Ranking Responses Oriented to Conversational Relevance in Chat-bots

Bowen Wu[1], **Baoxun Wang**[1], **Hui Xue**[2]
[1]Microsoft AI and Research Group, Beijing, China
[2]Microsoft Research, Beijing, China
{bowenwu, baoxwang, xuehui}@microsoft.com

Abstract

For automatic chatting systems, it is indeed a great challenge to reply the given query considering the conversation history, rather than based on the query only. This paper proposes a deep neural network to address the context-aware response ranking problem by end-to-end learning, so as to help to select conversationally relevant candidate. By combining the multi-column convolutional layer and the recurrent layer, our model is able to model the semantics of the utterance sequence by grasping the semantic clue within the conversation, on the basis of the effective representation for each sentence. Especially, the network utilizes attention pooling to further emphasis the importance of essential words in conversations, thus the representations of contexts tend to be more meaningful and the performance of candidate ranking is notably improved. Meanwhile, due to the adoption of attention pooling, it is possible to visualize the semantic clues. The experimental results on the large amount of conversation data from social media have shown that our approach is promising for quantifying the conversational relevance of responses, and indicated its good potential for building practical IR based chat-bots.

1 Introduction

There exist two query intentions in Intelligent Agents: the task completion oriented intention and the open-domain chat intention. As the applications of dialog systems, task completion oriented agents are designed to accomplish users' requirements in a few rounds of conversations. This kind of intentions reflect users' basic needs on the agents, thus studies on dialog systems have a longer history and achieved great process (Weizenbaum, 1966; Ferguson et al., 1996; Shawar and Atwell, 2007; Williams, 2010).

The open-domain chat intention, by contrast, represents users' communicating needs. Apparently, automatic chatting systems with good using experience will significantly attract people's interests, even make people form the habits of communicating with agents, hence it is possible to be a new platform for any task-oriented services to plug in (see Duer[1]). One challenge directly brought by open-domain Chat-bots is, indeed, user queries can be related to any topic in any possible forms, that is, it's NOT wise to transform chatting queries into slot-value sequences to further trace users' intentions within the conversation, as task-oriented dialog systems do.

The even more essential challenge chat-bots have to face is to guarantee the semantic and logic continuity of conversations, that is, a response from bots should be relevant with both the adjacent query and the corresponding short conversation history. Actually, such "context-aware" chatting ability is the critical feature of a human-like chat-bot, thus much attention has been paid on this task. The basic requirement for chat-bots is to semantically understand conversations like humans, which is abstracted as the conversation modeling problem. This paper discusses the approaches to addressing the context-aware chatting problem, by investigating and simulating the inner mechanism of human conversations.

Basically, two methodologies can be utilized to provide responses according to the given query, or more complicated, conversation history as discussed in this paper. The first method is to directly generate

[1]http://duer.baidu.com/

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 652–662, Osaka, Japan, December 11-17 2016.

Q0: 今天*天气好差*，天气**转凉**的**跨度好大**!
A0: 是不是*很恐怖*，今天已经**结冰了**。
Q1: 那你最近要怎么度过？
A1: 那就冬眠吧。
A2: 我一会去上班。

(a) Raw case in Chinese.

Q0: The *weather is so bad* today, the **temperature drops so much**!
A0: *Terrible*, isn't it? The temperature **goes down to freezing**.
Q1: Then, what are you going to do?
A1: Let's go to hibernate then.
A2: I'll go to work later.

(b) Translated to English.

Figure 1: A conversation example.

responses for a given query and its context (Ritter et al., 2011; Vinyals and Le, 2015; Shang et al., 2015), which provides an end-to-end solution for chat-bots. Despite its meaningful integrated architecture, it seems still a great challenge for generation based approaches to give responses with good readability and diversity. This problem can be directly addressed by another option, that is, finding proper methods to rank candidate responses selected from large amount of human dialog utterances by information retrieval (IR) method (Ji et al., 2014; Xian et al., 2016). Such Candidate Re-Ranking based solutions (Lowe et al., 2015; Sordoni et al., 2015; Hu et al., 2014) are of great value for building the practical chat-agents like Duer and XiaoIce[2], for which readability and diversity of responses are critical metrics.

