VIRTUAL ASSISTANT USING LSTM NETWORKS IN INDONESIAN

Mirwan
Faculty of Computer Science
Universitas Narotama
Surabaya, Indonesia
mirwan.14@fasilkom.narotama.ac.id

Aryo Nugroho
Faculty of Computer Science
Universitas Narotama
Surabaya, Indonesia
aryo.nugroho@narotama.ac.id

Ferial Hendarta
Faculty of Computer Science
Universitas Narotama
Surabaya, Indonesia
ferial.hendrata@narotama.ac.id

Rumaisah Hidayatillah
Faculty of Computer Science
Universitas Narotama
Surabaya, Indonesia
rumaisahhidayatillah.14@fasilkom.narotama.ac.id

Firdaus Hassan
Faculty of Computer Science
Universitas Narotama
Surabaya, Indonesia
firdaushassan.15@fasilkom.narotama.ac.id

Kristovel Printo Nana
Faculty of Computer Science
Universitas Narotama
Surabaya, Indonesia
kristovelprintonana.14@fasilkom.narotama.ac.id

***Abstract*—Researches into the development of virtual assistants that are human-like today were an active research that has been developed over the past few decades. Currently the generative model virtual assistant is a few steps towards artificial general intelligence, the smart computer with complex feelings and characteristics. One method for building the generative model virtual assistant is by using the LSTM Networks. This method has been popular for building English virtual assistant. Our aim is to test whether this method could be applied in Indonesian or not. We used the dataset from many movie subtitles in Indonesian languages. The LSTM variant that we used was the sequence-to-sequence model embedded with word2vec. The results show that the model could answer appropriately to the simple questions such as greetings. Although the models were failed to answer some complex questions, the results still give potential works on the future.**

Keywords— Artificial General Intelligence, Generative Models, Indonesian Languages, LSTM Networks, Virtual Assistant,

I. INTRODUCTION

The generative model virtual assistant is a smart virtual assistant who can have conversations on a broad topic. Unlike the retrieval-based model that generates an answer by repositories based on the decisions tree, this model does not use predefined responses. They generate new responses from scratch [1].

One of many deep learning methods that are capable of developing this model is a long-short-term-memories (LSTM). LSTM is a variant of the recurrent neural network, a deep learning algorithm that is capable of learning data representations in the form of sequences [2]. For the virtual assistant's needs, the data representation used is a conversation dataset which is pairs of questions and answers.

.This model is still very difficult to be applied especially for non-English languages. These are because architectural designs are complicated to build and the need of the dataset which is the conversation corpus with a large size. The dataset requirement is a crucial factor because this model need good context understanding and world knowledge to generate meaningful and relevant response [3].

This research tries to solve these problems by using the architecture's framework provided by the deep learning library and using conversation dataset collected from many movie subtitles in Indonesian. The movie subtitles are good sources for building dialogue system. It is because the movie scripts span a wide range of topics, contain long interactions with few participants and relatively few spelling mistakes and acronyms [4].

The results showing that the model could answer some questions with basic contexts such as "*halo*" (hello) and "*bagaimana kabarmu*" ("how are you"). Sometimes, the model could also generate a complete sentence in Indonesian languages. These facts indicate that this method still has further potential development in Indonesian languages.

A. Evaluation

The model was evaluated by user testing on the participant. We let the participant talk with this model and compare it with an English model. Then we give the participant five statements to be assessed on the score range of 0-10.

B. Related Work

Building virtual assistant using LSTM networks in Indonesian languages were still difficult task. There was once successful research that succeeded in developing a dialogue system which is only focused on responses generated by analogous relationships, but this research focuses only on the sensical response by the conversation [5].

II. METHOD

A. Dataset Preprocessing

The dataset was taken from several movie's subtitles in Indonesian languages. The genre mostly is Korean drama, because they contain many dialogues than other genres. But some TV series and western science fiction were also included.

The total movies that used were 8 movies, with the total conversation that consisted of 5297 pairs of questions and answers.

