IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 19, 2020, accepted June 10, 2020, date of publication June 22, 2020, date of current version July 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3004152

Retrieval-Polished Response Generation for Chatbot

LIANG ZHANG[1], **YAN YANG**[1,2], **JIE ZHOU**[1], **CHENGCAI CHEN**[3], **AND LIANG HE**[1,2]

[1]Department of Computer Science, East China Normal University, Shanghai 200062, China
[2]Shanghai Key Laboratory of Multidimensional Information Processing, East China Normal University, Shanghai 200241, China
[3]Xiaoi Research, Shanghai 201803, China

Corresponding author: Yan Yang (angel_yangyan@qq.com)

This work was supported in part by Xiaoi Research, in part by the Shanghai Key Laboratory of Multidimensional Information Processing, East China Normal University, under Grant 2020KEY001, and in part by the Fundamental Research Funds for the Central Universities.

ABSTRACT Chatbot communication, in which a robot communicates with a human being in natural language in an open domain, has achieved significant progress. However, it still suffers from problems such as a lack of diversity and contextual relevance. In this paper, we propose a retrieval-polished (RP) model for response generation that polishes a draft response based on a retrieved prototype. In particular, we first adopt a prototype selector to retrieve a contextually similar prototype. Then, a generation-based polisher is designed to obtain a polished response. Finally, we introduce a polished response filter to choose whether the final reply should be the retrieved response or the polished response. Extensive experiments on a dialog corpus show that our method outperforms retrieval-based and generation-based chatbots with respect to fluency, contextual relevance, and response diversity. Specifically, our model achieves substantial improvement compared with several strong baselines.

INDEX TERMS Response generation, chatbot, dialogue system, neural network.

I. INTRODUCTION

Chatbot communication, an essential task in both natural language processing and artificial intelligence fields, involves a robot communicating with human beings using natural language in open domains. Chatbots play a critical role in many real-world applications, such as smart speakers, customer service systems [1], and social robots. Research on chatbots began in the 1960s [2]; initially, researchers used sets of handwritten rules and templates. However, such rule-based models require significant human effort and lack flexibility. In recent years, as large-scale dialog corpora and high-speed computational resources have become available, the early rule-based models have been rapidly replaced by data-driven models. The existing data-driven models can be categorized as retrieval-based or generation-based models. Retrieval-based models [3], [4] return responses from a corpora by computing contextual similarity. The retrieved responses are usually fluent and informative because they are written by humans; however, they may contain irrelevant content. In contrast, generation-based models [5] return responses generated based on language rules learned during the training process. Thus, the generated responses are relevant but may suffer from problems such as being "safe responses" [5], lacking fluency or including grammatical errors. An excellent conversational robot should return replies that are contextually relevant, informative, and fluent. Therefore, researchers have proposed using retrieved results as a basis for response generation [6], [7] and this approach has made progress. However, the combined methods do not actually take advantage of both methods, and their results tend to be similar to retrieved responses.

To address this issue, we introduce a polishing process into the generation model that was inspired by writing articles. Human authors typically write an early draft and then polish it in detail. When polishing a specific sentence, authors tend to adopt writing styles and techniques from existing literature. In terms of response generation, the context, that consists of the sentences that immediately precede the response, provides background knowledge to generate a draft response, while retrieved responses provide information about the language style and techniques that can be used to polish the draft response.

Inspired by this idea, we propose a retrieval-polished (RP) response generation model for chatbots. The main idea

The associate editor coordinating the review of this manuscript and approving it for publication was Imran Sarwar Bajwa.

behind this model is to use the retrieval response to polish the generated response to enhance its information and fluency. RP consists of a prototype selector (PS), a generation-based polisher (GP), and a polished response filter (PRF). Specifically, we first design a PS to retrieve a contextually similar prototype. Then, we propose a GP to obtain a polished response. The PRF is adopted to select the final reply: either the polished response or the prototype according to a context-sensitive score.

To integrate contexts and prototypes, we introduce a variation of the encoder-decoder architecture to build the GP. First, a context encoder is used to convert the context sentences into a fixed vector representation. Second, a prototype encoder transforms the retrieved response into a fixed vector that integrates the context vector representation through a context-attention mechanism. The main function of this integration operation is to filter out context free content from the prototype. Finally, a decoder predicts a sequence of symbols by successively applying the outputs of the context encoder and the prototype encoder. The innovation of this model is that the decoder of GP generates responses in two stages: first generating based on the context and then polishing based on the prototype. While other methods don't have the polishing process. Specifically, the decoder generates a draft response based on the context through a context-attention. Usually, the context can provide background information; therefore, the draft response can include context-related content such as topic words. Then, the decoder polishes the draft response using the phrasing provided by the prototype as an example through a prototype-attention. The motivation is that the prototypes retrieved directly from the real corpus are informative, diverse and grammatical [7]; therefore, they can be helpful in improving the fluency and informativeness of the generated response. In this paper, we use the transformer [8] architecture to build the encoder and decoder.

In our experiments, we use a large-scale Chinese dialog corpus to verify the effectiveness of the proposed model. A series of experimental results shows that our method outperforms both retrieval-based and generation-based models in terms of relevance and establishes a new state-of-the-art relevance score.