For both generation and re-ranking approaches, indeed, their very basis is capturing the semantic clues within conversations, so as to select proper responses or generate them directly, and this procedure is generally named as "conversation modeling". As denoted by Grosz and Sidner (1986), the sequential utterances' structure, purposes and the state of focus of attention are the key components in a discourse. Correspondingly, to provide a conversationally reasonable response for a given session, the following abilities are needed for conversation modeling approaches: a) achieving the semantic representations of short sentences with the oral style; b) obtaining the focus of the entire dialog session; and c) selecting or generating responses based on the modeling of utterance sequences.

This paper aims to explore an integrated model framework to achieve the above goals, so as to find the context-aware responses from the candidates given by IR modules. Especially, our model will pay much attention to obtaining dialog focuses, that is, the model is designed to capture the semantic clues implicitly existing in human conversations. Such clues are always composed of phrases scattered in the utterances of conversations, and play a significant role in determining whether a given candidate is context-aware or not. Take the session in Figure 1 for example, some implicit clues flowing throughout the context (marked in **bold**) can be observed. Some of them like "temperature drops", "goes down to freezing", are more about the conversational topic, meanwhile, the words "weather", "temperature" stand for the key ingredient of the sentences. But, the keywords marked in *italic* maybe not helpful for judging A1 as a better response than A2. By contrary, we should task focus on some relatively meaningless phrases, such as "so much". So the clues detection is related with the end-to-end modeling task, and it is difficult and useless to treated this as a pre-processing.

This paper presents a convolution neural network (CNN) with attention pooling strategy to capture semantic clues within conversations by performing the sequential learning, so as to pick out context-aware replies based on the corresponding dialog history. According to the analysis on semantic relationships of the historical contexts, present-posted query and candidate responses, we propose to employ attention mechanism to model sentence upon convolutional layers, so as to provide meaningful semantic representations of contexts. After that, a Gated Recurrent Unit (GRU) based layer accomplishes the sequence modeling for response selection. Experiments with various structures of sentence and sequence modeling are conducted on the dataset from a Chinese Social Network Service (SNS), which has shown the good potential of our approach. Especially, due to the utilization of attention pooling, the obtained conversational clues can be visualized, which is very useful for the conversational state tracing and the interpretability of ranking results.

[2] http://www.msxiaoice.com/

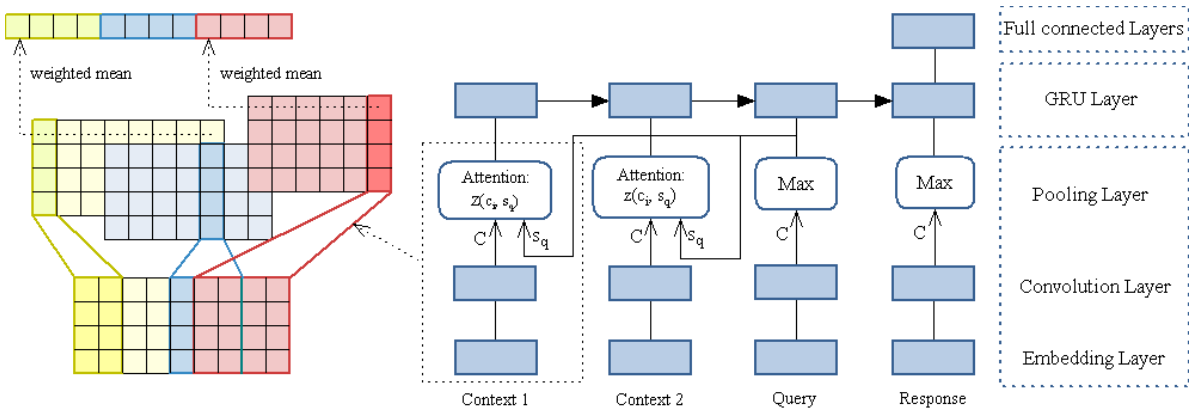


Figure 2: Architecture for modeling the conversation.

2 CNN with Attention Pooling for Quantifying Conversational Relevance

As mentioned in Section 1, the three abilities are needed for modeling open domain conversations, and our motivation is to design an integrated neural network framework to provide these abilities for response selection. This section will detail our approach that mainly takes Chinese characters as basic input elements[3], as well as the specially designed pooling strategy.

2.1 Model Architecture

As illustrated by Figure 2, the architecture of our model is composed of the following three modules:

Sentence Representing: As the essential part of our deep learning architecture, the Sentence Representing module aims to basically map short sentences into the real-valued semantic space. Moreover, the sentence modeling part in our work has to be able to locate the conversationally essential phrases of utterances, which can be jointly absorbed by the upper layer as the semantic clue for modeling conversations. For this purpose, we take the multi-width convolutional function and special designed pooling functions performing on the character-embedding layer, to build the sentence representing part.

