Context-to-Session Matching: Utilizing Whole Session for Response Selection in Information-Seeking Dialogue Systems

Zhenxin Fu*
WICT, Peking University
fuzhenxin@pku.edu.cn

Shaobo Cui
DAMO Academy, Alibaba Group
yuanchun.csb@alibaba-inc.com

Mingyue Shang
Northwestern University
mingyueshang95@gmail.com

Feng Ji
DAMO Academy, Alibaba Group
zhongxiu.jf@alibaba-inc.com

Dongyan Zhao
WICT, Peking University
zhaodongyan@pku.edu.cn

Haiqing Chen
DAMO Academy, Alibaba Group
haiqing.chenhq@alibaba-inc.com

Rui Yan†
[1] WICT, Peking University
[2] Young Fellow in Beijing Academy of
Artificial Intelligence
ruiyan@pku.edu.cn

ABSTRACT

We study the retrieval-based multi-turn information-seeking dialogue systems, which are widely used in many scenarios. Most of the previous works select the response according to the matching degree between the query's context and the candidate responses. Though great progress has been made, existing works ignore the contexts of the responses, which could provide rich information for selecting the most appropriate response. The more similar the query's context and certain response's context are, the more likely they are to indicate the same question, and thus, the more likely this response is to answer the query. In this paper, we consider the response and its context as a whole **session** and explore the task of matching the query's context with the sessions. More specifically, we propose to match between the query's context and response's context and integrate the context-to-context matching with context-to-response matching. Experiment results prove that our proposed context-to-session method outperforms the strong baselines significantly.

CCS CONCEPTS

• **Computing methodologies → Discourse, dialogue and pragmatics**; • **Information systems → Question answering**.

KEYWORDS

Response selection, Text matching, Graph attention network

ACM Reference Format:
Zhenxin Fu, Shaobo Cui, Mingyue Shang, Feng Ji, Dongyan Zhao, Haiqing Chen, and Rui Yan. 2020. Context-to-Session Matching: Utilizing Whole Session for Response Selection in Information-Seeking Dialogue Systems. In *26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20), August 23–27, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394486.3403211

1 INTRODUCTION

Information-seeking dialogue system [7, 24, 27] aims at satisfying the information needs of users through conversations. It has attracted increasing attention due to its wide range of application scenarios, such as customer services in online shopping and personal digital assistant [19, 30]. Different from the generation-based open-domain dialogue systems [2, 20], which generate response word by word, the retrieval-based information-seeking dialogue systems select the response from a candidate set. The quality of the selected response is a determining factor for the information-seeking dialogue systems. Within the dialogue interactions with users, the system takes the query's context, which consists of the previous utterances and the current query, as input to retrieve the most appropriate response from the candidate response set. In this paper, we improve the performance of the information-seeking dialogue systems by utilizing the session information.

The canonical retrieval methods for selecting the superior responses require two steps (Figure 1): 1) Coarse-grained candidate set construction: constructing a rough candidate set from the whole dialog sessions. 2) Fine-grained selection: selecting the best response from the candidate set. To be more specific, the first step is a coarse-grained relevance searching process between the query's context and the response's context (response's history as in Table 1), which is usually accomplished by TF-IDF based methods for efficiency [37]. The second step is a fine-grained reranking process by computing the matching degree between the query's context and the response where most of the existing works focused on [9, 21, 22, 26, 38].

Encouraged by the success of neural networks in natural language processing tasks, in recent years, researchers have adopted

*This work was done when Zhenxin Fu was an intern at Alibaba Group.
†Corresponding author: Rui Yan (ruiyan@pku.edu.cn)
[0] WICT is the abbreviation for "Wangxuan Institute of Computer Technology".

KDD '20, August 23–27, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-7998-4/20/08…$15.00
https://doi.org/10.1145/3394486.3403211

Table 1: An example for the query's context and the corresponding candidate session which is composed of the response and response's context. CUST stands for customer. STAFF represents the customer service staff.

	Query's Context
CUST:	Which kind of express do you use?
STAFF:	EMS.
CUST:	Can the package be handed over to me this week?
	Response's Context
CUST:	Is there additional fee for package delivering?
STAFF:	Nothing, Sir!
CUST:	When will the package be delivered?
	Candidate Response
STAFF:	The product will be delivered within three days.

neural networks to perform fine-grained matching in the retrieval-based dialogue systems. One line of research focuses on the context-to-response matching (CRM) methods (the upper right part in Figure 1), modeling the relationship between the query's context and the response [3, 25, 40, 41]. However, these methods ignore the response's context, which may benefit the response selection tasks. It is intuitional that if the response's context is similar to the query's context, the corresponding response has a high probability to answer the given query. Table 1 shows an example for better understanding. The query's context and the response's context in Table 1 are similar enough to conclude that the corresponding response is appropriate even if not seeing the response.

Inspired by the aforementioned observation, we investigate how to leverage the response's contexts to help the response selection task. We take the whole dialogue session into consideration and introduce a context-to-session matching (CSM) model (the lower right part in Figure 1). To be specific, we propose to match the query's context with the response's context and response in the candidate session respectively: context-to-context matching (CCM) and context-to-response matching (CRM), and then integrate the two kinds of matching representations to get the final matching score. The response in the session with the highest score will be selected as the final response.

To build a context-to-session matching (CSM) model, two main issues need to be addressed. The **first issue** is how to match the query's context with the response's context appropriately. Since the query's context and response's context are both long sequences of dialogue utterances, and utterances at different positions play different roles, matching between query's context and response's context is quite challenging. The **second issue** is how to effectively integrate context-to-context matching and context-to-response matching. In this paper, we propose to use graph attention network (GAT) [32] with role-aware attention aggregation and a gating mechanism to solve these two issues. GAT models the relationships between the utterance pairs while the role-aware attention aggregates the output of the GAT. The gating mechanism dynamically decides the weight between CCM and CRM.

We compare our model with other competitive models. Experimental results show that our model achieves the best performance, which verifies that CSM could benefit response selection. The code and data are released. [1]

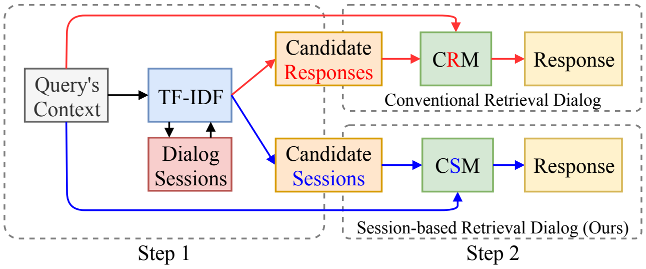

Figure 1: Comparison between conventional retrieval dialogue system (black and red lines) and our proposed session-based retrieval dialogue system (black and blue lines).