The main contributions of our approach can be summarized as follows:

- We introduce the generate-then-polish process into the response generation to improve the informativeness and fluency of the generated response.
- We propose a GP that contains two encoders, one to represent context and one to represent the prototype, and one decoder, which performs response generation and polishing.
- A series of experimental results on a Chinese dialog corpus demonstrates the substantial advantage of our proposed model. In particular, our model achieves state-of-the-art results on the Douban dialog dataset.

II. RELATED WORKS

A large number of studies on chatbots exist that can be divided into three groups. We focus on the works that are most closely related to our research.

A. RETRIEVAL-BASED METHODS

One simple implementation of a retrieval-based model involves computing the cosine similarity between the input utterances and the candidate replies [9]. To improve the relevance of the responses, researchers have proposed a series of matching algorithms that consider contextual information. SMN [3] matches a response with each utterance in the context of multiple levels of granularity. DAM [4] not only constructs text segment representations at different granularities solely using stacked self-attention but also extracts well-matched segment pairs with attention across the context and response.

B. GENERATION-BASED METHODS

Most generation-based methods are based on a variant of the sequence-to-sequence (seq2seq) framework [10], [11]. Studies on improving informativeness can be divided into two categories. The first category directly optimizes the seq2seq model: Li *et al.* [5] proposed the use of maximum mutual information (MMI) as the objective function, and Zhang *et al.* [12] optimized a variational lower bound on the pairwise mutual information between context and response. The second category introduces external elements such as topics [13], [14], keywords [15], and external knowledge [16], [17] to seq2seq models to increase the informativeness. With respect to improving the contextual relevance, [18]–[20] applied a hierarchical neural network to model the context. Mei *et al.* [21] proposed a dynamic attention model to combine each generated word with its most related words in the conversation history. Similarly, ReCoSa [22] was proposed to detect the relevant context via a self-attention mechanism. Reinforcement learning [23] and adversarial learning [24] have also been applied in generation-based models.

C. COMBINED METHODS

Recently, some researchers have combined the above methods, for example, 1) ranking two types of responses and returning the top-1 result or 2) feeding retrieved responses to a generation-based model to enhance the informativeness and diversity of the generated response. Qiu *et al.* [25] proposed that when the top retrieved response achieved a score above a certain threshold, it should be taken as the final response; otherwise, the response should be obtained from a generation-based model. Similarly, Song *et al.* [6] reranked the two kinds of responses but first concatenated the retrieved responses into the context to generate a response. Wu *et al.* [7] designed a response-editing model that modified a prototype using guidance from an edit vector. Zhu *et al.* [26] and Zhang *et al.* [27] cast response generation as

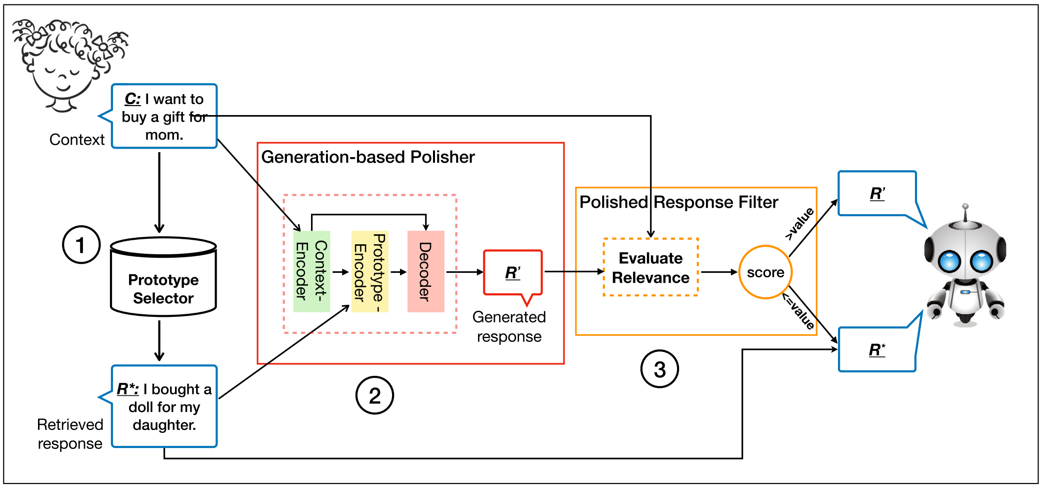


FIGURE 1. Architecture of the retrieval-polished chatbot. A prototype selector retrieves a prototype *R; then, a generate-then-polish model outputs a polished response *R*′, and a filter determines the final reply.**

a reinforcement learning process; however, Zhu *et al.* [26] used the N-best retrieved responses as evidence to compute the reward for generator, while Zhang *et al.* [27] applied the retrieved responses as additional information to both discriminator and generator. The authors of [28], [29] employed a pipeline approach for skeleton extraction and response generation; however, the response generator in [28] relied on the output of the learned skeleton extractor, while the response generator in [29] was trained with target-specific skeletons.

Most of these combined methods failed to balance the relationship between context and prototypes, and their results tended to be similar to prototypes. Therefore, we propose a RP model which take advantage of both context and prototypes.