CNN based sentence models have achieved success in some NLP tasks (Collobert et al., 2011; Kalchbrenner et al., 2014; Hu et al., 2014), especially, the recent character-level CNN (Kim et al., 2015; Zhang and LeCun, 2015) has even got some state-of-the-art results. Our sentence representing module continues this series of work, employing an one-dimensional valid convolutional layer over char embeddings.

Suppose there are n characters in a sentence, and let $x_i \in \mathbb{R}^k$ be the k-dimensional char vector corresponding to the i-th character, X present a $n \times k$-dimensional matrix made up of x_i, and $w \in \mathbb{R}^{1*m}$ is the weight of a convolutional filter (we'll use $m \in \{2, 3, 5\}$ to represent the bi-gram, tri-gram and 5-gram level abstraction). C stands for the output of this feature map, and represents the meaning of sub-phrases, each element vector c_i is computed by:

$$c_i = f(x_{i::i+m-1} \cdot w + b) \tag{1}$$

Where $b \in \mathbb{R}$ indicates the bias term and f stands for a non-linear function, e.g., the rectifier. Various potential features of the words or phrases are generated by multiple filters. For each of these candidate features will be screened by higher level pooling layers.

At present, the max pooling and average pooling are widely applied. In image modeling scenarios, max pooling can depict the texture better, while average pooling results represent more information about background (Boureau et al., 2010). The heuristics can be applied to NLP tasks similarly, that is, the whole meaning of sentences can be obtained by average strategy; on the other hand, the max pooling concentrates on the significant points relied on established tasks, and this is the reason for max pooling being utilized for solving many challenging text classification problems (Collobert et al., 2011; Kalchbrenner et al., 2014; Zhang and LeCun, 2015).

[3]Different from the European languages such as English, each Chinese character may keep special semantic independently.

In the conversation scenario, context-aware response selection basically relies on two major aspects: a) the relevance between the present query and the candidate response; and b) the additional background information provided by the dialog history. For quantifying the semantic relevance between the present query and candidate, the key phrases within them are playing the great role, and the irrelevant words should be ignored. By contrast, the phrases in a history utterance tend to act as a whole background, to supply and maintain a topic for the conversation. Consequently, as illustrated by Figure 2, we employ the max pooling for extracting the key points in the present query and the candidate for relevance judgment, and for the context sentences, the average pooling strategy is taken to introduce complete background.

Indeed, for each word in the dialog history (previous two sentences as shown by Figure 2), its contribution varies for selecting context-aware candidates, thus it is reasonable to give different weights to the words in the dialog history. For this purpose, in the sentence modeling module demonstrated in the left colorized part of Figure 2, we further present a new attention pooling strategy to learn the weights of each word, according to its contribution for determining whether a candidate is conversationally relevant. This attention pooling strategy will be detailed in Section 2.2.

Conversation modeling: Generated by the sentence modeling layer, the sentence vectors are adopted by a GRU layer for sequentially modeling the entire conversation. The motivation for selecting GRU is to naturally utilize its internal memory to process sequential inputs, and its performance is comparable with LSTM by controlling the gradient vanishing/exploding problems of ordinary RNNs (Bengio et al., 1994; Chung et al., 2014). To further investigate the balance between computational complexity and modeling ability, we also explored the RNN with identity initialized weights (iRNN for short) in practice as Mikolov et al. (2014) did.

Candidate Ranking: Given a sequence representation from conversation modeling, the candidate ranking module takes the full-connected layer to quantify the relevance of candidate responses. We employ the cross entropy as the point-wise ranking loss, and various ranking objective functions can be used to learn the parameters of the whole model.

2.2 Attention Pooling

As mentioned in the previous section, we wish to enhance the modeling of the context utterances for better understanding of the whole conversation, by employing the attention mechanism to learn the weights of words reasonably, meanwhile, it is possible to visualize the semantic clues in conversations according to the learnt weights. The attention strategies have been widely used in machine translation (MT) (Bahdanau et al., 2014; Meng et al., 2015; Li et al., 2015) and question answering (Weston et al., 2014; Hermann et al., 2015; Kumar et al., 2015). Especially for the Encoder-Decoder framework, the attention mechanism may introduce weighting functions of the encoding state and current decoding hidden state, so as to determine the elements that should be focused on.