TABLE I. MOVIE LIST ON THE DATASET

Movie Title	Genre	Total Conversations
Work of Love	Korean Drama	209
About Time	Korean Drama	380

978-1-5386-7422-2/18/$31.00 ©2018 IEEE

What's Wrong with Secretary Kim	Korean Drama	416
Revenge Note	Korean Drama	213
Blue Bird House E.1-5	Korean Drama	2453
Friends E.7-9	Western TV Series	577
A.I.	Science Fiction	615
Ex-Machina	Science Fiction	527
Total		**5390**

Unlike an English dataset, this dataset has a problem because it does not have a mapping in each dialog. To overcome this problem, we create a mapping by using a pair of one question and one answer in each index.

We preprocess the data by cleaning the text from any punctuation and rarely words. Next, we also tokenize every word on the dataset to be the unique number.

After being cleaned, the most occurred words on the dataset were "*aku*" (I) with the total numbers 635 words, "*kau*" (you) with the total numbers 525, and "yang" (which is) with the total numbers 444 words. These words are the subject part on Indonesian sentences. These indicated that the model would have a high probability for using these words as a personals' pronoun.

B. Word Embedding

Word embedding is a computational process to build low-dimensional vector representations of the corpus text, and maintain the similarity of contextual words [6]. These words are converted into vectors by the lookup table method used for input to the artificial neural network. The word embedding process is needed because many algorithms requiring an input in the form of a vector with continuous values [7].

Compared to one-hot vector, word embedding has more advantage including not involving dense matrix multiplications. This will make the training process have more efficient and does not require a large amount of data [8].

Word embedding could also hold some information by capture meaningful syntactic and semantic regularities in a very simple way [9]. For example, if already known the words men and women are relations, then the representation vectors, "King" - "Men" + "Women" = "Queen".

One of many word embedding's architectures that support non-English language is word2vec with skip-gram variant [10]. This architecture using the middle's word on the window as a context, then predicts the surrounding words as a target. For example, in the window there is a sentence "*Halo apa kabar*" ("Hello, how are you"), the context's word would be "*apa*" and the target would be "*halo*" and "*kabar*".

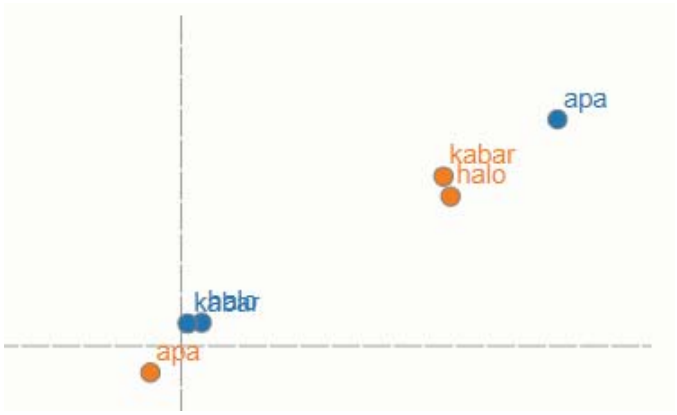


Fig. 1 The context and target words on the window

The probability of target after the context is obtained by using.

$$p(w_0|w_1) = \frac{exp(v'^{\top}_{w0}\, v_{wI})}{\sum_{w=1}^{W} exp(v'^{\top}_{w}\, v_{wI})} \qquad (1)$$

