Paraphrase Detection Using Manhattan's Recurrent Neural Networks and Long Short-Term Memory

Achmad Abdul Aziz, *Esmeralda C Djamal, Ridwan Ilyas
Department of Informatics
Universitas Jenderal Achmad Yani
Cimahi, Indonesia
*Corresponding Author Email: esmeralda.contessa@lecture.unjani.ac.id

***Abstract*- Natural Language Processing (NLP) is a part of artificial intelligence that can extract sentence structures from natural language. Some discussions about NLP are widely used, such as Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) to summarize papers with many sentences in them. Siamese Similarity is a term that applies repetitive twin network architecture to machine learning for sentence similarity. This architecture is also called Manhattan LSTM, which can be applied to the case of detecting paraphrase sentences. The paraphrase sentence must be recognized by machine learning first. Word2vec is used to convert sentences to vectors so they can be recognized in machine learning. This research has developed paraphrase sentence detection using Siamese Similarity with word2vec embedding. The experimental results showed that the amount of training data is dominant to the new data compared to the number of times and the variation in training data. Obtained data accuracy, 800,000 pairs provide accuracy reaching 99% of training data and 82.4% of new data. These results are better than the accuracy of the new data, with half of the training data only yielding 64%. While the amount of training data did not effect on training data.**

Keywords—Siamese Similarity; RNN; paraphrase; natural language processing;

I. INTRODUCTION

Many scientific works in digital form are easily accessible, making it possible for plagiarism. Plagiarism or plagiarism is an act intentionally or unintentionally in obtaining or trying to get credit or value for scientific work, by quoting part or all of the work and scientific work of other parties that are recognized as experimental works, without expressing the source appropriately and adequately [1]. Some applications have been used for plagiarism detection, such as Unplug and Writecheck [2]. Some applications can be accessed for free but with some limitations. But some of it is paid, which is very expensive.

Scientific writing consists of several paragraphs. Each paragraph consists of several sentences. Meanwhile, the smallest structure of the essay that has meaning is the word. Several studies have examined the similarity between documents by calculating the number of words that have similarities using TF-IDF or CF-IDF [3] to identify the sentence. However, the similarity of sentences is only seen from the same percentage of words that is less precise considering the complexity of the meaning in a sentence cannot be separated from each word [4].

The limitations of the word similarity approach can be increased by modeling the underlying semantic similarities between sentences or phrases. That approach allows the boundaries of the case with the many variations in the wording used in expressing the same meaning. The method that plays an essential role in sentence analysis with the semantic approach is Natural Language Processing (NLP). NLP is part of artificial intelligence that can extract sentence structure and its meaning from natural language. Some NLP approaches are widely used, such as Long Short-Term Memory (LSTM) to summarize papers with many sentences in them [5]. However, the use of LSTM is more prevalent in learning certain words and patterns. Each sentence is seen as a set of data, and all the words contained in it are analogous to features in machine learning.

Several studies have been found using the NLP approach to information retrieval [6], text summaries [7], answering several text questions [8], language translation [9] and plagiarism detection [10]. In the detection of plagiarism, it is an identification of paraphrases. Paraphrasing means re-expressing a speech and a level or type of language into other sentences without changing the meaning. Paraphrase can also be interpreted as a breakdown of a text in another form, to be able to explain the hidden meaning [11]. Paraphrasing is used as a technique to describe something using different sentences but has the same purpose [12].

Recurrent neural network (RNN) is a type of artificial neural network architecture whose processing is repeatedly called to process input which is usually sequential data. RNN is included in the deep learning category because data is processed through many layers. The RNN has experienced rapid progress and has revolutionized fields such as natural language processing (NLP) [13]. This LSTM is part of the architecture of the RNN, which is a recurring network, so LSTM is used to model the contents of the weight of the networks. Another variation is the Manhattan LSTM method, so the identification process is more straightforward [14].

In this study establish similarity detection systems sentence in Britain language scientific papers using Recurrent Neural Networks (RNN) and Manhattan LSTM. The process starts with extracting sentence text documents, labeling sentences, changing sentences into vectors, and calculating weights using Manhattan LSTM.

II. METHODS

Previous research used a Recurrent Neural Network to classify text [15] [8]. The study uses Siamese Similarity to detect the similarity of sentence structure using two sentences in pairs as input, which is then passed to the LSTM process to give weight, and determine the structure of sentence similarities [16].

