2023 The 23rd International Conference on Control, Automation and Systems (ICCAS 2023)
Yeosu Venezia Hotel & Utop Marina Hotel, Yeosu, Korea, Oct. 17∼20, 2023

Persona-based Dialogue Generation with Sentence Embedding for Prolonged Human-Robot Communication

Chen Yue, Armagan Elibol, and Nak Young Chong*

School of Information Science,
Japan Advanced Institute of Science and Technology
Ishikawa, Japan

* Corresponding author,(nakyoung@jaist.ac.jp)

Abstract: It is one of the most fundamental and challenging problems for current dialogue generation systems to maintain the consistency of dialogue logic during the conversation. Also, the lack of an open-source annotated persona-based dialogue dataset may lead to insufficient training volume for the model. Besides, the computational time and computational memory required by the attention mechanism have become drastically large. In order to overcome these problems, in this work, we propose a vertical-structure model based on the BERT model with the sentence embedding method. The model generates a raw response based on the sentence embeddings of context and persona and finally revises the raw response according to the persona. Moreover, an understanding task is designed for the BERT decoder to have a better revision ability. Considering the difference between the generation and the understanding models, three kinds of input methods are designed for each part of the model. Comparative and experimental results are presented using publicly available datasets.

Keywords: Human-Robot Interaction, Dialogue generation.

1. INTRODUCTION

Human-computer interaction (HCI) [1], as a technical bridge for communication between humans and computers in the current information age, has received widespread attention from academia and industry. Thanks to the development of new-generation artificial intelligence technologies such as computer vision [2] and natural language processing [3], HCI technology has been developed, leading to significant improvements in both the interaction process and user experience. However, it is still an important and challenging task in natural language processing to make machines have an engaging communication ability on par with humans, as illustrated in Fig. 1.

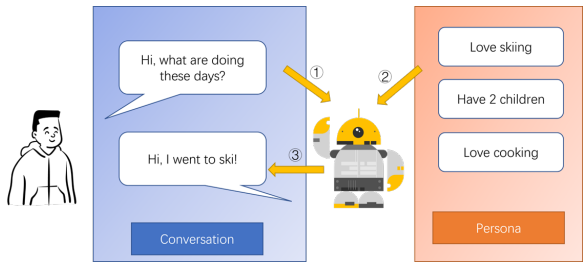

Fig. 1. Example of persona-aware human-robot conversation

According to specific system construction goals, conversational systems in natural language processing can be broadly classified into task-oriented closed-domain [4] and non-task-oriented open-domain conversational systems [5]. Among them, task-oriented closed-domain conversational systems, such as QA systems [6], are designed to accomplish specific tasks and are mostly used for website customer service and cell phone assistants. Non-task-oriented open-domain dialogue systems, on the other hand, allow the system to be “freer” and generate meaningful responses based on the relevant information received.

The main objective of our work focuses on an open-domain dialogue system. In some previous work, all contexts including the persona and dialogue history are concatenated as input. Although some good experiment results can be achieved, this kind of method dealing with the data may lead to a problem that the response generated by the model cannot establish a long-term logical relationship with the context and contradict the input. For example, in LSTM [7], particular hidden state inputs are set separately to solve the problem of forgetting historical conversation information. But this method is only suitable for short-term or medium-term conversations. Therefore, we focus on the usage method of the context, reuse the persona information and try to make our model understand the content of the preceding and following texts and establish corresponding logical dependencies. Recent studies use a variant based on the Transformer model [8]. Then based on the Transformer’s features, the longer the model input is, the larger the computational resources required. For regular use, the excessive computational resources are also more detrimental to the layout of the model and cannot be used on daily devices, and this reduces the applicability of the work. In order to overcome such computational issues, the sentence embedding method is used in this work. Moreover, applying the sentence embedding method benefits the response speed in real-time communication essential for social robots naturally interacting with people. Also, richer context information is expected to give a less unambiguous response. Borrowing ideas from the previous work in [9], a weighted average of the first and last layer outputs of BERT [10] is performed in order to obtain sentence embedding. As the computation of the BERT

model is conducted layer by layer, the BERT model is ensured to pay attention to the whole sentences instead of words. In this sentence embedding method, a sentence embedding that has a balance between words and the whole sentence can be obtained.