III. APPROACH

A. MODEL OVERVIEW

In this work, we have a dataset $\mathcal{D} = \{\sum_{i=1}^{n}(C_i, R_i)\}$. (C_i, R_i) that forms a complete dialogue, where R_i is the ground-truth response and C_i is the context, which consists of the sentences that immediately precede R_i. Notably, context C can be either a single- or a multi-turn input. As an initial attempt at the generate-then-polish idea, we address only single-turn inputs in this work. Our goal is to obtain a contextually relevant, fluent and informative polished response R_i' based on (C_i, R_i^*) and return the best response among R_i^* and R_i' to the user, where R_i^* represents a prototype whose context C_i^* is similar to C_i. Fig.1 depicts an overview of our method, which consists of a prototype selector (PS), a generation-based polisher (GP), and a polished response filter (PRF). Given a context C_i:

1) First, we use the PS to retrieve a prototype based on its similarity to the context.
2) Second, the dual encoders in the GP separately encode the context and prototype into vectors.
3) Third, the decoder in the GP generates a draft response from the context vector and then polishes the draft response with the help of the prototype vector.
4) Finally, we introduce the PRF, which calculates a score for the polished response. If the score exceeds the threshold level, the robot returns the polished response as the final reply; otherwise, it returns the retrieved response.

The following subsections discuss these three components.

B. PROTOTYPE SELECTOR (PS)

A useful prototype selector (PS) is essential because it supports the whole model. We use the strategy reported in [7]. During training, because we know the ground truth R, we retrieve the top-N prototypes based on response similarity rather than context similarity. We adopt this strategy because similar contexts may have entirely different responses that make polishing training more difficult due to lexical gaps [7]. Next, we extend $\mathcal{D}$ to a larger scale $\mathcal{D}' = \{\sum_{i=1}^{n}(C_i, R_i, C_i^*, R_i^*)\}$. Note that during testing, we retrieve a prototype based on context similarity. Please refer to [7] for more details.

C. GENERATION-BASED POLISHER (GP)

A GP, as shown in Fig. 2, is a variant of the encoder-decoder architecture that consists of three components: context- and prototype encoders to compute representations of the context and prototype, respectively, and a decoder to compute the generative probability distribution of the response words. As the context often includes several items, it is essential to capture dependency over long distances. Therefore, we extend the Transformer [8], which has a stronger ability to consider long-distance dependencies than do RNNs [30], [31], to compute the sentence representation. In the following, we introduce our model in detail.

1) CONTEXT ENCODER

As shown in Fig.2 (a), we use a self-attentive encoder to compute the context representation. The input to the encoder is a matrix C_{in} that contains all the context word vector representations:

$$C_{in} = [c_1; \ldots; c_m] \tag{1}$$

where c_i denotes the sum of the i-th word embedding and its positional embedding.

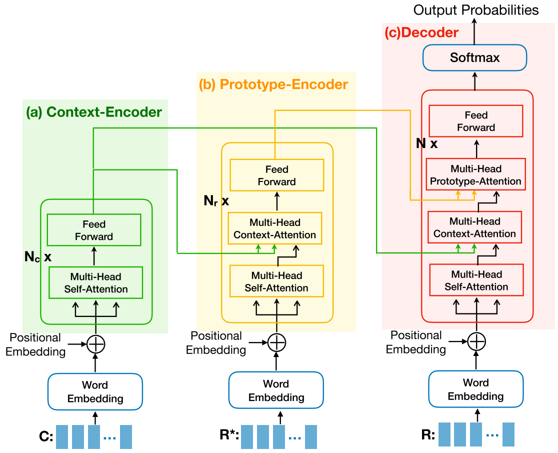


FIGURE 2. **The generate-then-polish model.**

The context encoder is composed of a stack of N_c identical layers, each of which has a self-attention sublayer and a position-wise fully connected feedforward sublayer. The first sublayer of the n-th identical layer is a multi-head self-attention mechanism:

$$C^n_{self-att} = MultiHead(A_1, \ldots, A_H) \tag{2}$$

where H is the number of parallel heads and A_i is the i-th self-attention based on the following equation:

$$A_i = Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d}})V \tag{3}$$

where d is the number of hidden units, Q is a query matrix, K is a key matrix, and V is a value matrix. In the first sublayer, $Q = K = V$, which is why it is called self-attention. In the current case, $Q = K = V = C^{n-1}$, where C^{n-1} is the output of the former layer computed by Equation 4. In the first layer, $C^0 = C_{in}$.

The second sublayer is a simple, position-wise fully connected feedforward network:

$$C^n = FFN(C^n_{self-att}) \tag{4}$$

where C^n is the representation of the context sentence, and $n = 1, \ldots, N_c$.

2) PROTOTYPE ENCODER

Directly employing the prototype to polish the draft response may introduce irrelevant information; therefore, we perform information filtering on the prototype according to the context. This step mainly relies on the addition of a sublayer called context attention.

As shown in Fig.2 (b), we use an attentive encoder to compute the representation of the prototype. The input of the encoder is a matrix $R_{in} = [r^*_1; \ldots; r^*_N]$ calculated by the same function shown in Equation 1.

The prototype encoder is also a stack of N_r identical layers in which each layer contains three components. In addition to the two sublayers mentioned in the context encoder, the prototype encoder has a context attention sublayer. Context attention allows every position in the prototype to attend to the overall positions in the context.

The self-attention layer is computed as $R^{*n}_{self-att}$ in Equation 2, and $Q = K = V = R^{*n-1}$ is the output of the previous layer computed by Equation 6. Similarly, $R^{*0} = R_{in}$ in the first layer.