The other studies conducted detection on text documents using the Karp Cabin method. The detection process uses TF-IDF weighting and matching between test documents and source materials. The document matching process uses the N-Gram Technique and Rabin Karp method. The N-Gram Technique method involves two steps, namely dividing the string into overlapping N-Gram (a set of substrings with length n) and checking to get a substring that has the same structure. Rabin Karp works with matching strings that use the hash function as a comparison between the search string (m) and substring in the text (n). If the hash values are the same, will compare with the characters. If the results of the two are not the same, then the substring will shift to the right [3].

This study used Word2vec for extracting sentences into vectors, LSTM to give weight in training, and Siamese Recurrent Neural Network to detect paraphrasing sentences.

A. *Word2vec*

This research uses Word2Vec with a keyed vector Gensim model. Genism keyed vector is a library whose contents are vocabulary available in vector form. This model will create a matrix to accommodate the marriage between the embedding of the Gensim library, and the text data it has the results of the marriage of the text will be returned to the embedding. Then will get a new vocabulary from the results in the form of a vector that will go into the next process [17]. The process, as shown in Fig. 1

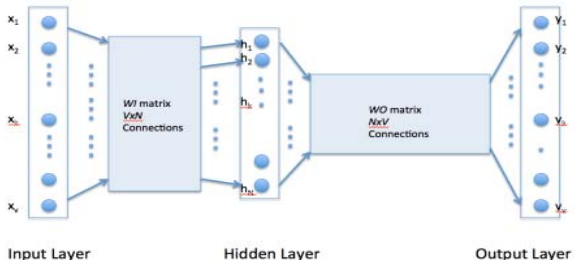

Fig. 1. Word Embedding

Secara tidak langsung model genism keyed vector ini akan membuat kamus yang menghasilkan dari library word2vec dan data teks yang dimiliki. Sehingga akan menghasilkan suatu kamus baru yang sesuai dengan kasus pada penelitian ini [18].

B. *Manhattan LSTM*

This Manhattan LSTM offers a relatively straightforward approach to common sentence similarity problems. The architecture is illustrated in Fig. 2 (not including the pre-sentence process). Because this is a conjoined network, so it is easier to train because it can share weights on both sides.

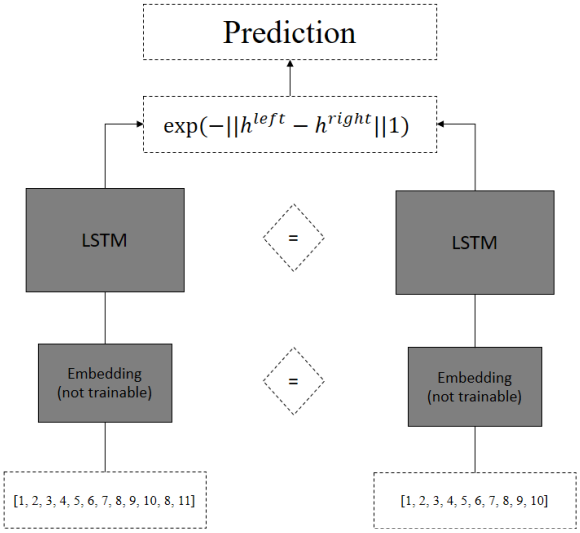

Fig. 2. Manhattan LSTM diagram

Siamese networks are networks that have two or more of the same sub-networks in them. This conjoined network works well for semantic sentence similarities, recognizes fake signatures, and more. This method gives the semantic meaning of words in vector representations.

Input to the network is a sequence of words without an index. This input is a vector of fixed length, where the first zero is ignored, and besides zero is an index that uniquely identifies the word.

The vectors are then inserted into the embedding layer. This layer looks for embedding that matches each word and summarizes everything into the matrix. This matrix represents the text to be given as a series of embedding [19]. The advantages of Manhattan LSTM are more comfortable in the pre-process, considering that the LSTM network shares the weight on both sides of each word in the sentence. Manhattan LSTM is also commonly used to process text, words, or sentences. So this Manhattan LSTM is very suitable for use in this problem

C. *Siamese Recurrent Neural Network*

The Siamese neural network is an architecture that contains two identical sub-networks that join the output [20], and are widely used in tasks finding similarities between two comparable patterns, for example, paraphrase identification.