2. RELATED WORK

2.1 BERT

BERT comes from Transformer's encoder and forms a new bidirectional language model independently. Before the BERT model was proposed, most language models were unidirectional language models that input a text sequence from left to right. Alternatively, some models combined the left-to-right training with the right-to-left training together (*e.g.*, bidirectional RNN [11]) to form a temporary bidirectional language model. There is a particular point where the BERT embedding layer accepts the input word vector, automatically generates the corresponding segment embedding and position embedding, and then adds them together to obtain a new word embedding. Adding the three is more like a feature fusion, where the information of the word vector, the segmentation information, and the position information are fused together to obtain a new word embedding. Also, with the introduction of the Transformer, the attention mechanism has been widely used. This attention mechanism is also used in conjunction with the mask mechanism. For some special goals, in the masked case, the attention mechanism cannot compute some of the later masked content, which enables unidirectional attention to the input. The idea, used in such as MASS [12], UniLM [13], and similar others., is to enable BERT to perform natural language generation by modifying the masking mechanism. These two kinds of research allow BERT to generate high-quality textual content while fully understanding the context. In this study, benefiting from such a masking mechanism, we can perform text generation with BERT and ensure coordinated cooperation among the various sub-models.

2.2 Encoder-Decoder Model

In the encoder-decoder model framework, different kinds of pre-trained language models can be chosen as encoders and decoders for different tasks as reported in [14]. This freedom also dramatically increases the scalability of the encoder-decoder model framework and solves the gradient disappearance problem mentioned in [15]. Compared with BERT, GPT2, and Roberta, the BERT2BERT model was successfully used in [16] to generate advertisement text. The quality of the generated text is comparably good.

2.3 Sentence Embedding Methods

Instead of word embedding methods, sentence embedding methods are an option to try to solve the problem of computational memory and weight allocation. Sentence embedding is essentially the same as word embedding. A word embedding uses a multi-dimensional vector to represent a word, while a sentence embedding uses a multi-dimensional vector to represent a sentence.

2.4 Persona-based Dialogue System

There are several works on persona-based dialogue systems, such as P^2BOT [17]. In P^2BOT, in order to be able to create chat agents that can generate conversations based on the robot's personality, a structure that combines a transmitter and a receiver is proposed. The transmitter receives personalization settings, conversation history, and current conversation utterances as input and then outputs a reply based on the conversation history and personalization settings. The receiver, meanwhile, will judge whether the reply is correct, which implies the way to revise the response generated by the model.

3. PROPOSED MODEL

The schematic flow diagram of the proposed dialogue generation model is given in Fig. 2, consisting of three modules described below.

3.1 BERT Encoder

The BERT encoder in the model is mainly designed to obtain sentence embeddings of the input utterances. Sentence embedding methods, like SentenceBERT [18] or SIF [19], are mostly used for natural language understanding and are not suitable for natural language generation. Compared with SentenceBERT or SIF, using original BERT to encode the input can lead to saving a huge amount of training resources, and the speed and quality of obtaining the sentence embedding of the input utterance remain similar.

3.2 Encoder-Decoder Model

The Encoder-Decoder model is designed to generate the corresponding responses based on historical conversation information and persona. In order to maintain the overall model's uniformity or to reduce the influence which may be caused by different vocabulary files of a different model, besides other models, like BART [20], T5 [21]. The encoder-decoder model structure of BERT2BERT is adopted and a cross-entropy loss function is taken to compare the generated responses with the standard labels to get the loss. Then the model is trained.