In summary, the contributions of our paper are:

- To our best knowledge, our paper is the first study that attempts to enhance response selection in the context-to-session matching fashion (CSM), which presents a new line of research in response selection in information-seeking dialogue systems.
- We design a simple and effective model to perform CSM, which models the query's context relationships with the response's context and the response separately.
- For the many-to-many matching problem involved in the context-to-context matching part of our CSM model, we innovatively use the graph attention network to capture the relationships among the utterance pairs and the role-aware attention mechanism to aggregate these utterance pairs' representations.

The rest of this paper is organized as follows: we review related work on retrieval-based dialogue systems and text matching in Section 2. The task formulation and our model structure is presented in Section 3. Section 4 is about the dataset construction, experiment settings, and baseline models. Detailed experimental results and analysis are elaborated in Section 5, after which we conclude in Section 6.

2 RELATED WORK

This paper explores to utilize knowledge of the whole session to boost the response selection task.

2.1 Retrieval-based Dialogue Systems

Most of the existing retrieval-based dialogue systems focus on the context-to-response matching that matches the query's context and the response directly. Zhou et al. [44] feed the query's context and response into a Recurrent Neural Network (RNN) respectively and measure the context-to-response relevance by the last hidden state of the RNN. Wu et al. [37] propose the Sequential Matching Network (SMN) which models the relationship between each utterance in the query's context and the response by a cross-attention matrix. Zhou et al. [45] propose the Deep Attention Matching (DAM)

[1] https://github.com/fuzhenxin/Context-to-session-Matching-KDD2020

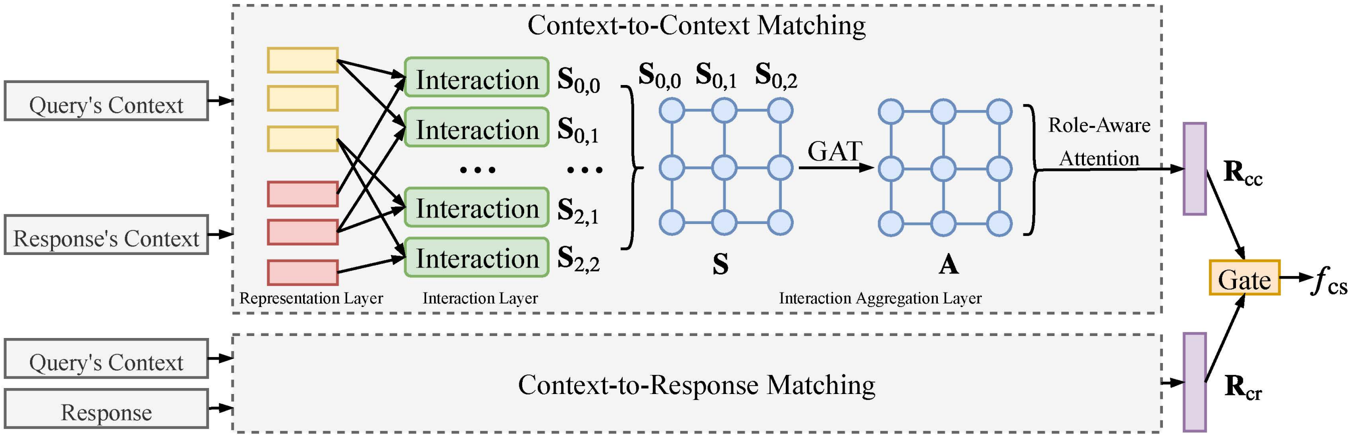

Figure 2: Context-to-Session Matching (CSM). As an example, the query's context and response's context both only contain three utterances. The context-to-context matching module is composed of the representation layer, interaction layer, and interaction aggregation layer. $\mathrm{S}_{i,j}$, $\mathrm{R_{cc}}$, and $\mathrm{R_{cr}}$ are all vectors.

model that encodes the query's context and response through self-attention. Cross-attention and 3-D convolution are also applied to predict the matching degree in DAM. Tao et al. [28] propose the Multi-Representation Fusion Network (MFRN) which considers the query-to-response matching with multiple kinds of representations. MFRN encodes queries and responses from the view of words, n-grams, and sub-sequences of utterances. Then it studies how to fuse them in a deep neural network architecture. Tao et al. [29] propose the Interaction over Interaction (IoI) model to make utterance-response interaction go deep by stacking multiple interaction blocks. Different from these methods that only consider the context-to-response matching and ignore the corresponding context of the response, we investigate the context-to-session matching problem where the response's context is also considered.

2.2 Text Matching

The key point of the response selection task is text matching. Along with the development of the neural networks, more and more researchers employ RNN or Convolution Neural Network (CNN) for the text matching tasks. These methods get the text representations and then manipulate those representations using techniques like cross-attention mechanism [10]. Previous text matching works mainly investigate the matching between two sentences (one-to-one matching) or the matching between a sequence of several sentences and one sentence (many-to-one matching), including Natural Language Inference [4], Paraphrase Identification [12], Context-response matching [37], and Information Retrieval [15]. However, seldom have they explored the matching between two sequences of sentences (many-to-many matching), which is a more challenging task. In this paper, we propose to match between the query's context and response's context, a many-to-many matching problem.

3 MODEL

In this section, we firstly introduce the task formulation in Section 3.1. Then, we present the model overview in Section 3.2 for better understanding. The attentive module adopted in our model is described in Section 3.3. The context-to-context matching (CCM) component and the context-to-response matching(CRM) component are described in Section 3.4 and Section 3.5 respectively. Finally, the integration component for integrating the CCM and CRM representation is presented in Section 3.6.

3.1 Task Formulation

We assume a training set of size N, denoted as $D = \{(C_i^q, C_i^r, R_i, l_i)\}_{i=1}^N$, where C_i^q is the i-th query's context. C_i^r and R_i are the response's context and response in the i-th candidate session respectively, and $l_i \in \{0, 1\}$ is the label denotes whether (C_i^r, R_i) is the correct session to match with C_i^q. To be specific, the query's context and response's context are both sequences of utterances (sentences), which can be formulated as $C^q = \{S_1^q, \cdots, S_{T_q}^q\}$ and $C^r = \{S_1^r, \cdots, S_{T_r}^r\}$. T_q and T_r are the corresponding max turn number. Given a query's context C_i^q and a candidate session (C_i^r, R_i), the model is trained to predict l_i correctly.

3.2 Model Overview

Our model is designed for the context-to-session matching, which is shown in Figure 2. Concretely, we divide the context-to-session matching into context-to-context matching (CCM) and context-to-response matching (CRM) to capture the session information from two different perspectives: 1) **CCM**: measuring whether query's context and response's context are asking the same question. We use graph attention network and role-aware attention aggregation to obtain the context-to-context matching representation (Section 3.4). 2) **CRM**: modeling the dialog pattern between the query's context and the response. In this paper, we use the IoI model [29] as the context-to-response matching component, which has shown a good performance (Section 3.5). The CCM representation and the CRM representation are then combined through a gating mechanism, see in Section 3.6.