The main feature of this conjoined structure is the sharing of weights across sub-networks, which reduces the number of parameters for training and trends that are more appropriate. Also, by processing inputs similar to similar models, sub-networks produce input representations that share the same semantics and are easily compared.

This model uses LSTM to read in words in vector shapes that represent each input sentence and use the presentation that was last processed by the previous submission. Furthermore, this

similarity between representations is used as a semantic similarity predictor [16]. This model is shown in Fig. 3.

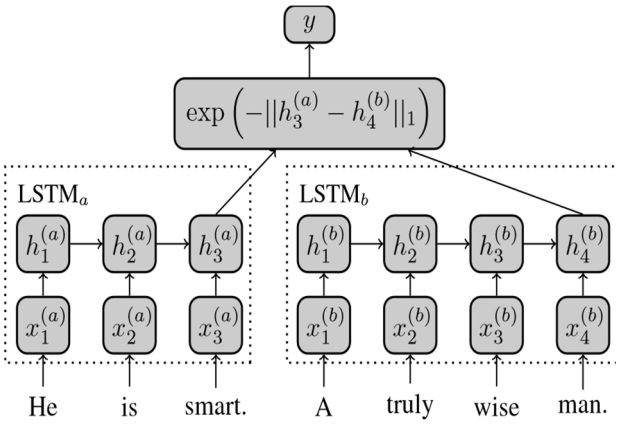

Fig. 3. Siamese RNN Manhattam LSTM model

The advantages of RNN itself are that they are useful in modeling sequential data, where they have memory storage that captures previous inputs and appropriate calculations, and allows information to be repeated in the network for a long time [21]. However, RNN has limitations in studying long-term dependencies, and decreasing performance with the increasing length of the input sequence. Given the boundaries of the RNN, the LSTM structure is widely used in sequence modeling, which is capable of studying long-term dependencies using four interacting layers, as opposed to one information processing layer in the RNN structure. In addition to better performance in long-term dependency modeling, LSTM provides more flexibility in controlling the amount of information stored as needed, making it the ideal choice for this modeling purpose.

This study extracted data in the form of sentences from English-language papers into vectors, which were previously labeled by the annotator. Extraction uses Word2vec to get a vector output from each word. The vector value will enter the weighting process with Manhattan LSTM. The weight of each side on LSTM will be classified using Siamese Similarity. Paraphrase or not paraphrase — data set before labeling as in Table I.

Labeling is carried out by annotator, namely an expert in English. Annotator predicts both sentences and concludes the similarity of meaning between the two sentences. Annotator will give the label "True" if the sentences compared have the same meaning or are called paraphrases and give "False" if the sentence compared is not paraphrase. So that you get supervised training data with the target class in the form of true or false (true/false) for the similarity between the two sentences compared, then the sentence is saved into the database. The labeling process can be seen in Fig. 4.

TABLE I. DATASET

Sentences1	Sentences2
A framework for human error analysis and error classification has been proposed in (Vilar et al., 2006) and a detailed analysis of the obtained results has been carried out.	A framework for human error analysis has been proposed in (Vilar et al., 2006) and a detailed analysis of the obtained results has been carried out.
For experiments reported in this paper, we use one of the largest, multi-lingual, freely available aligned corpus, Europarl (Koehn, 2005).	For each source and target pair in the English- Spanish portion of the Europarl corpus (Koehn, 2005), we initialize a sparse random vector.
Most of existing lexical-semantic networks have been built by hand (like for instance WordNet (Miller et al., 1990)) and, despite that assisting tools are generally designed for consistency checking, the task remains time consuming and costly.	Most of existing lexical-semantic networks have been built by hand (like for instance WordNet (Miller et al., 1990)) and, despite that tools are generally designed for consistency checking, the task remains time consuming and costly
…..	…..
Our system uses the architecture from (Lee et al., 2016) where a character-level neural MT model maps the source character sequence to the target character sequence.	Our system uses the architecture from (Lee et al., 2016) where a character-level neural MT model that maps the source character sequence to the target character sequence.

