Noname manuscript No.
(will be inserted by the editor)

ReBoost: A Retrieval-Boosted Sequence-to-Sequence Model for Neural Response Generation*

Yutao Zhu[1,2,3] · **Zhicheng Dou**[1,2,3] · **Jian-Yun Nie**[3] · **Ji-Rong Wen**[1,2,3]

Received: date / Accepted: date

Abstract Human-computer conversation is an active research topic in natural language processing. One of the representative methods to build conversation systems uses the Sequence-to-sequence (Seq2seq) model through neural networks. However, with limited input information, the Seq2seq model tends to generate meaningless and trivial responses. It can be greatly enhanced if more supplementary information is provided in the generation process. In this work, we propose to utilize retrieved responses to boost the Seq2seq model for generating more informative replies. Our method, called ReBoost, incorporates retrieved results in the Seq2seq model by a hierarchical structure. The input message and retrieved results can influence the generation process jointly. Experiments on two benchmark datasets demonstrate that our model is able to generate more informative responses in both automatic and human evaluations and outperforms the state-of-the-art response generation models.

Keywords Retrieved results · Seq2seq model · Response generation

1 Introduction

Conversational information retrieval system, which can allow users to answer a variety of information needs naturally and efficiently, has attracted more and more attention. Such a system usually contains an open-domain conversation

* This is a post-peer-review, pre-copyedit version of an article published in Information Retrieval Journal. The final authenticated version is available online at: http://dx.doi.org/10.1007/s10791-019-09364-x

1 School of Information, Renmin University of China, Beijing, P.R. China.
2 Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, P.R. China.
Beijing Key Laboratory of Big Data Management and Analysis Methods, P.R. China.
E-mail: {ytzhu,dou,jrwen}@ruc.edu.cn
3 DIRO, Université de Montréal, Québec, C.P. 6128, Succ Centre-Ville Montréal, Québec, Canada.
E-mail: nie@iro.umontreal.ca

Input Message	Gold Response	Retrieved Results
I'm **bored**, is **anyone** awake? Anyone want to **chat**? **无聊**中，**有没**有没睡觉的？**聊**下？	*Let's go, I haven't slept yet.* *来来，我也还没睡觉*	I'm **bored**! Is there **anyone** who can **chat** with me? I will reply to anything. **无聊**中！**有**陪**聊**的**没**？*有来必应。* *It seems that many people are not sleeping.* *看来不少人都没睡觉哦*
A fresh grown **watermelon**, **have you ever seen one?** 刚结出来的**西瓜**，**你见过吗**？	*This is too tiny. Such a small watermelon. Is it edible?* *这未免太迷你了，好袖珍的西瓜，能吃吗？*	A blue **watermelon**, is it beautiful? **Have you ever seen one?** 蓝色的**西瓜**，美不美？**你见过吗**？ *Such a weird watermelon. Is it edible?* *好奇怪的西瓜，这个能吃吗？*
The **Beijing** today **is most suitable for sleeping** 今天的**北京最适合睡觉**	*Sleeping on rainy days is refreshing!* *下雨天睡觉爽！*	It's too humid in **Beijing** … This type of day **is most suitable for sleeping** **北京**湿透了…这天**最适合睡觉**了 *Today's plan: sleep all day.* *今天的计划，全天睡觉。*

Fig. 1: Sample input messages and corresponding responses from Weibo dataset. The original text is in Chinese, and we translate it into English here. Similar conversations are retrieved by our retrieval module in the training data. The words in bold appear in both input messages and retrieved results, while ones with underlines appear in both gold response and retrieved results.

module to generate a response, which is a hot research topic in natural language processing. Modern open-domain conversation systems often use data-driven approaches due to the availability of large amounts of conversation data and the recent progress made by neural methods.

There are two main categories of approaches to building an open-domain conversation system: retrieval-based methods and generation-based methods. **Retrieval-based systems** maintain a large repository of conversation data and search for a most reasonable response by information retrieval approaches [5, 8, 20, 2]. A clear advantage of retrieval-based approaches is that the responses returned are usually fluent and grammatically correct since they are selected from a repository of real human dialogues. However, as retrieval-based systems do not generate new responses, but only select a response from a repository, the repository must have a large coverage of conversations. This is difficult to guarantee in practice, as the conversation topics can vary greatly and the conversation repositories are usually limited samples of real-world conversations.

On the other hand, **generation-based systems** try to generate a response other than retrieving an existing one. Variants of sequence-to-sequence (Seq2seq) neural network models [15, 13, 7, 10, 19, 16, 18] have been successfully applied for building conversation systems. The models typically incorporate an encoder and a decoder. The encoder aims to represent one message in a vector and the decoder generates a reply based on it. An attention mechanism is often used to improve the model on learning patterns from the data [1, 9].

The Seq2seq model is able to generate new replies for new messages. However, it is often observed that the Seq2seq model is liable to generate short, trivial and meaningless replies such as "something" and "I don't know" [7]. This problem is believed to stem from insufficient source information for generating meaningful targets [16]. Despite providing a large number of data, only message-response pairs are used to capture the information and based on which all parameters are learned. In the absence of more information, trivial

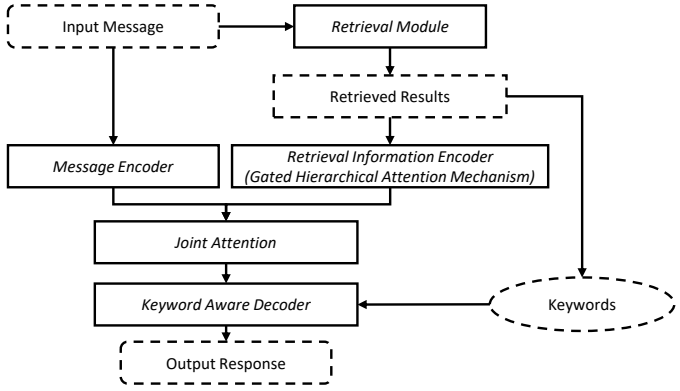

Fig. 2: The overview of ReBoost. All data are marked in dash lines. The remain parts are four modules, i.e., a retrieval module, a message encoder, a retrieval information encoder and a keyword aware decoder.

replies are often "safer" solutions. It is believed that this problem can be alleviated by introducing additional information to the generation process [19, 10]. Our work is also an attempt in this direction.

In previous studies, additional information provided by a pre-trained external model such as a commonsense knowledge graph, a topic model or an emotional classifier is proved to help generate more informative responses [4, 23, 19, 22]. However, such external knowledge is not always available in real applications and the effectiveness of external models also influences the generation results. In this work, we propose a framework, called **ReBoost**, **that uses the retrieved results as additional inputs to the Seq2seq model to boost the generation**. These retrieved results are returned by an information retrieval (IR) system on the training data, thus avoid involving any external knowledge. Let us use some examples to explain what is retrieved results and to motivate our idea. As shown in Figure 1, in a Weibo dataset, there are many similar dialogues (message-response pairs). These pairs are retrieved from an IR system by using the input message as the query. The IR system ranks the results based on the matching degree between the input message and each message in the repository. Thanks to these similar messages, the responses in retrieved results can provide some information contained in gold responses that should be generated. We call these retrieved responses as **retrieved results**. As can be seen, in the first example, the gold response and the retrieved results share some words such as "sleep". If we offer this retrieved result to response generation process, the model is possible to generate a response more related to "sleep". Therefore, we hypothesize that retrieved results can provide useful prior knowledge for generating responses.

