Anna: A Dapper Open-Domain Dialogue Agent Based on a Joint Attention Network

Itsugun Cho† and Hiroaki Saito†

We constructed a high-quality open-domain dialogue generation model called Anna that is composed of a hierarchical self-attention network with multiple convolution filters and a topic-augmented network. During daily conversations, humans typically respond by understanding a dialogue history and assembling their knowledge regarding the topic. However, existing dialogue generation models are weak at capturing the dependencies among words or utterances, resulting in an insufficient understanding of context and the generation of irrelevant responses. Previous works have largely ignored topic information modeling in multi-turn dialogue, making responses overly generic. Although pre-training using large-scale transformer models has recently resulted in enhanced performance, large parameter sizes complicate such models. Anna effectively captures contextual dependencies and assigns greater weight to important words and utterances to compute context representations. We incorporate topic information into our model as prior knowledge to synthesize topic representations. Two types of representations jointly determine the probability distributions of responses, which effectively simulates how people behave in real conversations. Empirical studies on both Chinese and English corpora demonstrate that Anna outperforms baseline models in terms of response quality, parameter size and decoding speed.

Key Words: Multi-turn Dialogue Response, Dialogue Generation System, Chatbot

1 Introduction

Dialogue agents can be divided into goal-driven and non-goal-driven agents. Goal-driven systems are constructed in vertical domains for specific tasks such as technical support services (Young et al. 2013; Shawar and Atwell 2007). Non-goal-driven systems aim to have natural conversations with people regarding a wide range of topics in open domains and include chatbots. This paper focuses on multi-turn dialogue response generation in an open domain in which we attempt to train a response generation model using large-scale context-response pairs. A context refers to several previous utterances (turns). In practice, a model takes the context as an input and generates a response in the next turn.

† Keio University
“dapper” is used in the title to indicate that Anna achieves a both light weight and good performance.
By summarizing the deficiencies of previous works, we determined that in this task, there are three primary challenges: (1) how to overcome the weakness of traditional recurrent neural networks (RNNs) for capturing contextual dependencies and accurately highlighting important parts of contexts to generate responses relevant to the context, (2) how to make a response less generic and generate a more specific response, and (3) how to balance the parameter size and model capability. To tackle these problems, we constructed contextual dependencies and topic information for multi-turn dialogue in a unified framework. Inspired by the self-attention mechanism, which performs better than a traditional RNN by effectively capturing sequence dependencies, particularly long-term dependencies, as demonstrated in (Vaswani et al. 2017), we transformed the self-attention mechanism into a hierarchical attention architecture with multiple convolution filters to model contexts reasonably. We define this architecture as a hierarchical self-attention network. We noticed that conversations typically revolve around a particular topic and that people give answers based on their understanding of the dialogue by assembling their knowledge regarding the topic. Therefore, this paper also proposes a topic-augmented network for modeling the topics of multi-turn dialogue. We call this new model Anna.

In the encoding phase, context information is encoded into a context representation by the hierarchical self-attention network, which operates in a bottom-up manner. More important words and utterances can contribute more to synthesizing context representations by capturing contextual dependencies. The topic words of a context are acquired from a pre-trained Latent Dirichlet Allocation (LDA) model (Blei et al. 2003). The weights between utterance representations and each topic word embedding are computed using a topic-augmented network. Topic words that are more relevant to the context will have a greater effect on the synthesis of topic representations. In the decoding phase, context representations and topic representations are projected jointly into the distribution space of the response vocabulary and topic vocabulary by a vanilla decoder. The probability distributions of the two types of vocabularies are used for training and inference. Therefore, generated responses not only provide insights into contextual dependencies, but also grasp correlated topic information. We conducted comprehensive evaluations of large-scale dialogue datasets in English and Chinese. The effectiveness and superiority of our model were demonstrated through these evaluations.

The main contributions of this paper can be summarized as follows: (1) a proposal of a hierarchical self-attention network that effectively models contextual dependencies in a few-parameter hierarchy, (2) a proposal of a topic-augmented network that incorporates topic information into multi-turn dialogue modeling, (3) an empirical verification of the effectiveness of the proposed model on both English and Chinese corpora through automatic evaluations and human judgment.
2 Related Work

2.1 Single-turn Dialogue Generation Models

Some studies have improved the anthropomorphic characteristics of generated responses. For example, the backgrounds of persona (Zhang et al. 2018; Mazaré et al. 2018; Madotto et al. 2019; Cui et al. 2018) and emotional information (Zhou et al. 2020; Huang et al. 2018; Rashkin et al. 2018; Li et al. 2019) have been incorporated into encoder-decoder architectures. These studies aimed to address the problem of generating generic responses, which is also one of the motivations for our study. This paper focuses on the sculpting of dialogue topic information. However, previous studies have focused on single-turn dialogue generation, whereas we focused on response generation for multi-turn dialogue. Previously, there have been some pioneering works on the modeling of topic information (Li et al. 2015; Mou et al. 2016; Xing et al. 2017) for single-turn dialogue, but most have merely fixed a single topic word in a response.

2.2 Multi-turn Dialogue Generation Model

2.2.1 RNN-based Model

Multi-turn dialogue response generation has attracted significant attention from academia in recent years. Various hierarchical recurrent models have been used in this area, including HRED\(^1\) (Serban et al. 2016). Under the architecture of HRED, additional variants such as MrRNN (Serban et al. 2017a) and VHRED\(^2\) (Serban et al. 2017b) have been proposed to incorporate stochastic latent variables to improve the diversity of responses. However, these models use all contexts indiscriminately, leading to unsatisfactory performance. To solve this problem, some researchers proposed HARN (Xing et al. 2018), which is a variant of HRED, by introducing a traditional attention mechanism into an RNN. However, RNN-based attention models are typically biased toward closer utterances in a context, meaning they suffer from the position bias problem (Hochreiter et al. 2001). Furthermore, double-layer dynamic attention changes the context representation in each step of the decoding process, which can lead to incoherent responses (Cui et al. 2018). (Zhang et al. 2020) detected topic-level attention relevance to handle the topic draft problem through the use of a pre-trained bi-term topic model. Because this method essentially integrates topic information into HRED, the shortcomings of the RNN-based networks discussed above are not resolved in their model. The main merit of our model compared to theirs is the effective capture of contextual dependencies.

