Contextual Modeling in Context-Aware Conversation Systems

Quoc-Dai Luong Tran[1*], Dinh-Hong Vu[1], Anh-Cuong Le[1], and Ashwin Ittoo[2]
[1] Natural Language Processing and Knowledge Discovery Laboratory, Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh city, Vietnam
Ho Chi Minh City 700000, Vietnam
[e-mail: tranluongquocdai.st@tdtu.edu.vn, vudinhhong@tdtu.edu.vn, leanhcuong@tdtu.edu.vn]
[2] HEC Liège, University of Liège, Belgium
Rue Louvrex 14, Liège 4000, Belgium
[e-mail: ashwin.ittoo@uliege.be]
[*]Corresponding author: Quoc-Dai Luong Tran

Received March 14, 2023; revised April 14, 2023; accepted May 2, 2023; published May 31, 2023

Abstract

Conversation modeling is an important and challenging task in the field of natural language processing because it is a key component promoting the development of automated human-machine conversation. Most recent research concerning conversation modeling focuses only on the current utterance (considered as the current question) to generate a response, and thus fails to capture the conversation's logic from its beginning. Some studies concatenate the current question with previous conversation sentences and use it as input for response generation. Another approach is to use an encoder to store all previous utterances. Each time a new question is encountered, the encoder is updated and used to generate the response. Our approach in this paper differs from previous studies in that we explicitly separate the encoding of the question from the encoding of its context. This results in different encoding models for the question and the context, capturing the specificity of each. In this way, we have access to the entire context when generating the response. To this end, we propose a deep neural network-based model, called the Context Model, to encode previous utterances' information and combine it with the current question. This approach satisfies the need for context information while keeping the different roles of the current question and its context separate while generating a response. We investigate two approaches for representing the context: Long short-term memory and Convolutional neural network. Experiments show that our Context Model outperforms a baseline model on both ConvAI2 Dataset and a collected dataset of conversational English.

Keywords: Conversation context, Conversation models, Conversation history, Deep neural networks.

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under Grant number 102.05-2020.26.

http://doi.org/10.3837/tiis.2023.05.005 **ISSN : 1976-7277**

1. Introduction

Interpersonal conversation is an integral part of our daily activities, enabling us to make informed decisions and to acquire a better understanding of various social phenomena [1]. In recent years, we have seen the emergence of conversation agents, also commonly known as spoken dialog systems (SDS) or chatbots. These agents enable a human-like user-machine communication, responding to human utterances in natural language.

In general, conversation agents can be categorized into two classes. First, task-oriented agents are designed to assist users in accomplishing specific tasks or goals, e.g., restaurant bookings or airline seat reservations [2]. The utterances generated by such agents are typically short texts aimed at clarifying users' requests. Popular examples include Siri, Cortana and Google Assistant. The second class of conversation agents are those not designed for any specific task. Instead, they are intended to engage in relatively longer "chit-chat"-like conversations with users. Chatbots constitute a classic example of such agents.

Conversation modeling is a Natural Language Processing (NLP) task concerned with developing models for generating a suitable response for a given user request. The basic premise of most models is that they take as input a user utterance *x*, and generate, as output, a response *y*, which maximizes the probability $P(y \mid x)$. As can be expected, these models (e.g. [3] and [4]) perform well for individual utterances issued by the user. However, their performance degrades when generating any response requiring information from previous utterances, i.e., the context. The example in **Table 1** illustrates the importance of the context in conversations. It can be seen that the context, i.e., previous utterances, is important for the model to understand "pro one" and "it".

To use context for response generation, many researchers have sought to combine the previous utterances with the current utterance into the model e.g. [1, 2, 5-7]. Some studies have used previous utterances to generate current ones or have used reinforcement learning methods to keep track of the conversation's history, such as [8]. These models compute the response *y* which maximizes $P(y \mid x, c)$, where *c* is the context, comprised of a concatenation of previous utterances. However, concatenation of these previous utterances can result in long sequences, which have to be subsequently truncated as in [7]. Doing so could lead to relevant information being discarded.

Other models encode the entire context and current utterance in dense vectors via Recurrent Neural Networks (RNNs) as in [9]. Alternatively, separate dense vectors for the context and utterances are learned separately, respectively *c* and *x*. These are fed to the models individually. While dense vectors enable the models to access past information (context), they tend to emphasize more recent information in the context while giving marginally less importance to information that occurred several time steps in the past. This is a common issue with RNNs, even with Long Short Term Memory (LSTM) [10] or Gated Recurrent Unit (GRU) memory units, as reported in [11]. Recently, Large language models like BERT [12] and ChatGPT [13] are designed to understand language in context, which makes them very effective at a wide range of natural language processing tasks. These models are trained on vast amounts of text data, which allows them to learn and build a rich representation of the relationships between words and concepts.

Other approaches attempt to capture all previous utterances, like [11]. Each utterance is passed through RNNs [9]. Then, the context is saved in a dense vector. However, the model will gradually forget long past utterances when conversations become longer. Dealing with conversations as illustrated in **Table 1** becomes challenging.

Table 1. Example of a short conversation between Bot and Customer

Bot	Hello, how can I help you?
Customer	How much is HP printer?
Bot	Which kind do HP you refer to, Pro or Basic?
Customer	The **pro one**, how much is **it**?
Bot	It is 300$, do you want to order now?

Our aim in this study is to address the aforementioned difficulties that arise when context is ignored in conversation models [3, 4]. Our work builds upon and extends recent studies, such as [1, 2, 5-7], which have explicitly attempted to incorporate context in the task of dialogue generation.

Similar to these studies, our model relies on an encoding of the context history. However, our main innovation lies in how the context is incorporated in the decoder, i.e., how we make the generator context-aware. The predominant approach in existing literature involves a sequence to sequence (Seq2Seq) [14] architecture, learning a joint representation of the current utterance and of the context at the encoder. The encoder's final hidden state is subsequently used to initialize the decoder. The joint representation of context and current utterance can be realized by concatenation or (cosine) similarity. Various attention mechanisms [2, 5] are also employed to help the decoder determine which parts of the input (current) utterance to attend to. These studies are characterized by encoding the current utterance and its context at the same time on the same model. However, the current question (utterance) and its context exhibit different characteristics. Specifically, the context is clearly much longer than the current utterance and requires different treatment. Furthermore, the roles of the question (current utterance) and its context are also different in generating the response, which is another motivation for separating the question and context when feeding them into the decoder to generate the response. To address the issue of context history and based on the foregoing discussion, we propose a model with a new architecture to handle the current utterance and its context separately. To model the context, we will investigate two different deep learning architectures, LSTM and Convolutional Neural Network (CNN). We combine the encoding results from the context with the current question, and then provide it to the generator to generate a response. In essence, the entire context is used to control the decoder when generating the response. Unlike most existing models, which take the whole conversation as input for the encoder, our proposed model distinguishes the role of the current question and its context in representation (encoding) and response generation. In particular, our model generates the response not only based on the current utterance but also considers all previous utterances. Our decoder is thus context-aware. In essence, we augment each current utterance with the entire context. Thus, this approach preserves the past conversation history.

