when we stop predicting the following character in a series (Sutskever et al., 2014).

Following the construction of the encoder and decoder network architectures in this typical end-to-end framework, a training approach may be utilized to obtain an optimal word pair translation model and to keep the character order C_t is referred to as a cell state or memory cell since the horizontal line going across the bottom of the diagram is in the source and target words, the input (Minangkabau) and output (Indonesia) sequence must be treated in time order.

3.4. SentencePiece Sequence to Sequence with Byte Pair Encoding (BPE)

The second method we presented is SentencePiece as subword tokenization. According to Kudo (2018), subword tokenization implements SentencePiece, subword-nmt, and wordpiece model features. Subword vocabulary is built by using the BPE segmentation method to train a SentencePiece tokenization model, which divides words into chunks of characters based on vocabulary size to make pattern detection easier.

BPE was added to our research methodology because Indonesian ethnic languages now utilize an alphabet script established by the Dutch despite having original scripts in the past. Dutch people appeared to assign a chunk of alphabets to phonemes of Indonesian ethnic languages when teaching the alphabets to them (Paauw, 2009). As a result, all Indonesian ethnic languages can use the same tokens.

Furthermore, with each phonetic development, languages belonging to the same language family descended from the same proto-language. As a result, we assume a phonetic-based strategy is preferable to a character-based method. The number of words to be processed into tokenization is known as vocabulary size, which in this case refers to the number of most often occurring characters, including the symbol like $<$ /unk$>$, and whitespace. We employ a wide range of vocabulary sizes. The following step is the same as the first method.

Figure 4 shows that the encoder and decoder input results as a result of character splitting from BPE in this illustration of the seq2seq model. This approach differs from Figure 3 in that the encoder (Minangkabau word) and decoder (Indonesian word) inputs are different. In the BPE method, we first set the vocabulary size for each language.

BPE builds a base vocabulary consisting of all symbols found in the set of unique words, then learns merge rules to combine two symbols from the base vocabulary to create a new symbol. It continues to do until the vocabulary has grown to the required size. BPE algorithm replaces the data byte pairs that occur most frequently with a new byte until the data can no longer be compressed since no byte pair occurs most frequently. The steps in the training procedure are as follows (Sennrich et al., 2016):

1) Gather a huge amount of training data.
2) Determine the vocabulary's size.
3) At identify the end of a word, add an identifier ($<$ /w$>$) to the end of each word, and then calculate the word frequency in the text.
4) Calculate the character frequency after dividing the word into characters.
5) Count the frequency of consecutive byte pairs from the character tokens for a predetermined number of rounds and combine the most frequently occurring byte pairing.
6) Repeat step 5 until performed the necessary number of merging operations or reached the specified vocabulary size.

