

Multi-Turn Response Selection in Retrieval-Based Chatbots with Iterated Attentive Convolution Matching Network

Heyuan Wang*

Key Lab of High Confidence Software Technologies(MOE), School of EECS,
Peking University
wangheyuan@pku.edu.cn

Ziyi Wu

Department of Probability and Statistics, School of Mathematical Sciences, Peking University
wuziyi@pku.edu.cn

Junyu Chen

Key Lab of High Confidence Software Technologies(MOE), School of EECS,
Peking University
chenjunyu@pku.edu.cn

ABSTRACT

Building an intelligent chatbot with multi-turn dialogue ability is a major challenge, which requires understanding the multi-view semantic and dependency correlation among words, n-grams and sub-sequences. In this paper, we investigate selecting the proper response for a context through multi-grained representation and interactive matching. To construct hierarchical representation types of text segments, we propose a refined architecture which exclusively consists of gated dilated-convolution and self-attention. Compared with the recurrent-based sentence modeling methods, this architecture provides more flexibility and a speedup. The matching signals of each utterance-response pair are extracted by integrating the interactive information from different views. Then a turns-aware attention mechanism is utilized to aggregate the matching sequence, so as to identify important utterances and capture the implicit relationship of the whole context. Experiments on two large-scale public data sets show that our model significantly outperforms the state-of-the-art methods in terms of all metrics. We empirically provide a thorough ablation test, as well as the comparison of different representation and matching strategies, for a better insight into how each component affects the performance of the model.

CCS CONCEPTS

- Information systems → Retrieval models and ranking;

KEYWORDS

multi-grained representation; matching; retrieval-based chatbot; multi-turn response selection; deep neural network

ACM Reference Format:

Heyuan Wang, Ziyi Wu, and Junyu Chen. 2019. Multi-Turn Response Selection in Retrieval-Based Chatbots with Iterated Attentive Convolution Matching Network. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM '19), November 3–7, 2019, Beijing, China*. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3357384.3357928>

*Corresponding Author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CIKM '19, November 3–7, 2019, Beijing, China

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-6976-3/19/11...\$15.00

<https://doi.org/10.1145/3357384.3357928>

	Context
Utterance-1	Speaker A: hendikins what do i need to do regarding the real-player ?
Utterance-2	Speaker B: for more details
Utterance-3	Speaker A: I have mplayer installed. I don't even need real player , I just want a real-player and windows media player plugin for playing online streams.
Utterance-4	Speaker B: there is a mozilla mplayer plugin .
Utterance-5	Speaker A: I have that installed but I cannot play any streams .
Utterance-6	Speaker B: what browser are you using?
Utterance-7	Speaker B: and what version?
Utterance-8	Speaker A: mozilla firefox _ number _ number
Response	Speaker B: hmm that's weird and can you download that stream and play it in mplayer ?

Figure 1: Example of multi-turn conversation from Ubuntu Corpus. Speaker A has a problem of playing streams with the mplayer plugin. A topic change occurs in utterance 6–8. Text segments with the same color can be regarded as matching pairs.

1 INTRODUCTION

Human-computer conversation systems on open-domain topics have attracted much attention for their functional roles in real-world applications in recent years. Representative real products include the AI companion XiaoIce [23] from Microsoft and the E-commerce assistant AliMe Assist from Alibaba Group [11]. Existing methods are data-driven and lie in two groups: the retrieval-based approach that selects a proper reply from a pre-build repository [9, 15, 35, 43–45], and the generation-based approach that directly generates a response via natural language generation techniques [19–22, 36]. Response selection, which aims to select the best response from a set of candidates given the conversation context, is an important and challenging task in building chatbots, especially for retrieval-based chatbots [8].

The key to response selection lies in input-response matching. While most early studies focus on single-turn conversation [32] which only considers the last input message for matching a reply, the multi-turn conversation scenario has attracted more attention in recent works [35, 43, 45]. Specifically, multi-turn response selection aims to detect the semantic clues of multiple utterances in previous turns, and select a reply that is natural and meaningful to the whole context. There are two major challenges of the task: (1) how to capture and utilize important information in dialogue segments for matching, over different granularities (words, phrases, sentences, etc); (2) how to model the relationship among multiple utterances in a context. Figure 1 shows a conversation from the Ubuntu Dialogue Dataset [14]. Firstly, as demonstrated, the semantic and functional relevance lies not only in lexical similarities of

words or phrases (e.g., “mplayer”-“real player”-“player plugin”, and “online streams”-“any streams”-“stream”), but also in the latent context dependencies. For example, the word “it” in the response refers to the previous word “stream”, which connects to the expression topic of “playing streams” in the third and fifth utterances. Secondly, the correlation between each utterance and response is different. Utterances 6-8 are about the browser version used by speaker A, while discussions in previous turns are about installing the mplayer plugin for playing streams. Such a topic change is natural in real-world human conversation, but in this dialogue, the new topic is less relevant to the reply and may introduce confusing information.

