



Increased Accuracy on Sequence-to-Sequence Models with the CNN Algorithm for Multi-Response Ranking on Travel Service Conversations Based on Chat History

Wahyu Wijaya Widiyanto^{1*}, Uli Rizki², Edy Susena³

^{1,3} Polytechnic Indonusa Surakarta

² Universitas Amikom Jogjakarta

^{1,3} Street KH. Samanhudi, Surakarta 57148, Central Java, Indonesia

² Street Ring Road Utara, Ngringin, Condongcatur, Kec. Depok, Kabupaten Sleman, Daerah Istimewa Yogyakarta 55281, Indonesia

*Corresponding email: wahyuwijaya@poltekindonusa.ac.id

Received 06 May 2020, Revised 20 May 2020, Accepted 22 May 2020

Abstract — Building a chatbot cannot be separated from the knowledge base. The knowledge base can be obtained from data that has been labeled by the developer, documents that have been converted into pre-processing data, or information taken from social media. In this case, the data used as knowledge is the chat history. In the chat history, there are certainly many variations of answers and allowing a question to give rise to many answers. The ranking system was made to overcome these multi-responses. The response provided for the user will be more desirable by utilizing the existence of ranking. The challenge in making the ranking is how to get the essence of a question and to find pairs of question-and-answer from the data. This can be solved by using a sequence-to-sequence model. However, the problem that will arise is the consistency of the answers. The existence of a lot of chat history certainly raises many explanations, even though the question's essence is the same. For this reason, the CNN algorithm will be used as a solution to the problem. This research uses convolutional sequence-to-sequence, which will be applied for ranking responses. The efficiency of the model is also compared. By making comparisons, this model shows an accuracy of 86.7%.

Keywords – chatbot, CNN algorithm, comparisons, sequence-to-sequence model, travel

Copyright © 2020 JURNAL INFOTEL

All rights reserved.

I. INTRODUCTION

Chatbot is a program derived from artificial intelligence for simulating interactive conversations with users (humans) through text, voice, or visuals. In recognizing and responding naturally to human speech, chatbot uses the Natural Language Processing (NLP) approach. The challenges in implementing NLP include the ambiguity of natural language, knowledge representation, levels of information, and various applications for language technology [1].

Various approaches and models are proposed to increase the efficiency and accuracy in recognizing conversation patterns and in providing answers quickly from the knowledge base of human and machine conversation. One of these approaches is the encoding of text pairs using lexical, syntactic, and semantic

features, which then the similarity degree between representations in a sentence are being calculated [2].

To respond conversation naturally like how human response, chatbot relies on a knowledge base that has been arranged or input from the developer. The problem is that it might need a lot of knowledge. To improve the accuracy, further labeling process on each sentence pattern will be performed. It might be a tiring job if the data size is medium-sized, but it might become impractical if the data size is in industrial level [3].

Another approach is to use a sequence-to-sequence framework or the context of the conversation context. This method is implemented when using data derived from group conversations or specific chat history [4]. Of course, there will be many different answers in a chat group conversation depending on the person

answering. The existence of a sequence will help to find proper answers from the same questions. But this sequencing method has a weakness in response consistency. The response given depends on the order at a time updated so that with the same question at a different time, different responses might be given [5].

The problem of consistency can be resolved by ranking the response using the CNN algorithm [3]. Another feature related to content approach is added to increase the relevance in accordance with user requests [6]. The content approach takes user profiles, both from the user information that has been inputted to the system and from the social media accounts related to the user's profile. User profiles that are linked to posted contents, responses, and personal information are used as consideration for the user's perspective so that the responses will be chosen more precisely for certain people.

The number of previous studies and the many references regarding ranking chatbot responses used the CNN algorithm with several models. However, from most prior studies, it can be understood that these studies use the dataset from a broader conversation and from the discussion that was not yet specific [7]. Therefore, this study uses the sequence-to-sequence approach model and the CNN algorithm with the data knowledge based on chat history in creating the ranking of conversation response [8].