This paper proposes a new pooling function with attention mechanism to model conversational contexts. Noticing that a posted utterance mainly pays attention to some specific points of the previous utterances in the same session, our pooling approach aims to emphasize such points while obtaining the whole meaning of sentences in the context. As mentioned in Section 2.1, the average pooling can cover the overall information, and our attention pooling tries to assign weights to the words and perform weighted averaging, to find the more important words or phrases in contexts.

The given c_i is the i-th combination of chars as defined previously, and $s_q \in \mathbb{R}^{(k(n-m+1))*1}$ presents the sentence embedding of the posted query upon the max pooling layer. The feature set $z(c_i, s_q)$ for weighting and the attended result a_i can be computed following:

$$z(c_i, s_q) = [c_i, s_q, c_i^T W^{(a)} s_q], \quad a_i = \frac{e^{(z(c_i,s_q)\cdot W^{(1)}+b^{(1)})}}{\sum_{j=0}^{n-m+1} e^{(z(c_j,s_q)\cdot W^{(1)}+b^{(1)})}} \tag{2}$$

Where $W^{(a)} \in \mathbb{R}^{(k(n-m+1))*k}$, $W^{(1)} \in \mathbb{R}^{(k(n-m+2)+1)*1}$, and $b^{(1)}$ is the bias term. The sentence vector

$s_j^{context}$ by the j-th convolutional filter is the weighted average of each c_i instead of ordinary mean:

$$s_j^{context} = \sum_{l=0}^{n-m+1} c_l^j \circ a_l^j \tag{3}$$

The $\circ$ indicates element-wise dot, c_l^j and a_l^j are computed by the j-th filter.

For similar purposes, there are some works take the attention strategy on RNN based dialogue generation (Yao et al., 2015; Shang et al., 2015), different from these works, the model described in this paper aims to apply the attention pooling upper the character level convolutional layer. Besides, Yin et al. (2015) applied an attention method upper convolutional layers to reflect the sub-phrases' interaction between sentence pairs, while we mainly focus on the sentence modeling about contexts in the conversations, and we proposed different attention function to model the relationship between context and query, other than matching units in two feature maps.

3 Experiments

Our proposed model is utilized on the response selecting task, that is, our goal is to distinguish the conversationally relevant responses from the irrelevant ones.

3.1 Dataset & Metrics

The dataset contains totally 1,025,000 sessions collected from the threads in a popular Chinese SNS, each session is composed of 4 utterances including a one-turn context, a present query and a context-aware response. Each sentence's character count varies from 3 to 50, with 10 as average. All the examples used in this paper are included in the dataset. For each conversation, we replace the response with another one randomly sampled from the corpus as the negative sample like (Hu et al., 2014; Lowe et al., 2015; Al-Rfou et al., 2016). This operation repeats 4 times, and we duplicate the real conversations 4 times as positive samples. For all the experiments, we split our dataset into training, validation and test sets, with 8,000,000, 100,000 and 100,000 conversations respectively.

Except evaluating the classified performance by accuracy, we introduce **1 in t P@k** to evaluate the ranking ability with $t-1$ negative cases, where P@k denotes the precision at top k.

3.2 Competitor Models & Parameter settings

The baseline approaches taken by our work can be basically categorized into two groups: the classic methods include the Logistic Regression (LR) models trained on Tri-Gram based TF-IDF features or LDA (Blei et al., 2003) based distributed representations of sentences. Besides, several neural network combinations of different components' implementation are adopted in our experiments, whose general frameworks are the same with the one illustrated in Figure 2, and their details are given as follows:

- GRU+MLP: This model takes GRU to model sentences, which is different from our CNN based sentence modeling layers. Above that, the multi-layer perceptron (MLP) is used to model the conversations without consideration about the sequential characteristic of conversations;

- GRU+iRNN/GRU: In these models, iRNN or GRU takes the position of conversation modeling module, and the sentences modeling part still employs GRU;

- Attention GRU+iRNN: Employing GRU with attention for modeling the sentences in contexts and iRNN for sequence modeling;

- CNN+iRNN: This model takes iRNN for conversation modeling, and in the CNN based sentence representation module, several pooling strategies are tried, including max pooling, mean pooling, and their mixture, to replace our attentional average pooling in the architecture in Figure 2;