3.3 BERT Decoder

The BERT decoder is designed to modify the responses generated by the encoder-decoder model based on the information from the persona. Nowadays, most chit-chat models still have the problem that the generated utterances contradict the persona in persona-based conversation. In our work, a separate BERT decoder is designed at the end of the whole model in order to make the utterances generated by the encoder-decoder model not conflict with the persona. If the relationship between generated sentence and persona is entailment or neutral,

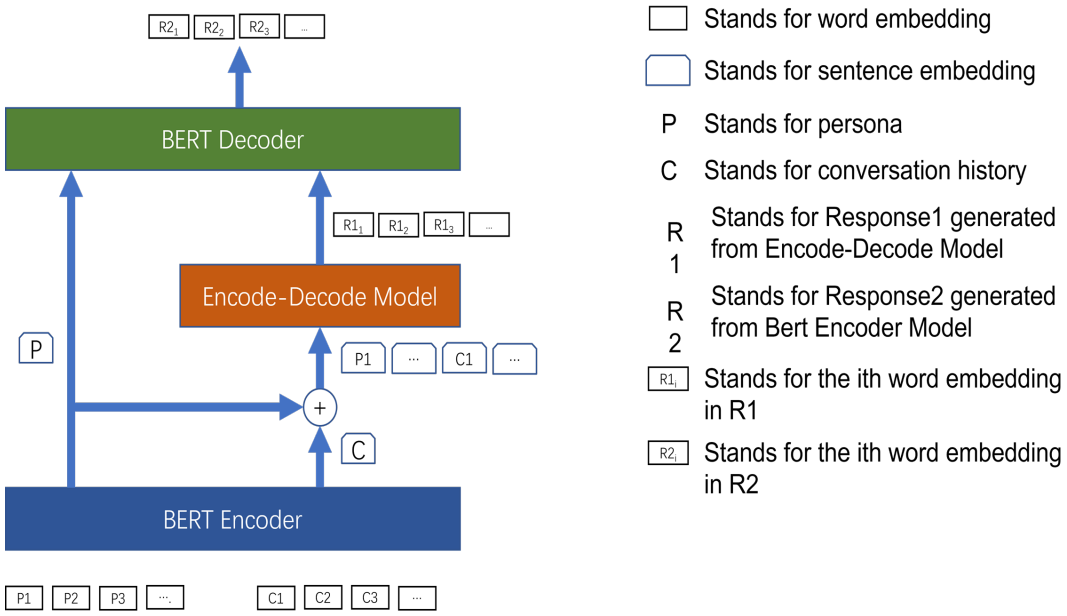

Fig. 2. The persona-based dialogue generation model has three parts: BERT encoder, BERT decoder, and Encoder-Decoder Model. The interaction scenario of the model is shown in Fig. 1. Since this study has three different sub-models, we designed three different input methods for the BERT decoder and Encoder-Decoder models to investigate how the input methods affect the generated results. Notably, in addition to the dialogue generation task, an additional understanding task is added to overcome the problem of too few data samples in the dataset and to be able to add some noise to the model

then the BERT decoder will directly output the generated utterance. If not, then the BERT decoder will revise the generated utterance. However, the BERT decoder at the end of the model is not the same as the BERT decoder in the encoder-decoder model. The BERT decoder in the Encoder-Decoder model is modified from a bidirectional encoder for natural language understanding to a causal LM for natural language generation with a specific masking mechanism, similar to GPT2. Nevertheless, the endmost BERT decoder does not use the masking mechanism, and it still uses the BERT model's original bidirectional attention mechanism. After encoding the persona and the generated utterance, the modified generated sentence is output through a fully connected neural network layer. In terms of the operation of this BERT decoder, it can be treated as a bidirectional decoder.

3.4 Persona-based Dialogue Generation

Persona-based dialogue generation is the main task of this study. First, in this task, the robot's persona and dialogue utterances are input into the BERT encoder. The BERT encoder encodes the robot's persona and dialogue utterances into word embeddings. Then, based on the obtained word embeddings, the corresponding sentence embeddings are obtained after a computation. After that, the sentence embedding of the robot's persona and the sentence embedding of the dialogue utterance are fed into the Encoder-Decoder model, which generates the relevant responses based on the information of both like Eq. 1.

$$R_{1,i} = \text{FNN}\left(\text{Enocder-Decoder Model}(p, c, R_{1,<i})\right) \quad (1)$$

where FNN stands for fully connected neural network, p stands for the sentence embedding of persona, c stands for the sentence embedding of dialogue history, $R_{1,<i}$ stands for the word embedding of generated response words which are before i^{th} words. Considering that the generated vectors need to be converted into text by looking up the vocabulary file, the representation of the generated responses is still word embedding rather than sentence embedding. Here, for the Encoder-Decoder model, the cross-entropy loss function is used as in Eq. 2.