The overview of ReBoost is illustrated in Figure 2. Specifically, given an input message, the retrieval module returns some relevant responses and their relevance scores. In our assumption, the information contained in these re-

trieved results can help generate better responses. As different words and different retrieved responses may play different roles in the generation process, we construct a hierarchical structure from word-level to sentence-level (each response contains only one sentence in our dataset). We design a **gated hierarchical attention** mechanism to integrate words, sentences, and their relevance scores to improve the generation process.

A **word-level attention** assigns different weights to words in retrieval results according to their importance in generation. The keywords which contain useful information are expected to get higher weights in this step. Then, each retrieved response is represented as a vector by the weighted sum of the word embeddings and fed into a **sentence-level attention**. Similarly, at this level, each retrieved result is assigned a weight based on its contribution to the generation process. Furthermore, we design a gate operation that utilizes the **relevance scores** as prior knowledge when assigning weights to the retrieved replies, to leverage the relevance information returned by the retrieval model. Consequently, the weighted sum of the sentence vectors constructs a supplementary vector which represents the retrieval information. In addition, to enhance the ability of the decoder, we extract some **keywords** from retrieved results to guide the generation process explicitly.

We conduct an empirical study on two large scale datasets. The first one is Sina Weibo dataset released by NTCIR-13 STC task [14]. It is a Chinese dataset constructed by users' posts and corresponding replies in Sina Weibo[1]. Another one is OpenSubtitles dataset proposed by Li, et al. [7]. This is an English dataset containing many scripted lines spoken by movie characters extracted from OpenSubtitles[2]. We compare our ReBoost model with the existing methods in both automatic and human evaluations and analyze the effectiveness of different modules in our model by a module ablation experiment. Experimental results show that ReBoost generates more informative and meaningful responses than state-of-the-art models. This confirms our assumption that utilizing retrieved results in training data is helpful in the generation process.

Our contributions are concluded as follows: (1) we present a retrieved results aware neural response generation model, which uses retrieved results as supplementary information to help the generation; (2) we design a novel gated attention mechanism to make use of relevance scores as a kind of prior knowledge to improve the learning process; (3) we conduct experiments on two widely used datasets and prove our assumption that the retrieved results are helpful in generating better responses.

The rest of the paper is structured as follows: Section 2 briefly describes recent works in neural response generation. Section 3 introduces background neural language models and text generation process. The details of our model are described in Section 4. Section 5 is a description of experiments and results.

[1] Sina Weibo, http://weibo.com
[2] OpenSubtitles, http://www.opensubtitles.org

Analysis and discussion are also given in this section. Finally, we conclude our paper in Section 6.

2 Related Work

In this section, we briefly introduce recent related works and compare them with our model. These studies are categorized into two groups: the retrieval-based system and the generation-based system.

Retrieval-based system Retrieval-based methods take the input message as a query and select a set of suitable responses by information retrieval (IR) techniques from a large conversation repository [5]. In addition to the basic information retrieval approaches, various additional features and deep networks have been used to rank and select replies. Some works focus on learning to rank responses according to their similarity with a given messages [17, 2, 20]. On the other hand of the spectrum, retrieved results from a basic IR system are further reranked by a deep learning based model [8].

Generation-based system Generation-based methods and, in particular, Seq2seq models have recently attracted increasing attention [15]. Initial works attempt to apply the Seq2seq model to response generation and the results have proved its effectiveness [13]. However, many researchers have reported that the Seq2seq model is liable to generate short, trivial and meaningless replies [7, 19, 16]. To tackle this problem, Li et al. proposed to modify the objective function in the training process, i.e. use mutual information instead of maximum likelihood when training the model [7]. Under this circumstance, the parameters in Seq2seq model are still learned from message-response pairs. With limited input information, the Seq2seq model can not generate more informative response substantially [16].

To incorporate more information into the generation process, many researchers proposed using external knowledge and models. For example, Xing et al. used a topic model to excerpt topic information and guide the generation process [19]. This model can generate more informative results with the help of topic information. However, it has two drawbacks. At first, training a usable topic model needs a large scale of text data. This external dataset is not always available. In early experiments, we trained a similar topic model on the conversation dataset (about 4 million pairs), but the results are extremely unreasonable. Secondly, given the limited number of topics, it is possible that no topic is specific enough to an input message, thus the approach is less useful in this case. Compared with this model, our method utilizes conversations in the training set that are related with input messages as supplementary materials, which can provide more specific information (such as some keywords, concepts, etc.) for response generation.

Many studies focus on facilitating response generation models with other external information such as commonsense knowledge and emotional class. Commonsense knowledge is vital to many natural language processing tasks and can also be helpful in a dialog system theoretically [4, 23]. Unfortunately,

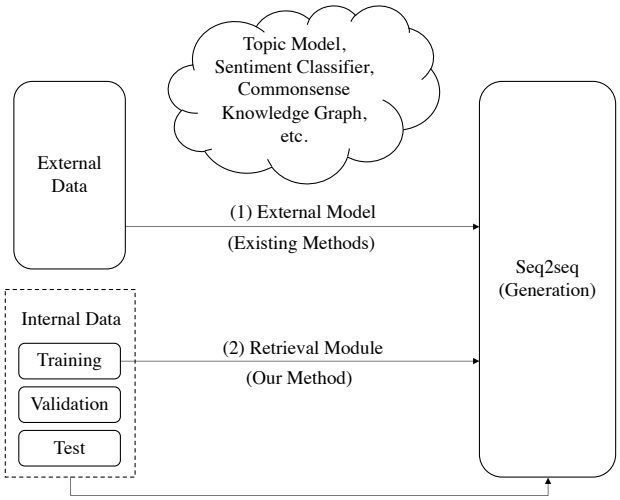

Fig. 3: Comparison between our method and existing methods. The above one (1) represents existing methods which use external data as supplementary information to improve the response generation. While the bottom one (2) is our method that uses the internal data to boost the generation process.

an open-domain commonsense knowledge graph is hard to obtain. In a recent work [23], only about 20,000 entities and their relations are used as commonsense information, which is far less compared with the number of conversational pairs (3 million) in their experiments. That is to say, only a small part of conversations can be augmented with the commonsense information. Thus the improvement is limited. Building an emotional conversation system is another interesting problem. The response can be more meaningful if the corresponding emotion is aware. Zhou, et al. proposed a chatting machine with such emotion information [22]. All conversation pairs are categorized into six groups of emotion and the classification accuracy is reported to be 64%. The generation results depend on the emotion class directly, if an inaccurate emotion is given, the generation process is affected.

In summary, as we show in Figure 3, all aforementioned models tend to improve the Seq2seq model by incorporating external data by external models. At the same time, the noise is also involved. Besides, the external data is not always available in a real application scenario. Compared with these studies relied on external knowledge, our method draws helpful information from the training set rather than an outside dataset. This is more applicable in a real scenario, and the noise in external data is also avoided meanwhile. Besides, our method that uses retrieved results to boost the generation moves a step further towards building an ensemble system combining both retrieval-based and generation-based methods.

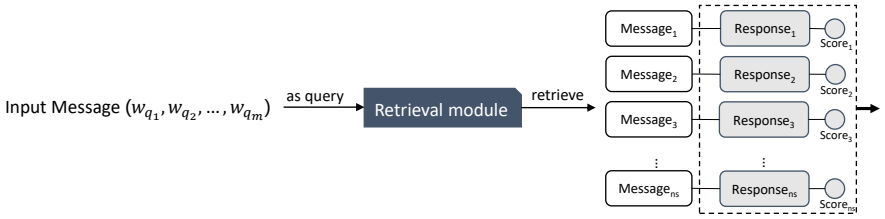

Fig. 4: Retrieval Module.