\(^1\) https://github.com/hsgodhia/hred
\(^2\) https://github.com/lipiiji/dialogue-hred-vhred
2.2.2 RNN and Self-attention Hybrid Model

ReCoSa\(^3\) (Zhang et al. 2019) focuses on modeling the relevance between response representations and contexts to allocate attention weights for utterances reasonably. However, ReCoSa is still inadequate in terms of context modeling because it uses an RNN-based utterance-level encoder that may lose important information in an utterance. Through experimentation, we also found that ReCoSa is prone to generating generic responses such as “Okay I know.” These responses carry little information, making it difficult to keep a conversation going.

2.2.3 Auxiliary Task Model

(Zhao et al. 2020) devised a group of self-supervised auxiliary tasks that help their models produce better features for response generation. They introduced maximum likelihood estimation during learning with four self-supervised auxiliary tasks. The key concept of their method is to transfer the burden of context understanding from modeling to learning. However, they directly concatenate the utterances in a dialogue as a long sequence and feed the sequence in an encoder. It is unclear how auxiliary tasks can remedy the loss of utterance-level dependencies in this form of non-contextual modeling. Additionally, this approach makes it easy for the matrix inputs for the model to become enormous as the utterance length increases. This makes it difficult to accelerate the decoding process. This could be why utterances were truncated in their study and only the first 25 words were kept. However, the length of an utterance often exceeds 25 words in daily life when people narrate something or explain a perspective. Particularly in languages with a convention of equivocal expressions such as Japanese or Chinese, more words typically appear in an utterance. Therefore, their study was only evaluated on an English corpus. In Section 5.6.5, some comparisons between Anna and the method from (Zhao et al. 2020) are presented. We then describe the advantages of our model in detail.

2.2.4 Large-scale Pre-training Model

Recently, task-agnostic pre-training using large-scale transformer models has achieved significant success in natural language generation. Meena (Adiwardana et al. 2020) scaled up the parameter size to 2.6 B to obtain a human-like chatbot. Blender (Roller et al. 2021) fine-tuned a pre-trained model using human-annotated datasets, making dialogue more personalized. PLATO-2\(^4\) (Bao et al. 2020) constructed an effective training schema via curriculum learning and exhibited state-of-the-art performance. However, the parameter size was more than triple that of ReCoSa.

\(^3\) https://github.com/zhanghainan/ReCoSa
\(^4\) https://github.com/PaddlePaddle/Knover/tree/master
Scilicet pre-training models improved performance by increasing the number of parameters, but this created obstacles when hardware resources were limited. A model with more parameters incurs a higher computational cost, leading to greater expenditure during research. In our experiments, we spent 385 h training PLATO-2 on a Chinese dataset, whereas the training of Anna only required 96 h. Memory overflow occurred when PLATO-2 was trained on an RTX5000 GPU based on the large parameter size, so we were forced to switch to an RTX8000 GPU.

3 Background

3.1 Self-attention Mechanism

Given a vector sequence $X = (x_1, x_2, \ldots, x_n)$, where $x_i$ is a $d$-dimensional vector, $(x_1, x_2, \ldots, x_n)$ is considered as a matrix. Matrices are denoted as $Q_{seq} \in \mathbb{R}^{n \times d}$, $K_{seq} \in \mathbb{R}^{n \times d}$, and $V_{seq} \in \mathbb{R}^{n \times d}$. Different linear projections are implemented in these matrices to obtain queries, keys, and values, which are denoted as $Q \in \mathbb{R}^{n \times c}$, $K \in \mathbb{R}^{n \times c}$, and $V \in \mathbb{R}^{n \times c}$, respectively.

$$Q = Q_{seq}W^Q, \quad K = K_{seq}W^K, \quad V = V_{seq}W^V,$$

where the parameter weight matrices $W^Q \in \mathbb{R}^{d \times c}$, $W^K \in \mathbb{R}^{d \times c}$, $W^V \in \mathbb{R}^{d \times c}$. $W^Q$, $W^K$, and $W^V$ are called attention head. The attention scores are calculated using the scaled dot product and the weighted sum of $V$ is obtained as

$$Attention(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V.$$

The self-attention mechanism adopts the multi-head approach which is multiple sets of weight matrices that project an original sequence into different representation subspaces to focus on information from different positions. For the $i$-th head, the weighted sum of $V_i$ is calculated as

$$Q_i = Q_{seq}W_i^Q, \quad K_i = K_{seq}W_i^K, \quad V_i = V_{seq}W_i^V,$$

$$\text{head}_i = Attention(Q_i, K_i, V_i),$$

where the parameter weight matrices $W_i^Q \in \mathbb{R}^{d \times c/h}$, $W_i^K \in \mathbb{R}^{d \times c/h}$, and $W_i^V \in \mathbb{R}^{d \times c/h}$, and $h$ is the number of heads. All heads are concatenated together into a single matrix and a linear projection is then used to mix the subspaces from different heads.

$$\text{MultiHead}(Q_{seq}, K_{seq}, V_{seq}) = \text{Concat(\text{head}_1, \ldots, \text{head}_h)}W^O,$$

where the parameter weight matrix $W^O \in \mathbb{R}^{c \times d}$. See (Vaswani et al. 2017) for more details.
3.2 LDA Model

LDA is a generative probabilistic model. LDA is typically used in the topic mining of text with unsupervised learning. The basic concept of LDA is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. This is similar to the standard bag-of-words model assumption. Fig. 1 presents a graphical representation of LDA. Given a topic \( k \in [1, K] \), \( K \) is the number of topics in the topic set. Given a document \( m \in [1, M] \), \( M \) is the number of documents in the corpus. Given a word \( n \in [1, N] \) in document \( m \), \( N \) is the number of words in the document. \( \alpha \) and \( \beta \) denote the parameters of a Dirichlet for document-topic distributions and topic-word distributions, respectively.

The LDA model is constructed as follows. (1) Draw a multinomial distribution \( \theta_m \) from a Dirichlet prior \( \text{Dir}(\alpha) \) that represents the topic distribution of document \( m \). (2) Sample from the topic multinomial distribution \( \theta_m \) to generate a topic \( z_{m,n} \) for the \( n \)-th word in document \( m \). (3) Draw a multinomial distribution \( \phi_k \) from a Dirichlet prior \( \text{Dir}(\beta) \) that represents the word distribution of topic \( z_{m,n} \). (4) Sample from word the multinomial distribution \( \phi_k \) to generate word \( w_{m,n} \). Each document in the corpus can be denoted as a joint probability distribution. 

\[
P(w, z | \alpha, \beta) = \prod_{m=1}^{M} \frac{\Delta(n_m + \alpha)}{\Delta(\alpha)} \prod_{k=1}^{K} \frac{\Delta(n_k + \beta)}{\Delta(\beta)},
\]

where \( n_m \) is the vector of the number of words in each topic in document \( m \) and \( n_k \) is the vector of the number of times each word appears in the corpus under topic \( k \). Model training can
be performed through the collapsed Gibbs sampling of $z$ to maximize the joint probability to estimate $\theta$ and $\phi$. We direct readers to (Blei et al. 2003) for additional details.