In this paper, we propose the Iterated Attentive Convolution Matching Network (IACMN) to deal with the two challenges. For text encoding, most existing work employ Recurrent Neural Networks (RNNs) [14, 35, 44], which are often time-costly due to the limitation of parallelism. As an alternative, IACMN constructs multi-grained representations of context utterances and response candidates by stacking a refined combination of *dilated-convolution* [42] and *self-attention* [29], namely the *Attentive Gated Dilated Residual* (AGDR) block. Our solution is inspired by [34] in the issue of reading comprehension, which achieves satisfying speedups using only dilated convolution with increasing receptive fields in place of RNNs. We extend the convolutional structure in two ways:

- We jointly introduce dilated-convolution and self-attention in one uniform architecture. Convolution operations have the ability to model different n-gram features based on the hierarchical composition of local interactions, while the self-attention mechanism is more helpful to enhance global interactions. Each AGDR block contains a layer-by-layer structure wrapped with the gated linear unit [5] and the residual network [6]. We use exponentially increasing dilation widths, so as to ensure the extensibility and consistency of input receptive fields. By making a sentence attend to itself at the beginning of each layer, each segment can be enriched with other similar segments, thus the textual relevance and intra long dependencies are better modeled.
- We hierarchically stack the unified AGDR blocks from word embeddings, the output of each layer is extracted and viewed as a feature map of the input. In this way, we gradually obtain the input representations of different granularities, including words, n-grams, and sentence levels.

IACMN calculates matching matrices for each representation view of each utterance-response pair. Then a convolution neural network is utilized to identify salient signals and integrate the multiple matching patterns into a fused vector. The refined matching information in IACMN is much richer than the two levels (word and recurrent-based sentence level) in [35]. Moreover, compared with the state-of-the-art method [45] which leverages Transformer [29] to encode input messages, our convolution-based representation layer takes the advantage of significantly reducing the token-pair memory consumption in position-wise Feed-Forward Network (FFN), which is an important component in Transformer, especially for long texts. To address the challenge of modeling relationships among multi-turn utterances in the context, IACMN leverages the bidirectional GRU [4] with a turns-aware attention design in the aggregation stage. By this means, both the temporal order of history utterances and their distinct correlations with the response

candidate are considered. Finally, the matching score is obtained via a single-layer perceptron with the hidden states of bi-GRU.

We test IACMN on two large-scale public datasets, the Ubuntu Dialogue Corpus v1 [14] and the Douban Conversation Corpus [35]. Experimental results show that our model significantly outperforms the state-of-the-art methods on both datasets in all evaluation metrics. The improvement to the best baseline on $\mathbf{R}_{10}@1$ for Ubuntu dataset is over 1.5%, and 2.1% on $\mathbf{P}@1$ for Douban dataset. Our code is available at <https://github.com/heyuanw/IACMN>.

To sum up, our contributions are four-folds: (1) the proposal of a new deep matching model for multi-turn response selection in retrieval-based chatbots; (2) empirical verification of the effectiveness of the model on public datasets; (3) publication of the source code; (4) the thorough ablation test, quantity analysis and case study to better compare different representation and matching strategies.

2 RELATED WORK

Building a chatbot that can interact with human beings has long been an attractive but challenging task in artificial intelligence (AI) [28]. Due to the great quantity of human conversation data available, a number of data-driven dialogue systems are designed. Existing work includes generation-based and retrieval-based methods.

The generation-based [19–22, 36] method aims at generating a response token by token according to conditional probabilistic language models (LM). Since the sequence-to-sequence model with attention [3] achieved great success in machine translation task, it has also been widely applied in generation-based chatbot system. There are plenty varieties discovered based on this model, including leveraging external knowledge [18, 37], applying reinforcement learning [12] and adversarial learning techniques [13, 38].

The retrieval-based method [9, 15, 35, 43–45] is the focus of this paper, which selects the best-matching response candidate from a large dialogue repository. These methods have the advantage of providing more decent and fluent replies since all the candidates are human-generated. Early studies of retrieval-based chatbots focus on single-turn conversation, which only leverages the last utterance in the context to select the response [7, 8, 16]. Recent works pay more attention to the multi-turn dialogue scenario. Lowe et al. [14] concatenate the whole context utterances together to match a response. Yan et al. [40] select previous utterances in different strategies and then combine them with the last utterance to form a reformulated context. Zhou et al. [44] investigate a multi-view model on word level and RNN-based utterance level to measure the relevance. These models lose much important information as the response cannot interact with the context until the final step in matching. To address this problem, Wu et al. [35] propose to interact each utterance-response pair at the beginning, and then accumulate the multi-turn matching vectors with a gated recurrent units (GRU). Zhou et al. [45] construct multi-grained representations of the inputs through self-attention and cross-attention based on the architecture of Transformer [29]. Unlike previous studies, our model uses a refined combination of dilated convolution and self-attention to build multi-view input representations fast and effectively. Considering the different importance of history utterances, we adopt a turns-aware attention design to aggregate the multi-turn matching information.