In the following subsections, we first discuss the structure of the attentive module which is a basic component of our method, and then introduce the other parts in detail.

3.3 Preliminary: Attentive Module

Inspired by the success of Transformer [6, 31], we adopt the attentive module following previous work [29, 42, 45] to learn the utterance representation. The attentive module is a variant of the encoder of the Transformer with single-head attention. The attentive module is composed of a single-head self-attention sublayer and a position-wise fully connected feed-forward sub-layer. A residual connection [13] is employed around each of the two sub-layers, followed by layer normalization [18]. It is abstracted as $f_{\text{att}}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \in \mathbb{R}^{t \times d_k}$, where $\mathbf{Q} \in \mathbb{R}^{t \times d_k}$, $\mathbf{K} \in \mathbb{R}^{t \times d_k}$ and $\mathbf{V} \in \mathbb{R}^{t \times d_k}$ are matrices representing the query input, the key input, and the value input respectively. t is the sentence length and d_k is the dimension of the word embedding in this paper.

The detailed implementation of the attentive module: an attention function is first applied to map the query set $\mathbf{Q}$ and key-value set pair $\{\mathbf{K}, \mathbf{V}\}$ to an output. The output is the weighted sum of the values where the weight assigned to each value is calculated by the relevance between the query and the corresponding key. In this paper, we adopt the scaled dot-product attention mechanism following Vaswani et al. [31], which is formulated as:

$$\text{Att}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Softmax}(\frac{\mathbf{Q}\mathbf{K}^{\mathsf{T}}}{\sqrt{d_k}})\mathbf{V} \quad (1)$$

After the attention layer, a residual connection [13] with summation and layer normalization [18] is applied on $\text{Att}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ to get an intermediate representation $\mathbf{X}$ in order to obtain better fused representation:

$$\mathbf{X} = f_{\text{norm}}(\mathbf{Q} + \text{Att}(\mathbf{Q}, \mathbf{K}, \mathbf{V})) \quad (2)$$

Then, a position-wise feed-forward network (FFN) is applied to each position of the intermediate representation separately and identically. The FFN with ReLU [17] activation is denoted as:

$$\mathbf{X}'_i = \max(0, \mathbf{X}_i \mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2 \quad (3)$$

where $\mathbf{W}_1, \mathbf{b}_1, \mathbf{W}_2$, and $\mathbf{b}_2$ are learnable parameters. $\mathbf{X}_i$ and $\mathbf{X}'_i$ are the i-th row of $\mathbf{X}$ and $\mathbf{X}'$ respectively.

Finally, the summation, applied with layer normalization, of intermediate representation $\mathbf{X}$ and $\mathbf{X}'$ is the final output:

$$f_{\text{att}}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = f_{\text{norm}}(\mathbf{X} + \mathbf{X}') \quad (4)$$

3.4 Context-to-Context Matching (CCM)

The context-to-context matching module is composed of a representation layer (Section 3.4.1), an interaction layer (Section 3.4.2), and an interaction aggregation layer (Section 3.4.3). The representation layer is designed to obtain an utterance level representation. The interaction layer is for securing the interaction matching representation between two utterances by cross-attention mechanism which is followed by a convolution layer. The interaction aggregation layer uses the graph attention network to model the relationships among the utterance pairs, and a role-aware attention mechanism is introduced to aggregate the interaction matching representations.

3.4.1 CCM: Representation Layer. The representation layer is designed to get self-attentive utterance representation. Figure 3 shows the representation layer and the following interaction layer. We employ the aforementioned attentive module to encode the input utterance S into the utterance representation $\mathbf{U} \in \mathbb{R}^{t \times d_k}$. Take the i-th utterance in the query's context as an example. First, the utterance S_i^q is transformed into word embedding representation $\mathbf{E}_i^q$ by looking up the word embedding table. Following Vaswani et al. [31], position embedding is added to the word embedding to identify the absolute position of the tokens in the sequence. Then we get the position-aware word representation $\widetilde{\mathbf{E}}_i^q \in \mathbb{R}^{t \times d_k}$, which is fed into the attentive module to get the utterance representation $\mathbf{U}_i^q$:

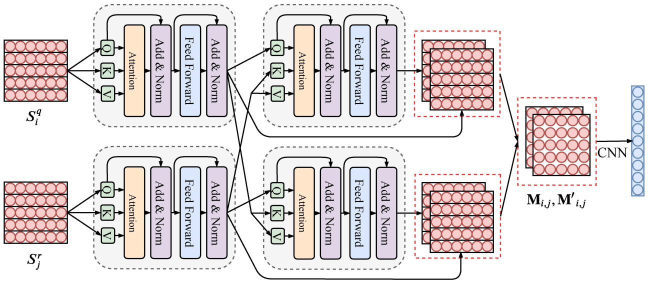

Figure 3: The representation layer and the interaction layer in the context-to-context matching module. S_i^q is the i-th utterance of the query's context and S_j^r is the j-th utterance of the response's context.

$$\mathbf{U}_i^q = f_{\text{att}}(\widetilde{\mathbf{E}}_i^q, \widetilde{\mathbf{E}}_i^q, \widetilde{\mathbf{E}}_i^q) \quad (5)$$

The representation layer can be seen as self-attention within the utterance, which provides integration of the local information. To obtain deep utterance representation, we adopt a multi-layer attentive module rather than a single layer.

3.4.2 CCM: Interaction Layer. After getting the utterance representation, following Zhou et al. [45], we employ the cross-attention mechanism which has shown great success in text matching [4, 11] to measure the relevance between utterances. The following two kinds of cross-attention mechanisms are adopted: (1). the cross attention between utterances representation from the representation layers: $\mathbf{M}_{i,j}$. (2) the cross attention between attentive representation of utterances:$\mathbf{M'}_{i,j}$.