3 The ReBoost Model

To incorporate the information contained in retrieved results to the generation process, we propose the ReBoost model. As illustrated in Figure 2, our model consists of a retrieval module, a message encoder, a retrieval information encoder, and a keyword aware decoder. As introduced in Section 1, given an input message, our idea is to generate a response by using retrieved results from training data. We first retrieve n_s message-response pairs and their relevance scores with the input message by the retrieval module. Both the input message and these retrieved responses are represented as fix-sized vectors by the message encoder and retrieval information encoder respectively. We call them a message vector and a retrieval information vector. In particular, when computing the retrieval information vector, we also take into account the relevance score provided by the retrieval module, which is proved to be a helpful prior knowledge in the learning process. For decoding, two vectors provided by the encoder guide the generation process jointly. And to convey the key information more directly, we extract some keywords from retrieved results and improve their generation probabilities explicitly. The details are introduced as follows.

3.1 Retrieval module

Our motivation is using retrieved results in training data to improve the generation process, thus the first problem is how to obtain retrieved results. We build a retrieval module to achieve this (as shown in Figure 4). In particular, we use the Apache Solr[3], an open-source search platform, for the retrieval implementation. We construct the indices on the message-response pairs in training data. Both the message and response are set as attributes separately to allow the directed queries.

Given an input message, the retrieval module would provide many pairs and score them according to the semantic matching degree. Here we retrieve n_s message-response pairs according to the relevance score (BM25 [12]) between the input and the message in each pair. These retrieved pairs are denoted

[3] Apache Solr, https://lucene.apache.org/solr/

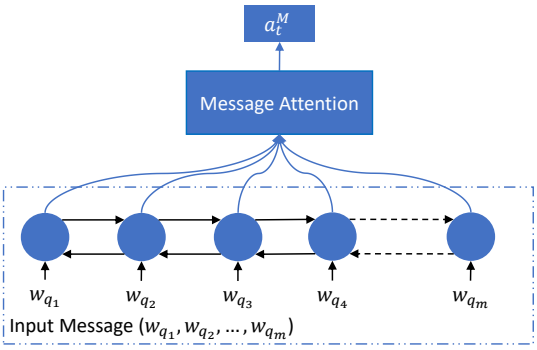

Fig. 5: Message Encoder.

as (m_k, r_k), $1 \leq k \leq n_s$. In this work, we use r_k as retrieved results. The information retrieval is a relatively mature technique, thus more sophisticated systems can be alternated as the retrieval module.

3.2 Message Encoder

The input message is represented by the input message encoder (as illustrated in Figure 5). We use a bi-directional RNN with GRU as the encoder to represent the input message.

Formally, assuming the input message with length m is $X = (x_1, x_2, \cdots, x_m)$, ReBoost first uses an embedding layer to map each word x to an d-dimension embedding $\mathbf{x}$:

$$x \Rightarrow \mathbf{x}. \tag{1}$$

Then the hidden states of the encoder are corresponding representations, i.e., $(\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_n)$, where $\mathbf{h}_i$ is computed as follows:

$$\mathbf{h}_i = [\overrightarrow{\mathbf{h}}_i; \overleftarrow{\mathbf{h}}_i], \tag{2}$$

$$\overrightarrow{\mathbf{h}}_i = \mathrm{GRU}_1(\mathbf{x}_i, \overrightarrow{\mathbf{h}}_{i-1}), \tag{3}$$

$$\overleftarrow{\mathbf{h}}_i = \mathrm{GRU}_2(\mathbf{x}_i, \overleftarrow{\mathbf{h}}_{i+1}), \tag{4}$$

where [;] is the concatenation operation. $\overrightarrow{\mathbf{h}}_i$ is the hidden state in the forward RNN, while $\overleftarrow{\mathbf{h}}_i$ is the hidden state in the backward RNN. The hidden state $\overrightarrow{\mathbf{h}}_0$ and $\overleftarrow{\mathbf{h}}_{m+1}$ are randomly initialized. The operations in a GRU cell of the

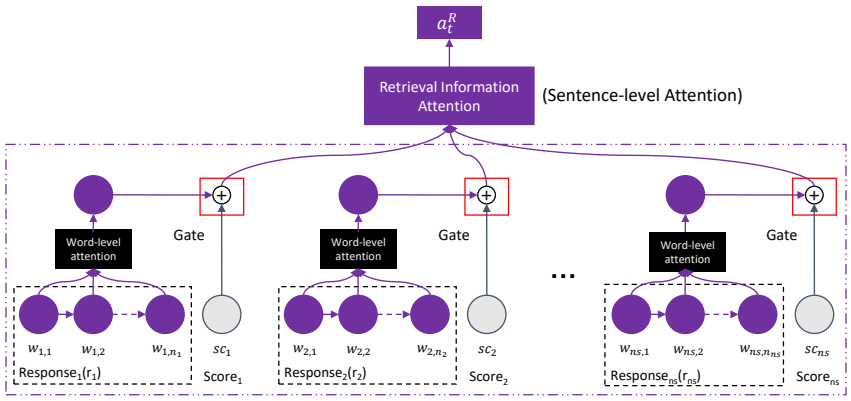

Fig. 6: Retrieval Information Encoder.

forward RNN are defined as follows:

$$\mathbf{z} = \sigma(\mathbf{W}_z \mathbf{x}_i + \mathbf{U}_z \overrightarrow{\mathbf{h}}_{i\ \ 1}), \tag{5}$$

$$\mathbf{r} = \sigma(\mathbf{W}_r \mathbf{x}_i + \mathbf{U}_r \overrightarrow{\mathbf{h}}_{i\ \ 1}), \tag{6}$$

$$\tilde{\mathbf{h}}_i = \tanh(\mathbf{W}_h \mathbf{x}_i + \mathbf{U}_h(\mathbf{r} \odot \overrightarrow{\mathbf{h}}_{i\ \ 1})), \tag{7}$$

$$\mathrm{GRU}_1(\mathbf{x}_i, \overrightarrow{\mathbf{h}}_{i\ \ 1}) = \mathbf{z} \odot \overrightarrow{\mathbf{h}}_{i\ \ 1} + (1 - \mathbf{z}) \odot \tilde{\mathbf{h}}_i, \tag{8}$$

where $\odot$ denotes element-wise product between vectors. $\tanh(\cdot)$ and $\sigma(\cdot)$ are the tanh and sigmoid function. $\mathbf{W}_h$, $\mathbf{W}_z$, $\mathbf{W}_r$, $\mathbf{U}_h$, $\mathbf{U}_z$ and $\mathbf{U}_r$ are parameter matrix. The backward RNN is defined likewise and we omit its definition here. Note that the parameters in the two RNNs are not tied together, but randomly initialized and trained separately. With the bi-directional RNN, the representation $\mathbf{h}_i$ for the word x_i can accumulate information from its context.

An attention mechanism is involved to summarized the input message representations into a fixed-size vector. To make it clear, we call it the input message vector and denote it as $\mathbf{a}_t^M$. The calculation of the input message vector is:

$$\mathbf{a}_t^M = \sum_{j=1}^{m} \alpha_{tj} \mathbf{h}_j, \tag{9}$$

$$\alpha_{tj} = \frac{\exp(e_{tj})}{\sum_{k=1}^{m} \exp(e_{tk})}, \tag{10}$$

$$e_{tj} = \tanh\left(\mathbf{W}_{\alpha_1}[\mathbf{s}_{t\ \ 1}; \mathbf{h}_j]\right), \tag{11}$$

where $\mathbf{s}_{t\ \ 1}$ is the hidden state of the decoder in the decoding time step $t-1$, which will be introduced later.