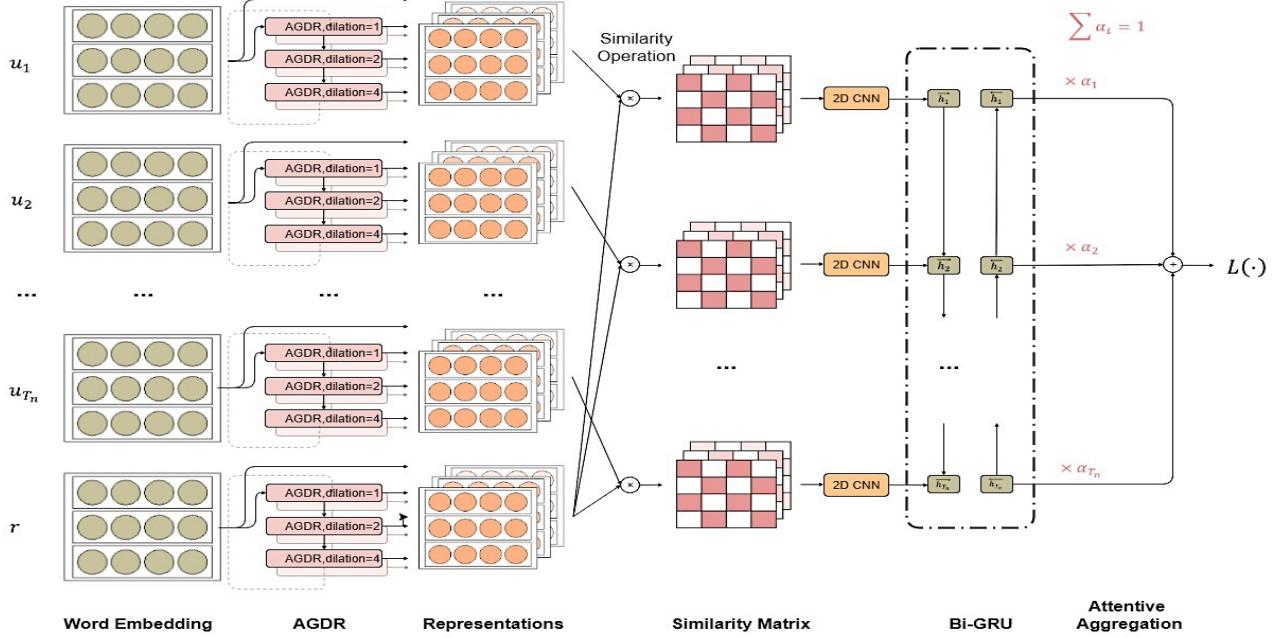


Figure 2: Architecture of IACMN.

3 PRELIMINARIES

3.1 Problem Formalization

The dialogue data set can be described as $\mathcal{D} = \{(c, r, y)_i\}_{i=1}^N$. For each conversation triplet $< c, r, y >$, $c = \{u_1, \dots, u_{n_T}\}$ represents the multi-turn context where u_k denotes the k -th utterance. r is a response candidate and $y \in \{0, 1\}$ is a binary label, indicating whether r is the proper response for c . The aim is to build a model $g(\cdot, \cdot)$ with \mathcal{D} . For any context-response pair (c, r) , $g(c, r)$ measures the matching score between c and r .

3.2 Model Overview

We propose the Iterated Attentive Convolution Matching Network (IACMN) to model $g(\cdot, \cdot)$. Figure 2 shows the overview. IACMN decomposes the context-response matching into multiple utterance-response pair matching, and then accumulates all pairs information to calculate the final matching score for (c, r) in an end-to-end way. Specifically, IACMN has three modules: *hierarchical representation*, *pair matching*, and *turns-aware aggregation*.