For our key evidence, we show that our approach achieves strong performance on an open-domain corpus that we collected. This dataset contains general English conversations for teaching English to non-native speakers. In addition, we also evaluate our model on the standard ConvAI2 Dataset. It outperforms a strong baseline model based on a Seq2Seq architecture, similar at its core to that of [5]. Furthermore, we show that LSTM can capture the dependence on long sentences better than CNN. Our results confirm that LSTM based models do indeed capture longer dependencies. However, we also show that CNN based models tend to train faster and perform better on smaller datasets. We did not investigate bi-LSTM in our models as it is well-known that it is slower than the classical LSTM. We wanted a model that was fast to train and efficient at inference time. Also, we did not consider more complex models, such as BERT, as our aim was to have an accurate model without being excessively large (in terms of the number of parameters).

Moreover, we investigate the influence of word2vec pre-trained embedding on context model's performance. Our results show that the LSTM Context model outperforms the baseline by 50.6%. Concerning the CNN Context model, it outperforms baseline 33.2% in BLEU 1 [15].

The rest of this paper is organized as follows: We talk about some related works in section 2. We show details of our models in section 3 and the datasets we used in section 4. We then evaluate our models and show the results in section 5. Finally, we state our conclusions in section 6.

2. Related Works

Conversation agents' architectures generally fall into two broad classes: rule-based systems and corpus-based systems. Rule-based systems, ELIZA [16] and PARRY [17] for example, match the users' message with rules and then transform the message into a response by those rules. These systems suffer from several well-known limitations. For instance, they are unable to answer questions that do not match any rules. Their answers tend to follow fixed formats and lack variation. Corpus-based systems mine conversations corpora of human-human or sometimes human-machine to build conversation models, which are then used to locate or generate suitable responses. Typical corpora used in past research include huge collections of Twitter conversations and movie dialogues, as described in [11].

Corpus-based systems adopt two main paradigms, namely information retrieval and machine learning. Information retrieval-based systems will respond to a user's message *m* by searching in the corpus to find turn *t* which is the most similar to *m*, and then return the message found at turn *t* or the following message. Given a data set of the pair (q_t, a_t) the model finds the answer to user's message *m* by measuring the similarity between *m* and q_t to find the pair (q_t, a_t) with the greatest similarity to *m*, q_t and return a_t (turn after *t*). Alternatively, it may measure the similarity between *m* and a_t and return a_t with the greatest similarity (turn *t*) [18, 19]. However, in the data there can be many pairs (s, q_t) or (s, a_t) with the greatest similarity. To address this issue, [20, 21] rely on other approaches such as statistical machine translation, filtering, and ranking to increase the accuracy of answer selection a_t. However, information retrieval models can only answer the questions *m* contained in the dataset (with similarity above a certain threshold), and the answer will also be one contained in the dataset. While the chatbot needs to answer the user's question, it must also reply on the chat history and the user profile.

Machine learning based systems respond to a user's message by generating new response text after being trained on large corpora. A recent and popular approach is to train these systems *end-to-end* using neural network architectures, such as Seq2Seq [22-24] or GAN [25, 26]. This idea was first proposed by [17] using phrase-based machine translation to translate a user's message into a system response. In recent years, deep reinforcement learning has garnered a lot of attention in NLP [27]. In chatbot systems, this approach suggested using a generation model in combination with a task-oriented model based on RL (reinforcement learning) to build a chatbot. This method results in more realistic conversations better at achieving task goals, as demonstrated by the experiments conducted [28]. Additionally, recent studies have focused on developing dialogue systems that track the conversation history to make better decisions for the following action or response selection, considering the user's goals and incorporating the dialogue history [8, 29].

Machine learning-based approaches for conversation models can be further divided into two categories: those that are *context aware* and those that are *oblivious to context*. The first

category takes the context into account when generating the response, as opposed to the second category that ignores it. Examples of context-oblivious models include those proposed by [3, 4, 30]. In these models, the last utterance alone is used to generate the response. Some models incorporate additional information, such as such as syntax (c), topic (z) and speaker embedding. The main limitation of these models is that they tend to forget distant previous utterances, and thus may be unable to generate suitable responses that require this past context.

Most context-aware models concatenate the entire set of previous utterances and current utterance. This concatenated input is then fed to the model, as in [7] and [23]. While these modes do achieve reasonable performance, their main limitation is that, over time, the context size will increase. Subsequently, it will have to be cut off to fit the models' input (vector size). This results in the discarding of important information, which may be critical in generating the response.

To capture longer context, [11] processes each utterance through an RNN. The context is encoded in a dense vector and is then used to decode the tokens when generating the response. The context is saved in a dense vector from the beginning of the conversation. When new utterances are created, they will be encoded into that same vector.

In a similar vein, [2] proposed a model consisting of two components, using a context encoder and a tagger. The former acts on utterances to produce a vector representation of the dialogue context. The latter then takes both the dialogue context encoding and the current utterance as input to generate an intent and slot annotations as outputs. They proposed three variants of the model, each adopting a different approach for representing context, e.g., using only the last utterance, using a softmax attention mechanism on the current and previous utterances, and using a specific sequential dialogue encoder network (SDEN) with temporal information. Their experimental evaluation was not focused on assessing the quality of the generated responses. Instead, they were concerned with performance during tasks including intent classification and slot filling. The best performance was achieved by the SDEN variant when working on several domain-specific crowdsourced datasets that covered covering movie ticket purchases and restaurants bookings.

Recently, BERT (Bidirectional Encoder Representations from Transformers), a neural network architecture based on transformer architecture designed to model data sequences like natural language text, has been rising in natural language processing [12]. BERT has been applied to various NLP tasks, such as machine translation [31, 32], language modeling [33], and chatbot [34]. Its training process utilizes next-sentence prediction to understand the relationship between two sentences, making it useful for question answering. Like BERT, ChatGPT is a language model designed to understand and process natural language. ChatGPT employs a transformer-based architecture that utilizes self-attention mechanisms and comprises several layers of transformers that can be fine-tuned for specific tasks. BERT and ChatGPT have also outperformed traditional neural network models for chatbot systems, particularly in terms of accuracy and naturalness of responses [34, 35].

Another study that proposes a context aware model is that of [5], which proposes two architectures to make the decoder context-aware. The first architecture concatenates the preceding user utterance to the current dialogue act (DA), before being fed to the encoder. Conversely, in the second architecture the hidden states of both the context and DA encoders are concatenated for initializing the decoder. Experimental evaluations performed on a domain-specific dataset concerning public transport showed that a combination of both approaches yielded a significant improvement in BLEU scores. These results were subsequently confirmed in a human pairwise preference test. However, only the last turn is used for context as opposed to the entire conversation history, resulting in potentially useful

information being discarded.