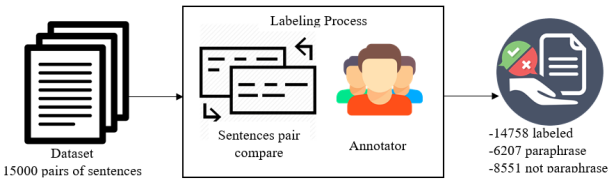

Fig. 4. Labeling Process

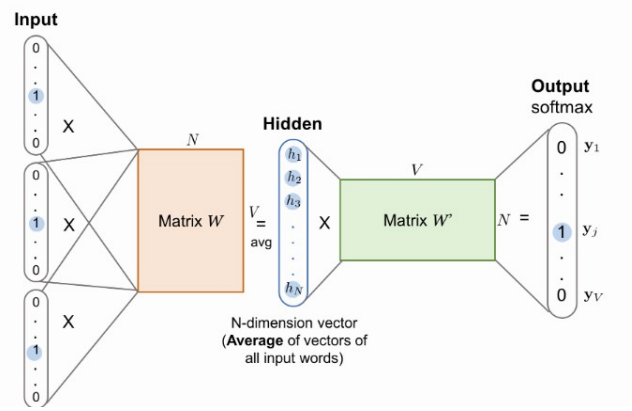

Fig. 5. After labeling, feature extraction is performed to get vector values from each sentence with word2vec, as shown in Fig. 5. Embedding Process

Inputs to the network are zero-padded sequences of word indices. These inputs are vectors of fixed length, where the first zeros are being ignored and the non zeros are indices that uniquely identify words.

Those vectors are then fed into the embedding layer. This layer looks up the corresponding embedding for each word and encapsulates all them into a matrix. This matrix represents the given text as a series of embeddings. Then feed into the LSTM

and the final state of the LSTM for each sentence is a 50-dimensional vector. This is shown in Fig. 6.

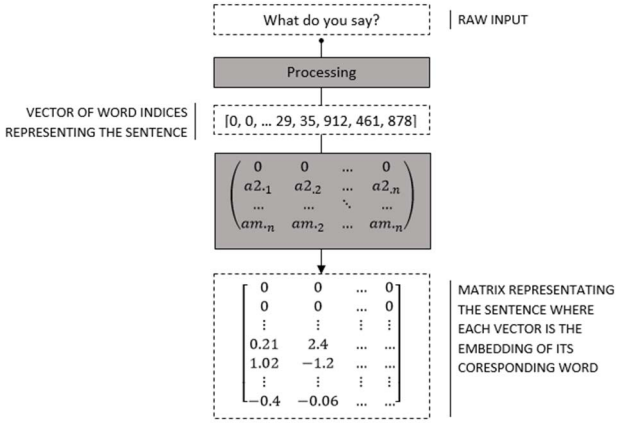

Fig. 6. Process Word2vec

For example, given a sentence: "*king brave man*" "*queen beautiful woman*". Then, after getting the vector value from each sentence, it will enter the next stage, namely the Siamese similarity model. As shown in Fig. 7.

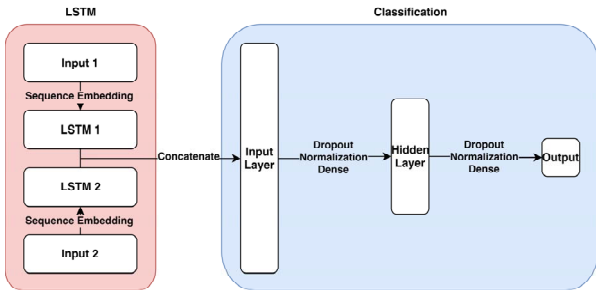

Fig. 7. Siamese Similarity RNN

A structure with two input layers, the sentence passes through the vector representation of sentence 1 with the other sentence pair to the embedding layer, which then results in embedding sentences and inserting embedding into the LSTM network. Next, the LSTM layer produces a vector representation of two sentences in the input sentence pair. The layered circuit is then applied to combine two input representations into a single vector representation, which is then used for the final classification.

The training data will then enter the learning process with the Siamese Similarity RNN to determine the similarity of the meaning of the sentence using (1).

$$exp\left(-\left|\left|h^{left} - h^{right}\right|\right|1\right) \qquad (1)$$

exp h^{left} is the left side network, and h^{right} is the right side network. Because using conjoined or twin network characters, the values from left to right are adjusted, to determine the significant increase of the two networks. The whole process can be seen, as shown in Fig. 8, which shows the proposed model.