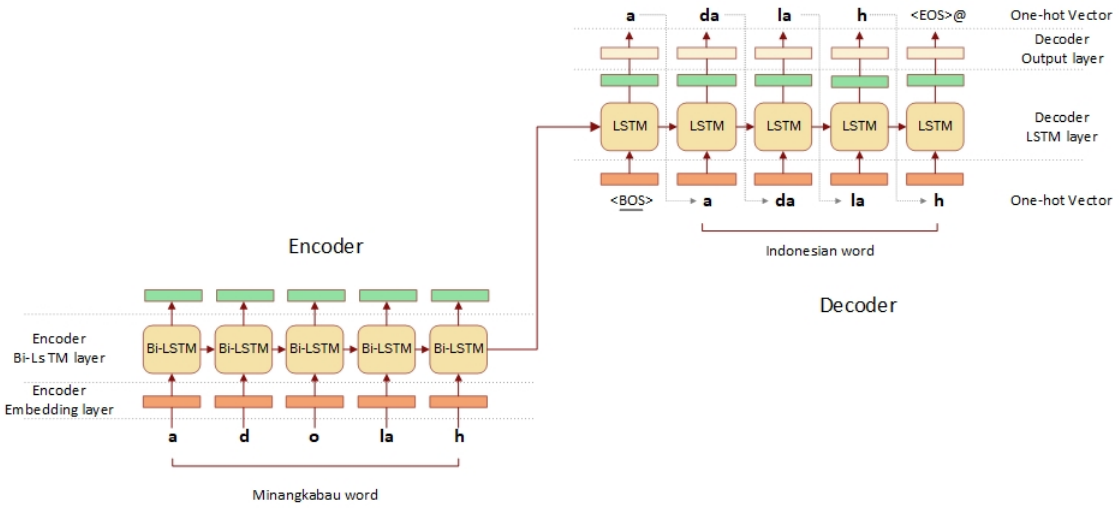

Figure 3: Character Level Sequence to sequence model

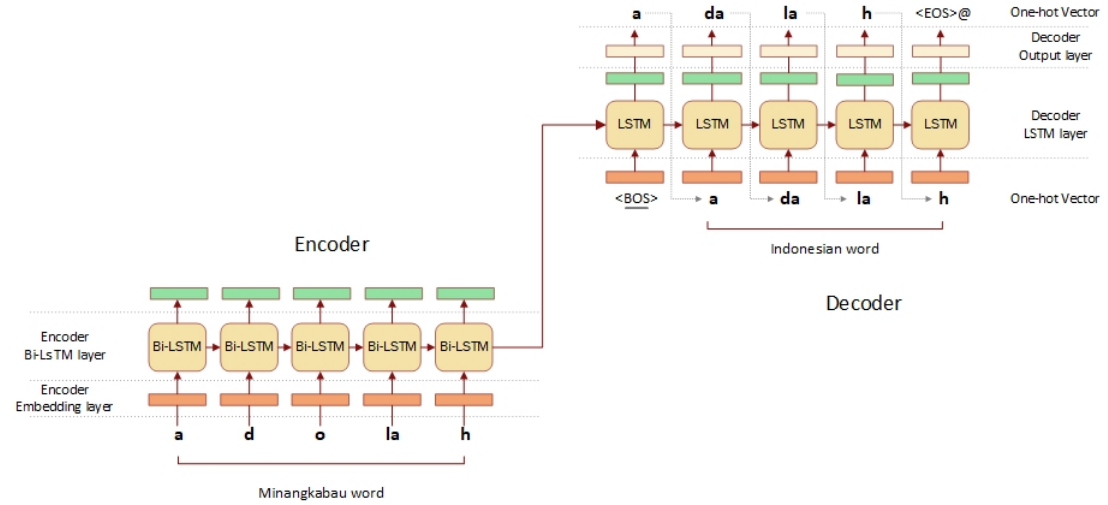

Figure 4: SentencePiece Sequence to sequence model

The input text is treated as a sequence of unicode characters by SentencePiece. Whitespace is also treated like any other symbol. SentencePiece expressly handles whitespace as a fundamental token by first escaping it with the meta symbol "___" (U+2581) (Kudo, 2018). Meanwhile the symbol of '\n' is the end of string. The results of the chunk of characters from the BPE will vary when utilizing a higher vocab size.

Except for alphabets, the vocabularies obtained from BPE 40 and 100 are summarized in the Table 1. For the Minangkabau language, there were 16 and 69 vocabularies obtained, respectively. Indonesian contains 9 and 69 vocabularies, respectively. According to the Table 1, character pieces are more obtained if use larger vocabulary sizes. The alphabet following the "_" symbol is a piece of characters at the beginning of the term in vocabulary that begins with the "_" symbol.

Example in the Minangkabau language, the difference between the character pieces sa and _ sa is that sa indicates that the character is not at the beginning of the word. Tokenization results refer to the Table 2 that shows the words in Minangkabau and Indonesia turned into a piece of characters from BPE.

The tokenization with vocab size=40 is done almost one by one like character-based tokenization except for "an", "ng", "pa" and "la" because vocab size=40 is nearly the same as the number of alphabets.

4. Experiment Design

In the first method, two models to find translation word pairs will be examined by Bidirectional Long Short-Term Memory, and also Long Short-Term Memory to improve and compare performance with previous research (Heyman et al., 2018). We utilize the parameters selected for both models in Table 3. Minangkabau and Indonesian are the language pairs, with a total dataset

Language	Vocab Size=40	Vocab Size=100
Minangkabau	an, _ma, ang, ng, _pa, _di, _ba, si, an, ng, kan, ta, si, ra, _men, nya	an, ng, ra, la, si, ta, _di, _ba, _pa, _ma, _ka, da, kan, nyo, li, ba, ang, ik, ri, ti, tu, ga, ka, bu, ja, ak _**sa**, ma, **sa**, ku, ku, ek, in, _man _ta, ah, di, su, to lu, ca, wa, du, pu, ro, mu, pa, bi, ran, en, lo, _pan, ju, tan, _pe ya, te, de, angan han, _me, gu, er _ke, do, po, gi, le, mi, _se
Indonesia	an, ng, kan, _di ta, si, ra, _men, nya	an, ng, kan, ta, ra, la, _di, da, nya, si, ke, _ber er, ti, ga, ba, li, in, ka, _se, ri, at, bu, tu, ja, ma, sa, en, _men, na di, _per, _a, ya, ku, pa, wa, is, lu _meng, _me, ca, _pen, _p, or, du, _ter, su, ru, ar, un, de, _ba, _mem, on, _ma, _ka, pu, ju, bi, _pe, al, _ko, ran, as, gu, tan, _sa, se