Group	Model	Accuracy	1 in 2 P@1	1 in 5 P@1	1 in 5 P@2
#1	Random	50.0%	50.0%	20.0%	40.0%
	TF-IDF+LR	53.2%	58.4%	24.2%	43.0%
	LDA+LR	59.7%	66.7%	35.6%	52.8%
#2	GRU+MLP	65.8%	72.0%	43.4%	67.6%
#3	GRU+iRNN	72.5%	80.3%	54.7%	78.5%
	bi-GRU+iRNN	**73.6%**	**81.7%**	**55.7%**	**80.2%**
	GRU+GRU	**75.8%**	**84.5%**	**63.1%**	**83.3%**
	Attention GRU+iRNN	70.3%	78.0%	51.5%	75.0%
#4	Max CNN+iRNN	73.1%	81.0%	55.1%	80.0%
	Mean CNN+iRNN	73.5%	81.7%	55.8%	80.8%
	Mix CNN+iRNN	74.2%	82.9%	56.9%	81.9%
	Attention CNN+iRNN	**75.7%**	**84.3%**	**60.5%**	**83.5%**
	Attention CNN+GRU	**78.6%**	**87.0%**	**65.1%**	**86.1%**

Table 1: Comparison of different approaches on the context-aware candidate selection task.

The implementation with online learning for LDA (Hoffman et al., 2010) is used for our experiments, with α and β fixed at 0.01 and the number of topics $K = 400$. In all the neural network based experiments, we initialize the learning rate with 0.005, and the network is trained with the Adam update rule (Kingma and Ba, 2014). Early stopping (Giles, 2001) and Dropout (Hinton et al., 2012) are taken to prevent overfitting. As recommend by Krizhevsky et al. (2012), we utilize ReLU as the non-linear active function of convolutional and full-connected layers, and $tanh$ is used for the hidden states of GRU. The dimension of character embedding is 100 for all the NN models. For CNN based sentence modeling layers, the widths of the filter windows are set to 2, 3 and 5 in parallel, and the pooling window covers all the element after convolutional function. The GRU based sentence modeling module holds a 100-unit hidden layer. For conversation modeling layers, the size of the hidden states of iRNN and GRU is set to 300. Finally, the size of the hidden layer of MLP is 50.

3.3 Results & Analysis

Table 1 details the results, and groups them into four categories for the following analysis perspectives:
(a) traditional methods vs. neural networks based ones for modeling short sentences with oral style;
(b) GRU vs. CNN on sentence representation;
(c) aligning sentence embeddings vs. modeling sentence sequences for conversation understanding;
(d) separately modeling sentences vs. sentence representation with attention mechanism;
(e) GRU with attention vs. CNN with attention for sentence representation.

From the results in Table 1, it can be observed that all the models adopting neural network components have notably outperformed TFIDF-LR and LDA-LR. This phenomenon reflects the difficulty of modeling the short sentences with oral style, since the information introduced by pure lexical features introduce is very limited for such text samples. By contrast, both GRU and CNN have the ability of catching the richer semantic information in short texts, according to the layer-by-layer learning upon the distributed character embeddings. Thus, the comparison of aspect (a) shows NN based sentence models are more suitable for conversation utterances.

Further, by comparing GRU+iRNN with CNN+iRNN, aspect (b) tries to figure out which deep learning architecture works better as the sentence modeling module, and our observation is CNN outperforms GRU on the whole task. We ascribe this result to the information bias of sentence embeddings generated by GRU, that is, GRU tends to pay more attention to the words in the end of a sentence. However, for the task discussed by this paper, complete semantics provide more help to context-aware candidate selection as discussed in Section 2.1. The limited improvement of CNN with max pooling also supports our inference. This problem has been partially solved by introducing bi-direction GRU (see bi-GRU+iRNN), it can also be seen that the special defined mixture pooling strategy (Mix CNN+iRNN) can achieve

context	The weather is so bad today, the temperature dips so much! Terrifying, isn't it? The temperature goes down to freezing.				
query	Then, what are you going to do?				
Mix CNN+iRNN	Attention CNN+iRNN	Label	Rank Label	No.	Response
0.782	**0.826**	1	1	#1_1	Let's go to hibernate then.
0.573	0.422	0	2	#1_2	I'll go to work later.
0.150	0.029	0	3	#1_3	How to prove it?
context	What's the point of keeping my phone if it can't connect to Wifi anymore? So what are we waiting for? Buy a new one!				
query	You sponsor me.				
0.808	**0.940**	1	1	#2_1	No phone, no money!
0.840	0.824	1	2	#2_2	How to sponsor?
0.384	0.226	0	3	#2_3	Curiously, this's across both 3G and Wifi.

Table 2: Samples of the response selection. Sentences are translated to English for better understanding.