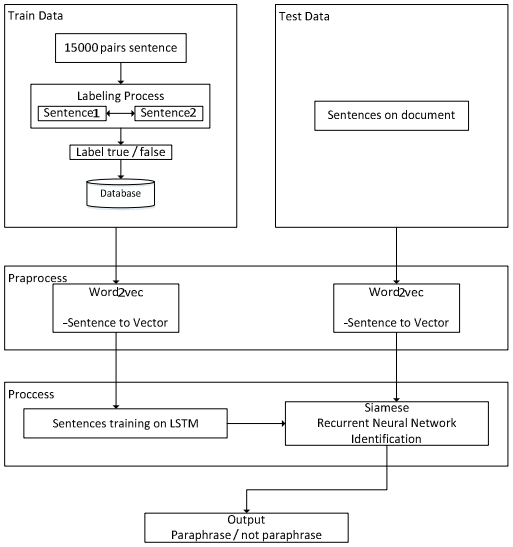

Fig. 8. Similarity identification model

III. RESULT AND DISCUSSION

The experiment consists of two parts, namely the influence of the amount of training data and the effect of the number of epochs. The model was developed using AdaDelta optimation.

A. Dataset Number Testing

The first experiment was to do a variation of the amount of training data. In general, the large amount of training data will certainly provide high accuracy. But a large amount of training data has the consequence of extensive training computing time. Therefore, it is necessary to test the amount of training data to provide optimal accuracy. This research used several configuration models as shown in Table II.

TABLE II. CONFIGURATION WITH VARIOUS OF TRAIN DATA NUMBER OF 25 EPOCHS

Configuration	Data	*Loss*	*Accuracy*
1	Train Data (3566 pairs)	0.2319	0.6955
	Test Data (1661 pairs)	0.3527	0.4660
2	Train Data (15000 pairs)	0.1587	0.7941
	Test Data (15000 pairs)	0.2297	0.6474
3	Train Data (400000 pairs)	0.1489	0.8473
	Test Data (800000 pairs)	0.1786	0.8254

In configuration 1 with training data, 3566 and test data 1661 produce the accuracy of the smallest new data between the two other settings. It is explained that there are fewer new data than configurations 2 and 3, which then affect generalizations in Siamese similarity training. In the process of Siamese similarity, a repetitive learning process is required, so variations in data will affect the machine training process. The difference in distance between accuracy between training data and test data can be seen in Fig. 9.

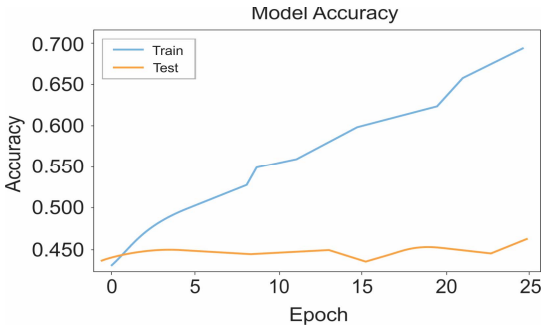

Fig. 9. Accuracy of 3566 pairs data set until 25 epochs

Configuration 2 shown that the accuracy increased with increasing of dataset 15000. The accuracy increased by about 0.16, as shown in Fig. 10.

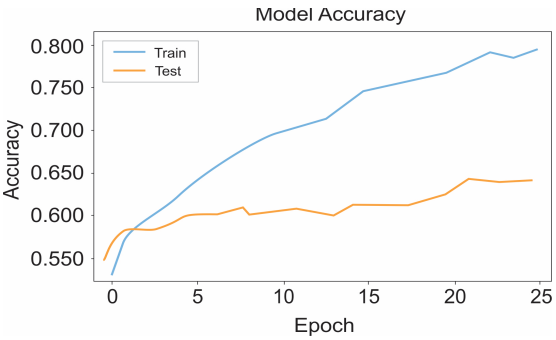

Fig. 10. Accuracy of 7132 pairs data set until 25 epochs

In the third configuration gave the results significantly higher accuracy. This way is because variations in data have a lot of influence on the process of Siamese similarity. The test results can be seen in Fig. 11.

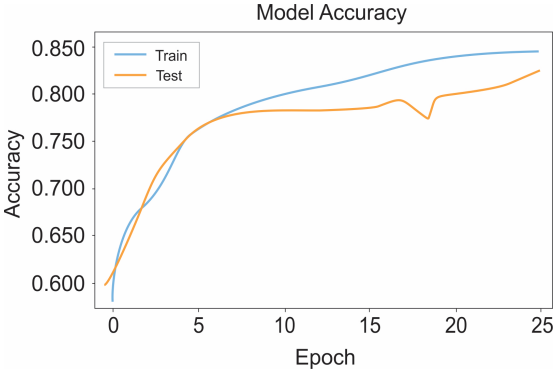

Fig. 11. Accuracy of 15000 pairs until 25 epochs

B. *Amount of Epoch Influence*

The second experiment was conducted to increase accuracy with variations in the number of epochs. The same configuration is tested for training with 500 epochs, as in Table III.