Let n_{u_k} and n_r denote the length of the k -th utterance and the response, u_k and r are first represented as sequences of word embeddings, namely $U_k^0 = [e_{u_k, 0}, \dots, e_{u_k, n_{u_k}-1}]$ and $R^0 = [e_{r, 0}, \dots, e_{r, n_r-1}]$, where $e \in \mathbb{R}^d$ refers to a d -dimension word embedding. Based on U_k^0 and R^0 , IACMN constructs multi-grained representations for u_k and r by hierarchically stacking the *Attentive Gated Dilated Residual (AGDR)* block. Each block layer takes the output of the previous layer as its input, so as to harvest more sophisticated lexical and semantic features among words, n-grams and sub-sequences in an utterance. Suppose that L layers are stacked from the word embeddings, u_k and r are then encoded as $[U_k^l]_{l=0}^L$ and $[R^l]_{l=0}^L$ respectively.

The second module distills rich matching patterns of a pair (u_k, r) from $[U_k^l]_{l=0}^L$ and $[R^l]_{l=0}^L$. Specifically, for each granularity $l \in \{0, L\}$, IACMN calculates a similarity matrix toward U_k^l and R^l , denoted as $M_{k,r}^l \in \mathbb{R}^{n_{u_k} \times n_r}$. All the $L + 1$ matrices $\{M_{k,r}^l\}_{l=0}^L$ are considered as different feature channels and merged into a 2D image, then the convolution & max pooling operations are utilized to extract the salient matching signals into a fused vector $s_{k,r}$. In this way, the utterance and response can fully interact with each other, based on the compositional hierarchy of multiple representation types.

In aggregation module, the multi-turn matching vectors $\{s_{k,r}\}_{k=1}^{n_T}$ are fed into the bidirectional GRU with a turns-aware attention design, following the chronological order of utterances in the context. The final score of (c, r) is calculated via a single-layer perceptron, using the weighted sum of hidden output in each step of the bi-GRU.

IACMN enjoys several advantages over existing models. Firstly, the representation module leverages the specific AGDR block in place of costly recurrent units to capture latent long-range dependencies, which is more convenient for latency critical applications. Secondly, multi-grained semantic or functional matching information can be sufficiently extracted with the hierarchical encoding and interaction architecture. Thirdly, with a turns-aware attention design in multi-turn aggregation, utterances more related to the context can be highlighted, while redundant noise can be ignored to some extent. In the following sections, we will describe the details of each module.

Figure 3 illustrates the structure of one layer in AGDR block, which is a refined combination of self-attention, gated linear unit, dilated convolution and residual network. Given an input sentence $X = \{x_1, \dots, x_n\}$, we first apply self-attention to model the intra

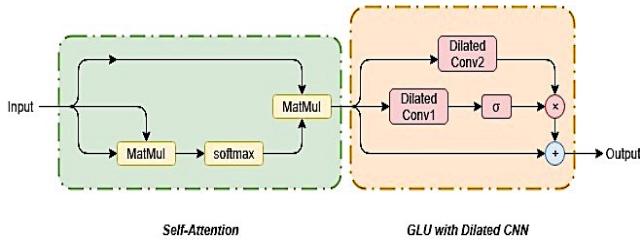


Figure 3: AGDR Layer.

element-wise dependencies. Specifically, each word in X (named the query) is enhanced by attending to all other words (named the keys), where the similarity is calculated by scaled dot product between the query and key [29]. The attention operation is formulated as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

where Q, K, V are the same for self-attention of a sentence. Each row in $\text{Attention}(Q, K, V)$, denoted as $\text{Att}[i]$, is a fused vector of all words in the value sentence, indicating their weighted correlation with the i -th word in the query sentence. Then the dilated convolution is applied to capture more information of text segments. Different from standard convolution that convolves a contiguous subsequence of the input at each step, dilated convolution has a wider receptive field by skipping over δ input elements at a time, where δ is the dilation rate. An example of dilated convolution is shown in Figure 4, the darker output is a weighted combination of the lighter regular spaced inputs in the previous layer. For a sequence x and a kernel W of size $2d + 1$, the standard / dilated convolution operations are separately defined as:

$$F(x_t) = W \bigoplus_{k=0}^d x_{t+k} \quad (2)$$

$$F(x_t) = W \bigoplus_{k=0}^d x_{t+k\delta} \quad (3)$$

where \oplus is vector concatenation. The output is wrapped by a Gated Linear Unit (GLU) that provides both linear and non-linear paths to ensure effective information propagation, defined as:

$$h_l(x) = (x * W + b) \otimes \sigma(x * V + c) \quad (4)$$

where σ is the sigmoid function and \otimes is the element-wise product between matrices. This gated convolution structure is then placed inside a residual block, followed by layer normalization [1] to prevent vanishing or exploding of gradients. The output is then fed into the next layer to build higher-level representations.