3.3 Retrieval Information Encoder

From the retrieval module, we can obtain several retrieved results and their relevance scores. The next question is how to utilize and incorporate them into the generation process. In real life, facing a new message, people often generate replies containing some keywords. The retrieved responses can be used to identify those keywords. If similar conversations happened before, the replies can even be reused. Based on this observation, we utilize the retrieved results at different levels and propose a gated hierarchical attention mechanism.

A simple way to implement our idea is to directly feed the keywords or retrieved responses into the decoder. However, this simple model cannot distinguish between more important and less important retrieved results during reply generation. Besides, each retrieved result is a natural language sentence consisting of multiple words. The contribution of these words in generating a corresponding response is different. Thus retrieved results should be modeled hierarchically, namely from word-level to sentence-level. Unfortunately, the simple model cannot extract the hierarchical information contained in retrieved responses. To address these issues, we design a gated hierarchical attention mechanism (as shown in Figure 6). This attention mechanism comprises a word-level attention layer and a sentence-level attention layer. They are used to assign different weights to the words in the retrieved results and the retrieved results according to their importance or contribution in generating a target response. In the sentence-level attention layer, we add a gate operation (red rectangular in Figure 6) to incorporate the relevance score provided by the retrieval module. The relevance score is used as prior knowledge to guide the calculation of weights for each retrieved response.

Formally, assume $(r_1, r_2, \cdots, r_{n_s})$ are responses provided by the retrieval module and $(sc_1, sc_2, \cdots, sc_{n_s})$ are their corresponding relevance scores. Similar to the input message, the k-th response $r_k = (w_{k,1}, w_{k,2}, \cdots, w_{k,n_k})$ is first mapped into d-dimension embeddings and then represented as $(\mathbf{h}_{k,1}, \mathbf{h}_{k,2}, \cdots, \mathbf{h}_{k,n_k})$ by an RNN with a GRU cell. At decoding time step t, the representation of r_k could be calculated using a traditional attention mechanism as follows:

$$\mathbf{r}_{k,t} = \sum_{j=1}^{n_k} \alpha_{k,t,j} \mathbf{h}_{k,j}, \tag{12}$$

$$\alpha_{k,t,j} = \frac{\exp\left(o_{k,t,j}\right)}{\sum_{l=1}^{n_k} \exp\left(o_{k,t,l}\right)}, \tag{13}$$

$$o_{k,t,j} = \tanh\left(\mathbf{W}_{\alpha_2}[\mathbf{s}_{t\ \ 1}; \mathbf{h}_{k,j}]\right), \tag{14}$$

where $o_{k,t,j}$ and $\alpha_{k,t,j}$ are the original and normalized weights of the j-th word in k-th retrieved result when generating the t-th word in target response. Note that the representation of the k-th response r_k is not fixed but changing in different decoding steps, thus we add a subscript to distinguish it, e.g., $\mathbf{r}_{k,t}$ for the representation in the time step t. $\{\mathbf{r}_{k,t}\}_{k=1}^{n_s}$ are then fed into the sentence

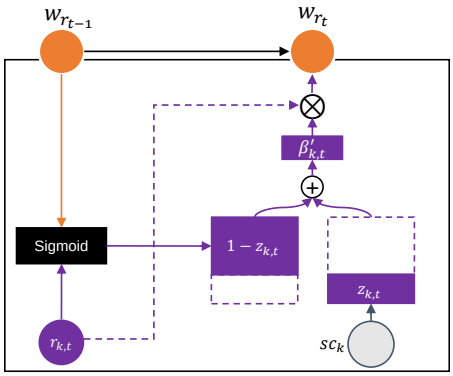

Fig. 7: The Gate Mechanism.

level attention layer and assigned a weight $\beta_{k,t}$ to form a context vector $\mathbf{a}_t^R$:

$$\mathbf{a}_t^R = \sum_{k=1}^{n_s} \beta_{k,t} \mathbf{r}_{k,t}, \tag{15}$$

$$\beta_{k,t} = \frac{\exp\left(o_{k,t}^{\emptyset}\right)}{\sum_{j=1}^{n_s} \exp\left(o_{j,t}^{\emptyset}\right)}, \tag{16}$$

$$o_{k,t}^{\emptyset} = \tanh\left(\mathbf{W}_{\beta}[\mathbf{s}_{t\ \ 1}; \mathbf{r}_{k,t}]\right), \tag{17}$$

where $\beta_{k,t}$ is the normalized attention weight of the k-th retrieved result which reflects its contribution (importance) in generating the t-th word. $o_{k,t}^{\emptyset}$ is the weight before normalization. These equations are used in the traditional attention mechanism, but they are not suitable in our sentence-level attention. Thus we modify the calculation of $\beta_{k,t}$ and $o_{k,t}^{\emptyset}$, which are introduced as follows.

We modify $\beta_{k,t}$ at first. This normalized weight of response is learned automatically. But in our case, when returning the retrieved results, the retrieval module also provides **relevance scores** for those results which measure their relevance with the given message. Obviously, these relevance scores are valuable prior knowledge for the attention mechanism when assigning a weight for each retrieved result. However, they are not always reliable. To take into account this factor, we also use alternative attention weights learned by the model itself. As both signals (given relevance scores and learned weights) are useful, we design a gate operation to automatically control their importance during the generation process.

The detail of this gate operation is shown in Figure 7. Formally, considering the process of assigning a weight for a retrieved reply $r_{k,t}$ at the time step t, the normalized weight of sentence $\beta_{k,t}$ is calculated by a given relevance score sc_k and an original weight $o_{k,t}^{\emptyset}$ learned by the model:

$$\beta_{k,t} = z_{k,t} \cdot sc_k + (1 - z_{k,t}) \cdot o_{k,t}^{\emptyset}, \tag{18}$$

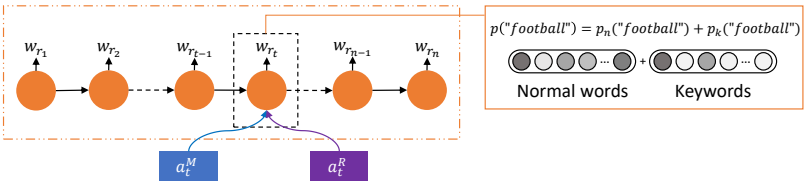

Fig. 8: Keyword Aware Decoder.

where $z_{k,t}$ is the *refer* gate that controls how much the overall weight refers to the relevance score. It is randomly initialized and tuned in the training process. A smaller $z_{k,t}$ means the weight learned by the model is more suitable to the case.

In traditional attention mechanism (presented in Equation (10)), the original weight $o^0_{k,t}$ is normalized as a probability distribution over a set of input vectors, i.e. all retrieved replies are assigned positive values (probabilities) and their sum is equal to one. However, this is not suitable to our case because: 1) there could be more than one relevant replies, all of them can be assigned high weights, thus the limitation on the sum of their weights is not suitable; 2) the retrieved responses are not always relevant, all irrelevant responses should be assigned small weights, i.e. be ignored in the generation process. We expect our model to have the ability to determine whether a retrieved reply is useful or not. Based on these considerations, we remove the softmax normalization in Equation (10) and modify the calculation of the weight $o^0_{k,t}$ as follows:

$$o^0_{k,t} = \text{sigmoid}(\mathbf{W}_\beta[\mathbf{s}_{t\ \ 1}; \mathbf{r}_{k,t}]). \tag{19}$$

The value of this weight is between 0 and 1. A higher value of $o^0_{k,t}$ indicates $\mathbf{r}_{k,t}$ is more important in the generation process.