4 Model

4.1 Problem Formalization

Given a context $C = (U_1, ..., U_m)$, response $R = (y_1, ..., y_n)$, and set of topic words $Z = (k_1, ..., k_l)$, $U_i$ is the $i$-th utterance, $y_i$ is the $i$-th response word, and $k_i$ is the $i$-th topic word. For all $i$, $U_i = (w_{i,1}, ..., w_{i,n_i})$, where $w_{i,j}$ is the $j$-th utterance word. We denote a corpus as $\mathcal{C}$, which consists of $(C, R, Z)$ pairs. Our goal is to learn a mapping function $P(y_1, ..., y_t | C, Z)$ from $\mathcal{C}$ that can generate a response $G = (g_1, ..., g_t)$ when a new dialogue context $C$ and its topic words $Z$ are given. The construction of Anna and the process for training the mapping function are detailed in Sections 4.2 and 4.3, respectively.

4.2 Encoder

4.2.1 Hierarchical Self-attention Network

Hierarchical self-attention consists of an utterance-level encoder and context-level encoder, both of which adopt a self-attention mechanism. For each word $w_{i,j}$ in an utterance $U_i$, the token embedding is calculated as the sum of a word embedding and its position embedding as

$$I(w_{i,j}) = WE(w_{i,j}) + PE(w_{i,j}),$$

where $WE$ is a function for obtaining a word embedding from an embedding table. $PE$ is a function that provides position information. \{I(w_{i,1}), ..., I(w_{i,n_i})\} is considered as a matrix $M_{U_i} \in \mathbb{R}^{n_i \times c}$. To obtain an utterance representation, $M_{U_i}$ is fed into the utterance-level encoder in the form of queries, keys, and values by using different linear projections. The multi-head attention computes the attention representation encoding $U_i$ in the form of hidden vectors.

$$h_{i,1}, ..., h_{i,n_i} = MultiHead(M_{U_i}, M_{U_i}, M_{U_i}).$$

A convolution operation involving a filter $f \in \mathbb{R}^{h \times 1}$ is then applied to a window with a height of $h$ and width of one. $(h_{i,1}, ..., h_{i,n_i})$ is considered as an image $M_{h_i} \in \mathbb{R}^{n_i \times c}$ containing the semantic information of $U_i$. Let $M_{h_i}^t \in \mathbb{R}^{h \times 1}$ represent the $t$-th window position in $M_{h_i}$, where the filter is scanning, and a feature $s_i^t$ is generated from the $t$-th window as

$$s_i^t = Filter(W^FM_{h_i}^t + b),$$

1190
where $b$ is a bias term and $\text{Filter}$ is a nonlinear function (Tanh was used in our experiments). $W^F \in \mathbb{R}^{1 \times h}$ denotes the parameter weight matrix. The stride of the filter along the height is $k$ ($k > 0.5h$) and that along the width is one. The filter is applied to each possible window in $M_{hi}$ to generate a feature map $Map_i = (s_1^i, s_2^i, ..., s_{n_i}^{i/k \times c})$ with $Map_i \in \mathbb{R}^{n_i/k \times c}$. Because the width of filter $w$ is one, this process performs convolution on each dimension of the hidden vectors to extract the semantic features of $U_i$. In our implementation, the feature map $Map_i$ is obtained using multiple convolution operations. A pooling operation is then applied to the resulting feature map and the utterance representation of $U_i$ is computed as $S_i$.

$$S_i = \text{MaxAbsPool}(Map_i),$$

where $\text{MaxAbsPool}$ is a function that calculates the maximum absolute value of the last dimension and then takes on the corresponding value. The description above covers the extraction of a feature map from one filter. Our model employs multiple filters with varying window heights to obtain multiple semantic features and calculates averages to derive utterance representations, as shown in Fig. 2. The matrices $\{M_{U,i}\}_{i=1}^n$ are fed into the utterance-level encoder to obtain $\{S_i\}_{i=1}^m$, where the computing process is a parallel process. Position information is included in $S_i$ to indicate which turn $S_i$ belongs to.

$$I(S_i) = S_i + PE(S_i).$$

$\{I(S_1), ..., I(S_m)\}$ is considered as a matrix $M_C \in \mathbb{R}^{m \times c}$. To obtain the context representation $O_C$, matrix $M_C$ is utilized as an input for the context-level encoder. The concept of the context-level encoder is similar to that the utterance-level encoder, which contains a multi-head attention
layer followed by a feed-forward neural network (FF).

\[ O_C = FF(\text{MultiHead}(M_C, M_C, M_C)). \]  

(12)

To facilitate matrix operations in the decoding phase, we do not conduct dimensional reduction on \( O_C \). Therefore, \( O_C \) is a vector sequence. To maintain the hierarchy of the utterance level and context level, parameter weight matrices between the utterance-level encoder and context-level encoder are not shared.

### 4.2.2 Topic-Augmented Network

We acquire the topic words of a dialogue from a pre-trained LDA model. Each context and its responses are concatenated to form a short document. Then, the collapsed Gibbs sampling algorithm is applied to each short document to estimate the parameters of the LDA model. The pre-trained LDA model allocates the topic \( z \) with the highest probability in \( \theta \) to a context. We select the top-\( n \) words with the highest probabilities in \( \phi \) under topic \( z \) as topic words. As discussed in Section 3.2, \( \theta \) represents the topic distribution and \( \phi \) represents the word distribution.