Table 1: Vocabularies obtained from BPE

Vocab Size= 40		Vocab Size= 100	
Minangkabau	Indonesia	Minangkabau	Indonesia
[_,n,an]	[_,y,a,ng,\'n']	[_,n,an]	[_,ya,ng,\'n']
[_pa,d,o]	[_,p,a,d,a,\'n']	[_pa,do]	[_,pa,da,\'n']
[_a,d,o,la,h]	[_a,d,a,l,a,h,\'n']	[_a,do,la,h]	[_a,da,la,h,\'n']
[_,s,a,g,i,r,o]	[_,s,e,g,e,ra,\'n']	[_,sa,gi,ro]	[_,se,ge,ra,\'n']
[_,d,a,s,an,y,o]	[_,d,a,s,a,r,nya,\'n']	[_,da,sa,nyo]	[_,da,sa,r,nya,,\'n']

Table 2: Example of tokenization BPE with different vocabulary size

of 13,761 language pairs split into 5 folds. Drop duplicated data is converted into 13,207 word translation pairs. Then, the total of training data is 10,565 and testing data is 2,642 language pairs.

5. Result and Discussion

This study uses two scenarios to find the optimal seq2seq model with the best performance. When comparing the character level and sentence piece approaches with the seq2seq model, the character level seq2seq method generates a more accurate translation of word pairs.

Character Level and SentencePiece with BPE		
Parameter	**Bi-LSTM**	**LSTM**
Embedding Size	512	512
Epoch	80	80
Batch Size	64	64

Table 3: Model's Parameter

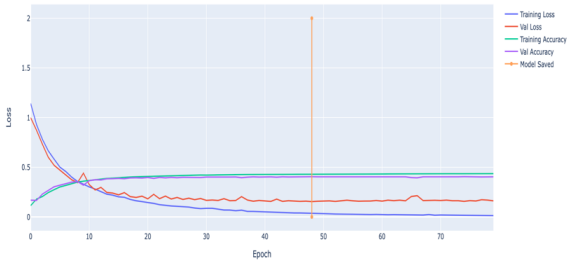

Figure 5: Epoch loss from train and validation on character level seq2seq model

Figure 5 shows the optimal process model that is saved and constructed to generate translation pairs based on the evaluation model using k-fold cross-validation. The model that will be utilized will be better if the loss value is smaller. The loss values for both train and validation remain high in the first epoch and gradually improve. The optimal validation loss value is identified in the 46th epoch using tensorflow's ModelCheckpoint feature, which only saves good models and does not save models in the following epoch if the validation loss value worsens.

Vocab Size	K-Fold Cross-Validation					
	K=1	K=2	K=3	K=4	K=5	Average
32	74.03	77.92	81.55	79.88	**86.27**	79.93
35	71.45	78.03	77.37	79.62	**85.87**	78.46
40	75,34	77,08	80,13	81,51	**83,36**	78,515
50	67.61	73.63	73.23	75.12	**79.99**	73.91
80	65.43	66.41	65.64	64.01	**72.22**	66.74
100	66.6	70.44	**70.91**	65.5	70.62	68.81
300	57.84	62.67	64.34	**67.9**	66.33	63.81

Table 4: Evaluation of SentencePiece with BPE model

The vocabulary size has a minimum and maximum value. The minimum number necessary for this experiment data is 32. The experiment was conducted seven times with various vocabulary sizes, with the largest of number vocab size is 300. As shown in Table 4, using vocabulary size=32, the highest generation of translation pairs accuracy is obtained at 86,27%. Perhaps, because the vector length is shortened, the data is likely to be less informative, making it more difficult for the

Method	K-Fold Cross-Validation					
	K=1	K=2	K=3	K=4	K=5	average
Bi-LSTM (encoder), LSTM (decoder)	78.85	82.23	82.67	86.48	**87.5**	83.55
LSTM (encoder decoder)	64.92	75.19	74.72	**77.01**	75.63	73

Table 5: Evaluation of character-level model

model to recognize. In general, the larger the vocabulary size, the higher the results. It is also probably because the data is word-to-word pairs translation instead of sentence to sentence.