(1) $\mathbf{U}_i^q$ and $\mathbf{U}_j^r$ are the utterance representations of the i-th utterance in the query's context and the j-th utterance in the response's context. For the cross-attention matrix $\mathbf{M}_{i,j} \in \mathbb{R}^{t \times t}$ between $\mathbf{U}_i^q$ and $\mathbf{U}_j^r$, each element of it is calculated by Equation 6, where $\{\mathbf{U}_i^q\}_a$ is the a-th row in $\mathbf{U}_i^q$, and $\{\mathbf{U}_j^r\}_b$ is the b-th row in $\mathbf{U}_j^r$.

$$\mathbf{M}_{i,j}^{a,b} = \{\mathbf{U}_i^q\}_a^{\mathsf{T}} \cdot \{\mathbf{U}_j^r\}_b \quad (6)$$

(2) $\mathbf{U}_i^q$ and $\mathbf{U}_j^r$ are also fed into the attentive model to obtain the attentive representation:

$$\mathbf{H}_i^q = f_{\text{att}}(\mathbf{U}_i^q, \mathbf{U}_j^r, \mathbf{U}_j^r) \quad (7)$$

$$\mathbf{H}_j^r = f_{\text{att}}(\mathbf{U}_j^r, \mathbf{U}_i^q, \mathbf{U}_i^q). \quad (8)$$

A new cross-attention matrix $\mathbf{M'}_{i,j}$ is generated by the cross-attention between the attentive sentence representation $\mathbf{H}_i^q$ and $\mathbf{H}_j^r$ as in Equation 9.

$$\mathbf{M'}_{i,j}^{a,b} = \{\mathbf{H}_i^q\}_a^{\mathsf{T}} \cdot \{\mathbf{H}_j^r\}_b \quad (9)$$

After that, the two cross-attention matrices which represent different views of the cross-attention are stacked to form the final cross-attention representation between the two utterances.

$$\mathbf{F}_{i,j} = f_{\text{stack}}(\{\mathbf{M}_{i,j}, \mathbf{M'}_{i,j}\}) \tag{10}$$

Following Wu et al. [37] and Wang et al. [34], we adopt a 2-layer 2-D CNN to extract matching features from the attention matrix $\mathbf{F}_{i,j}$. The output of the CNN are flattened and mapped into low-dimension vector representation $\mathrm{S}_{i,j}$. $\mathrm{S}_{i,j}$ indicates the interaction feature between the i-th utterance in query's context and the j-th utterance in response's context:

$$\mathrm{S}_{i,j} = f_{\text{flatten}}(f_{\text{CNN}}(\mathbf{F}_{i,j})) \tag{11}$$

3.4.3 CCM: Interaction Aggregation. After the interaction layer, we obtain the interaction matching representations $\{\mathrm{S}_{i,j}\}_{i=0,j=0}^{T_q,T_r}$. The key problems to get the context-to-context matching representation are: 1) how to model the relationships between elements in S and 2) how to aggregate them. In this part, we employ graph attention network and role-aware attention aggregation to solve the two problems respectively.

Considering how to model the relationships between elements, it should be noted that S is a matrix of vectors. So, it is nontrivial to use the traditional encoding methods such as RNN, Transformer encoder, whose input is always a sequence of vectors. The canonical encoding methods cannot well capture the interacted information among the elements in S. Instead, we treat the matching representation S as an undirected graph. The graph network technologies can help us model the relationships between elements in S. Each element of S is a node in the graph. Two nodes are connected to each other when they are neighbors in S. For example, the neighbors of $\mathrm{S}_{i,j}$ are $\{\mathrm{S}_{i-1,j}, \mathrm{S}_{i+1,j}, \mathrm{S}_{i,j-1}, \mathrm{S}_{i,j+1}\}$ as in Figure 2. It means that two utterance pairs are connected when the two pairs own the same query utterance and their response's context utterances are neighbors, and vice versa. We take the graph attention network (GAT) to process the constructed graph and model the relationships in the graph. GAT has shown great success in graph processing [5, 14, 35]. It stacks layers in which nodes are able to attend over their neighborhoods' features. In each layer, it assigns different weighting coefficients to different nodes within a neighborhood to obtain a new node representation.

The representation of the node in the l-th layer is represented as $\mathrm{G}^l_{i,j}$, which corresponds to the matching relationship between i-th utterance in query's context and j-th utterance in response's context. The representation $\mathrm{G}^0_{i,j}$ of the first layer is initialized by $\mathrm{S}_{i,j}$. For the following layers, the nodes is computed as follows: 1) the similarity between $\mathrm{G}^l_{i,j}$ and $\mathrm{G}^l_{a,b}$ is first calculated as the attention weight in the l-th layer:

$$\alpha^l_{i,j,a,b} = \frac{\exp(\text{LeakyReLU}(\text{MLP}_g([\mathrm{G}^l_{i,j}, \mathrm{G}^l_{a,b}])))}{\sum_{c,d\in\mathcal{N}_{i,j}} \exp(\text{LeakyReLU}(\text{MLP}_g([\mathrm{G}^l_{i,j}, \mathrm{G}^l_{c,d}])))}, \tag{12}$$

where $\mathcal{N}_{i,j}$ represents the neighbours of the node $\{i, j\}$ and MLP_g is a multi-layer perceptron (MLP) whose output is a scalar. 2) The next layer representation of node $\{i, j\}$ is calculated by aggregating the representations of its neighbours:

$$\mathrm{G}^{l+1}_{i,j,a,b} = \text{Sigmoid}(\sum_{a,b\in\mathcal{N}_{i,j}} \alpha^l_{i,j,a,b}\mathbf{W}^l\mathrm{G}^l_{a,b}), \tag{13}$$

where $\mathbf{W}^l$ is the corresponding input linear transformation's weight matrix. The aggregation is controlled by the calculated attention weights $\alpha^l_{i,j,a,b}$. The aggregation learns to dynamically fuse representations of its neighbours, consequently, to discover and model the relationships between the matching representations.

Different from standard graph attention network, for each node, we concatenate the output of each layer as the final output to capture different level representations: $\mathbf{A}_{i,j} = [\mathrm{G}^1_{i,j}, \cdots, \mathrm{G}^L_{i,j}]$, where L is the number of layers for GAT. The processing phrase of GAT can be abstracted as:

$$\mathbf{A} = f_{\text{GAT}}(\mathbf{S}). \tag{14}$$

A has the same form with S. $\mathbf{A}_{i,j}$ is also a vector. Due to the page limitation, more details on GAT can be found in Veličković et al. [32].