With the above gated hierarchical attention mechanism, we can selectively use the retrieved replies and the words contained in them. The vector $\mathbf{a}^R_t$ is used as our context vector.

3.4 Keyword Aware Decoder

From the aforementioned two encoders, both the input message and retrieval information are represented as vectors $\mathbf{a}^M_t$ and $\mathbf{a}^R_t$ respectively. Then the message vector $\mathbf{a}^M_t$ and the retrieval information context vector $\mathbf{a}^R_t$ are concatenated together and sent to the keyword aware decoder.

$$\mathbf{a}_t = [\mathbf{a}^M_t; \mathbf{a}^R_t], \tag{20}$$

where [;] is the concatenation operation.

The modules we proposed above manipulate retrieved information in the encoder step. We also consider making use of retrieval results to directly guide the generation process in the decoder. Specifically, we modify the generation

probability in the decoder to make it biased towards some keywords in related responses. We call it the keyword aware decoder. The intuition is that the keywords which appeared frequently in related responses are more relevant and may contain helpful information. To implement this idea, we first extract some nouns as candidate keywords in responses according to their TF-IDF values. Then sort them by their frequency and the top N_k of them are remained as selected keywords.

Formally, at decoding time step t, for a target word y_t, the generation probability p_t is:

$$p_t = p_n + p_k, \tag{21}$$

$$p_n = \text{softmax}(\mathbf{W}_s \mathbf{s}_t + \mathbf{b}_s), \tag{22}$$

$$p_k = \text{softmax}(\mathbf{W}_k [\mathbf{s}_t; \mathbf{a}_t^R] + \mathbf{b}_k), \tag{23}$$

$$\mathbf{s}_t = \text{GRU}(\mathbf{y}_{t\ \ 1}, [\mathbf{s}_{t\ \ 1}; \mathbf{a}_t]), \tag{24}$$

where $\mathbf{W}_s$, $\mathbf{W}_k$, $\mathbf{b}_s$ and $\mathbf{b}_k$ are parameters. It is worth noting that the probability p_k is only computed for the selected keywords, and the probability for other words in this vector is masked as zero. In this way, the generation probability is biased to the selected keywords. For a non-keyword, the generation probability is the same as that in the standard Seq2seq model. But for a selected keyword, there is an extra probability term that increases its generation probability. This extra term is determined by the current hidden state of decoder $\mathbf{s}_t$ and the retrieval information attention vector $\mathbf{a}_t^R$. When a keyword is relevant to the generated parts and the input message, it will be more possible to appear in a response.

In conclusion, in our keyword aware decoder, the retrieval information can guide the generation process through the joint attention vector (implicitly) and the keywords (explicitly).

One advantage of our model is that it will be trained to learn how to use different levels of retrieval information through the gated hierarchical attention mechanism. If such information turns out to be unreliable, the gated attention mechanism is able to assign a small weight to it or ignore it. On the other hand, the extracted keywords can influence the generation process directly, which further helps the model to generate more informative replies.

Overall, the retrieved replies and the input message provide complementary information to the response generation module. Our framework offers a new way to integrate retrieval-based and generation-based approaches.

4 Experiments

4.1 Dataset and Preprocessing

We use the Chinese Sina Weibo dataset released by NTCIR-13 STC task [14] and the English OpenSubtitles dataset proposed by Li, et al. [7].

For the Weibo dataset, the user's posts are used as messages and the comments as responses. Following the existing approach [19], we randomly select 4.3 million pairs as the training set, 50,000 pairs as the validation set and 5,000 pairs as the test set. There is no overlap among the three sets. The retrieval module is built on the training set and provides related responses for training, validation and test set. To avoid the model "seeing" the ground-truth response, we remove the original response (the ground-truth) from the retrieved results in the training set. The messages in the test set are used as inputs to generate responses and the corresponding original responses are used as the ground-truth to calculate evaluation metrics. All the text are segmented by Jieba[4], a Chinese word segmentation tool. We construct two vocabularies for posts and responses by using 40,000 most frequent words, covering 97.01% and 95.65% usage of words respectively. The words not in the dictionary are replaced by a special token "⟨unk⟩".

For the OpenSubtitles dataset, it containing many scripted lines spoken by movie characters. As the dataset does not specify which character speaks each subtitle line, following the same assumption as [7], each line of the subtitle is used as a full speaker turn. And our models are trained to predict the next turn given the current ones based on the assumption that two consecutive turns belong to the same conversation. Consequently, we randomly select 5 million pairs as the training set, 50,000 pairs as the validation set and 50,000 pairs as the test set. Other settings are the same as the Weibo dataset. And the dataset is preprocessed by the author[5].

4.2 Baseline Models and Experiment Setup

We compare our models with the following baseline models and the state-of-the-art models:

- S2SA: the standard Seq2seq model with an attention mechanism. This is the basic model for response generation.
- NRM-hyb: the best model in [13] using two encoders to represent messages in local and global schemes. In the local information encoder, attention mechanism is used to aggregate and summarize the information in the input message and the attention vector is used as the local representation. In the global information encoder, the hidden state of the last word in the input message is used as the global representation. The two representations are concatenated together and fed to the decoder. This model uses more complex encoders to get better representations of the input message, which is an easy way to improve the informativeness of the generated response.
- MMI: the best model in [7] which uses a diversity-promoting objective function to train the Seq2seq model. It first trains a Seq2seq model for generating responses based on the given input message. Then, another

[4] Jieba, https://github.com/fxsjy/jieba

[5] https://github.com/jiweil/Neural-Dialogue-Generation

Seq2seq model is trained for generating input messages based on the given response. The first model is used to generate a list of responses for a given input, and the second model is used to rerank the list based on their probability of generating the given input. This model modifies the objective function of the Seq2seq model which is different from us that uses supplementary information. We select this model as a baseline to compare which way is better in generating informative responses.

– TA-Seq2seq: the model proposed by Xing et al. [19] which uses a topic model to extract topic information and utilizes it to boost the Seq2seq model. For each input message, the pre-trained topic model assigns a topic for it and the corresponding topic words are fed into the decoder by the attention mechanism. In the experiments, we train a topic model on the training set to make a fair comparison.

We use the same settings for the training on two datasets. The common settings for all models are introduced at first followed by the specific settings for each model respectively. (1) Common settings: for all models, including ReBoost and the baselines, the dimension of the hidden states of both encoder and decoder is 1,000 and the dimension of the word embeddings is 300. All model parameters are initialized with uniform distribution in [-0.1, 0.1] and trained with the Adam algorithm [6] and mini-batch of size 128 on NVIDIA Tesla K40 GPU. The initial learning rate is 0.001, which decays dynamically in the training. We also use the validation set for an early stop. Beam search with a beam width of 10 is used for predicting the results.

(2) Specific settings: a) NRM-hyb contains two RNNs as the encoder, both of them have the same hidden size (1,000) but the parameters are not shared. b) MMI trains two Seq2seq models and they have the same settings as the common settings. c) The topic model for TA-Seq2seq is trained by Biterm [21], which is a state-of-the-art topic model for short texts. Following the original experimental setting, the number of topics is 200 and the top 100 words in each topic are selected. For each input message, 15 topic words with the highest probability (topic probability multiply word probability) are selected as supplementary information for decoding. 4) In ReBoost, we use Apache Solr 6.5 and its default ranking function BM25 as the retrieval module. The number of retrieved results is ten. 15 words with the highest TF-IDF values in retrieved results are provided to the decoder with a biased generation probability. Zero paddings are used if there are less than 15 keywords. As retrieved results are from the training set, we should avoid providing the original response for an input message. Therefore, the response that is the same as the original one is removed from the retrieved results and this forces the model to learn how to use the retrieved results rather than simply copy a ground-truth for the generation. All datasets and codes will be released later[6].