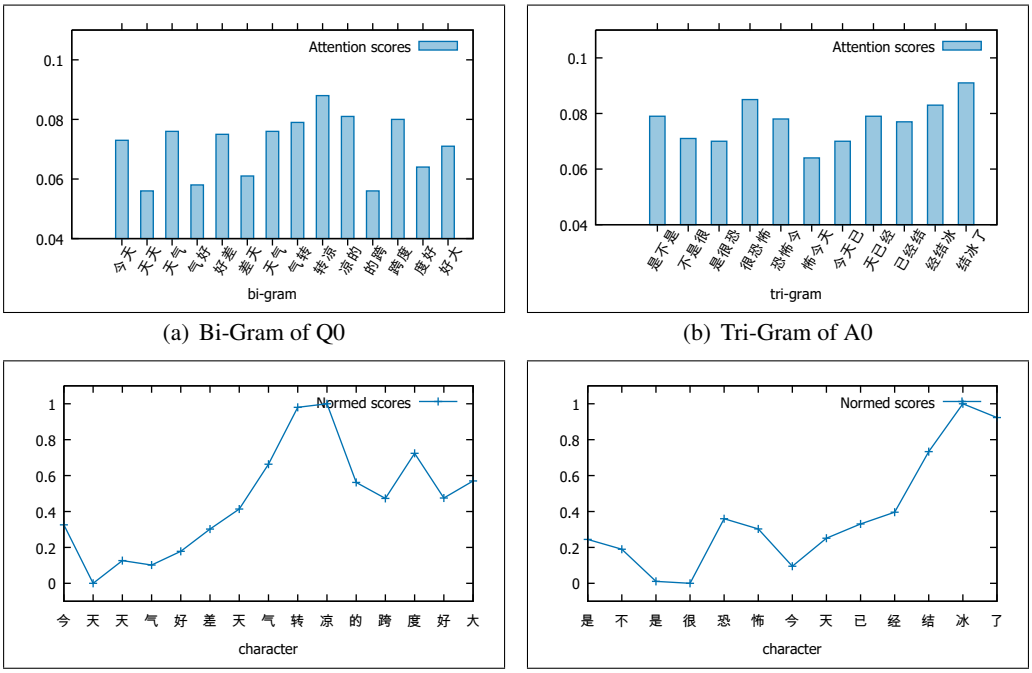
more competitive results, because the advantages of different pooling methods have been integrated for complete semantic representation.

We can easily observe the huge gap between the performances of models in group #2 and group #3. All of these methods take GRU as the sentence modeling layer, but the ones in group 3 adopt RNN based layers for conversation modeling. Since the conversation modeling task is naturally a sequential modeling problem, it is reasonable that models with GRU components achieve better results, which is the motivation of aspect (c). Another observation is iRNN performs fairly well as the conversation modeling layer, with good potential for practical usage. Besides, the comparisons suggest that GRU is indeed more powerful for modeling conversations.

As shown in the results of group #3 and #4, the attention pooling is helpful to improve the precisions, especially on 1 in 5 P@1, which meets the expectation of aspect (d). Nevertheless, when considering aspect(e), it is noticed that GRU with attention (attention GRU+iRNN) gets unsatisfying performance comparing with the ones without attention function. This observation is different from the general impression, since quite a number of studies adopting attention mechanism have good results (Bahdanau et al., 2014; Yao et al., 2015; Hermann et al., 2015; Kumar et al., 2015). We attribute the performance gap to the character-level inputs. In detail, since the attention function is applied on each hidden state, which mainly contains the information of the current input, despite a small amount of previous information involved. Meanwhile, a single Chinese character keeps very limited semantic information, thus the information obtained by the attention function of GRU is incomplete, reflecting some single characters in fact. By contrast, the convolution layer can extract the combinations of characters indicating words or even phrases, and the attention function performed upon such combinations is possible to figure out the important segments with complete semantics, for the upper layer to understand the whole sequence. This is the main reason for our proposed "Attention CNN+GRU" model finally get the best results.