After GAT layer, we need to aggregate $\{\mathbf{A}_{i,j}\}_{i=0,j=0}^{T_q,T_r}$ to obtain the context-to-context matching representation. A simple way to aggregate them is to take the element-wise mean and max pooling over them. Then a multi-layer perceptron (MLP) is used to get the context-to-context matching representation $\mathbf{R}_{\text{cc}}$. However, not all utterance pairs should be treated equally. The utterances in different positions show different roles and importance. Inspired by such observation, we design the role-aware attention mechanism which takes the position and speaker information into consideration to aggregate the interaction representations:

We define the role embedding as $\mathbf{E} \in \mathbb{R}^{T\times d_p}$ where T is the max turn number of the context[2] and d_p is the dimension of the role embedding. Because in the odd positions the speaker is customer service staff, and in the even positions the speaker is customer, the role embedding also contains speaker information. The role-aware attention is formulated as:

$$\beta_{i,j} = \text{MLP}_r([\mathbf{A}_{i,j}, \mathbf{E}_i, \mathbf{E}_j]) \tag{15}$$

$$\mathbf{A}_a = \sum_{i=0}^{T_q}\sum_{j=0}^{T_c} \mathbf{A}_{i,j}\frac{\exp(\beta_{i,j})}{\sum_{a=0}^{T_q}\sum_{b=0}^{T_r}\exp(\beta_{a,b})} \tag{16}$$

$$\mathbf{A}_{\max} = \text{Max-pooling}(\{\mathbf{A}_{i,j}\}_{i=0,j=0}^{T_q,T_r}) \tag{17}$$

$$\mathbf{A}_{\text{mean}} = \text{Mean-pooling}(\{\mathbf{A}_{i,j}\}_{i=0,j=0}^{T_q,T_r}) \tag{18}$$

$$\mathbf{A}_p = [\mathbf{A}_{\max}, \mathbf{A}_{\text{mean}}, \mathbf{A}_a] \tag{19}$$

where [,] denotes concatenation, $\mathbf{E}_i \in \mathbb{R}^{d_p}$ and $\mathbf{E}_j \in \mathbb{R}^{d_p}$ are the i-th and the j-th row of the role embedding E which provides the global attention information indicating the position and speaker information of the utterances in the query's context and response's context. The output of MLP_r is a number which is used to calculate the attention weight. E is randomly initialized and tuned in the same way with other parameters in this model. $\mathbf{A}_{i,j}$ provides the case level attention and it can dynamically influence the attention weight. The importance of different utterance pairs are shown in the analysis part.

[2] T, T_q, and T_r are all the same in this paper.

The overall process of the context-to-context matching module is abstracted as:

$$\mathbf{R}_{\mathrm{cc}} = f_{\mathrm{cc}}(C^q, C^r) = \mathrm{MLP}_{\mathrm{cc}}(\mathbf{A}_p) \tag{20}$$

3.5 Context-to-Response Matching (CRM)

For the CRM module, which has been well studied, we adopt the IoI model here. The overall process of CRM is: $\mathbf{R}_{\mathrm{cr}} = f_{\mathrm{cr}}(C^q, R)$, where $\mathbf{R}_{\mathrm{cr}}$ is the output of the last layer of the final MLP in IoI and it has the same dimension with $\mathbf{R}_{\mathrm{cc}}$ in this paper. Besides, the essential local loss of IoI which supervises blocks directly is also added to our final loss for fair comparison and the prediction score of each layer is also added to the final prediction score [29]. Inspired by IoI, the loss and prediction score of CRM are also added to CSM to strength the context-response matching which is more important as discussed in the Result Section. More details on IoI can be found in Tao et al. [29]. One of the advantages of our model is that the CRM module is configurable which can be replaced by the new state-of-the-art CRM models to improve the performance.

3.6 Matching Representation Integration

After getting the CCM representation $\mathbf{R}_{\mathrm{cc}}$ and the CRM representation $\mathbf{R}_{\mathrm{cr}}$, we need to integrate them to predict the final matching score. How to balance the weight between $\mathbf{R}_{\mathrm{cc}}$ and $\mathbf{R}_{\mathrm{cr}}$ is crucial since in some cases the context-context pair is similar enough to predict whether the corresponding response is an appropriate response, however, in other cases, the context-to-response matching plays a more decisive role. So, we adopt the gating mechanism to integrate the two matching representations into the final context-to-session matching representation $\mathbf{R}_{\mathrm{cs}}$ (γ is the weighting coefficient):

$$\gamma = \mathrm{Sigmoid}(\mathrm{MLP}_g([\mathbf{R}_{\mathrm{cc}},\mathbf{R}_{\mathrm{cr}}])) \tag{21}$$

$$\mathbf{R}_{\mathrm{cs}} = \gamma\mathbf{R}_{\mathrm{cc}} + (1-\gamma)\mathbf{R}_{\mathrm{cr}} \tag{22}$$

Finally, an MLP with sigmoid activation is applied on the CSM representation $\mathbf{R}_{\mathrm{cs}}$ to predict the final matching score. The CSM model is abstracted as $f_{\mathrm{cs}}(C^q, C^r, R) = \mathrm{MLP}(\mathbf{R}_{\mathrm{cs}})$. The model is trained using the cross-entropy loss.

4 EXPERIMENT SETUP

In this section, we introduce our experiment setup. Section 4.1 is about the datasets we use and their construction methods. Experimental settings is detailed in Section 4.2. The baselines we use are introduced in Section 4.3. We also describe the evaluation metrics in Section 4.4.

4.1 Dataset

Original Dataset. To test our CSM model, we conduct experiments on the E-commerce dialogue corpus [43]. This corpus contains real-world dialogues between customers and customer service staffs from Taobao[3]. It contains 500,000 positive context-response pairs and 500,000 negative context-response pairs. We did not conduct experiments on the open-domain dataset because when constructing the context-session pairs, a response needs to have multiple types of contexts (query's context and response's context). With this requirement, the open-domain dialogue corpus is unsuitable since there are only the dull and useless responses that are corresponding to multiple contexts. The E-commerce dialogue corpus for information-seeking dialogue fits our requirement well. The service staff of E-commerce platform usually use the same response to answer the query whose context locate in the same domain. For instance, a response about pre-defined after-sale policy can answer multiple types of queries related to customers' after-sale concerns. We compose the following training set and two kinds of test set.

Algorithm 1 Context-session pairs construction. The negative pairs construction is not shown in the pseudo-code.

```
CR: All the training context-response pairs.
CS = ∅: Context-session pairs.
for c, r in CR do
    CC=FindCorredpondingContext(r)    ▷ Find the contexts whose response is r.
    if |CC| > 0 then
        c_r = TF-IDF(c, CC)    ▷ Find the most similar context to c in CC.
        CS = CS ∪ {{c, c_r, r}}    ▷ c is the query's context, c_r is the response's context, and r is the response. c_r and r compose the session.
    end if
end for
Return CS
```

Dataset Construction: TestRandNegCand. To train CSM model, we construct the context-session pairs based on the positive context-response pairs. For each context-response pair, we need to find a response's context to form the {query's context, response and response's context} pair, i.e., the context-session pair. Specifically, we firstly collect at most n contexts whose response is the same with the response from the whole context-response training pairs as the coarse candidate response's context set CC [4]. Then, we rerank response's contexts in CC by the TF-IDF score between the element of this set with the query's context. We select the element with the highest score as the response's context to avoid that the query's context and the selected response's context are too dissimilar to train the CCM module. The context-response pairs are dropped when we can not find the coarse candidate response's context set CC. n is set to 20 for efficiency.