In the topic-augmented network, given a set of topic words \( Z = (k_1, ..., k_l) \) for the context \( C \), the embedding of topic words is calculated using the function \( \text{WE} \), as discussed in Section 4.2.1. With a minor exploitation of notation, we also use \((k_1, ..., k_l)\) to denote the embeddings of the topic words. These embeddings are linearly combined as the topic embedding by a topic-augmented network. The combination weight of \( k_n \) and the topic embedding \( O_z \) are given by

\[ \alpha_n^i = \frac{\exp(\eta(W^Z k_n, S_i))}{\sum_{m=1}^l \exp(\eta(W^Z k_m, S_i))}, \]  

(13)

\[ O_{z_i} = \sum_{n=1}^l \alpha_n^i W^Z k_n, \]  

(14)

where \( \eta \) is a multilayer perceptron. The topic-augmented network is different from a traditional attention network and the parameter matrix \( W^Z \) is introduced to reduce the effects of different words with similar cosine similarities on the weight calculation. The topic embedding of each utterance in \( C \) is denoted as \( \{O_{z_1}, ..., O_{z_m}\} \) and considered as a topic representation matrix \( O_Z \in \mathbb{R}^{m \times c} \). Fig. 3 presents the architecture of Anna. In the next section, we will elaborate on \( O_C \) and \( O_Z \), which jointly affect the likelihood of the response sequence in the decoding phase.
Fig. 3  Architecture of the Anna model.

4.3 Decoder

For training, the hidden vectors of the response $R$ are calculated using Eq.(7) and Eq.(8). A mask matrix operator is applied to $R$ to obtain the response representation $O_R$. The mask operation aims to prevent positions from attending to its subsequent positions. Another multi-head attention feeds in the $O_R$ as queries, context representation $O_C$ as keys, and $(O_C \oplus O_Z)$ as values to calculate the hidden vectors. The operator $\oplus$ denotes element-wise addition.

$$O_{d_1}, ..., O_{d_n} = FF(MultiHead(O_R, O_C, O_C \oplus O_Z)).$$  \hfill (15)

The probability distribution $P(y_1, ..., y_n \mid C, Z)$ is defined as

$$P(y_1, ..., y_n \mid C, Z) = P(y_1 \mid O_{d_1}) \prod_{i=2}^{n} P(y_i \mid O_{d_i}, y_1, ..., y_{i-1}),$$  \hfill (16)

where $P(y_i \mid O_{d_i}, y_1, ..., y_{i-1})$ is given by

$$P(y_i \mid O_{d_i}, y_1, ..., y_{i-1}) = \Psi_{y_i} \cdot \text{softmax}(O_{d_i}W^E).$$  \hfill (17)
The $W^E$ is a shared weight matrix composed of learned word embeddings, which is similar to the method proposed in (Press and Wolf 2017). $W^E$ converts $O_{d_i}$ into vectors of dimension $V$, where $V$ is the response vocabulary size. $\Psi_{y_i}$ is a one-hot vector for $y_i$ in the response vocabulary. Suppose that $k$ ($k < n$) elements in the set of topic words $Z$ appear in response $R$, meaning that $k$ words in the response are topic words. The order in which these topic words appear in $R$ is denoted as $\{y'_i\}_{i=1}^k$. The corresponding hidden vector of $\{y'_i\}_{i=1}^k$ is denoted as $O_{d_x} \in \{O_{d_i}\}_{i=1}^n$. Because the positions of the topic words appearing in $R$ are random, $O_{d_x}$ is an unfixed element.

For example, if the topic word $y'_1$ is the third word in $R$, its hidden vector $O_{d_x}$ is $O_{d_3}$. If the topic word $y'_2$ is the seventh word in $R$, its hidden vector $O_{d_x}$ is $O_{d_7}$. Therefore, the index of $O_{d_x}$ depends on where the topic word appears in $R$. The distribution $P(y'_1, ..., y'_n | C, Z)$ is defined as

$$P(y'_1, ..., y'_n | C, Z) = P(y'_1 | O_{d_x}) \prod_{i=2}^k P(y'_i | O_{d_x}, y'_1, ..., y'_{i-1}),$$  
(18)

where $P(y'_i | O_{d_x}, y'_1, ..., y'_{i-1})$ is given by

$$P(y'_i | O_{d_x}, y'_1, ..., y'_{i-1}) = \Psi_{y'_i} \cdot \text{softmax}(O_{d_x}W).$$  
(19)

The $W$ is a parameter that converts $O_{d_x}$ into vectors of dimension $H$, where $H$ is the topic vocabulary size. $\Psi_{y'_i}$ is a one-hot vector for $y'_i$ in the topic vocabulary. We let $\Theta$ denote the parameter set of Anna and estimate $\Theta$ from the corpus $\mathcal{C} = \{(C_i, R_i, Z_i)\}_{i=1}^N$ by minimizing the following loss function, where $N$ refers to the number of dialogues in the corpus $\mathcal{C}$.

$$\text{Loss}_{C_i} = \log(P(y_{i,1}, ..., y_{i,n_i} | C_i, Z_i)),$$  
(20)

$$\text{Loss}_{Z_i} = \log(P(y'_{i,1}, ..., y'_{i,n_i} | C_i, Z_i)),$$  
(21)

$$\Theta = \arg\min_{\Theta}(- \sum_{i=1}^N (\text{Loss}_{C_i} + \text{Loss}_{Z_i})).$$  
(22)

One can see that the probability distribution of the response vocabulary and topic vocabulary jointly determine the parameters of Anna based on the above two loss items. Similarly, this joint probability distribution is used for inference. Suppose that the generated response $G = \{g_1, ..., g_t\}$. Then, the generation probability $P(g_i)$ is defined by

$$P_C(g_i) = \Psi_{g_i} \cdot \text{softmax}(O_{d_i}W^E),$$  
(23)

$$P_Z(g_i) = \Psi_{g_i} \cdot \Psi_{\text{mat}} \cdot \text{softmax}(O_{d_i}W),$$  
(24)

$$P(g_i) = P_C(g_i) + P_Z(g_i),$$  
(25)

where $\Psi_{\text{mat}}$ is a one-hot matrix that projects the generation probability of the topic vocabulary.
into the corresponding position in the response vocabulary. Words in the response vocabulary that do not appear in the topic vocabulary corresponding to the columns of $\Psi_{mat}$ all have values of zero. Therefore, if $g_i$ is a topic word, then $P(g_i)$ is the joint generation probability from the response vocabulary and the topic vocabulary. If $g_i$ is not a topic word, then $P(g_i)$ is the generation probability of the response vocabulary. Therefore, topic words have a greater likelihood of appearing. For inference, the response words generated in each step are fed into the decoder as queries in the next time step.

5 Experiments

We comprehensively evaluated the quality of the responses generated by Anna and baseline models through automatic evaluations and human evaluations. Subsequently, we further investigated the effectiveness and superiority of Anna.

5.1 Datasets

In our experiments, Anna was trained on both Chinese and English public multi-turn dialogue datasets, which were extracted from open-domain social dialogue. Table 1 presents the statistics of the datasets. The details are discussed in the following subsections.