[6] https://github.com/DaoD/ReBoost

Ground-truth	Green is a restful and quiet color. 绿色/是/一种/让人放松/和/安静的/颜色/。	
Result	Green is my favorite color. 绿色/是/我/最喜欢的/颜色/。	Green, green, hahaha 绿色/，/绿色/，/哈哈哈
Distinct-1	6 / 6 = 1.00	3 / 5 = 0.60
Distinct-2	5 / 5 = 1.00	3 / 4 = 0.75
BLEU-1	47.77	10.98
BLEU-2	37.00	0
BLEU-3	0	0
BLEU-4	0	0

Fig. 9: Examples for demonstrating the metrics.

4.3 Evaluation Metrics

To evaluate the performance of our model and baseline models, we follow existing studies and employ several standard metrics: perplexity, distinct and BLEU-N.

Distinct-1 and Distinct-2 These two metrics are proposed by Li et al. [7] to measure the degree of diversity according to the ratios of distinct unigrams and bigrams in generated responses.

Higher values of these metrics indicate the replies contain more different words and more information potentially. Let us use an example to demonstrate the metrics. As shown in Figure 9, all unigrams and bigrams in the left case are distinct, therefore the values of Distinct-1 and Distinct-2 are both 1.00. As for the right case, there are 5 unigrams and 4 bigrams in the sentence but only 3 of them are distinct, thus the results are 0.60 and 0.75 respectively.

BLEU-N BLEU is a metric that is originally used in machine translation [11]. It evaluates the output by using n-gram matching between the output and the reference. BLEU-1, BLEU-2, BLEU-3, and BLEU-4 are commonly used.

Formally, BLEU-N score is calculated by:

$$\text{BLEU-N} = \exp\left(\min\left(1-\frac{r}{c},0\right)+\sum_{n=1}^{N} w_n \log p_n\right), \tag{25}$$

where r and c are the lengths of the reference response and candidate ones respectively, p_n is the modified n-gram precision, and N means using n-grams up to length N and $w_n = 1/N$. Based on the formula, we can see that the BLEU value depends on both the length of the response and the n-gram precision. Higher BLEU values mean that the output response and the reference have more sharing words and are more similar. As shown in Figure 9, comparing the two cases, the left one is much close to the ground-truth sentence since they share more words, thus its BLEU values are much higher. The trigrams

Table 1: Automatic Evaluation Results.

(a) Results on Weibo Dataset

	Distinct-1	Distinct-2	BLEU-1	BLEU-2	BLEU-3	BLEU-4
S2SA	.0107	.0499	7.25	2.77	1.46	0.93
NRM-hyb	.0142	.0699	11.66	4.69	2.70	1.90
MMI	.0132	.0683	12.70	4.66	2.52	1.69
TA-Seq2seq	.0133	.0671	12.30	4.59	2.52	1.85
ReBoost	**.0302**	**.2112**	**12.73**	**5.68**	**3.55**	**2.62**

(b) Results on OpenSubtitles Dataset

	Distinct-1	Distinct-2	BLEU-1	BLEU-2	BLEU-3	BLEU-4
S2SA	.0025	.0078	6.84	2.59	1.4	0.75
NRM-hyb	.0025	.0080	6.57	2.7	1.46	0.77
MMI	.0015	.0062	**8.83**	3.36	1.76	0.9
TA-Seq2seq	.0026	.0089	5.04	2.03	1.16	0.7
ReBoost	**.0027**	**.0090**	8.57	**3.93**	**2.21**	**1.49**

and four-grams in these two cases are all different from the ground-truth, thus the BLEU-3 and BLEU-4 are equal to 0.

4.4 Overall Performance

We compare our ReBoost model with all baselines and the results are showed in Table 1. The performance improvements of ReBoost on all metrics are statistically significant (p-value < 0.01) and Bonferroni correction is applied for counteracting the problem of multiple comparisons. Based on the results, we can find:

On the Weibo dataset, ReBoost achieves higher performance on all metrics. Based on the results in terms of Distinct-1 and Distinct-2, we can conclude that ReBoost can generate more different words. This partially indicates the responses are more diverse and informative. This result proves our assumption that the retrieved results are useful supplementary information in the response generation. As for the BLEU scores, a higher BLEU score usually indicates a higher similarity between the generated responses and the ground truth. All BLEU values of the results demonstrate our ReBoost model outperforms other baselines in response generation.

On the OpenSubtitles dataset, the conclusions are similar except for two points: (1) All values are lower than that on Weibo dataset. After comparing these two datasets, we find that the sentences in OpenSubtitles are usually incomplete. This may because of the ellipses in English. The incomplete sentences are much more difficult for the model to learn the mapping. (2) MMI achieves the best results in terms of BLEU-1 among all models. We check the generated responses and find that there are many long and repeated sentences

Table 2: Human evaluation results on Weibo dataset.

(a) Absolute Scores

Models	+2	+1	0	Kappa
S2SA	23.60%	36.50%	39.90%	.326
NRM-hyb	27.80%	40.70%	31.50%	.335
MMI	23.40%	43.30%	33.30%	.291
TA-Seq2seq	28.00%	44.70%	27.30%	.339
ReBoost	33.00%	34.40%	32.60%	.372

(b) Side-by-side Comparisons

Models	Win	Tie	Lose	Kappa
ReBoost vs. S2SA	37.50%	44.00%	18.50%	.311
ReBoost vs. NRM-hyb	34.80%	42.90%	22.30%	.347
ReBoost vs. MMI	39.30%	37.10%	23.60%	.322
ReBoost vs. TA-Seq2seq	30.30%	45.60%	24.10%	.315

such as "I don't know what you're thinking". These results can achieve better BLEU values but are very boring and trivial, which leads to lower Distinct values.

In summary, our ReBoost model outperforms other baseline models in almost all automatic evaluation metrics. These results prove that incorporating retrieved responses can improve the performance of the Seq2seq model.

4.5 Human Evaluation

4.5.1 Results and Analysis

In addition to evaluating the models with automatic metrics, we also conduct a human evaluation. We randomly selected 200 messages from the test set and collect the corresponding results generated by each model. Then we invited 5 evaluators with rich experience of Sina Weibo to do two kinds of evaluations: absolute scoring and side-by-side comparison. In both evaluations, Fleiss's kappa [3] is used to evaluate the degree of agreement.

The first human evaluation is **absolute scoring**. Following the criterion of [13], the labelers are asked to judge a result based on 5 criteria: grammar correctness, fluency, logic consistency, semantic relevance, and scenario dependence. Responses from different models are shuffled and mixed and the evaluators are required to assign a score from 0 to +2 for each response independently. A suitable (+2) response means the response is appropriate, natural and informative. A neutral (+1) one is a reply that is either suitable only in a specific scenario or trivial and universal that can be used for many messages. And an unsuitable (0) response means it is impossible to find a scenario where this response is suitable, i.e., it is irrelevant to the input message or contains

grammar errors. To ensure consistency, before labeling, the annotators are trained with some examples.