II. RESEARCH METHOD

A. Type of Research

A chatbot is not a new thing in machine learning nowadays; it has been implemented in many online service industries. Lots of research tries to emulate a conversation system that acts like a natural human conversation. There is research with various models and approaches that can retrieve information from the knowledge base provided within the chatbot system and have taken information from the outside of the system even taking personal profiles from users' social media accounts embedded in the chatbot system. In this case, this research was conducted by applying the model to chatbot responses ranking using chat history as the chatbot's knowledge base.

B. Nature of Research

This research has causality nature, where the model applied gives the results of how this approach works. The accuracy of the model used in the study also is demonstrated [9].

C. Research Approach

In the learning systems, many methods can be used depending on the problem. Let's start with Natural Language Processing and Deep Learning. These two techniques can be mapped as solution branches of chatbots problem, as shown in Fig. 1.

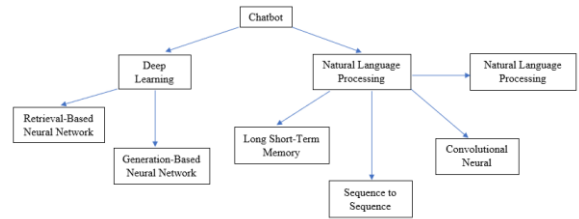


Fig. 1. Conceptual Map of Topics

There are many methods of reading the information in a chatbot, as shown in Fig. 1. Still, this research will be devoted to the use of the sequence-to-sequence, and Convolutional Neural Network methods. The sequence-to-sequence method will be used to process the order of a sentence and the CNN algorithm will be used its hidden layer to count or to rank sentences as the most relevant response candidate [10].

The first step taken to rank responses to a conversation system is to break up a sentence based on word order. The next step is to filter them using the CNN algorithm with several hidden layers to see or to find the most relevant words to the knowledge base, in this case, the chat history, as shown in Fig. 2.

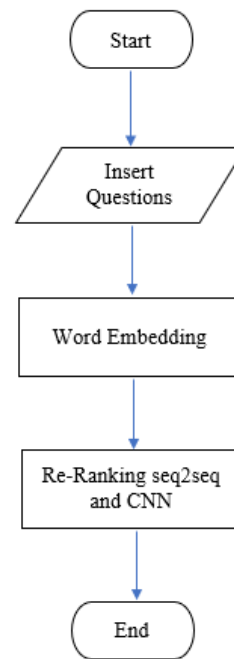


Fig. 2. Ranking Process Flow

The re-ranking model based on sequence-to-sequence, and CNN adopts the research of Abstract Text Summarization with a Convolutional sequence-to-sequence. First, a sentence from the user enters the input layer, where the sentence processing will be performed in this stage. Next, the result will enter the Hidden State Word for ranking, which is supported by the Hidden State Sentence Layer for the filtering process, as shown in Fig. 3.

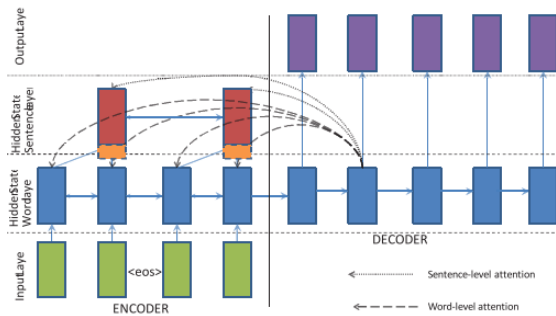


Fig. 3. Ranking Response Mechanisms

D. Data Collection Method

In researching ranking the responses, various supporting components are needed for the smooth running of research. One of the supporting components that cannot be separated is data as test material. One of the trial materials used in this study was a chat history dataset from online ticket sales services at a startup company in Yogyakarta [8].

There are more than 10000 conversations for airline ticket sales services. The chat history data consists of 10 customer service chat history from 2018.

E. Data Analysis Method

Let a dataset with N data components $(x^{(i)}, t^{(i)})$ consists of source text represented as $x^{(i)}$ and target keywords M_i or key sentences symbolized as $t^{(i)} = (t^{(i),1}, t^{(i),2}, \dots, t^{(i),M_i})$.