Following the use of dilated convolution in reading comprehension and entity recognition [26, 34], we double the dilation rate of each layer in a block, starting with dilation_1 (equals to standard convolution) for the first layer, to ensure that no element of the input sequence is excluded. With iterated layers, the actual receptive input width grows exponentially, while the number of parameters

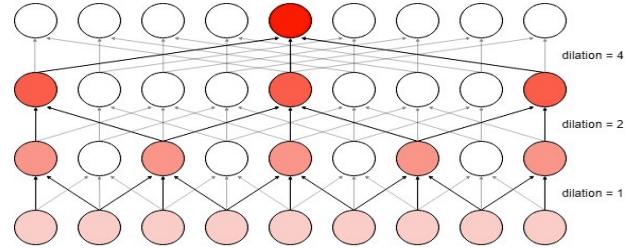


Figure 4: Dilated CNN.

increases only linearly. For a 4-layer block with convolution kernel size of 3 and dilation rates of [1-2-4-8], the input n-gram ranges perceived by the kernels of each layer are [3-7-15-31]. Therefore, the AGDR block can easily incorporate a broader context to capture long dependencies, while greatly shortening the computational path. It is superior to the recurrent units in parallelization and the entirely attention-based approach in reducing token-pair memory consumption.

4 METHODOLOGY

4.1 Hierarchical Representation

Take word-level embeddings U_k^0 and R^0 as inputs, IACMN hierarchically stacks the AGDR block to harvest multi-grained representations of u_k and r . The output of each layer in a block is reserved as a feature type, denoted as $U_k^{b,l}$ and $R^{b,l}$, namely the l -th layer output of the b -th block. By this means, the semantic and functional relationships among words, n-grams and sub-sequences are iteratively constructed, and more and more sophisticated information are distilled. Let $L = n_b \times n_l$, where n_b and n_l are the number of blocks and internal layers within a block, we define the multi-grained representations as $[U_k^0, \dots, U_k^L]$ and $[R^0, \dots, R^L]$.

4.2 Utterance-Response Matching

A segment-segment matching matrix is calculated for each granularity, i.e., $M_{k,r}^l$, where $l \in \{0, L\}$. The (i,j) -th element of $M_{k,r}^l$ is defined as:

$$M_{i,j} = U_k^l[i] \cdot R^l[j]^T \quad (5)$$

indicating the relevance between the i -th segment in u_k and the j -th segment in r for the l -th representation type. The $L+1$ matching matrices are grouped together like a 2D image, where each matrix acts as a feature channel. Then important matching information is extracted by two-layer convolution & max-pooling operation. Suppose that $M^{(c,l)} = [M_{i,j}^{(c,l)}]_{I^{(c,l)} \times J^{(c,l)}}$ denotes the feature map of the l -th granularity at c -th layer, where $M^{(0,l)} = M_{k,r}^l$. $\forall l \in \{0, L\}, c \in \{1, 2\}$, the convolution transformation is defined as:

$$M_{i,j}^{(c,l)} = \text{ELU} \left(\sum_{l'=0}^{n_{c-1}} \sum_{s=0}^{k_w^{(c,l)}} \sum_{t=0}^{k_h^{(c,l)}} K_{s,t}^{(c,l)} \cdot M_{i+s,j+t}^{(c-1,l')} + b^c \right) \quad (6)$$

where n_{c-1} is the number of feature maps at the $(c-1)$ -th layer, $\mathbf{K}_{(c,l)} \in \mathbb{R}^{k_w^{(c,l)} \times k_h^{(c,l)}}$ is a 2D convolution kernel with the size of $k_w^{(c,l)} \times k_h^{(c,l)}$, and \mathbf{b}^c is the bias for the c -th layer. A max-pooling operation is then adopted as follows:

$$\widehat{\mathbf{M}_{i,j}^{(c,l)}} = \max \left(\mathbf{M}_{[i:i+p_w^{(c,l)}-1], [j:j+p_h^{(c,l)}-1]}^{(c,l)} \right) \quad (7)$$

where $p_w^{(c,l)}$ and $p_h^{(c,l)}$ are the width and height of 2D max-pooling. Outputs of the final layer are flattened and concatenated to form a fused matching vector of (u_k, r) , denoted as $s_{k,r} \in \mathbb{R}^v$.