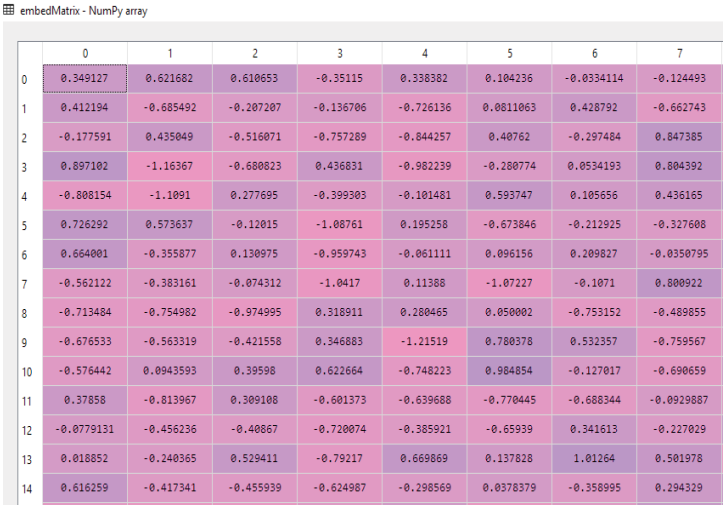
Where v_w and v'_w are the "input" and "output" vector representations of w, and W is the number of words in the vocabulary [8].

Word embedding takes about 5 minutes to train with the dataset that consisted of 5679 unique words in Indonesian language.

TABLE II. HYPERPARAMAETERS IN WORD EMBEDDING

Hyperparameter	Size
Context Window	3
Dimension	100
Negative Sample	64
Batch	128
Iteration	100000

After word embedding, the 100-dimensional trained word2vec from 5349 words In Indonesian language has been saved.


embedMatrix - NumPy array

	0	1	2	3	4	5	6	7
0	0.349127	0.621682	0.610653	-0.35115	0.338382	0.104236	-0.0334114	-0.124493
1	0.412194	-0.685492	-0.207207	-0.136706	-0.726136	0.0811063	0.428792	-0.662743
2	-0.177591	0.435049	-0.516071	-0.757289	-0.844257	0.40762	-0.297484	0.847385
3	0.897102	-1.16367	-0.680823	0.436831	-0.982239	-0.280774	0.0534193	0.804392
4	-0.808154	-1.1091	0.277695	-0.399303	-0.101481	0.593747	0.105656	0.436165
5	0.726292	0.573637	-0.12015	-1.08761	0.195258	-0.673846	-0.212925	-0.327608
6	0.664001	-0.355877	0.130975	-0.959743	-0.061111	0.096156	0.209827	-0.0350795
7	-0.562122	-0.383161	-0.074312	-1.0417	0.11388	-1.07227	-0.1071	0.800922
8	-0.713484	-0.754982	-0.974995	0.318911	0.280465	0.050002	-0.753152	-0.489855
9	-0.676533	-0.563319	-0.421558	0.346883	-1.21519	0.780378	0.532357	-0.759567
10	-0.576442	0.0943593	0.39598	0.622664	-0.748223	0.984854	-0.127017	-0.690659
11	0.37858	-0.813967	0.309108	-0.601373	-0.639688	-0.770445	-0.688344	-0.0929887
12	-0.0779131	-0.456236	-0.40867	-0.720074	-0.385921	-0.65939	0.341613	-0.227029
13	0.018852	-0.240365	0.529411	-0.79217	0.669869	0.137828	1.01264	0.501978
14	0.616259	-0.417341	-0.455939	-0.624987	-0.298569	0.0378379	-0.358995	0.294329

Fig. 2 Word2vec in 100-dimensional Vector

To facilitate the analysis process, it requires representation in the form of data visualization. Word2vec can be visualized by reducing its vectors but keep it with a similar point using embeddings projection [11]. This visualization shows the closest point based on the similarity of the contexts of several words. For example, the word "*kau*" ("you") has the closest point to the words "*mereka*" ("they") , "*kita*" ("us"), and "*aku*" ("me"). The visualization of these words also shows that plural pronouns have a higher point than singular pronouns.