3.4 Case study

To get a better intuition for what the model and attention pooling is learning, we give some cases to illustrate the details. Table 2 gives two groups of query-response pairs, with the predicted scores by Mix CNN+iRNN (MCNN) and Attention CNN+iRNN (ACNN) models. **Label=1** indicates the suitable response to the given context, and the **Rank Label** reflects the candidates' recommendation degrees. It can be seen that scores of both models are aligned with the overall ranking trend, which reflects the models having the ability to quantify the conversational relevance reasonably. It should be noted that the scores given by ACNN are more closed to the labels. More specifically, all the predictions of ACNN are correct, while MCNN makes some incorrect decisions on #1_2 and #2_2. Different from other candidates, the sentences (#1_2 and #2_2) are very sensitive to contexts, in other words, they are natural to answer the corresponding query without considering the contexts. #1_2 has wandered off topic, by contrary, #2_2 can also be a suitable response even it seems #2_1 have better maintenance of the information in the conversation flow. The results of ACNN reflect these phenomena, as the ranking scores express the right disposition and offer higher scores to the more appropriate responses. Besides, the gaps within the



(a) Bi-Gram of Q0

(b) Tri-Gram of A0

(c) Relative attention on chars of Q0

(d) Relative attention on chars of A0

Figure 3: Attention values on the Bi-Gram of Q0 and Tri-Gram of A0 are detailed in sub-figure (a) and (b), where y-axis is the attention value. Sub-figure (c) and (d) give the focus on each char.

groups of ACNN are larger than those scores by MCNN. All these enhancement of the model's ability can be ascribed to the capability of differentiating the matching degree not only correspond to the posted query but also corresponding with to the whole session, obtained by attention pooling to lay particular emphasis on the phrases that the previous conversation focused on.

In order to further illustrate the effect of attention pooling, Figure 3 details the distribution of probabilities given by the attention function to the phrases of the case in Figure 1. Actually, the extra advantage of our framework is that we can locate the key information for candidate selecting, by visualizing the attention weights of phrases and performing proper transforming on them. In detail, we firstly visualize the pooling weights for each character combination in each convolutional kernel as shown by Figure 3(a)-(b), then we assign the weights averaged by the frequencies of the characters in each kernel, and get the curves in Figure 3(c)-(d). According to this operation, we can clearly see which positions in the context considered to be more important when given a query. While it can be seen that, obvious higher weights appeared along the positions of significant words and phrases (marked in **bold**) in Figure 1. Another observation from the histograms is the overall scores of the essential words and phrases are higher than the other char-combinations, which indicates the sentence embedding is mostly draw from the meaningful words. This group of results shows the attention pooling, rather than simple mean pooling, are effective to draw focus on the words and phrases composing the semantic clue in a given conversation.

4 Related Work

Before open-domain chat agents, the task-completion oriented dialog system has been a subject of study for a long time, and most of these studies pay attention to particular vertical domains. Such as *ELIZA* based on simple text parsing rules (Weizenbaum, 1966). Ferguson et al. (1996) built a rule-based system to solve problems in transportation domain; Shawar and Atwell (2007) leverage answer template in generating the *ALICE*; Williams (2010) focus on tracing pre-defined dialogue state. These systems rely on pre-designed rules or templates, which can be hardly generalized on open domain chat-style robots.

Until recently, some works such as (Ritter et al., 2011) demonstrate that the sentences can be generated corresponding to a given post or context using MT techniques. Since the encoder-decoder based Recurrent Neural Networks (RNN) outperforms other methods on MT tasks in the past two years (Bahdanau

et al., 2014; Sutskever et al., 2014), several approaches are directly applied on the conversation modeling task by concatenating the context that modeled by one recurrent encoder (Vinyals and Le, 2015; Shang et al., 2015). Yao et al. (2015) and Serban et al. (2015) model the sentences of context separately by an encoder, and address the sequential embeddings by cumulative hidden units. The main drawback of these approaches is they can't guarantee the readability and variety of generated sentences.

By contrast, the response ranking strategies can avoid the problems caused by direct generation, since this methodology tries to pick up reasonable responses from the human-generated sentences. Ji et al. (2014) proposed an IR approach to generate candidates, and rank them with many kinds of features such as MT, keywords, similarity, etc; Sordoni et al. (2015) and Luan et al. (2016) directly utilize the generating loss of the response for ranking, with the adjusted RNN based encoder-decoder framework. Generally, the architectures introducing CNN or RNN to learn representations of sentences and modeling the relevance of context and candidate response on hyper layers, tend to achieve state-of-the-art performance (Hu et al., 2014; Lowe et al., 2015).

5 Conclusion

In this paper, we have presented a deep learning architecture to quantify the conversational relevance of responses for candidate ranking. The contributions of this paper can be summarized as follows: a) According to the investigation on the role of contexts in conversations, this paper proposes the attention pooling to provide more reasonable context representations, by taking the phrases' different contributions to the semantic clue into consideration. b) We have combined the multi-column convolutional layer and the GRU based layer to build the candidate ranking model, so as to take the advantages of both CNN on sentence modeling and RNN on sequence modeling. c) The proposed model enables the visualization of the achieved essential phrases, and our analysis on them shows the importance of capturing semantic clues for finding conversationally relevant responses.