The second sublayer is a multi-head context-attention mechanism denoted as R^{*n}_{c-att}. It is also computed by Equation 2, but it takes the context representation C^{N_c} as the key and value and the self-attention representation $R^{*n}_{self-att}$ as the query:

$$R^{*n}_{c-att} = MultiHead(Attention(R^{*n}_{self-att}, C^{N_c}, C^{N_c})). \tag{5}$$

The third sublayer is computed as follows:

$$R^{*n} = FFN(R^n_{c-att}), \tag{6}$$

where R^{*n} is the representation of the prototype sentence and $n = 1, \ldots, N_r$.

3) DECODER

As shown in Fig. 2, the decoder considers that the generate-then-polish process relies primarily on two rounds of information integration. The decoder first integrates the output of the context encoder through the context attention and then integrates the output of the prototype encoder through the prototype attention.

The decoder predicts the sequence word by word; it cannot operate in parallel like the encoder. Therefore, when predicting the i-th word, we have only the word sequence $\{r_1, \ldots, r_{i-1}\}$ as input. We follow [8] to offset the response word embeddings by one position and denote the matrix representation of $\{r_1, \ldots, r_{i-1}\}$ as follows:

$$R^0 = [r_0, r_1, \ldots, r_{i-1}], \tag{7}$$

where r_0 is the vector representation of the start-of-sentence token.

As shown in Fig.2 (c), we use a stack of N identical layers to compute the decoder-side representations, and each layer has four sublayers. In addition to the two sublayers in the context encoder, we insert context attention and prototype attention into each layer of the decoder. This fourfold structure allows every masked position in the decoder to focus on all the positions in C and R^*.

The first sublayer is a self-attention mechanism, computed as $R^{n}_{self-att}$ in Equation 2, where $Q = K = V = R^{n-1}$, which is the output of the previous layer calculated by Equation 10.

The second sublayer is a generating process that integrates contextual information C^{N_c} into the decoder-side self-attention representation via the context attention:

$$R^{n}_{c-att} = MultiHead(Attention(R^{n}_{self-att}, C^{N_c}, C^{N_c})). \quad (8)$$

The third sublayer is a polishing process that integrates prototype information R^{*N_r} into the representation of the former decoding results via the prototype attention:

$$R^{n}_{r-att} = MultiHead(Attention(R^{n}_{c-att}, R^{*N_r}, R^{*N_r})). \quad (9)$$

The fourth sublayer is a simple, position-wise fully connected feedforward network:

$$R^{n} = FFN(R^{n}_{r-att}). \quad (10)$$

Finally, a softmax layer over the vocabulary is used in the last layer to compute the generative probability distribution of the word r_i:

$$P(r_i|C, R^{*}, r_1, \ldots, r_{i-1}; \theta) = softmax(W_{\theta} R^{N}_{i}), \quad (11)$$

where θ and W_{θ} are model parameters, and R^{N}_{i} is a hidden state for the i-th generated word calculated by the last decoder layer. These probabilities are then scored using a negative log-likelihood loss, and we train the model by minimizing the loss:

$$\mathcal{L} = -\sum_{i=1}^{L} logP(r_i|C, R^{*}, r_{1:i-1}). \quad (12)$$

D. POLISHED RESPONSE FILTER (PRF)

Normally, the polished response is suitable, but sometimes, it may contain unsuitable content such as ‘ha ha ha’ may be returned. To ensure smooth human-robot dialog, we define a filter to determine whether the polished response should be the final reply. The filter scores the polished response R^{+} based on (C, R^{+}). When the resulting score is exceeds a threshold, R' is the final reply; otherwise, the retrieved response is selected as the final reply. In general, context-sensitive sentences are separated by a small distance in semantic space. Therefore, the filtering function is defined as follows:

$$R'_{score} = -Distance(C_{emb}, R'_{emb}), \quad (13)$$

where C_{emb} and R'_{emb} are the sentence embeddings of the context and polished response, respectively. In this work, the sentence embeddings are derived from BERT [32]. We use the Euclidean distance to calculate the distance and take the Euclidean distance result to the opposite number. In this way, the semantic distance between two sentences with similar semantics is small, but their score is high. Notably, this step is applied only during the testing process.

IV. EVALUATION

A. EXPERIMENTAL SETUP

1) DATASET

The Douban dialog corpus is a Chinese single-turn dialog corpus crawled from a popular forum in China named Douban [7]. The dataset is divided into three parts, consisting of 10,000,000 pairs for training, 150,000 pairs for validation and 10,000 pairs for testing. We utilize Chinese characters as input and limit the size of the character set to 5,000.

2) PARAMETER SETTINGS

Our GP has a hidden layer size of 512, and the hidden dimension of the feedforward layers is 1,024. We employ 8 parallel attention heads for the multi-head attention layers and set the number of identical layers to $N_c = N_r = N = 6$. We limit the length of sentences to no more than 150 characters and set the dimension of the character embedding size to 512. For training, we use Adam for optimization and set the dropout probability to 0.1 for all layers. For testing, the beam size is 4, and the default value of the polished response filter is -13.17, which is the 3rd quartile distance calculated from 3,000 randomly selected (C, R^{+}) dialog pairs.