For the input sequence $x = (x_1, \dots, x_m)$, we first state it as a low dimensional vector $u = (u_1, \dots, u_m)$, where $u_j \in R^d$. As for position embeddings, we first obtain a vector to record the absolute position of an element in a sequence. Then we use the embedding layer to transform these rare and discrete representations into continuous embeddings as $p = (p_1, \dots, p_m)$ where $p_j \in R^d$, so it is possible to know which part of the sequence is being processed. d is the dimension of the embeddings position and the input sequence element embeddings. We use a combination of both embeddings to form input $e = (u_1 + p_1, \dots, u_m + p_m)$ element embeddings. We also add the embeddings position to the decoder output element embeddings, and then return to the decoder

For the Convolutional Sequence-to-Sequence Model, we divide the convolutional layer architecture to the encoder and decoder, which will calculate the intermediary status through the input element. We represent the output of the 1st layer as $Z^1 = (Z^1_1, \dots, Z^1_m)$ for the encoder and for the decoder. Each layer consists of one-dimensional convolution and $h^1 = (h^1_1, \dots, h^1_m)$ non-linearity. If the decoder has one layer with the width of the kernel being k , then the output h^1 will compress the input element information k . To increase the length of input elements, we stack blocks with each other, for example, stacking six blocks with $k = 5$ can represent

25 input elements. If needed, non-linearity allows our model to handle the entire sequence of inputs or only a few elements. The computational advantage of our model is that it performs parallel computing, which is far more efficient than traditional RNN models that are computed one element by one element. As mentioned in Part 1, to represent sequences in words, CNN only requires operations $O_{(n/k)}$, whereas RNN requires operations. However, this hierarchical CNN model is not as efficient as traditional CNN models because our model has to stack layers of CNN to represent a more expressive sequence. In contrast, traditional CNN models only require one layer to explore the whole sequence.

F. Research Flow

The research stages are shown in the following Fig. 4. The first stage is to take references from books, journals, papers, and existing research as literature studies. After getting the references from the literature study, the researchers then collected the dataset. The collected dataset itself came from the downloaded chat log data in e-mail. The dataset is then preprocessed for a conversation test. The conversation log is then merged, and the results are stored in a database. The data stored in the database is then carried out to separate between questions and answer pairs using the sequence-to-sequence and the CNN. The results of the separation process are compared between the sequence-to-sequence and the CNN to get the accuracy results.

Severyn et al. conducted a study of the CNN architecture to compile short text pairs and its similarity functions to note the words that form the basis of knowledge based on the available training data. The method used can improve the previous state-of-art system, which is 3% absolute points in Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR). This research also answers the challenges of the TREC question and answer forum [11].

Wu, et al also conducted research using the Sequential Matching Network (SVM) approach. This study matches the response with each context in a statement from the user and filters relevant matches as vectors with collection operations. The vector is accumulated in a sequence through the Recurrent Neural Network (RNN), which models the relationship between speech [5].

Yin et al conducted a study combining convolutional layers by understanding semantic cues in conversation based on effective representations for each sentence. In this case, the researcher also added a GRU-based layer to build the candidate ranking model so that it took sentence modeling using CNN and RNN as sequence modeling. This model gets an accuracy of 78.6% [12].

The related research also implements the question-and-answer technique in e-commerce. In a question-and-answer conversation, the problem that arises is to

match the source of knowledge with natural language processing. For this reason, researchers propose a new framework that can effectively and efficiently adjust the knowledge shared from domains rich in knowledge sources to domains lacking in knowledge sources. As a result, the performance of e-commerce services increased with $F1 = 0.681$, with a time of 80.7 per query [13].