4.3 Attentive Aggregation

We employ a bidirectional GRU with turns-aware attention to aggregate the multi-turn matching vectors $\{s_{k,r}\}_{k=1}^{n_T}$ (suppose the conversation has n_T turns). The structure of GRU is described as:

$$\begin{aligned} z_t &= \sigma(W_z x_t + V_z h_{t-1}) \\ r_t &= \sigma(W_r x_t + V_r h_{t-1}) \\ \tilde{h}_t &= \tanh(W_h x_t + V_h (r_t \odot h_{t-1})) \\ h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \end{aligned} \quad (8)$$

where σ is the sigmoid function, z_t and r_t are the update and reset gates, \odot denotes the element-wise multiplication, and $W_z, W_r, W_h, V_z, V_r, V_h$ are parameters. The matching sequence is fed into the bi-GRU in two chronological orders:

$$\begin{aligned} \overrightarrow{h}_k &= \overrightarrow{\text{GRU}}(s_{k,r}), k \in [1, n_T], \\ \overleftarrow{h}_k &= \overleftarrow{\text{GRU}}(s_{k,r}), k \in [n_T, 1]. \end{aligned} \quad (9)$$

We first obtain the annotation for the k -th pair by concatenating both directions, i.e., $h_k = [\overrightarrow{h}_k, \overleftarrow{h}_k]$. To reward utterances that are vital clues for selecting the proper reply, we follow [41] and employ an attention mechanism to integrate the hidden output at each step. The attentive aggregation procedure is formulated as:

$$\begin{aligned} u_k &= \tanh(W_s h_k + b_s) \\ \alpha_k &= \frac{\exp(u_k^T u_s)}{\sum_k \exp(u_k^T u_s)} \\ v_o &= \sum_k \alpha_k h_k \end{aligned} \quad (10)$$

where $W_s \in \mathbb{R}^{q \times a}$, $b_s \in \mathbb{R}^a$ and $u_s \in \mathbb{R}^a$ are jointly learned parameters. Concretely, the k -th annotation h_k is first fed through a one-layer perceptron to get a hidden representation u_k . Then the importance of each utterance is measured under the supervision of a global vector u_s , which can be regarded as a virtual high-level query guideline “what are the important utterances” over the whole context. The normalized weights $\{\alpha_k\}_{k=1}^{n_T}$ are calculated by the similarities of $\{u_k\}_{k=1}^{n_T}$ and u_s through a softmax function. The weighted sum of the pair annotations is then taken as the final aggregated matching vector v_o .

The final matching score $g(c, r)$ is obtained via a single-layer perceptron, formulated as:

$$g(c, r) = \sigma(W_o v_o + b_o) \quad (11)$$

where σ is the sigmoid function, and W_o, b_o are parameters. We learn $g(\cdot, \cdot)$ by minimizing cross entropy with \mathcal{D} . Let Θ denotes the parameters of IACMN, the objective function $\mathcal{L}(\mathcal{D}, \Theta)$ is:

$$-\sum_{i=1}^N [y_i \log(g(c_i, r_i)) + (1 - y_i) \log(1 - g(c_i, r_i))] \quad (12)$$

5 EXPERIMENTS

5.1 Experimental Setup

5.1.1 Dataset. We test IACMN on two large-scale public data sets, the Ubuntu Dialogue Corpus V1 [14] and the Douban Conversation Corpus [35]. The statistics are shown in Table 1.

- **Ubuntu Corpus** This corpus consists of English multi-turn dialogues constructed from Ubuntu IRC chat logs. The dataset contains 1 million context-response pairs for training, and 0.5 million pairs for both validation and test sets. For each context, the original human-generated response is labeled as positive while negative responses are randomly selected. The ratio of positive and negative is 1:1 in training, and 1:9 in validation and testing. We use the copy shared by [39] in which numbers, urls, and paths are replaced with special placeholders.

- **Douban Corpus** An open domain dataset constructed from Douban group, which is a popular Chinese social network service. The training set contains 1 million instances and the validation set contains 50k instances, both with 1:1 positive-negative ratio. The test set consists 10k instances, where each context corresponds to 10 responses retrieved from a standard search engine *Apache Lucene*¹ and the labels are manually annotated.

5.1.2 Evaluation Metric. Following [35, 43, 45], we employ $R_n@k$ as the main evaluation metric, which stands for the recall of true positive replies among k selected ones from n available candidates, namely $R_n@k = \frac{\sum_{i=1}^k y_i}{\sum_{i=1}^n y_i}$, where y_i is the binary label. In addition, we also employ MAP (Mean Average Precision) [2], MRR (Mean Reciprocal Rank) [30], and Precision-at-one P@1 for Douban Corpus, as the setting of previous work.

5.2 Baseline

We compare IACMN with the following baselines:

- **Single-matching models:** Models first concatenate all utterances in the context, and then match the long text with a response candidate, including DualEncoder_{lstm/bilstm} [14], MV-LSTM [31], Match-LSTM [33], and Attentive-LSTM [27].
- **Multi-view** [44]: The model leverages a hierarchical recurrent neural network to distill textual information, from both word view and utterance view.
- **DL2R** [40]: The model proposes a contextual query reformulation framework, which combines history utterances in different

¹<http://lucene.apache.org/>

Table 1: Statistics of Ubuntu & Douban datasets.