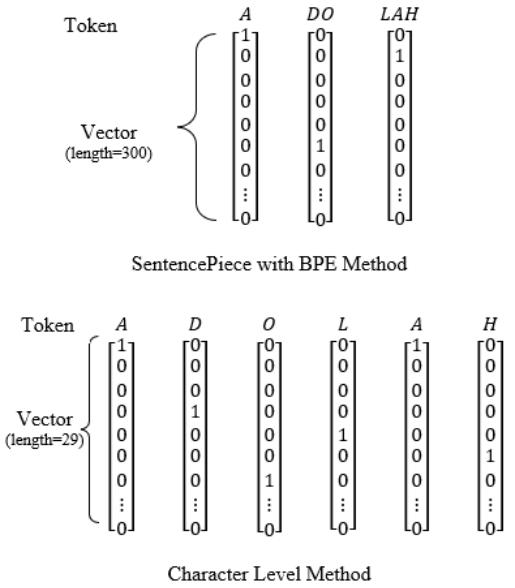

Figure 6: Comparison between SentencePiece with BPE and character level method

However, when we use a small vocab size, it's almost the same as the basic character level. As shown in Table 5, because the Bi-LSTM executes the input in two ways, backward to forward and vice versa, the outcome is better than when LSTM is used as both encoder and decoder at an average precision of 83.55%.

6. Conclusion

According to the comparison of the two approaches used, the character level seq2seq method (Bi-LSTM as encoder and LSTM as decoder) with an average precision of 83.55% outperforms the sentence piece byte pair encoding (vocab size of 32) with an average precision of 79.93%. The model can recognize patterns in both Minangkabau and Indonesian languages, indicating that the two languages are related. In the future, we will adapt the approach utilized in this research to other ethnic languages depending on the translation data pairs, add more experiments and analysis, and find the patterns from generated translation model.

7. Acknowledgements

This research was partially supported by a Grant-in-Aid for Scientific Research (B) (21H03561,2021-2024) and a Grant-in-Aid Young Scientists(A)(17H04706,2017-2020) from the Japan Society for the Promotion of Science(JSPS). This research was also partially supported by the Ministry of Education and Culture of Indonesia, Ministry of Research and Technology of Indonesia.

8. Bibliographical References

Heyman, G., Vulić, I., and Moens, M.-F. (2018). A deep learning approach to bilingual lexicon induction in the biomedical domain. *BMC Bioinformatics*, 19(1), jul.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780, nov.

Koto, F. and Koto, I. (2020). Towards computational linguistics in minangkabau language: Studies on sentiment analysis and machine translation. In *Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation*, pages 138–148.

Kudo, T. (2018). Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Murakami, Y. (2019). Indonesia language sphere: an ecosystem for dictionary development for low-resource languages. *Journal of Physics: Conference Series*, 1192:012001, mar.

Nasution, A. H., Murakami, Y., and Ishida, T. (2016). Constraint-based bilingual lexicon induction for closely related languages. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3291–3298.

Nasution, A. H., Murakami, Y., and Ishida, T. (2017a). A generalized constraint approach to bilingual dictionary induction for low-resource language families. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 17(2):1–29.

Nasution, A. H., Murakami, Y., and Ishida, T. (2017b). Plan optimization for creating bilingual dictionaries of low-resource languages. In *2017 International Conference on Culture and Computing (Culture and Computing)*, pages 35–41. IEEE.

Nasution, A. H., Murakami, Y., and Ishida, T. (2019). Generating similarity cluster of indonesian languages with semi-supervised clustering. *Interna-*

tional Journal of Electrical and Computer Engineering (IJECE), 9(1):531, feb.

Nasution, A. H., Murakami, Y., and Ishida, T. (2021). Plan optimization to bilingual dictionary induction for low-resource language families. *Transactions on Asian and Low-Resource Language Information Processing*, 20(2):1–28.

Paauw, S. (2009). One land, one nation, one language: An analysis of indonesia's national language policy. *University of Rochester working papers in the language sciences*, 5(1).

Schuster, M. and Paliwal, K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Yulita, I. N., Fanany, M. I., and Arymuthy, A. M. (2017). Bi-directional long short-term memory using quantized data of deep belief networks for sleep stage classification. *Procedia Computer Science*, 116:530–538.

Zhang, H., Li, J., Ji, Y., and Yue, H. (2016). A character-level sequence-to-sequence method for subtitle learning. In *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*. IEEE, jul.