5.1.1 Chinese Dataset

The Chinese dataset was collected from the Douban Conversation Corpus (Wu et al. 2019), which was crawled from the Chinese social networking Douban group. For each dialogue, the last turn is considered as the response and the previous turns are considered as the context. We performed meticulous preprocessing on these context-response pairs. Specifically, duplicates were first removed because they would dominate the generated results if they were included in the training data. We then used the Jieba tool\(^5\) for word segmentation and removed pairs with

| Statistics                        | Douban | DailyDialog |
|-----------------------------------|--------|-------------|
| Dialogues for Training            | 1,000,000 | 76,726      |
| Dialogues for Validation          | 50,000  | 7,069       |
| Dialogues for Testing             | 10,000  | 6,740       |
| Avg. Speaker Turns Per Dialogue   | 6.5     | 7.9         |
| Avg. Tokens Per Utterance         | 19.3    | 14.6        |

\(^5\) https://github.com/fxsjy/jieba
any turn longer than 50 words. Finally, long pairs were split into multiple instances to make the length of each pair less than 20 turns. After preprocessing, we randomly split the pairs into training, validation, and testing sets containing 800,000, 10,000, and 1,000 pairs, respectively. We separately constructed two vocabulary sets that were formed from all words appearing in the training data. The response vocabulary contained 56,330 words. The topic vocabulary contained 5,000 words. Words outside the response vocabulary were denoted as “UNK.”

5.1.2 English Dataset

The English dataset was extracted from DailyDialog (Li et al. 2017). The raw data in DailyDialog were crawled from websites focusing on English dialogue practice and resemble human communications in typical scenarios. The English dataset was preprocessed in a manner similar to the Chinese dataset. We randomly split the pairs into training, validation, and testing sets that contained 74,083, 1,500, and 1,000 pairs, respectively. We also constructed two vocabulary sets. The response vocabulary contained 20,251 words and the topic vocabulary contained 1,004 words. Words outside the response vocabulary were denoted as “UNK.” In contrast to previous works, contexts and responses share a vocabulary set in Anna and the topic vocabulary can also be considered as a subset of the response vocabulary.

5.2 Baselines

The following four baseline methods were considered in our experiments. (1) HRED: a hierarchical RNN-based encoder-decoder model proposed in (Serban et al. 2016). (2) VHRED: an extension of HRED in which a variational autoencoder is incorporated to increase the diversity of responses (Serban et al. 2017b). (3) ReCoSa: an RNN-based utterance-level encoder and self-attention context-level encoder model proposed in (Zhang et al. 2019). (4) PLATO-2: a chatbot trained using a large-scale transformer that exhibits state-of-the-art performance on several benchmarks. In our experiments, we compared Anna to the 333 M parameter model (Chinese model) and 310 M parameter model (English model) proposed in (Bao et al. 2020).

5.3 Evaluation Metrics

For automatic evaluation, we employed perplexity, distinctness, parameter size, and decoding time. For human evaluation, we recruited human annotators to perform side-by-side judgment.

---

(Xing et al. 2018) and (Liu et al. 2016) proved that BLEU (Papineni et al. 2002) is unreliable as a metric of generation quality in multi-turn dialogue. The units for perplexity and distinctiveness are percentages.
5.3.1 Perplexity (PPL)

Perplexity is a measure of how well a model generates a response. A lower perplexity generally indicates that the probability distribution learned by the model is close to that of the training data. Therefore, perplexity is typically used to evaluate whether a generated response is human-like in terms of syntax and semantics. The perplexity of a response $R$ is defined as

$$\text{PPL} = \exp\left\{ -\frac{1}{n} \sum_{i=1}^{n} \log(P(y_i)) \right\},$$  \hspace{1cm} (26)

where $-\sum_{i=1}^{n} \log(P(y_i))$ is a term of cross-entropy. Therefore, we obtain the PPL by calculating an exponent based on $e$ for the loss function. In our experiments, the PPL for the validation sets were used to determine when to stop training and for the testing sets were used for evaluation.

5.3.2 Distinctness

We evaluated the degrees of informativeness and diversity of the generated responses using metrics denoted as distinct-1 and distinct-2. These metrics were calculated as the numbers of distinct unigrams and bigrams divided by the total number of generated words, respectively. Higher ratios indicate more content in generated responses and ensure that a dialogue is scalable.

$$\text{Distinct-1} = \frac{\text{Count(unique unigram)}}{\text{Count(generated words)}},$$ \hspace{1cm} (27)

$$\text{Distinct-2} = \frac{\text{Count(unique bigram)}}{\text{Count(generated words)}}.$$ \hspace{1cm} (28)

5.3.3 Parameter Size and Decoding Time

A model should use a few parameters as possible to maximize efficiency. Therefore, parameter size is a helpful metric to evaluating the performance of a model. The number of all trainable variables in a model is considered as the parameter size.

The length of the decoding (response) time is also an important indicator for assessing response quality. A long response time may negatively affect user experiences when a model is applied in daily life. The average response time per word was calculated as the decoding time.

5.3.4 Human Annotation

For human evaluations, we recruited six native speakers (three Chinese speakers and three English speakers) as human annotators. We randomly sampled 100 pairs from the Chinese and English testing datasets, and each annotator was asked to judge which response was better between Anna and the baselines. The responses were selected using a greedy search and randomly
shuffled. The criteria for response A being better than B were defined as follows: (1) A is syntactically and semantically correct (fluent), whereas B is not; (2) both A and B are fluent, but A is relevant and logically consistent with the context, whereas B is not; (3) both A and B satisfy (1) and (2), but A is more informative and interesting than B. If an annotator could not judge which response was better, they were required to record a tie. Because the annotators independently judged 100 examples for each Anna-baseline pair, each annotator labeled 400 examples from the testing dataset. The agreement among annotators was calculated using the Fleiss Kappa (Fleiss 1971).

5.4 Implementation Details

In our model, word embeddings were learned by a network initialized using the Xavier method (Glorot and Bengio 2010). The number of hidden nodes was set to 512 and the multi-head number for Anna was set to eight. The number of topics for pre-training the LDA model was set to 100 and the number of topic words was set to 50 for each topic. The open-source platform TensorFlow-Gpu 1.14.1 was utilized for model training. The Adam algorithm was utilized for optimization with a learning rate of 0.0002. The batch size was set to 64. For fair comparisons between the baselines and Anna, the maximum number of turns for all dialogues was set as 10 and the maximum length of each utterance was set to 50. We executed all the baselines on an NVIDIA Quadro RTX8000 GPU. All models were tuned using the validation sets based on perplexity. If the perplexity did not drop for five consecutive epochs, we considered that the algorithm had reached convergence and terminated training.