	Ubuntu Corpus			Douban Corpus		
	train	val	test	train	val	test
# context-response pairs	1M	500K	500K	1M	50K	10K
# candidates per context	2	10	10	2	2	10
Avg. # turns per context	10.13	10.11	10.11	6.69	6.75	6.45
Avg. # words per utterance	11.35	11.34	11.37	18.56	18.50	20.74

Table 2: Evaluation results of IACMN and other comparison baselines on the two datasets.

	Ubuntu Corpus				Douban Conversation Corpus					
	R ₂ @1	R ₁₀ @1	R ₁₀ @2	R ₁₀ @5	MAP	MRR	P@1	R ₁₀ @1	R ₁₀ @2	R ₁₀ @5
DualEncoder _{lstm}	0.901	0.638	0.784	0.949	0.485	0.527	0.320	0.187	0.343	0.720
DualEncoder _{bilstm}	0.895	0.630	0.780	0.944	0.479	0.514	0.313	0.184	0.330	0.716
MV-LSTM	0.906	0.653	0.804	0.946	0.498	0.538	0.348	0.202	0.351	0.710
Match-LSTM	0.904	0.653	0.799	0.944	0.500	0.537	0.345	0.202	0.348	0.720
Attentive-LSTM	0.903	0.633	0.789	0.943	0.495	0.523	0.331	0.192	0.328	0.718
Multi-View	0.908	0.662	0.801	0.951	0.505	0.543	0.342	0.202	0.350	0.729
DL2R	0.899	0.626	0.783	0.944	0.488	0.527	0.330	0.193	0.342	0.705
SMN _{dynamic}	0.926	0.726	0.847	0.961	0.529	0.569	0.397	0.233	0.396	0.724
DUA	-	0.752	0.868	0.962	0.551	0.599	0.421	0.243	0.421	0.780
DAM	0.938	0.767	0.874	0.969	0.550	0.601	0.427	0.254	0.410	0.757
IACMN	0.944	0.782	0.886	0.973	0.571	0.621	0.448	0.269	0.453	0.783
IACMN _{ensemble}	0.948	0.795	0.895	0.976	0.580	0.626	0.451	0.272	0.462	0.787

Table 3: Evaluation results of ablation variants on the two datasets.

	Ubuntu Corpus				Douban Conversation Corpus					
	R ₂ @1	R ₁₀ @1	R ₁₀ @2	R ₁₀ @5	MAP	MRR	P@1	R ₁₀ @1	R ₁₀ @2	R ₁₀ @5
IACMN	0.944	0.782	0.886	0.973	0.571	0.621	0.448	0.269	0.453	0.783
IACMN _{last_layer}	0.939	0.770	0.880	0.971	0.557	0.603	0.430	0.255	0.439	0.757
IACMN _{last_block}	0.933	0.751	0.862	0.965	0.555	0.599	0.425	0.252	0.436	0.761
IACMN-Self Attention	0.931	0.754	0.868	0.963	0.554	0.600	0.412	0.248	0.440	0.771
IACMN-Residual	0.940	0.772	0.881	0.969	0.562	0.607	0.427	0.260	0.440	0.768
IACMN-GLU2ReLU	0.941	0.776	0.882	0.970	0.565	0.611	0.440	0.262	0.449	0.781

strategies with the input message to obtain a reformulated query for response selection.

- **SMN_{dynamic}** [35]: The model presents a comparison-aggregation architecture, which makes the response interacts with each utterance and then accumulates all the pair matching vectors.
- **DUA** [43]: The model formulates previous utterances in the context using a proposed deep utterance aggregation model to form a fine-grained context representation.
- **DAM** [45]: The model utilizes stacked self-attention and cross-attention learned from Transformer [29] to construct multi-grained input representations.

5.3 Ablation Test

In addition to comparing IACMN with state-of-the-art techniques, we further provide five ablation tests, in order to get a better understanding of whether each key component in the hierarchical AGDR blocks plays a crucial role in the proposed model. The ablation variants are as follows:

- **IACMN_{last_layer}**: In this variant, we extract and integrate the feature maps of the last layer in each stacked block as the multi-view representations of utterances and response candidates.
- **IACMN_{last_block}**: This variant uses only the output of the top layer in the last block (namely, the L -th embedding type) to represent each utterance and response for matching. Due to