Other studies that are tangentially related to ours include those of [6, 36, 37]. For instance, [6] proposed a model for generating a natural language dialog response when given a question, a corresponding dialog act and target semantic slots. The model of [36] attempts to predict hash tags based on conversation context. Study [37] addresses the problem of thread detection in conversations. The approach of encoding the context separately has been shown to improve upon the method of simply concatenating the context with the current utterance. However, utterances that occurred at a time step far back in the conversation will be forgotten by the model. Additionally, when new conversations are started, the generated responses will still be affected by the (context of) previous conversations. To address these issues, our proposed model innovates by taking the *entire* context into consideration at every turn when generating the response.

For a full review of the current state of the art in chatbots/conversation modeling, we refer the reader to [38]. Our model handles previous utterances and input separately so that our Context Model can handle longer history when capturing its main information. Besides the context associated with each input, the model will not be affected by the context of the previous conversation.

3. Proposed model

A conversation consists of multiple turns. We denote a conversation by n_C turns $T_1, T_2, \dots, T_{n_C}$, where each turn k^{th} has I_k words denoted by $T_k = \{w_{k,1}, w_{k,2}, \dots, w_{k,I_k}\}$. Considering a conversation, suppose that T_i is the current utterance and T_{i+1} is its response, while $\{T_1, T_2, \dots, T_{i-1}\}$ is the context for generating T_{i+1} from T_i. For the convenience of further use, we denote $M = T_i$, $R = T_{i+1}$ and $C = \{T_1, T_2, \dots, T_{i-1}\}$. For example, in **Table 1** we have 5 turns. If current utterance is the 4^{th} turn in the conversation, then the last turn is its response and the three preceding turns are its context.

3.1 The baseline model

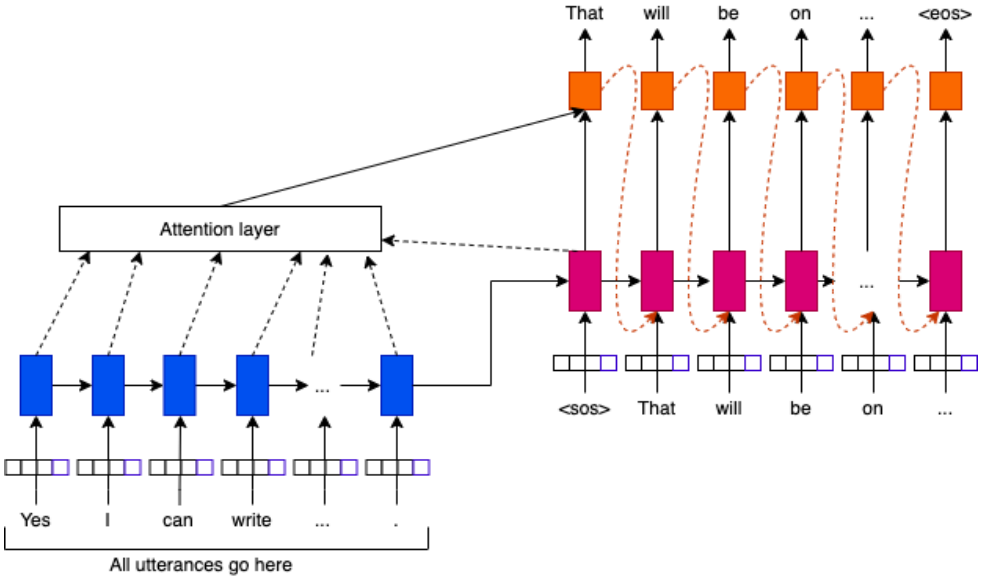

Fig. 1. Baseline model, all utterances are fed into encoder.

Before presenting our proposed model, we first present the baseline against which the performance of the proposed model will be assessed. The baseline we used is similar to the model [5]. At its core, the architecture is based on Seq2Seq [14], incorporating Bahdanau's attention [39] mechanism. LSTM [10] are used both at the encoder and the decoder, with hidden size of 200. In this model all previous utterances and the current utterance are concatenated and then fed into the encoder. The input of the encoder will be the concatenation of all turns in the context *C* and the current utterance *M*. Then decoder uses the encoder's result and attention to generate a response *R* (See **Fig. 1**).

3.2 The proposed model

Our model is based on the Seq2Seq architecture [14], which is by far the most popular paradigm for conversation modeling and dialog generation employed in previous studies. It consists of 3 modules: Current utterance encoder, Context encoder and Response generator. A general overview of our models is shown in **Fig. 2**. The current utterance, *M*, will be encoded into h_e by the Current utterance encoder module. The context *C* will be encoded into h^C by the context encoder module. Finally, the Response generator module will use h_e, h^C and Bahdanau's attention mechanism [39] to generate the response *R*. We describe these modules in detail below.

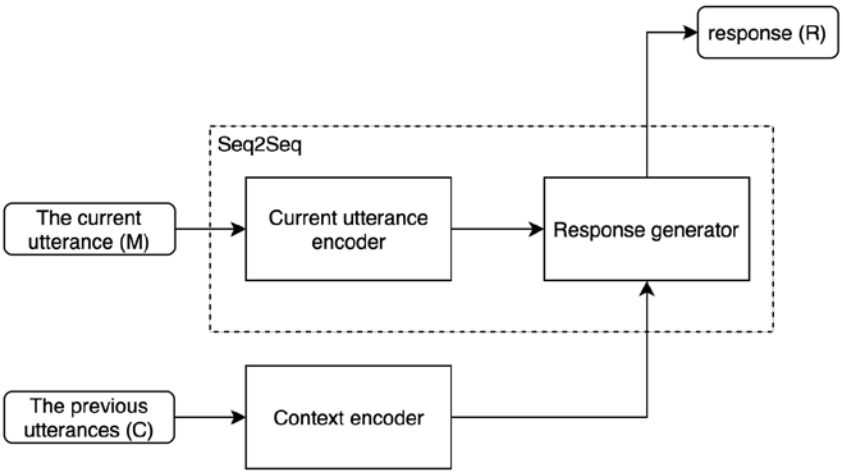

Fig. 2. Context model overview: The current utterance goes though the Current utterance encoder module and the context goes though the Context encoder module. The Response generator module uses the results to generate the response.

3.2.1 The utterance encoder module

The utterance encoder module uses LSTM to encode the current utterance into a fixed length vector. We use this vector to initialize the weights for the generator module. We also apply Bahdanau's attention [39] to the outputs of this module at the response generation stage.