Table 2(a) shows the results. The kappa scores indicate that labelers are in fair agreement with the quality of responses. The results demonstrate clearly that our ReBoost model generates much more informative responses (+2) and less trivial responses (+1). This indicates that additional retrieval information can help generate more informative replies. However, comparing with TA-Seq2seq, ReBoost generates more results with label 0. We analyze the results generated by ReBoost and find that ReBoost tends to use more diverse words to synthesize informative responses. This may involve some noise and hurt the coherence of the response. In the future, we plan to add more constraints to the decoder for generating more coherent responses. Among the baseline models, TA-Seq2seq introduces topic information as prior knowledge and it generates the most informative responses (28%). Both ReBoost and TA-Seq2seq utilize additional information into the generating process, thus the results consistently prove that incorporating more information can help alleviate the trivial replies problem.

We further conduct a **side-by-side comparison** evaluation on generated results. For the 200 samples, we created 800 triplets (message, response 1, response 2) where one response is generated by ReBoost and the other is generated by a baseline. In each triplet, the two responses are randomly shuffled so that the evaluators cannot easily guess which response is generated by ReBoost. The evaluators follow the same 5 criteria in the former annotation to judge the quality of each response. They are required to compare the two results and make a decision among win, lose and tie (win: response 1 is better; loss: response 2 is better; tie: they are equally good or bad).

The side-by-side annotation results are showed in Table 2(b). We find: (1) ReBoost model outperforms all the baselines, which indicates our model can generate much more suitable results. (2) ReBoost model outperforms TA-Seq2seq. This confirms that our method using retrieved replies is more effective than TA-Seq2seq, which selects a set of topic words to enhance response generation.

4.5.2 Discussions

We find that the Kappa is not high in the human evaluation results. To investigate the reason, we sample some cases which cause disagreements among annotators. These cases are shown in Figure 10. The generated responses are marked with an underline.

In the first case, two annotators think that the generated response has grammatical errors and it is difficult for them to understand the response. On the contrary, another three annotators consider the response as a suitable one since it mentions the key information “cut hair” in the input message. As for the second and third examples, things are similar. One annotator cannot well understand the response and annotate it with “0” score. Some remain-

Input Messagae	Labelers				
	#1	#2	#3	#4	#5
I have lost a lot of hair recently. So I make up my mind and cut my hair which has been kept for more than ten years. 最近掉头发，狠狠心把留了十几年的长发剪了 I also cut it. I have cut it for several years. 我也剪剪了，剪了这么多年。	+2	0	+2	0	+2
Five essential things for the university students doing in summer vacation. 大学生暑假应该做的5件事 OK, I have to admit that I'm a university student. 好吧，我承认我是大学生。	+1	0	+2	+1	+1
This is man's friendship! I read it for 10 minutes but I kept silent for an hour! 这就是男人的友谊！我看了10分钟却沉默了一小时！ This is my friendship! 这就是我的友谊！	+2	+1	0	+2	+2

Fig. 10: Cases of disagreement among annotators.

Table 3: Module Ablation Results

(a) Results on Weibo Dataset

Models	Distinct-1	Distinct-2	BLEU-1	BLEU-2	BLEU-3	BLEU-4
ReBoost	.0302	.2112	**12.73**	**5.68**	**3.55**	**2.62**
ReBoost-gate	.0247	.1719	12.34	5.03	2.86	1.95
ReBoost-keywords	.0231	.1411	11.73	4.88	2.82	1.95
ReBoost-retrieval	.0212	.1352	10.51	4.36	2.61	1.88
Retrieval	**.2138**	**.7091**	10.63	4.03	2.36	1.71

(b) Results on OpenSubtitles Dataset

Models	Distinct-1	Distinct-2	BLEU-1	BLEU-2	BLEU-3	BLEU-4
ReBoost	.0027	.0090	**8.57**	**3.93**	**2.21**	**1.49**
ReBoost-gate	.0022	.0070	7.79	3.19	1.83	1.02
ReBoost-keywords	.0022	.0073	8.32	3.63	2.09	1.15
ReBoost-retrieval	.0021	.0071	7.52	3.14	1.79	1.00
Retrieval	**.0364**	**.3242**	6.17	1.74	0.88	0.59

ing annotators think the response is trivial and can be used for many input messages, while others consider the response is proper.

Based on the examples, we can find that it is difficult to make a gold standard in the evaluation of response generation. In the future, we plan to conduct the evaluation from different angles such as informativeness and appropriateness and perform the annotation respectively. This may help to improve the degree of agreement among different annotators.

4.6 Module Ablation

In our model, we design a new gated hierarchical attention mechanism to encode the retrieved results. And we also modify the decoder to make the generated responses biased to some keywords in retrieved results. To investigate the effectiveness of these two strategies and the performance of the retrieval module, we conduct a module ablation experiment.

At first, we remove the gate mechanism in the retrieval information encoder. In other words, the relevance scores returned by the retrieval module are not provided to the model. The weights of different retrieved results are learned in the training process without any prior knowledge. We denote this model as **ReBoost-gate**. Then, we remove the additional probabilities for the keywords in the decoder. All words are treated as normal words and their generation probabilities are computed by Equation (29). This model is denoted as **ReBoost-keywords**. Third, to investigate the performance of retrieval information encoder, we remove this encoder and only the input message encoder and the keyword aware decoder remain. This model is denoted as **ReBoost-retrieval**. Finally, as the retrieval module could provide many related responses, we can use the top one result as the reply. And this model is denoted as **Retrieval**.

The results are reported in the Table 3. Based on the results, we can find: (1) Except for Retrieval in terms of Distinct, the full ReBoost model achieves the best results on all metrics. This demonstrates that all modules in ReBoost are useful in boosting the Seq2seq model. (2) The retrieval information encoder is the most important module in ReBoost since the performance drops most after removing it. (3) The effectiveness of the gate mechanism and the keyword aware decoder is not definite since the results are different on two datasets. We think the performance is related to the data if more accurate keywords can be extracted, the keyword aware decoder would contribute more. (4) Retrieval can achieve extremely good results on Distinct but failed on BLEU values. We check the corresponding results and find that they are fluent and informative but not so relevant to the input message. This is because these responses are human written which are much longer and more natural. And this also indicates that directly using retrieved results as replies is not reliable and they are more suitable to be used as supplementary information.

4.7 Case Study and Error Analysis

Figure 11 shows examples generated by ReBoost, TA-Seq2seq and S2SA. The sentences with underlines are one of the retrieved results. From the figure, we can observe a few findings:

(1) Based on the first example, we can see that compared with S2SA, ReBoost and TA-Seq2seq can generate more suitable results. The S2SA model even generates a confused response. This is consistent with the basic assump-

Input Message	ReBoost	TA-Seq2seq	S2SA
I have to close my Weibo **account** temporarily. Stop embezzling my **account**, **hacker**! 我必须暂时关闭微博了，别再**黑**我的**号**了！ Block this **account**. 锁定这个**号**	Is there someone **hacked** your **account**? 有人**盗**你**号**了吗？	What? 什么？	I'm your Weibo. 我是你的微博
This is the **Shanghai**'s most **beautiful** moment. 这是**上海**最美的时刻。 I haven't seen the **sky** in this color. 我从未看过这种颜色的**天空**	The **sky** is so **beautiful** in **Shanghai**. **上海**的**天空**真**漂亮**啊	Where are you? Beautiful! 你在哪里啊？真好看！	Beautiful! Beautiful! Beautiful! 好看！好看！好看！
People in different **zodiac signs** have different sleeping postures. 不同**星座**的人有不同的睡姿 I'm Leo but I like **Aries** guys. 我是狮子座但我喜欢**白羊座**	I'm an **Aries**. 我是白羊座	Me, too. 我也是	Me, too. 我也是

Fig. 11: Case study samples.