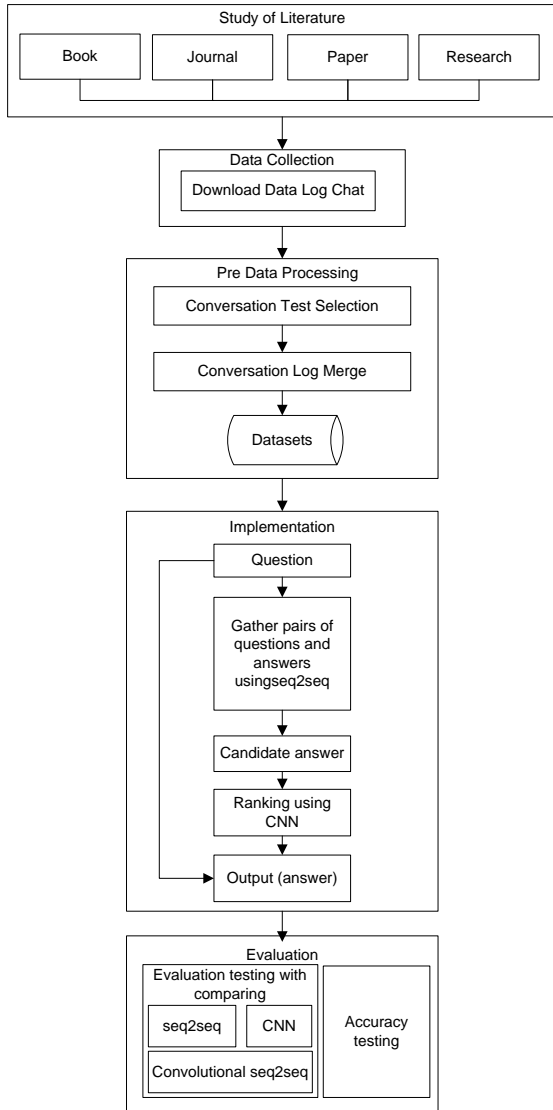


Fig. 4. Research Flow

III. RESULT

A. Dataset

First, we collect the data from the email chat logs with the extension .eml; it can be seen in Fig. 5.

```

[04:03] HENIWATY: hai
[04:03] HERY vitrianadewi: selamat pagi, saya Heryv travel assistant anda sekarang ada yang bisa kami bantu?
[04:04] HENIWATY: saya mau tiket.....nama heniwaty
[04:04] HERY vitrianadewi: pilihan nama depan heniwaty dan belakang tetap heniwaty ya?
[04:04] HERY vitrianadewi: iya kalau memang sesuai ktp namanya hanya 1 kata
[04:04] HERY vitrianadewi: ditulis sama
[04:05] HENIWATY: confirm ya jadi ditulisnya heniwaty heniwaty?
[04:05] HERY vitrianadewi: nama depan : heniwaty
[04:05] HERY vitrianadewi: nama belakang : heniwaty
[04:05] HENIWATY: okey...tks
[04:05] HERY vitrianadewi: baik, sama - sama.
    
```

Fig. 5. Chat Datasets from The Email Log

To merge the conversation texts, the authors use python language for making the code scripts. After the merging process, the data format can be seen in Fig. 6.

```

Message: hai
Response: selamat pagi, saya Heryv travel assistant anda sekarang ada yang bisa kami bantu?
Message: saya mau tiket.....nama heniwaty pilihan nama depan heniwaty dan belakang tetap heniwaty ya?
Response: iya kalau memang sesuai ktp namanya hanya 1 kata ditulis sama
Message: confirm ya jadi ditulisnya heniwaty heniwaty?
Response: nama depan : heniwaty nama belakang : heniwaty
Message: okey...tks
Response: baik, sama - sama.
    
```

Fig. 6. Chat that has been Combined

B. Dataset

To produce word vectors, we use the classic approach of the Word2Vec model. The basic idea is that the model creates word vectors by looking at the context with words that appear in sentences. Words with the same context will be placed close together in vector space.

C. Sequence-to-sequence Model

After pre-processing the data, it enters the implementation by identifying questions raised by chatbot users. For example, users ask questions like "Does the price include baggage?"

In this process the question will be converted that can be seen in Table 1 as follows.

Table 1. Identify the Question

Does	The	Price	Include	Luggage	?
h1	h2	h3	h4	h5	<EOS>

Then by using the formula described in the data analysis method, the question will be used as an encoder. And then the machine will find the pairs of question-and-answer. The answer is formed on the decoder format that can be seen in Table 2.

Table 2. Sequence-to-Sequence Encoder and Decoder Form

Does	The	Price	Include	Luggage
h1	h2	h3	h4	h5
?	Citilink	Has	Include	Luggage
<EOS>	y1	y2	y3	y4

IV. DISCUSSION

After checking the pair of question-and-answer, the candidate responses will be ranked. However, in the ranking process, the questions will be filtered again using the CNN algorithm.