$$Loss_1 = \text{Cross Entropy}\left(\text{FNN}(R_1, Label)\right) \quad (2)$$

Finally, the persona of the robot represented by the sentence embedding and the obtained raw response represented by the word embedding are both inputs to the BERT decoder, which computationally determines the textual relationship between the two and subsequently outputs the modified final response, like Eq. 3.

$$R_2 = \text{FNN}\left(\text{BERT Decoder}(p, R_1)\right) \quad (3)$$

where R_1 stands for the generated response by Encoder-Decoder Model, and R_2 stands for the modified response by BERT Decoder. Similarly, a cross-entropy loss function is also used for the BERT decoder as Eq. 4

$$Loss_2 = \text{Cross Entropy}\left(\text{FNN}(R_1, Label)\right) \quad (4)$$

3.5 Understanding sub-task

To revise the generated response, the true response and generated response should be the input for the model. However, this cannot be conducted since the true re-

sponse cannot be input for the model. Therefore, to overcome such a problem and help the model better understand the relationship between persona, context, and response, the understanding task is designed. We convert the generation task into an understanding task. With the utilization of the NLI datasets, this is rather straightforward. First, the premise and hypothesis in MNLI are input into the BERT encoder to obtain the premise's sentence embedding and the hypothesis's word embedding. Second, the sentence embedding of the premise and the word embedding of the hypothesis are put together and input into BERT Decoder. Third, separate the word embedding of the special token [CLS] from the output of the BERT Decoder and input it into another 2-layer Fully Connected Network (FCN) to determine what kind of relationship they have. A cross-entropy loss function is also used for the BERT decoder in the understanding task as Eq. 5

$$Loss_{understanding} = \text{Cross entropy}(\text{FNN}([CLS], label)) \quad (5)$$

3.6 Loss function Design for the model

Our work has two tasks: the main task of person-based dialogue generation and the auxiliary task of understanding. In order to combine the two tasks and train the model together, the final loss function is the sum of the three loss functions as in Eq. 6

$$Loss_{final} = Loss_1 + Loss_2 + Loss_{understanding} \quad (6)$$

3.7 Sentence Embedding Method

Weighted Average Operation: Similarly to SIF [19], each word's word embedding is first obtained through BERT. Then the sentence embedding is obtained by a weighted averaging operation on each word vector as depicted in Table 1.

Weighted Average Operation on 1st and last layer: Apply weighted average operation on the hidden state of the 1st and last layer with a different ratio, which is 0 to 1. It is shown in Table 2.

BERT Special Token: The special characters [CLS] and [SEP] of BERT are added at the beginning and at the end of the sentence, respectively. After BERT's computation, these two special characters keep the information of the whole sentence and thus can be considered sentence embedding.

3.8 Input Method

For Encode-Decode Model, these three input methods are designed:

[CLS] P1 [SEP] P2 [SEP] P3 [SEP] C1 [SEP]
[CLS] P1 P2 P3 C1 [SEP]
P1 P2 P3 C1

where P stands for persona. C stands for context. P_i stands for ith persona sentence embedding. C_i stands for ith context sentence embedding. For BERT Decoder, these three input methods are designed:

[CLS] P1 [SEP] P2 [SEP] P3 [SEP] C1 [SEP]
[CLS] P1 P2 P3 C1 [SEP]
[CLS] P1 P2 P3 C1

4. EXPERIMENTAL RESULTS

4.1 Experimental Setup

We conduct experiments based on two datasets. The ConvAI2 dataset [22] is one of the mainstream persona-based datasets publicly available. In this dataset, each sample contains a user profile, a bot profile, and a set of multi-turn conversations. Both the user and robot profiles contain at least three profile sentences describing the user or the robot. And MNLI dataset [23] is chosen for the understanding task of this work. The dataset contains 3 parts: premise, hypothesis, and label. The label shows the relationship between the premise and hypothesis (which can be neutral, entailment, or contradiction). The parameters used in our model are summarized in Table 3. For the evaluation of our model, we choose perplexity [24] and distinct-1 and distinct-2 [25] as our evaluation method. The perplexity metric is defined as the inverse probability of the test set, normalized by the number of words. For a test set $W = w_1 w_2 \cdots w_n$:

$$perplexity(W) = \sqrt[n]{\frac{1}{P(w_1 w_2 \cdots w_n)}} \quad (7)$$

Distinct-1 and distinct-2 are defined as the number of distinct unigrams and bigrams divided by the total number of generated words. We compare our model with Transformer, GPT2, and BERT2BERT. The experiment is conducted on A40 and A100 GPUs.

4.2 Baseline

We tested 7,801 sets of data for Transformer, GPT2, BERT2BERT, and the proposed model, respectively, and selected the best results for presentation. For evaluation, Table 4 shows the perplexity of these 4 models. From the table, the encoder-decoder model of our model performs better than Transformer in terms of the perplexity of generation tasks, and worse than GPT2 and BERT2BERT models. This is mostly due to three main reasons. First, since the sentence embedding method is used instead of the word embedding method, there is a difference in the input content of the model. Compared with the word embedding method, the sentence embedding method may lose some critical contents of the sentences. Whether it is the syntactic information in the sentence, the word information, or the semantic information of the sentence itself, any missing piece of content significantly impacts the generation task. Second, due to the sentence embedding method used in this study, all the sentence embedding vectors still maintain the same dimensionality as the hidden state of the BERT model. 768-dimensional vectors may not meet the expression needs of all sentence embeddings. Third, since there is still a BERT decoder downstream of the model to modify the generated content, this may generate some non-essential noise interference, which leads to some performance degradation of the Encoder-Decoder model. However, after the text correction by the BERT decoder, the perplexity is greatly improved. This result proves the BERT decoder's vital role in this study.

Table 1. Example of weighted average sum of last layer/special token

Output of last layer				
[CLS]	1	2	3	4
H_i	5	6	7	8
!	9	10	11	12
[SEP]	13	14	15	16
Sentence Embedding	(1+5+9+13)/4	(2+6+10+14)/4	(3+7+11+15)/4	(4+8+12+16)/4

Table 2. Example of weighted average sum of 1st and last layers

Output of last layer					
[CLS]	1	2	...	3	4
H_i	5	6	...	7	8
!	9	10	...	11	12
[SEP]	13	14	...	15	16
Sentence 1	(1+5+9+13)/4	(2+6+10+14)/4	...	(3+7+11+15)/4	(4+8+12+16)/4
Output of first layer					
[CLS]	11	12	...	13	14
H_i	15	16	...	17	18
!	19	20	...	21	22
[SEP]	23	24	...	25	26
Sentence 2	(11+15+19+23)/4	(12+16+20+24)/4	...	(13+17+21+25)/4	(14+18+22+26)/4

Sentence = ratio *Sentence 1 + (1-ratio) * Sentence 2, ratio $\in [0, 1]$

Table 3. Parameter used for the proposed model

Total Epochs	6
Warm-up Steps	30000
Warm-up Learning Rate	1.00E-05
Learning Rate	2.00E-05
Batch Size	4
Ratio for Weighted Average Sum	$[0, 1]$, step 0.1
Max Length for Sentence	16