5.5 Evaluation Results

Table 2 presents the automatic evaluation results. One can see that Anna outperforms all the baselines on both the Chinese and English public datasets. The reason why Anna achieves a PPL drop is that hierarchical self-attention network can model contextual dependencies reasonably. For distinct-1, Anna outperforms ReCoSa by 0.973 points on the Chinese dataset. For distinct-2, Anna outperforms PLATO-2 by 1.241 points on the English dataset. This further supports our claim that leveraging topic information as prior knowledge is helpful for generating responses. Regarding decoding time, our model is the fastest, meaning that Anna can generate responses more rapidly than the baselines. Regarding the human annotation results, the win-loss ratios were calculated by combining the annotations from judges, as shown in Table 3 (the rates of wins, losses, and ties are rounded). The win percentage is always greater than the loss percentage for Anna, meaning our model generates more high-quality responses than the baselines. All the
Kappa values exceed 0.46, indicating moderate agreement among annotators. Among the four baselines, the RNN-based models HRED and VHRED perform worse than the attention-based models ReCoSa and PLATO-2 because they lose information regarding segments of the context that should receive focus. PLATO-2 is the closest to our model, but the averaged parameter size of Anna is one-fifth of that of PLATO-2. In other words, because we construct context-topic information optimally, even though Anna makes use of fewer parameters, it still achieves good performance. We conducted a statistical test on the results of the automatic evaluations and human annotations. The results revealed significant differences between our model and the baselines on both the Chinese and English datasets (Anna versus ReCoSa for human annotation on the English dataset has a \( p \)-value < 0.05 and the other comparisons have \( p \)-values < 0.01). We also scrutinized the cases in which PLATO-2 lost to Anna. The results revealed that in 22 cases, PLATO-2 generated irrelevant responses and in 28 cases, PLATO-2 generated generic responses, whereas the responses generated by Anna were more relevant and diverse.
5.6 Discussions

In this section, some case studies are presented to facilitate a better understanding of Anna. To verify the effects of different components of Anna further, we discuss the following topics: Q1 how hierarchical self-attention lets Anna produce high-quality responses, Q2 how the encoders for the utterance level and context level affect performance differently, and Q3 how the topic-augmented network affects the performance of our model. Finally, analysis and additional experiments are presented to compare our model to the method proposed in (Zhao et al. 2020).

5.6.1 Case Study

Fig. 4 presents four cases extracted from the English testing dataset used to compare Anna to the baselines. One can see that Anna provides answers that are not only relevant to the context, but also diverse. In case 1, topic information provides prior knowledge that overseas study typically gives the impression of loneliness and is hard for people, which helps generate to a response that targets the topic of the dialogue (i.e., “study abroad,” “lonely,” and “friends”). In case 2, omitted mentions appear in the last turn (i.e., drinking beer will make a person fat and act stupidly). Clearly, Anna caught these dependencies and generated a proper response. In contrast, HRED and VHRED generate irrelevant responses. Although the responses from ReCoSa and PLATO-2 also echo the context, they carry little information. In case 3, Anna notices that sports are mentioned in u1, so it naturally asks the question “which sports do you like to play?” This response can extend the dialogue. In case 4, Anna associates travel and shopping in the context with relevant topic elements such as saving money, allowing it to generate an intelligent response of “I need to save money for travel.” In contrast, the responses from the baselines ignore the topic information and contain less specific responses related to “go-to travel.” The manner in which Anna focuses on which utterances and words are important in the context will be explained in Section 5.6.2.

5.6.2 Answer to Q1

We went one step further to analyze how Anna produces high-quality responses by visualizing the hierarchical self-attention network. The results are presented in Fig. 5. In each row and the leftmost column, a darker color indicates greater importance, which is computed from the average weight assigned by the hierarchical self-attention network. This visualization can represent the overall contributions of words and utterances. Responses are given at the bottom of each subfigure. Regarding the implementation of this visualization, we followed the method in (Xing et al. 2018). One can see that the hierarchical self-attention network pays more attention
Fig. 4 Case study.
Fig. 5 Hierarchical self-attention visualization.
to important parts of the context. In Fig. 5(b), the word “beers” in u1 is more highlighted than the other words because the hierarchical self-attention network determined that all of the other words were strongly dependent on “beers.” In this context, u1 and u4 are more important than the other elements and the word “fat” in u4 is the most important, which echoes the word “beers” in u1. This explains why the omitted information in u4 does not degrade the answer quality. Instead, Anna answered with the phrase “drinking a little red wine,” which is closely related to the context. Additionally, the highlighted word “it” in u3 refers to the “beers” mentioned in u1. The hierarchical self-attention network is well aware of this co-reference in the context. In Fig. 5(c), the hierarchical self-attention network assigns greater weights to u2 and u3, which contain words such as “sports,” “football,” “game,” and “courts.” This leads to a response that asks about favorite sports. Overall, the parts of the context that the hierarchical self-attention networks focuses on are consistent with human intuition.

5.6.3 Answer to Q2

We kept the main architecture of Anna and replaced the utterance-level encoder and context-level encoder with an RNN. These models as “w/o UL” and “w/o CL” respectively. Additionally, the hierarchical self-attention network was completely replaced with a hierarchical RNN. We denoted this model as “w/o HS.” We employed PPL as an automatic evaluation metric and conducted human judgment between these models and the full Anna model on both the Chinese and English testing datasets. Table 4 reports the results of model ablation. The encoder is important and useful for our model whether the utterance level or context level. Removing either of them causes a performance drop. The utterance-level encoder is more important for Anna and achieves greater win-loss ratios (8% on the English testing dataset and 16% on the Chinese testing dataset) for human judgment. When the full encoder is replaced, the approach degenerates into a two-layer RNN model. Because it does not pay attention to important components of the

| Model          | Douban       | DailyDialog  |
|----------------|--------------|--------------|
|                | Win  | Loss | Tie  | PPL   | Win  | Loss | Tie  | PPL   |
| Anna vs. w/o UL | 36   | 20   | 44   | 38.869 | 40   | 32   | 28   | 23.606|
| Anna vs. w/o CL | 34   | 25   | 41   | 36.804 | 39   | 33   | 28   | 22.183|
| Anna vs. w/o HS | 43   | 20   | 37   | 41.940 | 34   | 26   | 40   | 25.302|

Table 4 Model ablation results for encoders (%).
context, “w/o HS” is clearly inferior to the other models in terms of both automatic evaluations and human evaluations.