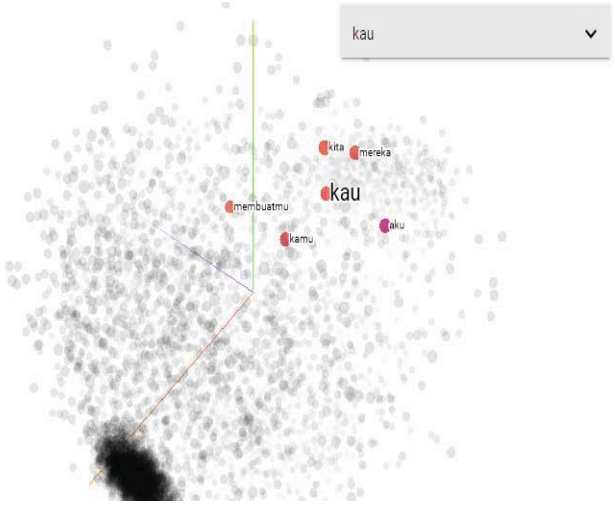


Fig. 3 Word2vec's Visualization in Indonesian Languages

C. Building The Model

The model was built using LSTM Networks with sequence to sequence variant. The sequence to sequence model is a form of framework in LSTM Networks that using the machine translation's structure [12]. This architecture consisted of two parts that are encoder to read a sentence and a decoder to generate a sentence.

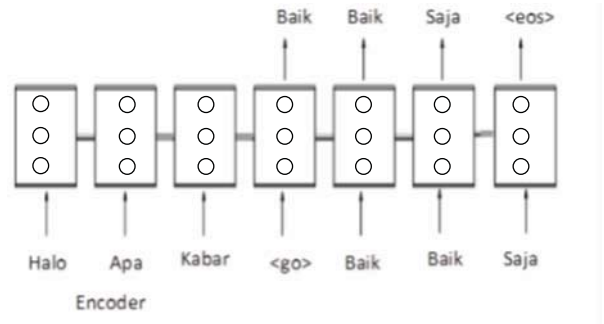


Fig. 2 Example of Sequence to Sequence in a Sentence

The goal of the LSTM is to estimate the conditional probability.

$$p(y_1,..,y'_T|x_1,..,x_T) = \prod_{t=1}^{T'} p(y_t|v,y_1,..,y_{1t-1}) \qquad (2)$$

Where $(x_1,..,x_T)$ is an input sequence and $(y_1,..,y'_T)$ is its corresponding output sequence whose length T'may differ from T[13].

The sentence in the encoder part is used in the reversed order, this would make the decoder part start generates prediction when the encoder part has been reached the first word on the sentence. The reversed sentence has been shown better for reducing the lag time and improving the prediction quality [14].

We use four special tokens that are used instead of several words and punctuation marks, namely:

TABLE III. SPECIAL TOKEN USED BY THE MODEL

Special Token	Function
<sos>	To start the sentence
<eos>	To end the sentence
<pad>	To fill the gap between words
<unk>	To replace the word that is not contained in the index.

D. Training The Model

The model was trained on the cloud using 24GB free GPU provided by Google Colab. The training process is completed in around 10 hours. We reduce the hyperparameters due to the small sizes of the dataset.

TABLE IV. HYPERPARAMETERS IN THE TRAINING PROCESS

Hyperparameter	Size
Batch sizes	12
Maximum decoder lengths	15
Maximum encoder lengths	15
Number of iterations	500000
LSTM units	112
LSTM layers	3

III. RESULTS

After the training process, the model was evaluated by user testing on the participant. The participant were limited to give only 10 questions. We also compare it with the model which is trained by the English dataset. Nevertheless, the result is pretty good due to the small sizes and unmapped dataset.

TABLE V. TESTING RESULTS

User	The Model
"*halo*" (hello)	"*hai*" (hi)
"*selamat pagi*" (good morning)	"*hai*" (hi)
"*gimana kabarmu*?" (how are you?)	"*saya sangat buruk*" (I am bad)
"*gimana kabarmu*" (how are you)	"*lepaskan* ("release")
"*bagaimana kabarmu*" (how are you)	"*saya baik*" (I am fine)
"*kamu kuliah dimana*" (where do you study)	"*terima kasih*" (thank you)
"*kamu kuliah di mana*" (where do you study)	"*mari kita bicara*" (let we talking about)
"*lalu*" (then)	"*lebih baik*" (better)

From the results on the table, the model could answer greeting such as hello and good morning appropriately. But when the question was turned into more complex, the model has missed its context, for instance, the question "where are you study" was responded by "thank you" which is nonsensical respond.