B. BASELINES

To comprehensively evaluate the performance of our model, we compare our models with the following baselines:

- **S2S-A**: A seq2seq model with an attention mechanism [33].
- **Edit-N-Rerank**: A response editing model proposed by [7], where the encoder takes N prototype responses as input and the decoder revises the prototype response with the help of an edit vector. The edit vector represents the differences between the prototype context and current context.
- **Transformer**: The original Transformer architecture introduced by [8].
- **Edit-Transformer**: A response-editing model based on the Transformer architecture that has the same context encoder and prototype encoder describe in Section III. The decoder has three sublayers: self-attention, prototype attention, and a fully connected feedforward network.
- **Retrieval**: We directly return the top-1 result provided by the Information Retrieval process, which ranks candidates based on context similarity.

Additionally, we design three variants of our model, named **RP w/o PRF**, **RP w/o GP&PRF**, and **RP**. RP is the complete model described in Section III. To verify the validity of the PRF, we design an RP w/o PRF model that responds with the polished response directly—without filtering. To investigate whether that the prototype encoder captures the connection between the context and prototype, we construct the RP w/o GP&PRF model, which regards the context and prototype as two separate entities. In this model, both the context encoder

TABLE 1. Results of the automatic evaluation. The underlined values represent the best results among the baselines.

	Relevance			Diversity		Fluency
	Average	Extrema	Greedy	Distinct-1	Distinct-2	PPL
Retrieval	0.288	0.130	0.309	*0.098*	*0.549*	–
S2SA	0.346	0.180	0.350	0.032	0.087	98.62
Transformer	0.352	0.182	0.361	0.037	0.124	96.80
Edit-N-Rerank	*0.386*	*0.203*	0.389	0.068	0.280	91.27
Edit-Transformer	0.384	0.203	*0.390*	0.070	0.277	92.00
RP w/o GP&PRF	0.373	0.199	0.375	0.063	0.278	94.61
RP w/o PRF	0.390	0.204	0.393	0.069	0.281	87.05
RP	**0.400(3.63%)**	**0.208(2.46%)**	**0.397(1.79%)**	**0.083(18.57%)**	**0.421(51.99%)**	**87.03(5.38%)**

and prototype encoder have two sublayers: self-attention and a fully connected feedforward sublayer.

We employed all the baseline models using PyTorch. All the hyperparameters for the Transformer and Edit-Transformer models are set to the same values as those of our model. For S2S-A and Edit-N-Rerank, we set the vocabulary size to 30,000 and the word embedding dimension to 512. The encoder and decoder are a 1-layer GRU with a hidden size of 1,024.

C. EVALUATION METRICS

To achieve automatic evaluation, we evaluated the methods with respect to three criteria: relevance, fluency, and diversity. To evaluate the response relevance, we use the following word embedding metrics: embedding greedy (Greedy), embedding average (Average), and embedding extrema (Extrema) [34], which correlate better with human judgments than do word overlap metrics. We employ perplexity (PPL), which is defined as the exponent of the average negative log-likelihood per word [35], to evaluate the response fluency. Finally, we evaluate the diversity of the response through the metrics Distinct-1 (Dist-1) and Distinct-2 (Dist-2) [5], which represent the ratios of distinct unigrams and bigrams in the responses, respectively. The larger the scores for the metrics relevance and diversity are better. In contrast, for fluency, smaller scores are better.

To perform a human evaluation, we evaluated the methods from two aspects: the fluency and informativeness of the response sentence and the contextual relevance of the context-response pairs. We recruited 12 volunteers from different scholastic majors from our school, all of whom are native Chinese speakers. For the response evaluation, seven volunteers scored 500 randomly selected responses from each model. We adopted three scores, 0, 1, and 2, where 0 means the sentence is difficult to understand; 1 means that either the sentence has some grammatical errors but can be understood by a human or that the sentence is fluent but meaningless; and 2 means the sentence is fluent and informative. For the context-response pairs evaluation, five other volunteers were given 100 randomly sampled contexts and asked to choose the better response between RP and each baseline. We adopt three possible results, namely, Win, Tie, and Loss, where Win means the response is selected; Loss means the response is eliminated; and Tie means the two responses have similar qualities, and it is difficult to make a choice.

D. EXPERIMENTAL RESULTS

1) AUTOMATIC EVALUATION

The results of the automatic evaluation are shown in Table 1. Our three models (RP w/o PRF, RP w/o GP&PRF, and RP) achieve excellent performances with respect to the three criteria. As shown in Table 1, retrieval obtains the worst embedding-based metrics, which means that the retrieved replies are less contextually relevant. RP w/o GP&PRF outperforms S2SA and Transformer in terms of relevance, but it performs worse than do Edit-N-Rerank and Edit-Transformer. The reason is that while the prototype introduces more features, it may also introduce noise due to a lack of content selection. Although the edit models (Edit-N-Rerank and Edit-Transformer) select relevant information, they are inferior to the polish models in terms of relevance because they predict responses based on only prototypes. RP w/o PRF and RP build a connection between context and prototype and then perform decoding based on both context and prototype. The results show that RP w/o PRF and RP improve the contextual relevance. In terms of diversity, the retrieval-based method is much better than the others because the replies are selected from a real dataset. The edit models and our models improve diversity by introducing a prototype. RP adds a filter to the model; therefore, some replies are taken directly from the corpus; thus, RP significantly outperforms the generation-based models in terms of diversity. The PPL scores achieved on the validation set are shown in Table 1. RP obtains the best PPL, which means that the polished replies are most fluent. RP adds a filter to RP w/o PRF; consequently they achieve the same PPL values.