5.6.4 Answer to Q3

We also used model ablation to confirm the effects of the topic-augmented network in Anna. The components of topic loss and topic extra probability were removed from Anna. Specifically, we replaced \((O_C \oplus O_Z)\) in Eq.(15) with \(O_C\), we removed Eqs.(21) and (24) from Eqs.(22) and (25), and denoted the resulting model as “w/o TA.” We performed automatic evaluation and human judgment between “w/o TA” and the full Anna model on both the Chinese and English testing datasets. ReCoSa was also added to the judgment for longitudinal comparisons to the baselines. Table 5 reports the results of model ablation. One can see that removing the topic-augmented network causes a performance drop for Anna, but “w/o TA” is still ahead of ReCoSa in terms of human judgment. This indicates that the hierarchical self-attention network contributes more to the improvement of the entire model than the topic-augmented network. Therefore, the topic-augmented network is useful for enriching generated responses when contextual dependencies are properly modeled.

5.6.5 Comparison of Anna to Zhao et al.’s Method

Our model possesses the following advantages. (1) The effectiveness of Anna has been verified using two languages (English and Chinese) from different language families. Therefore, our model is more versatile than their model. (2) The performance improvements of Anna are more convincing than that of their model because the superiority of Anna has been verified through comparisons to PLATO-2, which is the state-of-the-art model. In contrast, the aforementioned study did not consider PLATO-2 in its experiments. Because it considered ReCoSa (state-of-the-art in 2019), we made an indirect comparison based on the results of their study and our results. We found that their model outperformed ReCoSa by 2.476 points on average for PPL, whereas Anna outperformed ReCoSa by 3.296 points on average for PPL. This proves that our

| Model               | Douban          | DailyDialog     |
|---------------------|-----------------|-----------------|
|                     | Win | Loss | Tie | Distinct-1/2 | Win | Loss | Tie | Distinct-1/2 |
| Anna vs. w/o TA     | 30  | 19   | 51  | 1.186/3.185  | 31  | 23   | 46  | 1.000/3.517  |
| w/o TA vs. ReCoSa   | 26  | 20   | 54  |             | 31  | 26   | 43  | 1.000/3.517  |

Table 5 Model ablation results for the topic-augmented network (%).
model surpasses their model in terms of learning ability. (3) Anna can be applied to a wider range of utterance lengths. The maximum utterance length of Anna is 50 words and the effectiveness of this length was verified through empirical studies. Their model limits the maximum length of an utterance to 25 words. We implemented their model\(^7\) and performed a comparative trial by setting the maximum utterance length of their study to the same as that used in Anna. Fig. 6 presents the results for the decoding times of both models on the English testing dataset when the maximum utterance length varies from 25 words to 60 words. The average decoding time of Anna is 60% of that of their model for a given utterance length. This reveals that the response speed of our model is faster than that of their model. As the utterance length increases, the gap tends to widen.

The parameter size of their model is 27.3 M on the English dataset and 64.3 M on the Chinese dataset in our implementation. Anna has more parameters than their model because it only reduces the parameter size marginally, which was not our main motivation. According to the three primary challenges mentioned in the introduction, we aimed to develop a balanced modeling design. Accordingly, Anna uses a reduced number of parameters while ensuring that contextual dependencies and topic information are effectively modeled. In summary, Anna surpasses the method from (Zhao et al. 2020) in terms of comprehensive performance according to the above analysis.

\(^7\) A transformer architecture, but we did not implement auxiliary tasks for a limited time.

Fig. 6 Decoding time comparisons.
6 Conclusions and Future Work

This paper presented a novel multi-turn dialogue generation model called Anna. Experimental results demonstrated that Anna achieves substantial improvements over state-of-the-art models. The effectiveness of our model on both English and Chinese dialogue was verified. Further analysis revealed that Anna focuses on the important parts of contexts, which is consistent with human intuition. All of the components of Anna are useful. Our model successfully addresses the three primary challenges mentioned in the introduction. It can be concluded that contextual dependencies combined with topic information are useful for improving the quality of multi-turn dialogue response generation.

In future work, the following directions will be considered. (1) Through the analysis of generation failure cases, we determined that the fluency of responses is affected by the excessive generation probability of the topic vocabulary in a few cases. Therefore, the tradeoff between topic information and the fluency of generated responses should be studied. (2) Anna utilizes the traditional LDA model for topic mining, but Twitter LDA (Zhao et al. 2011) should also be considered. Twitter LDA belongs to a family of probabilistic topic models and its strength lies in the topic detection of short texts that are more likely to focus on one topic. Therefore, Twitter LDA may be more suitable for topic mining in multi-turn dialogue. (3) A greedy search was employed to generate responses in this study, but the beam search (Tillmann and Ney 2003) could be incorporated into our method.

Acknowledgement

We would like to acknowledge the significant contributions of the annotators in their careful annotation of the evaluation results. We also thank Yu Zhao for his help in recruiting native English annotators.

References

Adiwardana, D., Luong, M.-T., So, D. R., Hall, J., Fiedel, N., Thoppilan, R., Yang, Z., Kulshreshtha, A., Nemade, G., Lu, Y., et al. (2020). “Towards a Human-like Open-Domain Chatbot.” arXiv preprint arXiv:2001.09977.
Bao, S., He, H., Wang, F., Wu, H., Wang, H., Wu, W., Guo, Z., Liu, Z., and Xu, X. (2020). “Plato-2: Towards Building an Open-Domain Chatbot via Curriculum Learning.” arXiv
Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). “Latent Dirichlet Allocation.” *Journal of Machine Learning Research*, 3 (Jan), pp. 993–1022.

Cui, Y., Zhang, W., Wang, Y., Zhu, Q., Li, L., Zhou, L., and Liu, T. (2018). “Context-sensitive Generation of Open-domain Conversational Responses.” In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 2437–2447.

Fleiss, J. L. (1971). “Measuring Nominal Scale Agreement among Many Raters.” *Psychological Bulletin*, 76 (5), p. 378.

Glorot, X. and Bengio, Y. (2010). “Understanding the Difficulty of Training Deep Feedforward Neural Networks.” In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, pp. 249–256.

Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J., et al. (2001). “Gradient Flow in Recurrent Nets: The Difficulty of Learning Long-term Dependencies.” *IEEE Press. A Field Guide to Dynamical Recurrent Neural Networks*.