In this stage, all sentences, both questions and candidate answers, are converted into vectors and padding to equalize sentence lengths. This process will get a word representation taken from the embedding layer; if the vocabulary is not included in the embedding, a value of 0 will be assigned. As explained previously in the word embedding stage, the length of the sentence is obtained from the longest sentence in the dataset. The longest sentence in the dataset is 31. So, the sentence matrix will have a form of 31x31 for all

questions and candidate answers. The changes in the embedding layer can be shown in Table 3.

Table 3. Matrix Question

Index	Word	p ⁰	p ¹	p ²	p ³	...	p ³⁰
0	If	0.09	0.04	0.03	0.09	...	0.01
1	I	0.08	0.03	0.01	0.01	...	0.08
2	want	0.13	0.02	0.01	0.01	...	0.08
3	refund	0.07	0.09	0.07	0.05	...	0.05
4	can	0.07	0.01	0.01	0.01	...	0.03
5	it	0.01	0.06	0.03	0.01	...	0.01
...
30	<pad>	0.05	0.06	0.04	0.01	...	0.08

Then the sentence is converted into a matrix. It will be introduced with the convolution feature maps, which use parameter filter 3 with stride 1. The filter used is a random number from -1 to 1. The matrix can be seen in Table 4.

Table 4. 3x30 Kernel Matrix

	k _{.,0}	k _{.,1}	k _{.,2}	k _{.,3}	...	k _{.,30}
k _{0,.}	-0.04	-0.23	-0.79	0.81	...	0.60
k _{1,.}	0.01	-0.09	0.41	0.83	...	-0.9
k _{2,.}	0.47	0.59	0.52	0.73	...	0.52

The results of multiplying with the kernel are shown in Table 5.

Table 5. Convolution Feature Question

Feature	Value
0	-0.013
1	-0.078
2	-0.014
3	-0.14
4	0.012
...
28	-0.08

The matrix takes a 2-columns format. For example, in Table 5, the column with a grey background is compared between index 0 and index 1. The highest value between the two numbers will then be taken. Thus, the first max-pooling result is obtained. The same calculation will be continued until index 28, which will produce a 1x14 matrix.

Table 6. Max Polling Question

Feature	Value
0	-0.013
1	0.120
3	0.014
4	0.040

Feature	Value
5	-0.020
...
13	-0.088

The last stage is the Fully Connected Layer, which is to calculate the join layer matrix, which will be used as a ranking value. At this stage, the join layer matrix will be multiplied by the initialized weight. Table 7 shows the calculation results of the join layer to the output for the first candidate.

Table 7. Full Connected Layer

Matrix	Weight	Hidden layer	Bias	ReLU
-0.013	-0.49	0.63	0.06	0.063
0.120	-0.23	0.29	0.02	0.029
0.014	-0.79	1.00	0.10	0.100
0.040	0.81	-1.0	-0.1	0
-0.020	-0.26	0.34	0.03	0.034

For the evaluation, the test data used contains 1000 question-and-answer pairs. Each question consists of two answers, one right, and the other wrong. In this case, the rating involves customer service to choose the right answer. Table 8 is a test data used to measure accuracy.

Table 8. Chatbot Test Result

No	Datatest	Result
1	I have bought a flight ticket from Makassar to Yogyakarta on April 8 with the Sriwijaya Air ticket. Can the ticket be replaced with Lion Air with the same emotion Sokusix how? R1: If you switch airlines, you cannot Q: Can there be additional costs for the booking code?	1
2	have bought a flight ticket from Makassar to Yogyakarta on April 8 with the Sriwijaya Air ticket. Can the ticket be replaced with Lion Air with the same emotion Sokusix how? R1: If you switch airlines, you cannot Q: Can there be additional costs for the booking code?	1
3	I Thia want to reschedule my Citilink ticket so it can be morning hours to CGK R1: Good night Mrs. Thia, you can help with the booking code R0: For the same date, only hours	1
...
1000	I Thia want to reschedule my Citilink ticket so it can be morning hours to CGK R1: Good night Mrs. Thia, you can help with the booking code R0: For the same date, only hours	0

$$\text{Accuracy} = \frac{\# \text{ of correctly reconigzed labels}}{\text{total \# of label}} \quad [14]$$

Using a dataset from the conversation history; the proposed model improves the accuracy of the system. If the system solely relies on the sequence-to-sequence, the accuracy becomes lower because of the many different candidate answers. By adding the CNN technique, the system achieves higher accuracy in response ranking, as shown in Table 9.