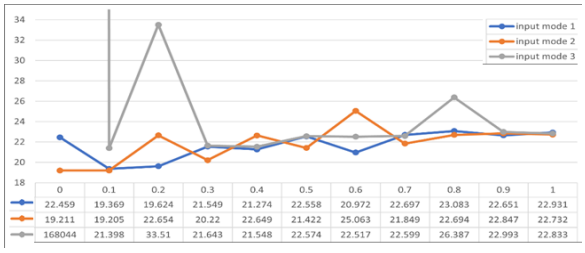

Fig. 3. Perplexity of Encode-Decode Model

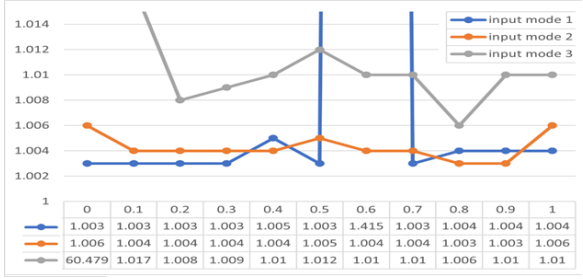

Fig. 4. Perplexity of BERT Decode Model

4.3 Sentence Embedding Method

In this work, we used four different sentence embedding methods, which are special characters [CLS], the summed average of the hidden states of the last layer or the first layer, and the weighted average of the first and last layers' hidden states. In order to evaluate their performance within the natural language generation task, we tested the embedding approaches on the test set. In the column of the weighted average of the first and last layers, the best performance is obtained with a ratio of 0.1 and reported in the table. The table shows that the weighted average of the first and last layers performed better than the other three tested sentence embedding methods. The BERT model obtains the vector of special characters [CLS] after calculating the attention of the whole utterance, and it contains the information of the whole utterance. However, more information is needed to help the downstream generative model for the text generation task. Similarly, the first layer of BERT pays too much attention to the syntactic structure and the semantics of individual words. At the same time, the last layer of BERT pays too much attention to the semantics of the whole sentence, thus resulting in biased information of the utterance, which is also not beneficial to the subsequent text generation. Combining the first layer's output with the last layer's results makes some improvement in model performance, which also demonstrates that using a weighted average of the first and last layers is a feasible way to obtain sentence embeddings suitable for text generation.

4.4 Sentence Embedding with different ratio

Combining the information from the first layer with the information from the last layer requires a particular ratio parameter. During our experiments, this parameter starts at 0 and ends at 1 with an increment of 0.1. The perplexity and Distinct 1/2 of the Encoder-Decoder model and BERT Decoder are shown in Figs. 3, 4, 5, and 6. For the perplexity of the Encoder-Decoder model, the model performs well with a ratio from 0 to 0.1. As the ratio gradually increases, the model performance starts to decrease. Furthermore, from the perplexity of the BERT Decoder, our model performance is improving or maintaining with the ratio, up to 0.9, and decreasing at 1. Similarly, as the Encoder-Decoder model, the Bert Decoder performs poorly on text correction as the ratio increases. Considering the results of perplexity and Distinct 1/2 together, the best ratio is between 0 and 0.2. From this re-

Table 4. Perplexity Comparison

Transformer Word Embedding	GPT2 Word Embedding	BERT2BERT Word Embedding	Encoder-Decoder Our Model	BERT Decoder Our Model
28.8	14.4	15.6	19.205	1.004

Table 5. Sentence Embedding Method Results

Parts of our Model		Perplexity	Distinct-1	Distinct-2
BERT2BERT		15.6	0.0243	0.0868
CLS	Encoder-Decoder	25.571	0.0001	0.0001
	BERT Decoder	1.003	0.0001	0.0001
summed average (last layer)	Encoder-Decoder	22.368	0.0001	0.0001
	BERT Decoder	1.003	0.0001	0.0001
summed average (first layer)	Encoder-Decoder	22.333	0.009	0.0267
	BERT Decoder	1.003	0.0094	0.0353
weighted average (ratio 0.1)	Encoder-Decoder	19.369	0.0165	0.054
	BERT Decoder	1.003	0.0178	0.0734