Huang, C., Zaiane, O. R., Trabelsi, A., and Dziri, N. (2018). “Automatic Dialogue Generation with Expressed Emotions.” In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics, Volume 2 (Short Papers)*, pp. 49–54.

Li, J., Galley, M., Brockett, C., Gao, J., and Dolan, B. (2015). “A Diversity-promoting Objective Function for Neural Conversation Models.” In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 110–119.

Li, Q., Chen, H., Ren, Z., Chen, Z., Tu, Z., and Ma, J. (2019). “Empgan: Multi-resolution Interactive Empathetic Dialogue Generation.” *arXiv preprint arXiv:1911.08698*.

Li, Y., Su, H., Shen, X., Li, W., Cao, Z., and Niu, S. (2017). “DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset.” In *Proceedings of the 8th International Joint Conference on Natural Language Processing*, pp. 986–995.

Liu, C.-W., Lowe, R., Serban, I. V., Noseworthy, M., Charlin, L., and Pineau, J. (2016). “How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation.” In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2122–2132.

Madotto, A., Lin, Z., Wu, C.-S., and Fung, P. (2019). “Personalizing Dialogue Agents via Meta-Learning.” In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5454–5459.

Mazaré, P.-E., Humeau, S., Raison, M., and Bordes, A. (2018). “Training Millions of Personalized Dialogue Agents.” In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2775–2779.
Mou, L., Song, Y., Yan, R., Li, G., Zhang, L., and Jin, Z. (2016). “Sequence to Backward and Forward Sequences: A Content-Introducing Approach to Generative Short-Text Conversation.” In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 3349–3358.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). “Bleu: A Method for Automatic Evaluation of Machine Translation.” In Proceedings of the 40th annual meeting of the Association for Computational Linguistics, pp. 311–318.

Press, O. and Wolf, L. (2017). “Using the Output Embedding to Improve Language Models.” In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, pp. 157–163.

Rashkin, H., Smith, E. M., Li, M., and Boureau, Y.-L. (2018). “Towards Empathetic Open-domain Conversation Models: A New Benchmark and Dataset.” In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 5370–5381.

Roller, S., Dinan, E., Goyal, N., Ju, D., Williamson, M., Liu, Y., Xu, J., Ott, M., Smith, E. M., Boureau, Y.-L., et al. (2021). “Recipes for Building an Open-Domain Chatbot.” In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pp. 300–325.

Serban, I. V., Sordoni, A., Bengio, Y., Courville, A., and Pineau, J. (2016). “Building End-to-end Dialogue Systems Using Generative Hierarchical Neural Network Models.” In Proceedings of the 30th AAAI Conference on Artificial Intelligence, Vol. 30, No. 1, pp. 3776–3783.

Serban, I. V., Klinger, T., Tesauro, G., Talamadupula, K., Zhou, B., Bengio, Y., and Courville, A. (2017a). “Multiresolution Recurrent Neural Networks: An Application to Dialogue Response Generation.” In Proceedings of the 31st AAAI Conference on Artificial Intelligence, Vol. 31, No. 1, pp. 3288–3294.

Serban, I. V., Sordoni, A., Lowe, R., Charlin, L., Pineau, J., Courville, A., and Bengio, Y. (2017b). “A Hierarchical Latent Variable Encoder-decoder Model for Generating Dialogues.” In Proceedings of the 31st AAAI Conference on Artificial Intelligence, Vol. 31, No. 1, pp. 3295–3301.

Shawar, B. A. and Atwell, E. (2007). “Chatbots: Are They Really Useful?” In Ldv forum, Vol. 22, pp. 29–49.

Tillmann, C. and Ney, H. (2003). “Word Reordering and a Dynamic Programming Beam Search Algorithm for Statistical Machine Translation.” Computational linguistics, 29 (1), pp. 97–133.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). “Attention Is All You Need.” In Advances in Neural Information Processing Systems, pp. 5998–6008.
Wu, Y., Wu, W., Xing, C., Xu, C., Li, Z., and Zhou, M. (2019). “A Sequential Matching Framework for Multi-turn Response Selection in Retrieval-based Chatbots.” *Computational Linguistics*, **45** (1), pp. 163–197.

Xing, C., Wu, W., Wu, Y., Liu, J., Huang, Y., Zhou, M., and Ma, W.-Y. (2017). “Topic Aware Neural Response Generation.” *The Association for the Advancement of Artificial Intelligence*, pp. 3351–3357.

Xing, C., Wu, W., Wu, Y., Zhou, M., Huang, Y., and Ma, W.-Y. (2018). “Hierarchical Recurrent Attention Network for Response Generation.” In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pp. 5610–5617.

Young, S., Gašić, M., Thomson, B., and Williams, J. D. (2013). “Pomdp-based Statistical Spoken Dialog Systems: A Review.” *Proceedings of the IEEE*, **101** (5), pp. 1160–1179.

Zhang, H., Lan, Y., Pang, L., Chen, H., Ding, Z., and Yin, D. (2020). “Modeling Topical Relevance for Multi-turn Dialogue Generation.” *arXiv preprint arXiv:2009.12735*.

Zhang, H., Lan, Y., Pang, L., Guo, J., and Cheng, X. (2019). “Recosa: Detecting the Relevant Contexts with Self-attention for Multi-turn Dialogue Generation.” In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3721–3730.

Zhang, S., Dinan, E., Urbanek, J., Szlam, A., Kiela, D., and Weston, J. (2018). “Personalizing Dialogue Agents: I Have a Dog, Do You Have Pets Too?” In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 2204–2213.

Zhao, W. X., Jiang, J., Weng, J., He, J., Lim, E.-P., Yan, H., and Li, X. (2011). “Comparing Twitter and Traditional Media Using Topic Models.” In *European Conference on Information Retrieval*, pp. 338–349. Springer.

Zhao, Y., Xu, C., and Wu, W. (2020). “Learning a Simple and Effective Model for Multi-turn Response Generation with Auxiliary Tasks.” In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 3472–3483.

Zhou, L., Gao, J., Li, D., and Shum, H.-Y. (2020). “The Design and Implementation of Xiaoice, an Empathetic Social Chatbot.” *Computational Linguistics*, **46** (1), pp. 53–93.

**Itsugun Cho:** is a first-year doctoral student at Keio University. His current research interests include natural language processing.

**Hiroaki Saito:** is currently an associate professor at Keio University. His research interests include natural language processing and music comprehension.