Table 9. Test Model

Model	Accuracy
Sequence-to-sequence	68.20%
CNN	84.10%
Conv-sequence-to-sequence	86.7%

V. CONCLUSSION

By combining the two models, the sequence-to-sequence and the CNN based on chat history, the accuracy can be improved by 86.7%. It is possible to get repeated answers according to the research if the system solely relies on the sequence-to-sequence technique.

REFERENCES

- [1] Y. P. Ruan, Z. H. Ling, X. Zhu, Q. Liu, and J. C. Gu, "Generating diverse conversation responses by creating and ranking multiple candidates," *Comput. Speech Lang.*, vol. 62, p. 101071, 2020.
- [2] R. Yan, D. Zhao, and E. Weinan, "Joint Learning of response ranking and next utterance suggestion in human-computer conversation system," *SIGIR 2017 - Proc. 40th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, pp. 685–694, 2017.
- [3] A. Chaturvedi et al., "CNN for Text-Based Multiple Choice Question Answering," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2019.
- [4] D. Tao, X. Li, X. Wu, W. Hu, and S. J. Maybank, "Supervised tensor learning," *Knowl. Inf. Syst.*, vol. 13, no. 1, pp. 1–42, 2007.
- [5] Y. Wu, W. Wu, C. Xing, Z. Li, and M. Zhou, "Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots," *ACL 2017 - 55th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf. (Long Pap.)*, vol. 1, pp. 496–505, 2017.
- [6] L. Yang et al., "Response ranking with deep matching networks and external knowledge in information-seeking conversation systems," *41st Int. ACM SIGIR Conf. Res. Dev. Inf. Retrieval, SIGIR 2018*, pp. 245–254, 2018.
- [7] H. Ouchi and Y. Tsuboi, "Addressee and response selection for multi-party conversation," *EMNLP 2016 - Conf. Empir. Methods Nat. Lang. Process. Proc.*, no. Section 3, pp. 2133–2143, 2016.
- [8] W. W. Widiyanto, S. Pariyanto, and D. Iskandar, "Prototype Analysis of Basic Words in Social Media in Indonesia," *Asian J. Res. Comput. Sci.*, vol. 5, no. 1, pp. 49–58, 2020.
- [9] B. Wu, B. Wang, and H. Xue, "Ranking responses oriented to conversational relevance in chat-bots," *COLING 2016 - 26th Int. Conf. Comput. Linguist. Proc. COLING 2016 Tech. Pap.*, pp. 652–662, 2016.
- [10] I. Shalyminov, O. Dušek, and O. Lemon, "Neural Response Ranking for Social Conversation: A Data-Efficient Approach," in *Proceeding of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI*, pp. 1–8, 2018.
- [11] A. Severyn, "Rank with CNN," 2014.
- [12] W. Yin, H. Schütze, B. Xiang, and B. Zhou, "Erratum: 'ABCNN: Attention-Based Convolutional Neural Network for Modeling Sentence Pairs,'" *Trans. Assoc. Comput. Linguist.*, vol. 4, pp. 566–567, 2016.
- [13] J. Yu et al., "Modelling domain relationships for transfer learning on retrieval-based question answering systems in E-commerce," *WSDM 2018 - Proc. 11th ACM Int. Conf. Web Search Data Min.*, vol. 2018-Febua, pp. 682–690, 2018.
- [14] H. Zhu, H. Chen, and R. Brown, "A sequence-to-sequence model-based deep learning approach for recognizing activity of daily living for senior care," *J. Biomed. Inform.*, vol. 84, no. February, pp. 148–158, 2018.