Supposing that *M* has *m* words denoted as $M = \{w_1, w_2, \ldots, w_m\}$, the LSTM will calculate input gate i_t, memory gate f_t, output gate o_t and hidden state h_t at each time step *t*.

$$i_t = \sigma(W_i w_t + V_i h_{t-1}) \quad (1)$$

$$f_t = \sigma(W_f w_t + V_f h_{t-1}) \quad (2)$$

$$o_t = \sigma(W_o w_t + V_o h_{t-1}) \quad (3)$$

$$l_t = \tanh(W_l w_t + V_l h_{t-1}) \quad (4)$$

where $\sigma(\cdot)$ is the sigmoid function. Then the unit computes c_t and final output h_t is computed:

$$c_t = f_k \cdot c_{t-1} + i_k \cdot l_k \quad (5)$$

$$h_t = o_k \cdot \tanh(c_k) \quad (6)$$

We denote the $h_e = h_m$ state, where h_m is the last hidden $W_{i,f,o,l} \in R^{D\times E}, V_{i,f,o,l} \in R^{D\times D}$, $x_t \in R^{E\times \mathbb{1}}$, E is embedding and D is hidden size. This module gives us states $\{h_1, h_2, \ldots, h_m\}$ to calculate attention scores and the final state h_e used to initialize the weights for the response generator.

3.2.2 The context encoder module

The context encoder module extracts key information in the context C into feature vector h^C. This module can be written in general form as (7), where the *ContextEncoder* function can be any feature extraction model such as LSTM, CNN, or BERT [12].

$$h^C = ContextEncoder(C) \quad (7)$$

In our experiments, we use LSTM and CNN as the context module to demonstrate the important role of context in response generation.

LSTM Context Encoder

We use LSTM for the context encoder module to encode the context C and get final output h^C using (6). Equation (7) can be written as (8):

$$h^C = LSTM(C) \quad (8)$$

CNN Context encoder

We use the CNN in context encoder module to extract the characteristics of the context sequence. So (7) can be written as (9). We use filter size 1, 2 and 3, which is equivalent to 1, 2, 3-gram in the language (see **Fig. 3**). For the last max pooling layer, we get a context vector h^C. We can assume that this vector is equivalent meaning to h^C in LSTM above.

$$h^C = CNN(C) \quad (9)$$

The context encoder module provides h^C for the response generator module and uses as (10) to control the generator.

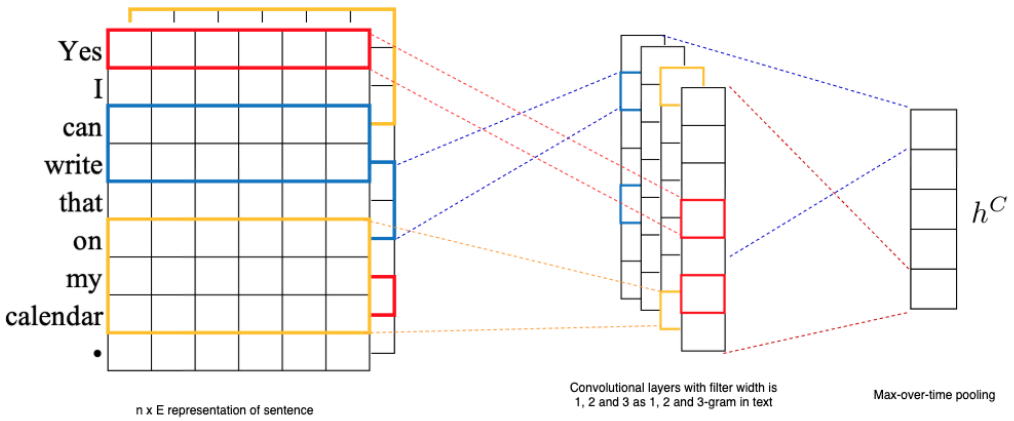

Fig. 3. CNN Context encoder module.

3.2.3 The response generator module

The response generator module uses h_e, h^C and attention scores (see **Fig. 4**) to generate response R as (10). So, estimating the probability p in (10) will depend not only on M, which encoded to h_e, but also on the context, which encoded into the h^C vector.

$$p(R,\theta) = \prod_{i=1} p(r_i \mid h_e, [r_1, h^C], [r_2, h^C], \ldots, [r_{i-1}, h^C], \theta, c) \quad (10)$$

where θ is the model's parameters, c is attention Bahdanau scores, r_i is decoder's output at time step i, word i^{th} of response, $[r_i, h^C]$ is concatenation of r_i and h^C vector, decoding process stops when token <*EOS*> is predicted. According to [40], concatenating two vectors gives better results than sum or average. As mentioned earlier, the main innovation of our model lies in proving the entire context at each timestep when generating the response. This enables the decoder to generate more meaningful responses as it has access to information from utterances much further back in time.

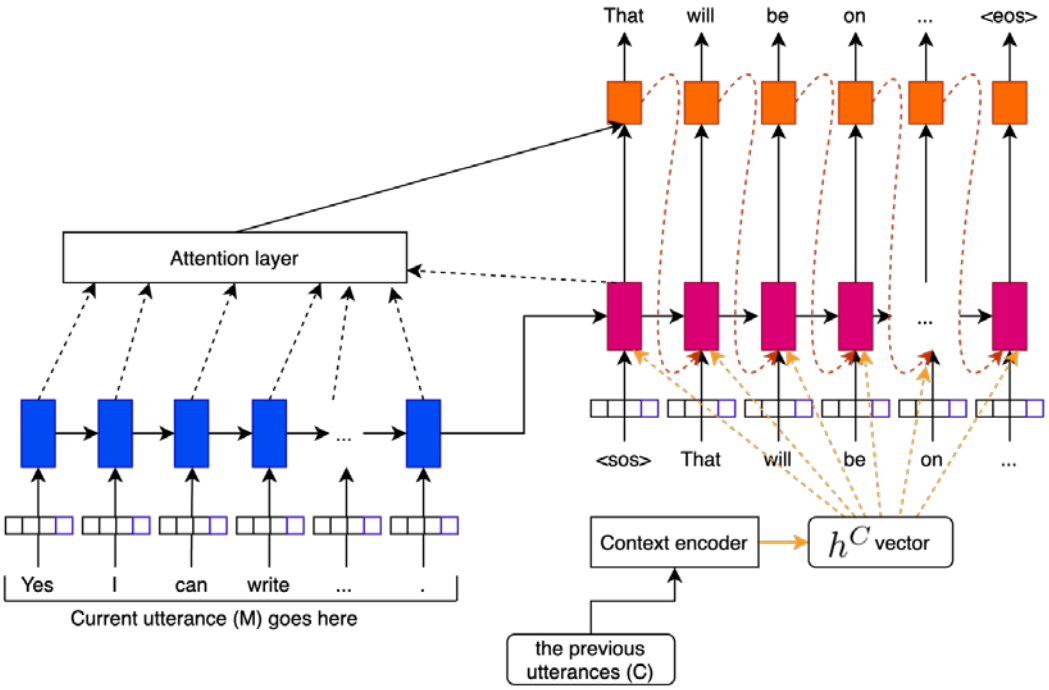

Fig. 4. The Context model, the current utterance is fed into the current utterance encoder module and the context is fed into the LSTM context encoder module.

4. Dataset

4.1 The PersonaChat ConvAI2 Dataset

We perform our experiments on the PersonaChat ConvAI2 dataset [22], which has been used in the past for conversation models [41-43]. We generate training and testing data by taking the current utterances and their corresponding context (ignoring the persona information available from ConvAI2). This results in 17,878 dialogs for training and 1,000 dialogs for testing.