Finally, we get 92,945 positive context-session pairs. For each query's context, we randomly select a session as the negative candidate session. All the positive and negative context-session pairs are divided into train, validation, and test set with size 181,890, 20,000 and 20,000. We call this test set TestRandNegCand (Random Negative Candidate). For each query's context in validation and test set, we randomly sample 9 negative candidate sessions. The pseudo-code of the construction process and data analysis are shown in Algorithm 1 for better understanding.

Dataset Construction: TestRetrvCand. In order to verify the effectiveness of our model in practical scenarios, we construct another test set called TestRetrvCand (Retrieved Candidate) where the response's contexts in the negative candidate sessions are stronger

[3] https://www.taobao.com/

[4] The coarse candidate response's context set does not contain the corresponding query's context.

than the randomly sampled contexts. Concretely, TestRetrvCand contains 1,000 query's contexts and each query's context has 10 response's contexts which are retrieved by TF-IDF from all the training set that are the most similar with the query's context. The retrieved context and its response constitute the candidate session, forming the context-session pair. We do not know whether the response of the retrieved session is an appropriate response. So we employ human annotators to label the responses. Three annotators are asked to label each case and the label most of them choose is treated as the final label. After filtering the cases which have no appropriate response, there are 8,840 context-session pairs left.

Data analysis. To better understand the motivation of dataset construction, we calculate the data distribution in terms of repeatability. The construction of our training data and TestRandNegCand relies on the assumption that some contexts correspond to the same response. 98,782 responses have one context. 10,256 responses have more than 2 contexts. More specifically, these 10,256 responses have 118,606 corresponding contexts totally. Therefore, the dataset we constructed contains enough context-session pairs to train our models.

To the best of our knowledge, our released test set is the first to select the response from the level of dialogue sessions which contain both the response's context and response. The way to construct the candidates of TestRetrvCand via TF-IDF is more practical.

4.2 Experimental Settings

We use Adam [16] optimizer with learning rate 0.0001 and batch size 100 to optimize the parameters. The word embedding dimension is 200. And we pre-train the word embeddings through GloVe [23]. The embeddings are tuned during the model training to get better performance. The vocabulary size is 36,105 which covers all the words in the training set. The max turn number of contexts is 5 and the max utterance length is 20, which is sufficient to cover most of the turns and words in the corpus. We use padding to handle the various lengths of the text. The best performing checkpoint on the validation set is selected for testing according to $P_{10}@1$.

The dimension d_p of the role-aware representation is 25 and the dimension of the matching representation $\mathbf{R}_{\text{cc}}$ and $\mathbf{R}_{\text{cr}}$ are both 50. The number of layers of the attentive module in the representation layer is 3. The kernel size of the 2-D convolution is (3, 3) with stride size 1. The channel size of the convolution operation is 32 for the first layer and 16 for the second layer. For the max pooling of the CNN model, the pool size is (3, 3) with stride size 3. For GAT, the number of layers is 4 and the dimension is 128 for the internal nodes.

4.3 Baselines and Models

To better evaluate the performance of our proposed methods, we consider three types of baselines and models: the conventional context-to-response matching (CRM) methods, our proposed context-to-session matching (CSM) model, and the ablated version of CCM to analyze the effect of our proposed context-to-context matching (CCM).

CRM. DAM and IoI are the strong baselines for the CRM task which do not take response's context into consideration. They are represented as CRM (DAM) and CRM (IoI).

CCM. As one of the ablation studies, we evaluate the performance of CCM. We also conduct ablation studies for the components in CCM. "CCM w/o Role" works as CCM without role-aware attention integration, which means there is no $\mathbf{E}_i$ and $\mathbf{E}_j$ in Equation 15. "CCM w/o GAT" means there is no GAT, and **S** are fed into the role-aware attention layer directly. Besides, to verify the effectiveness of our proposed CCM model, we also compare our results with CCM (Con) and CCM (BiMPM). They concatenate the utterances in the context and treats the context as one sentence. CCM (Con) model can be seen as a special case of our proposed CCM that the number of turns is 1. CCM (BiMPM) takes BiMPM [36] to calculate the matching degree between the concatenated contexts.

CSM. CSM denotes our proposed context-to-session matching model. In addition, CSM (CR-Con) works as a baseline which concatenates the candidate session into one sentence. Then an IoI model is applied to model the query's context and the concatenated session.

4.4 Evaluation Metrics

Following previous work [8, 24, 37, 39], we evaluate the models in terms of MRR (Mean Reciprocal Rank) [33], MAP (Mean Average Precision) [1], $P_{10}@1$, $R_{10}@1$, $R_{10}@2$, $R_{10}@5$ and $R_2@1$, which are all frequently-used metrics in response selection tasks. $R_n@k$ calculates the recall of the true positive responses among the k selected candidates from n available candidates, and $P_n@k$ refers to the precision. MRR is adopted to evaluate TestRandNegCand where there is only one ground-truth in the candidates. For the TestRetrvCand where a query's context has multiple right responses, we employ MAP metric.

5 RESULTS AND ANALYSIS

In this section, we present the experimental results and give our analysis. Section 5.1 is to compare our CSM model with its CCM and CRM counterparts. The ablation study, which is used to verify the effectiveness of proposed GAT and role-aware attention components, is introduced in Section 5.2. The discussion of the context-to-context matching module's role is given in Section 5.4. To better demonstrate the usefulness of different utterances in the dialogue session, the analysis of role-aware attention is given in Section 5.5.

5.1 CSM v.s. CRM v.s. CCM.

The evaluation results and ablation studies are shown in Table 2. Compared with the baselines including CRM and CCM, we can find that our proposed CSM model achieves the best performance across most of the evaluation metrics on the two test sets. On the TestRetrvCand test set which is more reliable, CSM gains 0.08 improvement on $P_{10}@1$ and 0.026 improvement on MAP. It reflects the promising prospect of applying our proposed context-to-session matching approach into industrial retrieval-based dialogue systems. As two important parts of CSM, CCM and CRM can work independently. The result of CCM indicates that CCM provides useful evidence for response selection. The comparison between CRM and CCM shows that CRM outperforms CCM. Possible explanations for the result may be: 1) CRM ranks response directly. However, CCM model cannot get access to the responses. 2) there is more data to train CRM model.

Table 2: Results for test set TestRandNegCand and TestRetrvCand. The results of CSM are significant with p-value < 0.05 measured by the Student's paired t-test over the baselines.