TABLE VI. QUESTIONNAIRE AND ASSESSMENT BY PARTICIPANTS WITH A SCORE RANGE OF 0-10

Statement	The Model	The Model using English dataset
"The virtual assistant was entertaining"	5	5
"The virtual assistant was natural"	5	5
"The virtual assistant has an empathy"	4	4
"The virtual assistant was easily understood"	3	4
" The virtual assistant could give an appropriate answer"	3	4
Averages	**4**	**4.4**

IV. CONCLUSIONS

After conducting research on the implementation of the LSTM networks model in the development of this virtual assistant, it can be concluded that the development of generative model can still be done in Indonesian. Although the model could only answer questions with a basic context, this still gives hope for the further development of this method. The difference in score between the English model and Indonesian were only 0.4 points. This fact shows that the output of model architecture doesn't affect by dataset language. Some errors such as context problems are caused by lack of training time and dataset preprocessing. The unmapped dataset also influenced the training process. Future works in this field should increase the dataset's size and training time.

REFERENCES

[1] A. Bhattacharya, "Generative Conversational Agents The State-of-the-Art and the Future of Intelligent Conversational Systems," Int. J. Recent Innov. Trends Comput. Commun., vol. 5, no. 5, pp. 817–821, 2017.

[2] A. Pradipta Gema and D. Suhartono, "Recurrent Neural Network (RNN) dan Gated Recurrent Unit (GRU)", School of Computer Science, 2018. [Online]. Available: http://socs.binus.ac.id/2017/02/13/rnn-dan-gru/. [Accessed: 11- Nov- 2018].

[3] M. K. Chinnakotla, "Building Conversational Agents using Deep Learning," Adv. Top. AI (Spring 2017) IIT Delhi Manoj, no. Spring, 2017.

[4] I. V. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau, "Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models," arXiv:1507.04808v3, p. 4, 2015.

[5] A. Chowanda and A. D. Chowanda, "Recurrent Neural Network to Deep Learn Conversation in Indonesian," Procedia Comput. Sci., vol. 116, pp. 579–586, 2017.

[6] J. Collis, "Glossary of Deep Learning: Word Embedding – Deeper Learning – Medium", Medium, 2018. [Online]. Available: https://medium.com/deeper-learning/glossary-of-deep-learning-word-embedding-f90c3cec34ca. [Accessed: 06- Aug- 2018]..

[7] J. Brownlee, "What Are Word Embeddings for Text?", Machine Learning Mastery, 2018. [Online]. Available: https://machinelearningmastery.com/what-are-word-embeddings/. [Accessed: 19- Oct- 2018].

[8] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," p. 1, 2013.

[9] T. Mikolov, W. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," Proc. NAACL-HLT, no. June, pp. 746–751, 2013.

[10] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning Word Vectors for 157 Languages," http://arxiv.org/abs/1802.06893v1, p. 1, Feb. 2018.

[11] Brownlee, "What Are Word Embeddings for Text?", Machine Learning Mastery, 2018. [Online]. Available: https://machinelearningmastery.com/what-are-word-embeddings/. [Accessed: 19- Oct- 2018].

[12] J. Brownlee, "How to Define an Encoder-Decoder Sequence-to-Sequence Model for Neural Machine Translation in Keras - Machine Learning Mastery," 2017. [Online]. Available: https://machinelearningmastery.com/define-encoder-decoder-sequence-sequence-model-neural-machine-translation-keras/. [Accessed: 21-Mar-2018].

[13] I. Sutskever, "Sequence to Sequence Learning with Neural Networks," eprint arXiv:1409.3215, pp. 1–9, 2014.

[14] K. Filippova, E. Alfonseca, C. A. Colmenares, L. Kaiser, and O. Vinyals, "Sentence Compression by Deletion with LSTMs," Proc. 2015 Conf. Empir. Methods Nat. Lang. Process., pp. 360–368, 2015.