Input Message	Top 1 Retrieved Result	Ground-truth Response	Generated Response
I will go back to Guangzhou tomorrow. Hah. My third brother will play a song for you. 明天回广州了，哈哈哈，三弟给大家弹一首	It turns out that Su Wei has such a skill! 原来苏伟还有这么一手呢！	Let's start with the song, 1155665.4433221. Hah 先来一首， 1155665. 4433221. 哈哈	Come on Suwei! Come on Dapeng! 苏伟加油！大鹏加油！
Many talented filmmakers were defeated by starving. Go to Sleep! 很多有才华的电影人，败在饥不择食。睡了！	Sometimes you feel hungry but there is no food. 有时候饥了，连食都没有	Confused, but I don't want to sleep. 纠结，但不能就此睡过	Hah, me too. 哈哈，我也是这样的。
The coldest joke is that when you already know the truth, he lies in front of you. 最冷的笑话是，当你已经知道真相，他却当着你的面撒谎。	The most ridiculous thing is to pretend that everything is true! 最最可笑的是还要装作一切是真的一样！	This is not the coldest joke. This is the most painful injury. 这不是最冷的笑话。这是最痛的伤害。	This is most most most most most 这是我最最最最最

Fig. 12: Bad responses with different type of errors.

tion that the Seq2seq model can be improved by incorporating more supplementary information.

(2) Based on the first two examples, the words in bold indicate that ReBoost can generate responses containing some keywords appeared in messages and retrieved results that make it more relevant and informative. This proves our assumption that the retrieved results can be used to boost the Seq2seq model in generating much more informative replies.

(3) Analyzing the last example, we can find the ReBoost model can better extract the semantic relationship between a message and a response (such as "zodiac signs" - "Aries"). This is achieved by providing the retrieved results to the model since the "Aries" appeared in the retrieved results.

To further investigate how to improve our model, we also do error analysis. We collect the samples that have more than three 0 labels and obtain 42 samples (the total number is 200). After checking their corresponding retrieved results, we categorize the errors into three types which are shown in Figure 12.

The first type of error is caused by irrelevant retrieved results. About 16.7% (7 of 42) bad responses are in this error. As shown in the first example, the retrieved result contains a name "Suwei" and ReBoost inserts this word into the generated response. Under this circumstance, the generated response can

only be suitable in some specific cases (e.g., the input message is from Suwei or Dapeng). This indicates that our model cannot distinguish how specific a word is. Too specific words may hurt the generated response. In the future, we can use some keywords extraction techniques to provide a weight of each word in each retrieved results. This may help the model to reduce this type of problem.

The second type of error is stem from neglecting the useful retrieved results. About 33.3% (14 of 42) errors are in this type. As we can see in the second example, the retrieval module provides a suitable response for the input message but ReBoost neglects it. In the future, we plan to collect all responses generated by ReBoost and retrieved by the retrieval module, and then rerank them to output a most suitable one as the reply.

The third type of error is caused by using the retrieved results incorrectly. There are 50% (21 of 42) bad responses in this type. In the third example, the top one retrieved result mentioned the word "most", but the generated response repeatedly uses this word and make a mistake. This indicates that we need to refine our keyword aware decoder to make sure the inserted keyword would not hurt the sentence.

5 Conclusion

In this paper, we propose to use retrieved replies to boost the Seq2seq model to generate more informative and interesting responses by a gated hierarchical attention mechanism. This is a novel way to combine the retrieval- and generation- based methods. Empirical results with both automatic and human evaluations confirm our model can generate better responses than the state-of-the-art models. The proposed framework can be improved in the future on several aspects: building a more advanced retrieval module, extracting other types of information from retrieved replies, etc.

Acknowledgements Zhicheng Dou is the corresponding author. This work was funded by the National Natural Science Foundation of China under Grant No. 61872370, National Key R&D Program of China No. 2018YFC0830703.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. CoRR **abs/1409.0473** (2014)
2. Bartl, A., Spanakis, G.: A retrieval-based dialogue system utilizing utterance and context embeddings. In: 16th IEEE International Conference on Machine Learning and Applications, ICMLA 2017, Cancun, Mexico, December 18-21, 2017, pp. 1120–1125 (2017)
3. Fleiss, J.L., Cohen, J.: The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. Educational and psychological measurement **33**(3), 613–619 (1973)

4. Ghazvininejad, M., Brockett, C., Chang, M., Dolan, B., Gao, J., Yih, W., Galley, M.: A knowledge-grounded neural conversation model. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018 (2018)
5. Ji, Z., Lu, Z., Li, H.: An information retrieval approach to short text conversation. CoRR **abs/1408.6988** (2014)
6. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR **abs/1412.6980** (2014)
7. Li, J., Galley, M., Brockett, C., Gao, J., Dolan, B.: A diversity-promoting objective function for neural conversation models. In: NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016, pp. 110–119 (2016)
8. Li, X., Mou, L., Yan, R., Zhang, M.: Stalematebreaker: A proactive content-introducing approach to automatic human-computer conversation. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016, pp. 2845–2851 (2016)
9. Luong, T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015, pp. 1412–1421 (2015)
10. Mou, L., Song, Y., Yan, R., Li, G., Zhang, L., Jin, Z.: Sequence to backward and forward sequences: A content-introducing approach to generative short-text conversation. In: COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan, pp. 3349–3358 (2016)
11. Papineni, K., Roukos, S., Ward, T., Zhu, W.: Bleu: a method for automatic evaluation of machine translation. In: ACL 2002, July 6-12, 2002, Philadelphia, PA, USA., pp. 311–318 (2002)
12. Robertson, S.E., Zaragoza, H.: The probabilistic relevance framework: BM25 and beyond. Foundations and Trends in Information Retrieval **3**(4), 333–389 (2009)
13. Shang, L., Lu, Z., Li, H.: Neural responding machine for short-text conversation. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers, pp. 1577–1586 (2015)
14. Shang, L., Sakai, T., Li, H., Higashinaka, R., Miyao, Y., Y., A., Nomoto, M.: Overview of the NTCIR-13 short text conversation task. In: Proceedings of the 13th NTCIR Conference on Evaluation of Information Access Technologies, December 5-8, 2017 Tokyo Japan (2017)
15. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada, pp. 3104–3112 (2014)
16. Tian, Z., Yan, R., Mou, L., Song, Y., Feng, Y., Zhao, D.: How to make context more useful? an empirical study on context-aware neural conversational models. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 2: Short Papers, pp. 231–236 (2017)
17. Wu, Y., Li, Z., Wu, W., Zhou, M.: Response selection with topic clues for retrieval-based chatbots. Neurocomputing **316**, 251–261 (2018)
18. Wu, Y., Wu, W., Yang, D., Xu, C., Li, Z.: Neural response generation with dynamic vocabularies. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018 (2018)
19. Xing, C., Wu, W., Wu, Y., Liu, J., Huang, Y., Zhou, M., Ma, W.: Topic aware neural response generation. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA., pp. 3351–3357 (2017)

20. Yan, R., Song, Y., Wu, H.: Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016, pp. 55–64 (2016)
21. Yan, X., Guo, J., Lan, Y., Cheng, X.: A biterm topic model for short texts. In: 22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013, pp. 1445–1456 (2013)
22. Zhou, H., Huang, M., Zhang, T., Zhu, X., Liu, B.: Emotional chatting machine: Emotional conversation generation with internal and external memory. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018 (2018)
23. Zhou, H., Young, T., Huang, M., Zhao, H., Xu, J., Zhu, X.: Commonsense knowledge aware conversation generation with graph attention. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden., pp. 4623–4629 (2018)