TABLE III. SCHEME WITH DIFFERENT TRAIN DATA

Scheme	Data 500 Epoch	Loss	Accuracy
4	Train Data (3566 pairs)	0.1319	0.9755
	Test Data (1661 pairs)	0.4527	0.4783
5	Train Data (7132 pairs)	0.0240	0.9827
	Test Data (3322 pairs)	0.3365	0.5138
6	Train Data (15000 pairs)	0.0289	0.9841
	Test Data (15000 pairs)	0.2367	0.6458

In scheme 4, using training data 3566 and new data, 1661 produce an accuracy of 0.9755. The accuracy value of the training data increases significantly from configuration 1 due to the difference in the number of epochs. In configuration 1, it uses 25 epochs, and scheme 4 uses 500 epochs, which can be seen in Fig. 12.

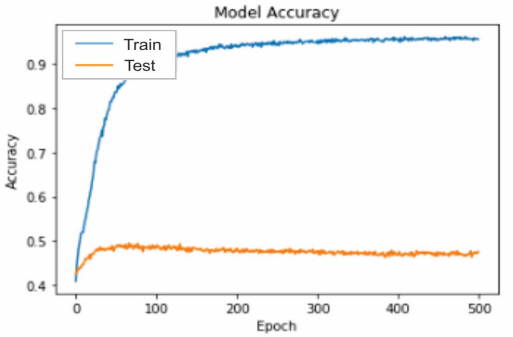

Fig. 12. Accuracy of 3566 pairs until 500 epochs

In scheme 5 uses duplicate data from plan 1, where the training data of 3566 are doubled to 7132, and the test data as many as 1661 are increased to 3322. However, duplication of the data does not affect the value of accuracy. Can be seen in Fig. 13.

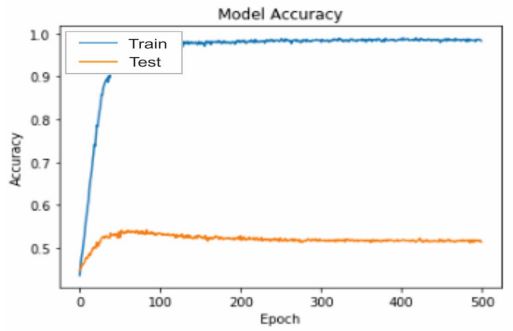

Fig. 13. Accuracy 3566 with duplicated into 7132 pairs until 500 epochs

Scheme 6 used the same data as schema 2; the difference is the number of epochs that use epoch 500. The addition of the epoch value affects the training data, which previously only reached 0.8474 to 0.9841. Can be seen in Fig. 13.

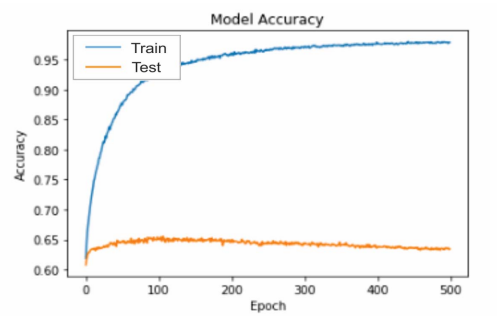

Fig. 14. Accuracy of 7132 pairs until 500 epcohs

IV. CONCLUSION

In the identification of sentences can be performed after machine learning using the RNN Siamese Similarity. Accuracy increased, and Losses decreased with an increasing number of data sets used. Of the three data set configurations obtained an accuracy of 85% of training data, and 82.54% of new data with as much training data as 800,000 pairs.

The results showed that the addition of epochs can increase accuracy from 85% to 99% of training data. The results also show that the machine learning process was repeated from random input data, and data duplication. Both of them did not affect the accuracy value of the Siamese Similarity RNN.

REFERENCES

[1] A. Kurniawati, A. K. Sekarwati, and I. W. S. Wicaksana, "Arsitektur Untuk Aplikasi Deteksi Kesamaan Dokumen," in *Konferensi Nasional Sistem Informasi*, 2012, pp. 297–302.