Acknowledgements

We thank the anonymous reviews, along with Deyuan Zhang and Zhen Xu for their valuable comments and suggestions that helped to improve the quality of this paper.

References

Rami Al-Rfou, Marc Pickett, Javier Snaider, Yun-hsuan Sung, Brian Strope, and Ray Kurzweil. 2016. Conversational contextual cues: The case of personalization and history for response ranking. *arXiv preprint arXiv:1606.00372*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

Y-Lan Boureau, Francis Bach, Yann LeCun, and Jean Ponce. 2010. Learning mid-level features for recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2559–2566. IEEE.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

George Ferguson, James F Allen, Bradford W Miller, et al. 1996. Trains-95: Towards a mixed-initiative planning assistant. In *AIPS*, pages 70–77.

Rich Caruana Steve Lawrence Lee Giles. 2001. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *Advances in Neural Information Processing Systems 13: Proceedings of the 2000 Conference*, volume 13, page 402. MIT Press.

Barbara J Grosz and Candace L Sidner. 1986. Attention, intentions, and the structure of discourse. *Computational linguistics*, 12(3):175–204.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Matthew Hoffman, Francis R Bach, and David M Blei. 2010. Online learning for latent dirichlet allocation. In *advances in neural information processing systems*, pages 856–864.

Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *Advances in Neural Information Processing Systems*, pages 2042–2050.

Zongcheng Ji, Zhengdong Lu, and Hang Li. 2014. An information retrieval approach to short text conversation. *arXiv preprint arXiv:1408.6988*.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, June.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2015. Character-aware neural language models. *arXiv preprint arXiv:1508.06615*.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Ankit Kumar, Ozan Irsoy, Jonathan Su, James Bradbury, Robert English, Brian Pierce, Peter Ondruska, Ishaan Gulrajani, and Richard Socher. 2015. Ask me anything: Dynamic memory networks for natural language processing. *arXiv preprint arXiv:1506.07285*.

Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. 2015. A hierarchical neural autoencoder for paragraphs and documents. *arXiv preprint arXiv:1506.01057*.

Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. *arXiv preprint arXiv:1506.08909*.

Yi Luan, Yangfeng Ji, and Mari Ostendorf. 2016. Lstm based conversation models. *arXiv preprint arXiv:1603.09457*.

Fandong Meng, Zhengdong Lu, Mingxuan Wang, Hang Li, Wenbin Jiang, and Qun Liu. 2015. Encoding source language with convolutional neural network for machine translation. *arXiv preprint arXiv:1503.01838*.

Tomas Mikolov, Armand Joulin, Sumit Chopra, Michael Mathieu, and Marc'Aurelio Ranzato. 2014. Learning longer memory in recurrent neural networks. *arXiv preprint arXiv:1412.7753*.

Alan Ritter, Colin Cherry, and William B Dolan. 2011. Data-driven response generation in social media. In *Proceedings of the conference on empirical methods in natural language processing*, pages 583–593. Association for Computational Linguistics.

Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2015. Building end-to-end dialogue systems using generative hierarchical neural network models. *arXiv preprint arXiv:1507.04808*.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. *arXiv preprint arXiv:1503.02364*.

Bayan Abu Shawar and Eric Atwell. 2007. Chatbots: are they really useful? In *LDV Forum*, volume 22, pages 29–49.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. *arXiv preprint arXiv:1506.06714*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.

Joseph Weizenbaum. 1966. Elizaa computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. *arXiv preprint arXiv:1410.3916*.

Jason D Williams. 2010. Incremental partition recombination for efficient tracking of multiple dialog states. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 5382–5385. IEEE.

Li Xian, Mou Lili, Yan Rui, and Zhang Ming. 2016. Stalematebreaker: A proactive content-introducing approach to automatic human-computer conversation. *arXiv preprint arXiv:1604.04358*.

Kaisheng Yao, Geoffrey Zweig, and Baolin Peng. 2015. Attention with intention for a neural network conversation model. *arXiv preprint arXiv:1510.08565*.

Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2015. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *arXiv preprint arXiv:1512.05193*.

Xiang Zhang and Yann LeCun. 2015. Text understanding from scratch. *arXiv preprint arXiv:1502.01710*.