2) HUMAN EVALUATION

Because the responses from Retrieval are taken directly from the corpus, we exclude Retrieval when evaluating the response sentences. The results of the human sentence evaluations are shown in Table 2. We list the number of sentences related to each score and average the scores of these 500 sentences (Avg.Score). Specifically, Avg.Score $= \frac{1}{500}\sum_{s=0}^{2} s \times n_s$, where s is the score and n_s is the number of sentences

TABLE 2. Results of the human evaluation of sentences. Avg. Score denotes the average score of the 500 sentences. K denotes Fleiss' Kappa, which reflects the level of agreement among human annotators.

	0	1	2	Avg. Score	K
S2SA	63	172	265	1.404	0.87
Transformer	47	126	327	1.560	0.88
Edit-N-Rerank	39	108	353	1.628	0.83
Edit-Transformer	30	142	328	1.596	0.80
RP w/o GP&PRF	44	113	343	1.598	0.87
RP w/o PRF	32	91	377	1.690	0.86
RP	21	89	390	**1.738**	0.81

TABLE 3. Results of the automatic evaluation of context-response pairs. When a row is labeled as "a v.s.b", the second column, "Win", means the ratio of responses given by "a" are better than those given by "b".

	Win	Tie	Lose	K
RP vs. Retrieval	56.3%	26.7%	17.0%	0.47
RP vs. S2SA	47.8%	34.1%	18.1%	0.42
RP vs. Transformer	43.6%	29.4%	27.0%	0.35
RP vs. Edit-N-Rerank	37.5%	33.6%	28.9%	0.31
RP vs. Edit-Transformer	42.4%	30.2%	27.4%	0.37

whose score is s. RP w/o PRF outperforms the baselines, which means that the GP improves the quality of generated sentences via polishing. RP further improves the quality of sentences by removing poor sentences. Table 3 shows the human evaluation results for the context-response pairs. The ratio of wins demonstrates that our method produces more appropriate responses than do the baselines. Moreover, Fleiss' Kappa values for the sentence evaluations of all the models are all above 0.8, but Fleiss' Kappa for the context-response pairs evaluation is approximately 0.4. This phenomenon indicates that humans achieve high agreement at the sentence level, but at the dialog level, a satisfactory level of agreement is difficult to achieve because more dimensions must be considered.

E. DISCUSSION

1) CASE STUDY

To better understand why our model performs well, some typical cases are shown in Table 4. The Retrieval model tends to return long sentences, increasing the risk of noise, such as the "Tianjin" and"Concert" in Example 1. This problem also causes the responses of Edit-N-Rerank, which edits the retrieved response directly, to contain some irrelevant contents. In contrast, most of the responses of S2SA and Transformer are relevant but are terse and overly safe, such as the first example "Yes."

In comparison, RP makes full use of the context and prototype information by means of the generate-then-polish process. As shown in Table 4, the average length of the RP results is second only to that of the retrieval model. Combined with the highest Distinct-1 and Distinct-2 values, we can be confident that RP returns response with more informativeness and greater diversity than the other generation-based models. RP introduces relevant information according to the context, such as the word "buy" in Example 3. In addition, RP removes irrelevant information from the prototype, such as "Tianjin" and "Concert" in Example 1 and "try" in Example 3, but it retains the useful information in the prototype, such as "flag raising" in Example 1. The above measures help ensure the contextual relevance of the polished responses.

TABLE 4. Examples of responses from the baselines and the proposed method.

Context	Are you in Beijing on the 27th?
Retrieval	I will go to Tianjin on the 29th to watch a concert and then go to Beijing in the early morning to watch the flag raising.
S2SA	Yes.
Transformer	Yes.
Edit-N-Rerank	I don't know. Are you going to watch the concert?
Edit-Transformer	I go to Tianjin.
RP	**I am in Beijing and I will go to see the flag raising.**
Context	Should I send an email to Uncle Chang in Chinese or English?
Retrieval	You can use Chinese or English.
S2SA	Chinese
Transformer	English
Edit-N-Rerank	In Chinese or English.
Edit-Transformer	Use Chinese or English.
RP	**You can use both Chinese and English.**
Context	We spend 50 RMB a month on dog food.
Retrieval	I heard that homemade dog food is not nutritionally balanced, although I also want to try.
S2SA	It is cheap.
Transformer	It is so good.
Edit-N-Rerank	I also want to try.
Edit-Transformer	I also want to try.
RP	**I also want to buy dog food.**

According to our observations, the context attention in the prototype encoder has a positive effect on removing unnecessary prototype words. Moreover, the prototype attention in the decoder can make use of some additional elements, such as function words, to polish a draft sentence and make it more fluent.

2) COMPUTATIONAL COMPLEXITY

The total computational complexity per attention layer of GP is $O(n^2 \cdot d)$, where n is the sequence length and d is the representation dimensionality. In terms of inference speed, RP can process 21 cases per second, which can be used in a real-time chatbot interaction.