First, we process all dialogs to create the model's inputs. With the baseline model, an input *x* will be a concatenation of current utterance *M* and its corresponding context *C*. With our proposed model, an input *x* will consist of two types of information *C* and *M*. The output *y* is the response *R* of *M* or next utterance. We refer to a pair *x* and *y* as a data point *I*.

More specifically, the PersonaChat ConvAI2 Dataset has many dialogs. Each dialog has many turns. In each turn the last history text will be considered the current utterance, while the rest of the history will be context. The last candidates will be output. For an example, we have

4 utterances $[t_1, t_2, t_3, t_4]$, and after reprocessing we have data points:

- $I_1 = [\{NULL, t_1\}, t_2]$. This data point has no context (NULL) because it is the first turn and t_1 is the current utterance, which will be fed to the current utterance encoder, while t_2 is output *y*.
- $I_2 = [\{t_1, t_2\}, t_3]$. In this data point t_1 is context, t_2 is current utterance and t_3 is output *y*.
- $I_3 = [\{t_1 + t_2, t_3\}, t_4]$. In this data point $t_1 + t_2$ is context, t_3 is current utterance and t_4 is output *y*.

After reprocessing we have 244,998 data points for training and 14,602 data points for testing. Next, we change all inputs from natural language to sequence numbers which are fed into the neural network. In our experiments we use a self-training word embedding network and Word2Vec embedding.

4.2 Conversation collection data

In addition to the standard ConvAI2 dataset, we also created a new dataset by collecting conversations from books and videos teaching conversational English[1] to non-native English speakers[2]. This data includes standard conversations about language, covering the most common conversation topics in communication from work to daily activities. Our collected dataset consists of 864 dialogs, which were preprocessed following the steps earlier for the ConvAI2 dataset. After pre-processing, the collected dataset was split into 6,710 data points for training and 2,237 data points for testing. Our aim in creating this dataset was twofold. First, it enabled us to investigate our model's performance on a smaller dialog collection consisting of everyday conversations. Second, we believe that the dataset could be a useful resource for future research in conversation modeling.

Table 2 shows minimum, maximum, and average number of words per turn; minimum, maximum, and average number of words of context; and minimum, maximum, and average number of turns of context. We use these statistics to choose maximum sequence length in experiments.

Table 2. Datasets statistics.

Dataset	Question words	Answer words	Context words	Context turns
ConvAI2 training set	Min: 1 Max: 60 Avg: 11.21	Min: 1 Max: 60 Avg: 11.57	Min: 1 Max: 561 Avg: 82.76	Min: 0 Max: 48 Avg: 6.46
ConvAI2 testing set	Min: 3 Max: 24 Avg: 11.76	Min: 3 Max: 24 Avg: 11.87	Min: 1 Max: 329 Avg: 93.97	Min: 0 Max: 24 Avg: 6.84
Our training data	Min: 3 Max: 69 Avg: 11.25	Min: 3 Max: 69 Avg: 11.45	Min: 1 Max: 278 Avg: 51.75	Min: 0 Max: 28 Avg: 6.11
Our testing data	Min: 3 Max: 43 Avg: 10.95	Min: 3 Max: 69 Avg: 11.56	Min: 1 Max: 251 Avg: 50.9	Min: 0 Max: 27 Avg: 6.0

[1] We collected from https://basicenglishspeaking.com/daily-english-conversation-topics/, https://www.eslfast.com, https://www.youtube.com/channel/UCVXM96yuiXY3ZT73Dy8HgCA and https://www.youtube.com/channel/UCV1h_cBE0Drdx19qkTM0WNw

[2] The dataset public at https://github.com/vudinhhong/conversation-dataset

5. Experiments

For comparing the performance of the baseline and of the proposed models, the models are tuned with the same hyper-parameters. Specifically, the learning rate is 10^{-3}, and word embedding size is 300. The current utterance encoder module and the decoder (generator) module (see **Fig. 2** and **Fig. 3**) are LSTM units, with a *tanh* activation function and a hidden vector size of 200. As mentioned earlier, we investigated two implementations of the context encoder, viz. LSTM and CNN. The configuration of the former is the same as that of the LSTM current utterance encoder. Concerning the CNN context encoder, we experiment with a window size of 1, 2 and 3, and set the number of filters to 128. All models are trained in 50 epochs. Our models are built based on Keras Tensorflow[3].

To choose the maximum sequence's length for the LSTM units, we rely on the descriptive statistics shown in **Table 2** for the ConvAI2 dataset and our own dataset of daily conversations. Consequently, for the ConvAI2 dataset, the maximum sequence length for the current utterance encoder and decoder was 62, while that of the context encoder was 561. For our own dataset of conversations, the maximum sequence lengths were of 23 for both the current utterance encoder and decoder, and 280 for the context encoder.

Regarding the baseline model, since its input is a concatenation of the current utterance and the context, the maximum sequence length (of the encoder) was 623 (=561+62) for the ConvAI2 dataset and 303 (=280+23) for our collected dataset.

Following [3] and [44], we used BLEU [15] as our evaluation metric. Furthermore, according to [45], "BLEU correlates well with human quality judgments of generated conversational responses".

Table 3. Experiment result of ConvAI2 Dataset

	Baseline model	Our model with LSTM context encoder	Inc.	Our model with CNN context encoder	Inc.
BLEU 1	0.0784	0.1181	50.6%	0.1044	33.2%
BLEU 2	0.0269	0.0454	68.8%	0.0400	48.7%
BLEU 3	0.0115	0.0212	84.3%	0.0188	63.5%
BLEU 4	0.0046	0.0091	97.8%	0.0080	73.9%

As can be seen in **Table 3** and **Table 4**, our proposed context-aware model outperforms the baseline model at all BLEU scores. Specifically, on the ConvAI2 dataset our model with an LSTM context encoder outperforms the baseline by 50.6% in BLEU 1. When using a CNN as context encoder, the performance is still better than the baseline, surpassing it by 33.2% in BLEU 1. The same behavior is observed on our collected dataset of conversation. Both of our model's variants, i.e., with the LSTM and CNN context encoder, outperformed the baseline, respectively by 27.6% and by 48.3% respectively in BLEU 1. An interesting observation with this dataset is that our proposed model outperforms the baseline by relatively large margins in BLEU 2, 3 and 4. In other words, the baseline performs poorly on the collected dataset. This could be attributed to the small size of the data, resulting in inputs (concatenation of current utterance and context) not sufficiently long and meaningful to the baseline model. This observation also shows that our proposed model performs well on datasets of different sizes.