[2] A. Wahyuningsih, "7 Aplikasi Online Pendeteksi Plagiarisme." .

[3] P. N. Technique, D. A. N. Rabin, and K. Pada, "Pendeteksi Plagiarisme Dokumen Teks Bahasa Indonesia," *Seminar Nasional APTIKOM (SEMNASTIKOM)*, vol. 10, no. 28, pp. 753–759, 2016.

[4] M. R. Pratama, E. B. Cahyono, and G. I. Marthasari, "Aplikasi Pendeteksi Duplikasi Dokumen Teks Bahasa Indonesia Menggunakan Algoritma Winnowing Dengan Metode K-Gram Dan Synonym Recognition," *Techno.COM*, vol. 3, pp. 21–26, 2011.

[5] K. Yoko, J. Hendryli, T. Informatika, and U. Tarumanagara, "Sistem Peringkas Otomatis Abstraktif Dengan Menggunakan Recurrent Neural Network," *Journal of Computer Science and Information Systems*, vol. 2, no. 1, pp. 65–75, 2018.

[6] J. Ponte and W.B. Croft, "A Language Modeling Approach to Information Retrival," *ACM SIGIR*, vol. 51, no. 2, pp. 202–208, 2017.

[7] M. Mustaqhfiri and Z. Abidin, "Peringkasan Teks Otomatis Berita Berbahasa Indonesia Menggunakan Metode Maximum Marginal Relevance," *Matics*, pp. 134–147, 2013.

[8] M. S. Ahmed and L. Khan, "SISC: A text classification approach using semi supervised subspace clustering," *ICDM Workshops 2009 - IEEE International Conference on Data Mining*, no. April, pp. 51–56, 2009.

[9] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Trans," in *Published as a conference paper at ICLR 2015*, 2015, pp. 1–15.

[10] R. Y. Dillak, F. Laumal, L. J. Kadja, and S. S, "Sistem Deteksi Dini Plagiarisme Tugas Akhir Mahasiswa Menggunakan Algoritma N-Grams dan Winnowing," *Jurnal Ilmiah*, vol. 2, 2016.

[11] B. I. Julianto, M. S. Mubarok, T. B. Batu, and J. Barat, "Identifikasi Parafrasa Bahasa Indonesia Menggunakan Naïve Bayes," *e-Proceeding of Engineering*, vol. 4, no. 3, pp. 4978–4982, 2017.

[12] R. Ilyas, "Building Candidate Monolingual Parallel Corpus from Scientific Papers," in *2018 International Conference on Asian Language Processing (IALP)*, 2018, pp. 230–233.

[13] B. Prijono, "Pengenalan Recurrent Neural Network (RNN)," 2018. .

[14] S. Akba, "Penerapan Algoritma Winnowing untuk Mendeteksi Kemiripan pada Karya Tulis Mahasiswa," *Jurnal Inspiraton*, vol. 7, no. 2, pp. 131–136, 2017.

[15] P. Liu, X. Qiu, and X. Huang, "Recurrent Neural Network for Text Classification with Multi-Task Learning," *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pp. 2873–2879, 2011.

[16] J. Mueller, "Siamese Recurrent Architectures for Learning Sentence Similarity," in *Proceedings of the 30th Conference on Artificial Intelligence (AAAI 2016)*, 2016, no. 2012, pp. 2786–2792.

[17] M. J. Kusner, K. Q. Weinberger, S. Louis, and K. W. Edu, "From Word Embeddings To Document Distances," *Washington University in St. Louis*, pp. 31–40, 2015.

[18] W. Zhu, T. Yao, J. Ni, B. Wei, and Z. Lu, "Dependency-based Siamese long short-term memory network for learning sentence representations," *PLoS ONE*, vol. 13, no. 3, pp. 1–14, 2018.

[19] Z. Chen, H. Zhang, X. Zhang, and L. Zhao, "Quora Question Pairs," pp. 1–7, 2017.

[20] J. BROMLEY *et al.*, "Signature Verification Using a 'Siamese' Time Delay Neural Network," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 07, no. 04, pp. 669–688, 1993.

[21] T. Mikolov, M. Karafiát, L. Burget, and S. Khudanpur, "Recurrent neural network based language model," *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH 2010)*, no. September, pp. 1045–1048, 2010.