V. CONCLUSIONS AND FUTURE WORK

In this work, we propose a retrieval-polished response generation method for human-robot dialog. In contrast to the existing approaches, our method polishes a draft response by considering a contextually similar prototype and finally chooses the better of the retrieved and polished responses as the final reply. Our method uses both the background provided by the context and the sentence style provided by the retrieved response. Experimental results on a large-scale dialog corpus show that our method outperforms the generation-based methods in terms of fluency, relevance, and diversity.

In future work we plan to investigate how to further remove irrelevant information from the prototype. Additionally, we will investigate how to introduce emotion into the generator so that the human-robot dialog can provide a better companion.

REFERENCES

[1] Z. Yan, N. Duan, P. Chen, M. Zhou, J. Zhou, and Z. Li, ''Building task-oriented dialogue systems for online shopping,'' in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 4618–4626.

[2] J. Weizenbaum, ''ELIZA—A computer program for the study of natural language communication between man and machine,'' *Commun. ACM*, vol. 26, no. 1, pp. 23–28, Jan. 1983.

[3] Y. Wu, W. Wu, C. Xing, M. Zhou, and Z. Li, ''Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots,'' in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 496–505.

[4] X. Zhou, L. Li, D. Dong, Y. Liu, Y. Chen, W. X. Zhao, D. Yu, and H. Wu, ''Multi-turn response selection for chatbots with deep attention matching network,'' in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 1118–1127.

[5] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan, ''A diversity-promoting objective function for neural conversation models,'' in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2016, pp. 110–119.

[6] Y. Song, C.-T. Li, J.-Y. Nie, M. Zhang, D. Zhao, and R. Yan, ''An ensemble of retrieval-based and generation-based human-computer conversation systems,'' in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 4382–4388.

[7] Y. Wu, F. Wei, S. Huang, Z. Li, and M. Zhou, ''Response generation by context-aware prototype editing,'' in *Proc. 33rd AAAI Conf. Artif. Intell.*, 2019, pp. 7281–7288.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, N. A. Gomez, L. Kaiser, and I. Polosukhin, ''Attention is all you need,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[9] X. Li, L. Mou, R. Yan, and M. Zhang, ''Stalematebreaker: A proactive content-introducing approach to automatic human-computer conversation,'' in *Proc. 25th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2016, pp. 2845–2851.

[10] I. Sutskever, O. Vinyals, and V. Quoc Le, ''Sequence to sequence learning with neural networks,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[11] O. Vinyals and Q. V. Le, ''A neural conversational model,'' in *Proc. 31st Int. Conf. Mach. Learn.*, Lille, France, vol. 37, 2015.

[12] Y. Zhang, M. Galley, J. Gao, Z. Gan, X. Li, C. Brockett, and B. Dolan, ''Generating informative and diverse conversational responses via adversarial information maximization,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1815–1825.

[13] C. Xing, W. Wu, Y. Wu, J. Liu, Y. Huang, M. Zhou, and W.-Y. Ma, ''Topic aware neural response generation,'' in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 3351–3357.

[14] X. Li, P. Li, W. Bi, X. Liu and W. Lam, ''Relevance-promoting language model for short-text conversation,'' in *Proc. 24th AAAI Conf. Artif. Intell.*, 2020, pp. 8253–8260.

[15] L. Mou, Y. Song, R. Yan, G. Li, L. Zhang, and Z. Jin, ''Sequence to backward and forward sequences: A content-introducing approach to generative short-text conversation,'' in *Proc. 26th Int. Conf. Comput. Linguistics*, 2016, pp. 3349–3358.

[16] M. Ghazvininejad, C. Brockett, M.-W. Chang, B. Dolan, J. Gao, W.-T. Yih, and M. Galley, ''A knowledge-grounded neural conversation model,'' in *Proc. 32nd AAAI Conf. Artif. Intell. (AAAI)*, 2018, pp. 5110–5117.

[17] T. Young, E. Cambria, I. Chaturvedi, H. Zhou, S. Biswas, and M. Huang, ''Augmenting end-to-end dialogue systems with commonsense knowledge,'' in *Proc. 32nd AAAI Conf. Artif. Intell. (AAAI)*, 2018, pp. 4970–4977.

[18] I. V. Serban, A. Sordoni, Y. Bengio, C. Aaron Courville, and J. Pineau, ''Building end-to-end dialogue systems using generative hierarchical neural network models,'' in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 3776–3784.

[19] I. V. Serban, A. Sordoni, R. Lowe, L. Charlin, J. Pineau, C. Aaron Courville, and Y. Bengio, ''A hierarchical latent variable encoder-decoder model for generating dialogues,'' in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 3295–3301.

[20] C. Xing, Y. Wu, W. Wu, Y. Huang, and M. Zhou, ''Hierarchical recurrent attention network for response generation,'' in *Proc. 32nd AAAI Conf. Artif. Intell. (AAAI)*, 2018, pp. 5610–5617.

[21] H. Mei, M. Bansal, and R. Matthew Walter, ''Coherent dialogue with attention-based language models,'' in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 3252–3258.