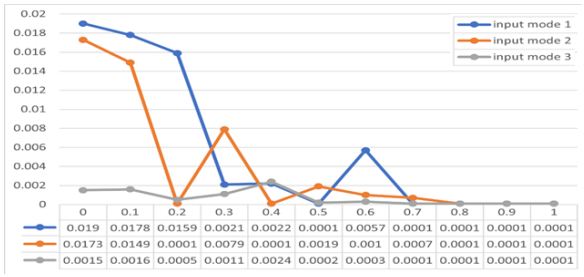

Fig. 5. Distinct-1 of BERT Decode Model

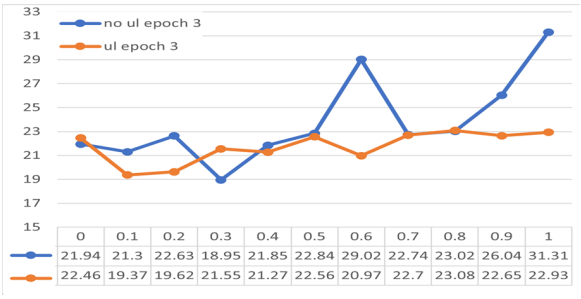

Fig. 7. Perplexity of the Encoder-Decoder Model

Fig. 8. Perplexity of BERT Decoder Model

sult, it can be determined that the output of the last layer of the BERT model can help improve the model's performance. In other words, using a portion of the output of the last layer of the BERT model can help obtain high-quality sentence embeddings.

4.5 Input Method

From Fig. 3 to Fig. 6, it can be noted that input mode 1 and input mode 2 are better than input mode 3 in general. In most cases, input mode 1 performed better than input mode 2. It should be taken into account that since we use sentence embeddings instead of word embeddings, each sentence embedding vector represents a sentence.

4.6 Understanding Task

Finally, we did an experiment and discuss this specially designed understanding task. In this test, we use input mode 3 and set the ratio from 0 to 1. We test how much the performance of the model changes with and without the understanding task. The perplexity and the Distinct 1/2 of the Encoder-Decoder model and BERT Decoder with or without understanding the task is shown in Figs. 7, 8, 9, and 10. From the experimental results, the Perplexity and Distinct-1/2 of the Encoder-Decoder model and the BERT decoder, with the help of the comprehension task, in most cases, the performance of the model has been improved to some degree. From our observations, two points need stressing. First, the understanding task helped the model to understand the relationship between sentences and improved the model's performance in the generation task. Second, the understanding task itself is at the very end of the model, and its loss function can be added as a special noise to the model's training, which helps the model have better robustness on the generation task.

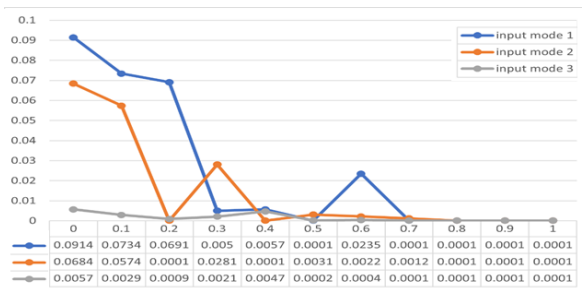

Fig. 6. Distinct-2 of BERT Decode Model

5. CONCLUSIONS AND FUTURE WORK

This research proposed a new sentence embedding-based personalized dialogue generation model. Specifically, we improved the model's persona-based dialogue generation capability, enabling the model to understand