Models	TestRandNegCand					TestRetrvCand				
	MRR	R_{10}@1	R_{10}@2	R_{10}@5	R_2@1	MAP	P_{10}@1	R_{10}@1	R_{10}@2	R_{10}@5
CCM (Con)	0.7366	0.6650	0.8225	0.9525	0.9015	0.5666	0.4446	0.1333	0.2534	0.5821
CCM (BiMPM)	0.7520	0.6865	0.8345	0.9605	0.9150	0.5749	0.4672	0.1422	0.2586	0.5850
CCM w/o Role	0.7536	0.6835	0.8285	0.9620	0.9085	0.5656	0.4389	0.1297	0.2557	0.5784
CCM w/o GAT	0.7584	0.6920	0.8470	0.9695	0.9195	0.5932	0.4729	0.1421	0.2792	0.6225
CCM	0.7732	0.7110	0.8530	0.9752	0.9290	0.5998	0.4977	0.1554	0.2945	0.6078
CRM (DAM)	0.8087	0.7540	0.9160	0.9920	0.9525	0.6484	0.5023	0.1592	0.3122	0.6935
CRM (IoI)	0.8308	0.7855	0.9315	0.**9945**	0.9610	0.6725	0.5430	0.1765	0.3474	**0.7193**
CSM (CR-Con)	0.6364	0.5360	0.7260	0.9355	0.8690	0.6126	0.4717	0.1412	0.2865	0.6487
CSM	**0.8456**	**0.8045**	**0.9380**	0.9905	**0.9635**	**0.6986**	**0.6222**	**0.2003**	**0.3767**	0.7168

5.2 CCM ablation study.

The GAT and role-aware attention are important parts of CCM model, thus we take the ablation study to verify their effectiveness. Results show that P_{10}@1 drops 0.248 when ablating the GAT, which proves that GAT can learn and model the relations of the matching representations. Additionally, P_{10}@1 drops 0.588 when ablating the role-aware attention. This shows that the role-aware attention can efficiently aggregate the matching representations.

5.3 Comparison with other CSM baseline.

CSM (CR-Con) performs worst across the metrics. CSM (CR-Con) treats the response's context and response as one sentence. The results show the necessity to model the context-to-context matching and context-to-response matching separately. Because the two parts correspond to different aspects: similar question detection and question-response dialog pattern detection. CSM (CR-Con) does not have the ability to separate these two aspects.

5.4 Why do we need context-to-context matching when the candidates are retrieved by TF-IDF?

In the real-world scenario, the candidate sessions are retrieved by TF-IDF based on the query's context, which is already a context-to-context matching. Why do we bother to model the context-to-context relationship with a new deep matching network? The reasons lie in two aspects. The first reason is that since deep matching has shown great success in text matching, it may improve the performance of context-to-context matching compared with TF-IDF. Another reason is that TF-IDF does not take { the position, speaker information, relationships between utterances} into consideration. The results in Table 2 verify that the position and speaker information are helpful to CCM and the neural network based CCM works fine. These are also the reasons why the response's context in CSM helps response selection.

5.5 Analysis on role-aware attention.

We design an interaction aggregation layer (Section 3.4.2) to aggregate the matching representations of utterance pairs. The proposed role-aware attention assigns different weights to the utterance pair matching representations. We calculate the mean of the attention weights across the test set TestRetrvCand as in Figure 4. The attention weights show that: 1) The CUST^{-5}-CUST^{-5} pair shows the highest score 0.1720. It means that the last utterance which is closest to response is the most important. 2) The scores of CUST-CUST pairs are generally larger than CUST-STAFF and STAFF-STAFF pairs. It indicates that what the customer asked is more important. This result is in agreement with our common sense: since information-seeking dialogue systems aim at satisfying information needs, what the customer asked is relatively more important than what has been answered. The analysis reflects the necessity to adopt role-aware attention.

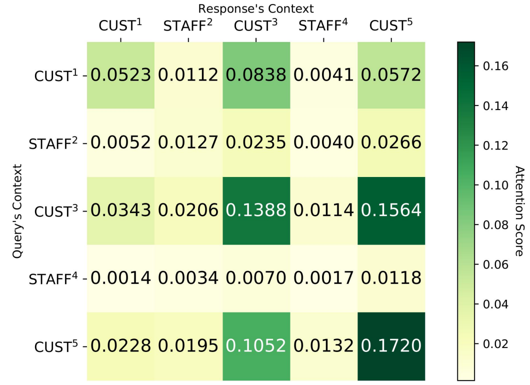

Figure 4: Attention weight for the role-aware attention. The superscript *i* means the utterance is the *i*-th utterance of the context.

6 CONCLUSION

In this paper, we propose a novel context-to-session matching (CSM) approach for the response selection task in information-seeking dialogue systems. Compared with its conventional context-to-response matching counterparts, this type of CSM approach takes full advantage of the knowledge of whole sessions: the response's context in the dialogue session can help choose the appropriate response by measuring whether the two query's context and response's context are discussing the same question. Furthermore, the graph attention network and role-aware attention in our proposed CSM model can help to model and aggregate the matching representations between query's context and response's context. We are surprised by the extent of improvement achieved by our proposed CSM approach and are excited about its future in response selection tasks.

7 ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their constructive comments. We thank Wenpeng Hu, Chang Liu for their helpful discussions. This work was supported by the National Key R&D Program of China (No. 2017YFC0804001), the National Natural Science Foundation of China (NSFC No. 61876196 and NSFC No. 61672058). Rui Yan is partially supported as a Young Fellow of Beijing Institute of Artificial Intelligence (BAAI). This work was supported by Alibaba Group through Alibaba Research Fellowship Program.

REFERENCES

[1] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. 1999. *Modern information retrieval*. Vol. 463. ACM press New York.

[2] Zhangming Chan, Juntao Li, Xiaopeng Yang, Xiuying Chen, Wenpeng Hu, Dongyan Zhao, and Rui Yan. 2019. Modeling Personalization in Continuous Space for Response Generation via Augmented Wasserstein Autoencoders. In *EMNLP-IJCNLP*. 1931–1940.

[3] Qian Chen and Wen Wang. 2019. Sequential Attention-based Network for Noetic End-to-End Response Selection. *arXiv preprint arXiv:1901.02609* (2019).

[4] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for Natural Language Inference. In *ACL*. 1657–1668.

[5] Weijian Chen, Yulong Gu, Zhaochun Ren, Xiangnan He, Hongtao Xie, Tong Guo, Dawei Yin, and Yongdong Zhang. 2019. Semi-supervised user profiling with heterogeneous graph attention networks. In *IJCAI*. AAAI Press, 2116–2122.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*. Minneapolis, Minnesota, 4171–4186.

[7] Devin Didericksen, Oleg Rokhlenko Kevin Small Li Zhou, and Jared Kramer. [n.d.]. Collaboration-based User Simulation for Goal-oriented Dialog Systems. ([n. d.]).

[8] Zhenxin Fu, Feng Ji, Wenpeng Hu, Wei Zhou, Dongyan Zhao, Haiqing Chen, and Rui Yan. 2019. Query-bag Matching with Mutual Coverage for Information-seeking Conversations in E-commerce. In *CIKM*. 2337–2340.