[3] https://www.tensorflow.org/

Table 4. Experiment result of our collected dataset

	Baseline model	Our model with LSTM context encoder	Inc.	Our model with CNN context encoder	Inc.
BLEU 1	0.0959	0.1224	27.6%	0.1422	48.3%
BLEU 2	0.0357	0.0772	116.2%	0.0817	128.9%
BLEU 3	0.0158	0.0609	285.4%	0.0580	267.1%
BLEU 4	0.0074	0.0487	558.1%	0.0422	470.3%

Comparing our two variants, the LSTM vs. CNN context encoder, it can be seen that the former (LSTM) achieves stronger performance on the ConvAI2 dataset. A similar performance is observed on our collected dataset, except at BLEU 1 and BLEU 2 where the CNN encoder achieves better results. A plausible explanation could be that the small size of the dataset and evaluation at BLEU 1 and BLEU 2 penalizes the LSTM model, which usually requires longer sequences for updating its weights.

With regards to training time, we observe that our CNN context encoder model trains faster than its LSTM counterpart. The training times on the ConvAI2 dataset are 1 hour and 43 minutes for the baseline (current utterance concatenated with context), 43 minutes for the LSTM context encoder model and 28 min for the CNN context encoder model (running on the same machine). It should be noted that the LSTM context encoder model trains faster than the baseline as it is able to handle the current utterance and context in parallel and applies attention only to the encoder.

The model accuracy and loss in training and testing are shown in **Fig. 5**, **Fig. 6** and **Fig. 7**. We can see that our models have better increment in accuracy of training and testing, and better decrement in loss of training and testing than baseline model.

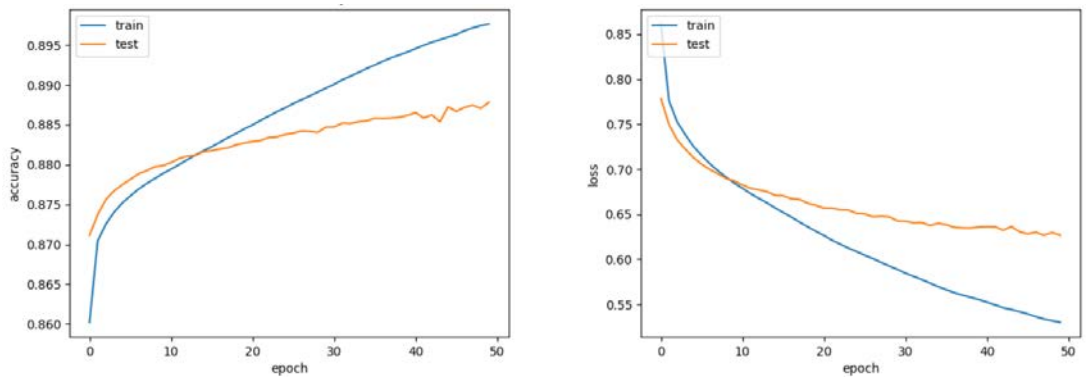

Fig. 5. Baseline model accuracy and loss in training and testing on ConvAI dataset, left: model accuracy, right: model loss.

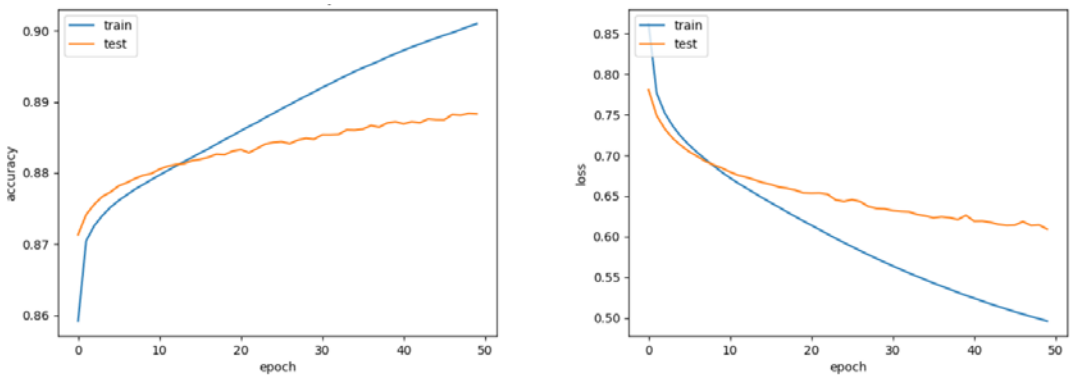

Fig. 6. LSTM context model accuracy and loss in training and testing on ConvAI dataset, left: model accuracy, right: model loss.

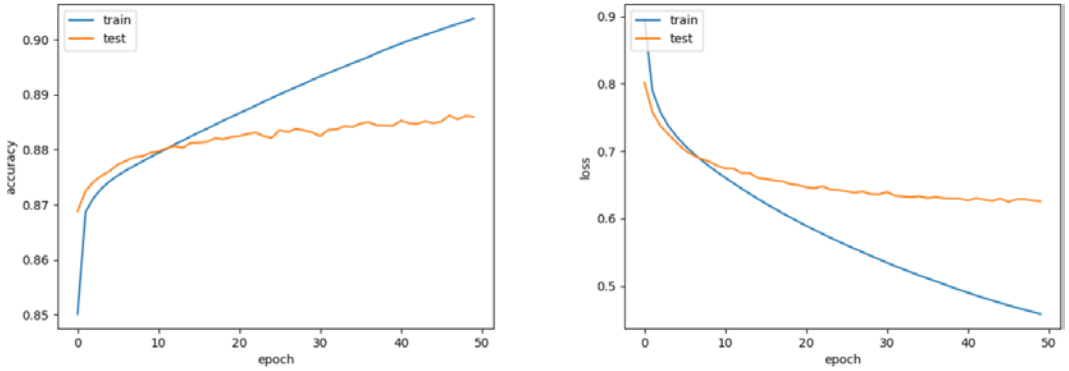

Fig. 7. CNN context model accuracy and loss in training and testing on ConvAI dataset, left: model accuracy, right: model loss.

Finally, instead of relying on the embedding layer of our model, we experimented with pre-trained word2vec embeddings (W2V)[4]. These embeddings are learned from Google News with a vocabulary size of around 3 million, and each embedding is of dimension 300. Results, depicted in **Table 5**, show that our proposed model (both LSTM and CNN context encoder) still outperforms the baseline. However, the BLEU scores did not improve. One reason could be that many words in the ConvAI2 dataset and in our collected conversation dataset did not have corresponding pre-trained embeddings, illustrating the well-known OOV problem. Our collected data has 164 OOV words, and the ConvAI2 dataset has 1808 OOV words.

Table 5. Experiment results using Word2Vec embedding in our model.