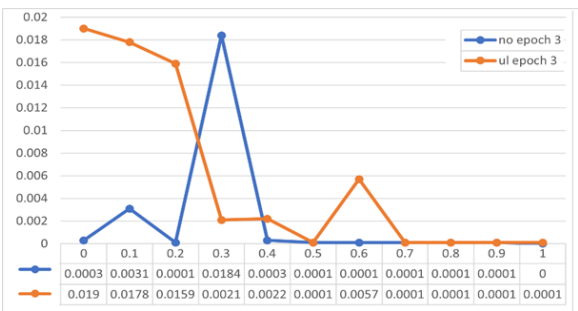

Fig. 9. Distinct-1 of BERT Decoder

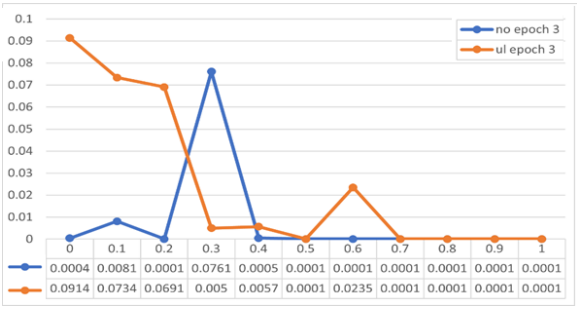

Fig. 10. Distinct-2 of BERT Decoder

the relationship between input utterances through a specially designed auxiliary task, which is called the understanding task. First, we designed four different approaches for sentence embedding: special characters, summation averaging of the one-layer hidden state output, and weighted averaging for the first and last layers. For the weighted averaging, a separate weight parameter was designed to change the weight of the sentence information components contained in the sentence embedding. These four sentence embedding methods were tested, and it was shown that the weighted average sentence embedding method works best, and the best ratio is between 0 and 0.2. Second, we considered the difference between sentence and word embedding and designed three input methods. The first input method stood out from the rest in a series of experiments. Finally, we tested and proved that the understanding task could help the model understand the relationship between utterances. Moreover, it could act as a special kind of noise to help the model achieve better generality on dialogue generation tasks. We are performing an extensive robustness study of the proposed model against various real-world conditions, including speech-to-text conversion errors, toward a persona-aware social robot or conversational agent.

REFERENCES

[1] Fakhreddine Karray, Milad Alemzadeh, Jamil Abou Saleh, and Mo Nours Arab. Human-computer interaction: Overview on state of the art. *International journal on smart sensing and intelligent systems*, 1(1):137–159, 2008.

[2] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, Eftychios Protopapadakis, et al. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018, 2018.

[3] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine*, 13(3):55–75, 2018.

[4] Zheng Zhang, Ryuichi Takanobu, Qi Zhu, MinLie Huang, and XiaoYan Zhu. Recent advances and challenges in task-oriented dialog systems. *Science China Technological Sciences*, pages 1–17, 2020.

[5] Yoshitaka Yamane, Y Sasaki, Y Fujisaku, Satoshi Muramatsu, Katsuhiko Inagaki, Daisuke Chugo, Sho Yokota, and Hiroshi Hashimoto. Development of non-task-oriented dialogue system for human friendly robots. In *2020 13th International Conference on Human System Interaction (HSI)*, pages 50–55. IEEE, 2020.

[6] Munazza Zaib, Wei Emma Zhang, Quan Z Sheng, Adnan Mahmood, and Yang Zhang. Conversational question answering: A survey. *Knowledge and Information Systems*, 64(12):3151–3195, 2022.

[7] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[9] Zeyu Ding, Armagan Elibol, and Nak Young Chong. Leveraging extended chat history through sentence embedding in multi-turn dialogue toward increasing user engagement. In *2022 22nd International Conference on Control, Automation and Systems (ICCAS)*, pages 642–649, 2022.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[11] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.

[12] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mass: Masked sequence to sequence pre-training for language generation. *arXiv preprint arXiv:1905.02450*, 2019.

[13] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. *Advances in neural information processing systems*, 32, 2019.

[14] Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*, 8:264–280, 2020.

[15] Juntao Li, Chang Liu, Chongyang Tao, Zhangming Chan, Dongyan Zhao, Min Zhang, and Rui Yan. Dialogue history matters! personalized response selectionin multi-turn retrieval-based chatbots. *arXiv preprint arXiv:2103.09534*, 2021.

[16] Kota Ishizuka, Kai Kurogi, Kosuke Kawakami, Daishi Iwai, and Kazuhide Nakata. Generating search text ads from keywords and landing pages via bert2bert. In *Advances in Artificial Intelligence: Selected Papers from the Annual Conference of Japanese Society of Artificial Intelligence (JSAI*

2021), pages 27–33. Springer, 2022.

[17] Qian Liu, Yihong Chen, Bei Chen, Jian-Guang Lou, Zixuan Chen, Bin Zhou, and Dongmei Zhang. You impress me: Dialogue generation via mutual persona perception. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1417–1427, Online, July 2020. Association for Computational Linguistics.

[18] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[19] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations*, 2017.

[20] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

[21] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[22] Varvara Logacheva, Valentin Malykh, Aleksey Litinsky, and Mikhail Burtsev. Convai2 dataset of non-goal-oriented human-to-bot dialogues. In *The NeurIPS'18 Competition: From Machine Learning to Intelligent Conversations*, pages 277–294. Springer, 2020.

[23] Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017.

[24] Dan Jurafsky and James H Martin. Speech and language processing (3rd ed. draft), 2019.

[25] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*, 2016.