[9] Shen Gao, Xiuying Chen, Chang Liu, Li Liu, Dongyan Zhao, and Rui Yan. 2020. Learning to Respond with Stickers: A Framework of Unifying Multi-Modality in Multi-Turn Dialog. In *WWW*. 1138–1148.

[10] Yichen Gong, Heng Luo, and Jian Zhang. 2017. Natural language inference over interaction space. *arXiv preprint arXiv:1709.04348* (2017).

[11] Jia-Chen Gu, Zhen-Hua Ling, and Quan Liu. 2019. Interactive Matching Network for Multi-Turn Response Selection in Retrieval-Based Chatbots. (2019).

[12] Hua He, Kevin Gimpel, and Jimmy Lin. 2015. Multi-perspective sentence similarity modeling with convolutional neural networks. In *EMNLP*. 1576–1586.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*. 770–778.

[14] Binxuan Huang and Kathleen M Carley. 2019. Syntax-Aware Aspect Level Sentiment Classification with Graph Attention Networks. In *EMNLP-IJCNLP*. 5472–5480.

[15] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *CIKM*. ACM, 2333–2338.

[16] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[17] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436.

[18] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).

[19] Feng-Lin Li, Minghui Qiu, Haiqing Chen, Xiongwei Wang, Xing Gao, Jun Huang, Juwei Ren, Zhongzhou Zhao, Weipeng Zhao, Lei Wang, et al. 2017. AliMe assist: an intelligent assistant for creating an innovative e-commerce experience. In *CIKM*. ACM, 2495–2498.

[20] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A Diversity-Promoting Objective Function for Neural Conversation Models. In *NAACL*. 110–119.

[21] 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. In *SIGDIAL*. 285–294.

[22] Wentao Ma, Yiming Cui, Nan Shao, Su He, Wei-Nan Zhang, Ting Liu, Shijin Wang, and Guoping Hu. 2019. TripleNet: Triple Attention Network for Multi-Turn Response Selection in Retrieval-Based Chatbots. In *CoNLL*. Association for Computational Linguistics, Hong Kong, China, 737–746.

[23] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*. 1532–1543.

[24] Minghui Qiu, Liu Yang, Feng Ji, Wei Zhou, Jun Huang, Haiqing Chen, Bruce Croft, and Wei Lin. 2018. Transfer Learning for Context-Aware Question Matching in Information-seeking Conversations in E-commerce. In *ACL*. 208–213.

[25] Chen Qu, Liu Yang, W Bruce Croft, Yongfeng Zhang, Johanne R Trippas, and Minghui Qiu. 2019. User intent prediction in information-seeking conversations. In *CHIIR*. ACM, 25–33.

[26] Chen Qu, Liu Yang, Minghui Qiu, W. Bruce Croft, Yongfeng Zhang, and Mohit Iyyer. 2019. BERT with History Answer Embedding for Conversational Question Answering. In *SIGIR* (Paris, France). ACM, New York, NY, USA, 1133–1136.

[27] Paul Solomon. 1997. Conversation in information-seeking contexts: A test of an analytical framework. *Library & Information Science Research* 19, 3 (1997), 217–248.

[28] Chongyang Tao, Wei Wu, Can Xu, Wenpeng Hu, Dongyan Zhao, and Rui Yan. 2019. Multi-Representation Fusion Network for Multi-Turn Response Selection in Retrieval-Based Chatbots. In *WSDM*. ACM, 267–275.

[29] Chongyang Tao, Wei Wu, Can Xu, Wenpeng Hu, Dongyan Zhao, and Rui Yan. 2019. One Time of Interaction May Not Be Enough: Go Deep with an Interaction-over-Interaction Network for Response Selection in Dialogues. In *ACL*. Association for Computational Linguistics, Florence, Italy, 1–11.

[30] Paul Thomas, Daniel McDuff, Mary Czerwinski, and Nick Craswell. 2017. MISC: A data set of information-seeking conversations. In *CAIR'17*, Vol. 5.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[32] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *ICLR*.

[33] Ellen M Voorhees et al. [n.d.]. The TREC-8 question answering track report. Citeseer.

[34] Heyuan Wang, Ziyi Wu, and Junyu Chen. 2019. Multi-Turn Response Selection in Retrieval-Based Chatbots with Iterated Attentive Convolution Matching Network. In *CIKM*. ACM, 1081–1090.

[35] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In *KDD*. 950–958.

[36] Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. In *AAAI*. AAAI Press, 4144–4150.

[37] Yu Wu, Wei Wu, Chen Xing, Ming Zhou, and Zhoujun Li. 2017. Sequential Matching Network: A New Architecture for Multi-turn Response Selection in Retrieval-Based Chatbots. In *ACL*. 496–505.

[38] Rui Yan, Yiping Song, and Hua Wu. 2016. Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In *SIGIR*. ACM, 55–64.

[39] Rui Yan and Dongyan Zhao. 2018. Coupled context modeling for deep chit-chat: towards conversations between human and computer. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2574–2583.

[40] Liu Yang, Minghui Qiu, Chen Qu, Jiafeng Guo, Yongfeng Zhang, W Bruce Croft, Jun Huang, and Haiqing Chen. 2018. Response ranking with deep matching networks and external knowledge in information-seeking conversation systems. In *SIGIR*. ACM, 245–254.

[41] Jianfei Yu, Minghui Qiu, Jing Jiang, Jun Huang, Shuangyong Song, Wei Chu, and Haiqing Chen. 2018. Modelling domain relationships for transfer learning on retrieval-based question answering systems in e-commerce. In *WSDM*. ACM, 682–690.

[42] Chunyuan Yuan, Wei Zhou, Mingming Li, Shangwen Lv, Fuqing Zhu, Jizhong Han, and Songlin Hu. 2019. Multi-hop Selector Network for Multi-turn Response Selection in Retrieval-based Chatbots. In *EMNLP-IJCNLP*. Association for Computational Linguistics, Hong Kong, China, 111–120.

[43] Zhuosheng Zhang, Jiangtong Li, Pengfei Zhu, Hai Zhao, and Gongshen Liu. 2018. Modeling Multi-turn Conversation with Deep Utterance Aggregation. In *COLING*. ACL, Santa Fe, New Mexico, USA, 3740–3752.

[44] Xiangyang Zhou, Daxiang Dong, Hua Wu, Shiqi Zhao, Dianhai Yu, Hao Tian, Xuan Liu, and Rui Yan. 2016. Multi-view response selection for human-computer conversation. In *EMNLP*. 372–381.

[45] Xiangyang Zhou, Lu Li, Daxiang Dong, Yi Liu, Ying Chen, Wayne Xin Zhao, Dianhai Yu, and Hua Wu. 2018. Multi-turn response selection for chatbots with deep attention matching network. In *ACL*. 1118–1127.