ConvAI2 dataset		**Our collected data**	
LSTM Context encoder	**CNN Context encoder**	**LSTM Context encoder**	**CNN Context encoder**
BLEU 1: 0.1036	BLEU 1: 0.0820	BLEU 1: 0.1107	BLEU 1: 0.1120
BLEU 2: 0.0428	BLEU 2: 0.0354	BLEU 2: 0.0392	BLEU 2: 0.0606
BLEU 3: 0.0216	BLEU 3: 0.0182	BLEU 3: 0.0166	BLEU 3: 0.0407
BLEU 4: 0.0104	BLEU 4: 0.0086	BLEU 4: 0.0074	BLEU 4: 0.0286

[4] https://code.google.com/archive/p/word2vec/

5. Conclusion

In this paper, we have proposed a model using deep learning with a new architecture to solve the problem of using context for automatic response generation in chatbots. In our model a module is dedicated to context representation and encoding. Other parts of the proposed model are based on Seq2Seq architecture. Our experiments results show that the results of the proposed model on different BLEU measures perform better than the method in which we encode the current utterance and its context concurrently. We have implemented LSTM and CNN models for context encoding and the experimental results also show that using LSTM is better than using CNN when handling long contexts. In summary, we have proposed an efficient model for automatic response generation in chatbot problems and demonstrated that the separation of context from the current utterance during coding has improved the quality of the model.

Acknowledgement

We would like to express our gratitude to Professor Soundararajan Ezekiel from the Department of Mathematics and Computer Science at Indiana University of Pennsylvania, USA, for his valuable discussions, which significantly improved the quality of this paper. Furthermore, we would like to thank Ms. Katherine E. Regan for providing invaluable assistance in correcting the English writing of the article.

References

[1] Zeng, X., Li, J., Wang, L., Wong, K.-F., "Joint effects of context and user history for predicting online conversation re-entries," in *Proc. of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2809–2818, 2019. Article (CrossRef Link)

[2] Bapna, A., Tür, G., Hakkani- Tür, D., Heck, L., "Sequential dialogue context modeling for spoken language understanding," in *Proc. of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pp. 103–114, 2017. Article (CrossRef Link)

[3] Li, J., Galley, M., Brockett, C., Spithourakis, G., Gao, J., Dolan, B., "A persona-based neural conversation model," in *Proc. of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 994–1003, 2016. Article (CrossRef Link)

[4] Baheti, A., Ritter, A., Li, J., Dolan, B., "Generating more interesting responses in neural conversation models with distributional constraints," in *Proc. of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3970–3980, 2018. Article (CrossRef Link)

[5] Dušek, O., Jurčíček, F., "A context-aware natural language generator for dialogue systems," in *Proc. of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp. 185–190, 2016. Article (CrossRef Link)

[6] Zhou, H., Huang, M., Zhu, X., "Context-aware natural language generation for spoken dialogue systems," in *Proc. of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 2032–2041, 2016. Article (CrossRef Link)

[7] See, A., Roller, S., Kiela, D., Weston, J., "What makes a good conversation? how controllable attributes affect human judgments," in *Proc. of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 1702–1723, 2019. Article (CrossRef Link)

[8] Pengshan Cai, Hui Wan, Fei Liu, Mo Yu, Hong Yu, and Sachindra Joshi, "Learning as Conversation: Dialogue Systems Reinforced for Information Acquisition," in *Proc. of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4781–4796, 2022. Article (CrossRef Link)

[9] Mikolov, T., Karafiát, M., Burget, L., Černocký, J., Khudanpur, S., "Recurrent neural network based language model," in *Proc. of Eleventh Annual Conference of the International Speech Communication Association*, 2010.
[10] Hochreiter, S., Schmidhuber, J., "Long short-term memory," *Neural computation*, vol. 9(8), pp. 1735–1780, 1997. Article (CrossRef Link)
[11] Serban, I.V., Sordoni, A., Bengio, Y., Courville, A., Pineau, J., "Building end-to-end dialogue systems using generative hierarchical neural network models," in *Proc. of Thirtieth AAAI Conference on Artificial Intelligence*, 30(1), 2016. Article (CrossRef Link)
[12] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proc. of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019. Article (CrossRef Link)
[13] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, et al, "Language models are few-shot learners," in *Proc. of the 34th International Conference on Neural Information Processing Systems (NIPS'20)*, pp. 1877–1901, 2020, Article 159. Article (CrossRef Link)
[14] Sutskever, I., Vinyals, O., V Le, Q., "Sequence to sequence learning with neural networks," *Advances in Neural Information Processing Systems*, vol. 4 (January), pp. 3104–3112, 2014. Article (CrossRef Link)
[15] Papineni, K., Roukos, S., Ward, T., Zhu, W., "Bleu: a method for automatic evaluation of machine translation," in *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics*, vol. 371(23), pp. 311-318, 2002. Article (CrossRef Link)
[16] Weizenbaum, J., "Eliza—a computer program for the study of natural language communication between man and machine," *Communications of the ACM*, vol. 9(1), pp. 36–45, 1966. Article (CrossRef Link)
[17] Colby, K.M., Weber, S., Hilf, F.D., "Artificial paranoia," *Artificial Intelligence*, 2(1), pp. 1–25, 1971. Article (CrossRef Link)
[18] Leuski, A., Traum, D., Npceditor, "NPCEditor: Creating virtual human dialogue using information retrieval techniques," *Ai Magazine*, vol. 32(2), pp. 42–56, 2011. Article (CrossRef Link)
[19] Ji, Z., Lu, Z., Li, H., "An information retrieval approach to short text conversation," *arXiv preprint arXiv:1408.6988*, 2014. Article (CrossRef Link)
[20] Ritter, A., Cherry, C., Dolan, W.B., "Data-driven response generation in social media," in *Proc. of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 583–593, 2011. Article (CrossRef Link)
[21] Jafarpour, S., Burges, C.J., Ritter, A., "Filter, rank, and transfer the knowledge: Learning to chat," *Advances in Ranking*, 10-15, 2009. Article (CrossRef Link)
[22] Zhang, S., Dinan, E., Urbanek, J., Szlam, A., Kiela, D., Weston, J., "Personalizing dialogue agents: I have a dog, do you have pets too?," in *Proc. of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2204–2213, 2018. Article (CrossRef Link)
[23] Ghazvininejad, M., Brockett, C., Chang, M.-W., Dolan, B., Gao, J., Yih, W.-t., Galley, M., "A knowledge-grounded neural conversation model," in *Proc. of Thirty-Second AAAI Conference on Artificial Intelligence*, 32(1), 2018. Article (CrossRef Link)
[24] Caldarini, Guendalina, Sardar Jaf, and Kenneth McGarry, "A Literature Survey of Recent Advances in Chatbots," *Information*, vol. 13, no. 1, 2022. Article (CrossRef Link)
[25] Yu, L., Zhang, W., Wang, J., Yu, Y., "Seqgan: Sequence generative adversarial nets with policy gradient," in *Proc. of the AAAI Conference on Artificial Intelligence*, vol. 31, pp. 2852–2858, 2017. Article (CrossRef Link)
[26] Kawano, S., Yoshino, K., Nakamura, S., "Neural conversation model controllable by given dialogue act based on adversarial learning and label-aware objective," in *Proc. of the 12th International Conference on Natural Language Generation*, pp. 198–207, 2019. Article (CrossRef Link)
[27] Uc-Cetina, V., Navarro-Guerrero, N., Martin-Gonzalez, A. et al, "Survey on reinforcement learning for language processing," *Artificial Intelligence Review*, vol. 56, pp. 1543–1575, 2023. Article (CrossRef Link)