[22] H. Zhang, Y. Lan, L. Pang, J. Guo, and X. Cheng, ''ReCoSa: Detecting the relevant contexts with self-attention for multi-turn dialogue generation,'' in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 3721–3730

[23] J. Li, W. Monroe, A. Ritter, D. Jurafsky, M. Galley, and J. Gao, ''Deep reinforcement learning for dialogue generation,'' in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 1192–1202

[24] J. Li, W. Monroe, T. Shi, S. Jean, A. Ritter, and D. Jurafsky, ''Adversarial learning for neural dialogue generation,'' in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 2157–2169.

[25] M. Qiu, F.-L. Li, S. Wang, X. Gao, Y. Chen, W. Zhao, H. Chen, J. Huang, and W. Chu, ''AliMe chat: A sequence to sequence and rerank based chatbot engine,'' in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 498–503.

[26] Q. Zhu, L. Cui, W.-N. Zhang, F. Wei, and T. Liu, ''Retrieval-enhanced adversarial training for neural response generation,'' in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 3763–3773.

[27] J. Zhang, C. Tao, Z. Xu, Q. Xie, W. Chen, and R. Yan, ''EnsembleGAN: Adversarial learning for retrieval-generation ensemble model on short-text conversation,'' in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2019, pp. 435–444.

[28] D. Cai, Y. Wang, W. Bi, Z. Tu, X. Liu, W. Lam, and S. Shi, ''Skeleton-to-response: Dialogue generation guided by retrieval memory,'' in *Proc. Conf. North*, 2019, pp. 1219–1228.

[29] D. Cai, Y. Wang, W. Bi, Z. Tu, X. Liu, and S. Shi, ''Retrieval-guided dialogue response generation via a matching-to-generation framework,'' in *Proc. IJCNLP*, 2019, pp. 1866–1875.

[30] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei and I. Sutskever, ''Language models are unsupervised multitask learner,'' 2019.

[31] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, and R. Salakhutdinov, ''Transformer-XL: Attentive language models beyond a fixed-length context,'' in *Proc. 57th Annu. Meeting Assoc. for Comput. Linguistics*, 2019, pp. 2978–2988.

[32] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, ''BERT: Pre-training of deep bidirectional transformers for language understanding,'' in *Proc. NAACL-HLT*, 2019, pp. 4171–4186.

[33] D. Bahdanau, K. Cho, and Y. Bengio, ''Neural machine translation by jointly learning to align and translate,'' in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–10.

[34] C.-W. Liu, R. Lowe, I. Serban, M. Noseworthy, L. Charlin, and J. Pineau, ''How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation,'' in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 2122–2132.

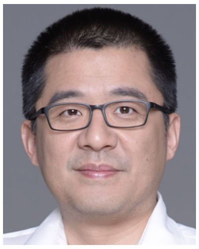
[35] G. Neubig, ''Neural machine translation and sequence-to-sequence models: A tutorial,'' 2017, *arXiv:1703.01619*. [Online]. Available: https://arxiv.org/abs/1703.01619

LIANG ZHANG was born in Baicheng, China, in 1995. She received the B.E. degree from the Department of Computer Science, East China Normal University, in 2017, where she is currently pursuing the master's degree. Her research interest includes natural language processing.

YAN YANG received the M.Sc. and Ph.D. degrees from East China Normal University, China. She is currently an Assistant Professor with the Department of Computer Science and Technology, East China Normal University. She has obtained five patents and published more than ten research papers in international conferences and journals. Her research interests include information extraction, dialogue systems, and knowledge processing. She was a recipient of the Shanghai Science and Technology Progress Award and won the Second Prize, in 2016.

JIE ZHOU is currently pursuing the Ph.D. degree with the Department of Computer Science and Technology, East China Normal University, China. His research interests include sentiment analysis, aspect-level sentiment classification, retrieval models, and neural networks. He was awarded a scholarship from the China Scholarship Council and has achieved Top-3 rank in the KDD Cup Competition several times. Since 2016, he has publishing more than ten cited papers in international conferences and journals such as AAAI, *Information Sciences*, DASFAA, and ICME.

CHENGCAI CHEN received the B.Eng. degree in industrial analysis from the Department of Applied Chemistry, Qingdao University of Science and Technology. He is currently the President at Xiaoi Research and the Vice President of technology at Xiaoi Robot. His research interests include QA systems and knowledge bases, NLP, data-driven machine learning, and knowledge representation and reasoning.



LIANG HE received the bachelor's and Ph.D. degrees from the Department of Computer Science and Technology, East China Normal University, China. He is currently a Professor and the Director with the Department of Computer Science and Technology, East China Normal University. He holds more than ten patents and has published two monographs and more than 70 refereed articles in national and international journals and conference proceedings. His current research interests include knowledge processing, user behavior analysis, and context-aware computing. He is also a Council Member of the Shanghai Computer Society and a member of the Academic Committee. He has been awarded the Star of Talent by Shanghai. He has received the Shanghai Science and Technology Progress Award for five times and winning the First Prize, in 2013, and the Second Prize, in 2015. He is also the Director of the Technical Committee of Shanghai Engineering Research Center of Intelligent Service Robot and a Technology Foresight Expert at the Shanghai Science and Technology in his focus areas. In recent years, he has hosted a number of National Science and Technology Supports and participated in the National 13th Five-Year Technology Support Programs, the Shanghai Science and Technology Long-Term Development Plan, and the Shanghai 13th Five-Year Science and Technology Plan.