[28] Yu-Ling Hsueh and Tai-Liang Chou, "A Task-oriented Chatbot Based on LSTM and Reinforcement Learning," in *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 22, pp. 1-27, 2022. Article (CrossRef Link)
[29] Derek Chen, Howard Chen, Yi Yang, Alexander Lin, and Zhou Yu, "Action-Based Conversations Dataset: A Corpus for Building More In-Depth Task-Oriented Dialogue Systems," in *Proc. of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3002–3017, 2021. Article (CrossRef Link)
[30] Li, J., Galley, M., Brockett, C., Gao, J., Dolan, B., "A diversity-promoting objective function for neural conversation models," in *Proc. of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 110–119, San Diego, California, 2016. Article (CrossRef Link)
[31] De Coster, Mathieu, and Joni Dambre, "Leveraging Frozen Pretrained Written Language Models for Neural Sign Language Translation," *Information*, vol. 13, no. 5, p. 220, 2022. Article (CrossRef Link)
[32] Yan, Rong, Jiang Li, Xiangdong Su, Xiaoming Wang, and Guanglai Gao, "Boosting the Transformer with the BERT Supervision in Low-Resource Machine Translation," *Applied Sciences*, 12, no. 14, p. 7195, 2022. Article (CrossRef Link)
[33] Eldar Kurtic, Daniel Campos, Tuan Nguyen, Elias Frantar, Mark Kurtz, Benjamin Fineran, Michael Goin, and Dan Alistarh, "The Optimal BERT Surgeon: Scalable and Accurate Second-Order Pruning for Large Language Models," in *Proc. of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 4163–4181, 2022. Article (CrossRef Link)
[34] Tianshu Shen, Jiaru Li, Mohamed Reda Bouadjenek, Zheda Mai, Scott Sanner, "Towards understanding and mitigating unintended biases in language model-driven conversational recommendation," *Information Processing & Management*, Vol. 60, no. 1, 2023. Article (CrossRef Link)
[35] Yogesh K. Dwivedi, Nir Kshetri, Laurie Hughes, Emma Louise Slade, Anand Jeyara, et al, "So what if ChatGPT wrote it? Multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI for research, practice and policy," *International Journal of Information Management*, Vol. 71, 2023. Article (CrossRef Link)
[36] Wang, Y., Li, J., King, I., Lyu, M.R., Shi, S., "Microblog hashtag generation via encoding conversation contexts," in *Proc. of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 1624–1633, 2019. Article (CrossRef Link)
[37] Tan, M., Wang, D., Gao, Y., Wang, H., Potdar, S., Guo, X., Chang, S., Yu, M., "Context-aware conversation thread detection in multi-party chat," in *Proc. of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 6456–6461, 2019. Article (CrossRef Link)
[38] Adamopoulou, E., Moussiades, L., "Chatbots: History, technology, and applications," *Machine Learning with Applications*, vol. 2, p. 100006, 2020. Article (CrossRef Link)
[39] Bahdanau, D., Cho, K., Bengio, Y., "Neural machine translation by jointly learning to align and translate," in *Proc. of 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–15, 2016. Article (CrossRef Link)
[40] Tian, Z., Yan, R., Mou, L., Song, Y., Feng, Y., Zhao, D., "How to make context more useful? an empirical study on context-aware neural conversational models," in *Proc. of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 231–236, 2017. Article (CrossRef Link)
[41] Xu, F., Xu, G., Wang, Y., Wang, R., Ding, Q., Liu, P., Zhu, Z. "Diverse dialogue generation by fusing mutual persona-aware and self-transferrer," *Applied Intelligence*, vol. 52, pp. 4744-4757, 2022. Article (CrossRef Link)
[42] Dinan, E., Logacheva, V., Malykh, V., Miller, A., Shuster, K., Urbanek, J., Kiela, D., Szlam, A., Serban, I., Lowe, R., et al., "The second conversational intelligence challenge (convai2)," *The NeurIPS '18 Competition*, pp. 187-208, 2019. Article (CrossRef Link)

[43] Logacheva, V., Malykh, V., Litinsky, A., Burtsev, M., "Convai2 dataset of non-goal-oriented human-to-bot dialogues," *The NeurIPS '18 Competition*, pp. 277–294, 2020. Article (CrossRef Link)

[44] Sordoni, A., Galley, M., Auli, M., Brockett, C., Ji, Y., Mitchell, M., Nie, J.-Y., Gao, J., Dolan, B., "A neural network approach to context-sensitive generation of conversational responses," in *Proc. of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 196–205, 2015. Article (CrossRef Link)

[45] Galley, M., Brockett, C., Sordoni, A., Ji, Y., Auli, M., Quirk, C., Mitchell, M., Gao, J., Dolan, B., "deltableu: A discriminative metric for generation tasks with intrinsically diverse targets," in *Proc. of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 445–450, 2015. Article (CrossRef Link)

Quoc-Dai Luong Tran received the BS and MS degrees of Information Technology from National University Ho Chi Minh, Vietnam, in 2009 and 2013, respectively. He is currently pursuing the Ph.D. degree in Computer Science at Ton Duc Thang University (TDTU) in Ho Chi Minh City, Vietnam. From 2018, he was a lecturer in the Department of Computer Science and was also a member of Natural Language Processing and Knowledge Discovery Laboratory at the TDTU in Ho Chi Minh City, Vietnam. His main research interests are natural language processing, deep learning, reinforcement learning and adversarial learning.

Dinh-Hong Vu received the B.S. degree in information technology from the VNU Ho Chi Minh City University of Science, Ho Chi Minh City, Vietnam, in 2005, and the M.Sc. degree in computer science from the VNU Ho Chi Minh City University of Science, in 2011. He is currently a Researcher with NLP-KD Lab, Ton Duc Thang University, Vietnam. His research interests include natural language processing, machine translation, and text clustering.

Anh-Cuong Le is now an Associate Professor of Computer Science and is also the Head of Natural Language Processing and Knowledge Discovery Laboratory at the Ton Duc Thang University (TDTU) in Ho Chi Minh City, Vietnam. He graduated with bachelor's and master's degrees of Information Technology from Vietnam National University in Hanoi in 1998 and 2001 respectively and received his Ph.D. degree of Information Science at Japan Advanced Institute of Science and Technology (JASIT) in 2007. His main research areas are in natural language processing and machine learning. He is responsible for several industrial projects of opinion analysis, text summarization and chatbot.

Ashwin Ittoo is a Full Professor, at the University of Liège, Belgium. He works in the area of NLP and machine/deep learning methods applied to business analytics. He is an Associate Editor with the Elsevier journal, Computers in Industry. His most recent research are in the areas of algorithmic collusion (AI &Law in general), NLP for requirements engineering & business process modelling, modelling customer behaviour, and temporal topic models. He obtained his PhD at the University of Groningen (The Netherlands), and his Masters and Bachelors (Engineering) from the Nanyang Tech University, and the National University of Singapore (both in Singapore).