- the hierarchical representation architecture, high-level features synthesize information from low-level features, and can model more complex lexical and semantic clues.
- **IACMN-Self Attention:** This variant discards the self-attention operation of each layer in the AGDR block.
 - **IACMN-Residual:** This variant removes the residual connections in AGDR that are applied to the self-attention and gated dilated-convolution operations.
 - **IACMN-GLU2ReLU:** In this variant, we replace the gated linear unit in Formula 4 with the ReLU activation, which is defined as:

$$\text{ReLU}(x) = \max(0, x) \quad (13)$$

5.4 Training Details

IACMN is implemented using tensorflow². We set at most 50 words for each utterance, and the number of turns is 15 for Ubuntu dataset and 10 for Douban dataset. Truncating and zero-padding were applied to the variable-sized inputs. Word embeddings for Ubuntu corpus were trained by word2vec [17] on the training data and the dimension is 200. For Douban corpus, word embeddings were the concatenation of 200-dimensional pre-trained embeddings from [24] and 200-dimensional embeddings learned from the training data via word2vec. All word embeddings were kept fixed during the training process. We tested stacking 1-3 AGDR blocks and 1-4 layers within each block. The reported results utilize 2×3 compositional hierarchy (i.e., 2 stacks of 3-layer blocks) as it gains the best scores on the validation set. Correspondingly, the dilation rates are [1-2-4] for each layer in a block. The width and channel of all convolution kernels are 3 and 150 respectively, and the window size of convolution & pooling is (3,3). The number of feature maps is 32 for the first layer and 16 for the second layer. For aggregation module, the hidden state of each direction in bi-GRU has a dimension of 128, and the dimension of the virtual global vector is 50. We use Adam Optimizer [10] to train IACMN and other ablation variants. The learning rate is initialized as 1e-3 and gradually decreased during training. The mini-batch size is 100, early-stopping and dropout [25] with a rate of 0.2 are applied.

5.5 Evaluation Results

5.5.1 Comparison with Baselines. Table 2 shows the evaluation results of our model, and we copy the reported results of all baselines for comparison. As demonstrated, IACMN outperforms other methods in terms of all metrics on both datasets, including the present state-of-the-art baseline, illustrating its ability to model the multi-turn context to select the best matching reply. The performance of single-matching models that concatenate all previous utterances together for matching is relatively poor, which indicates the advantage of the *comparison-aggregation* architecture in synthetically detecting fine-grained interactive matching information from multiple utterance-response pairs. One notable point is that, IACMN is about 1.5 faster than DAM and 3.5 faster than SMN_{dynamic} in our implementation, benefiting from the high parallelism and simple computational path of dilated-convolution and self-attention. This reveals that IACMN is more suitable for deployment in real-time applications.

²<https://www.tensorflow.org>

In addition, we further examine the performance of an ensemble method IACMN_{ensemble}, based on the average outputs of five individual models with identical structures and different random initializations. The results show that the ensemble model achieves a significant improvement in all evaluation metrics.

5.5.2 Comparison with Ablation Variants. Experimental results of different ablation variants are shown in Table 3. We can find that:

- (1) Both IACMN_{last_layer} and IACMN_{last_block} show a decrease in performance compared to IACMN. Note that the former with more embedding types is better than the later, which indicates that although higher level representation embeddings can extract and gather useful information from low levels through the hierarchical construction mechanism, the efficiency of performing interactive matching on the representation at each granularity is still obvious for this task.
- (2) Removing each component of AGDR will affect the performance of the entire model. The self-attention mechanism has the greatest impact, which illustrates the importance of modeling long-term dependency information from a global perspective. Besides, though the residual connection and the gated linear unit are both designed to prevent the gradient from vanishing or exploding, the latter has only a slight effect on the results after being replaced with ReLU. This might due to the effectiveness of the residual network for the multi-layer stacked structure.

6 ANALYSIS

6.1 Quantity Analysis

IACMN involves several parameters (i.e., context and utterance length, number of stacked blocks, number of layers within a block, hidden state dimension of GRU in the aggregation module, with or without attention mechanism). To investigate how different choices of parameters affect the performance of IACMN, we provide comparative experiments and detailed analysis of the results. Except for the parameter being tested, we set the other parameters to their default values.

6.1.1 Context Length & Utterance Length. We study how the number of history turns in context and the length of utterances influence the performance of IACMN. Figure 6(a), 6(b) and 6(c) show the changes of different evaluation metrics on Douban and Ubuntu datasets with respect to the number of history turns. As demonstrated, there is a similar trend for all curves: the performance increases significantly with longer context in the initial phase and then shows slight fluctuations after the length of context reaches 10. It indicates that considering only a few previous utterances will lead to the loss of much important information for response selection, and when the number of turns expands, more matching clues as well as irrelevant noise will be introduced to the model. Besides, we observe that the influence on P@1 and R₁₀@1 is relatively more dramatic across different evaluation metrics. The reason might be that they are the most significant and difficult measurements for evaluating the matching score of the selected response candidate.

Figure 6(d) illustrates how the performance of R₁₀@1 on Ubuntu and Douban datasets changes with respect to different utterance

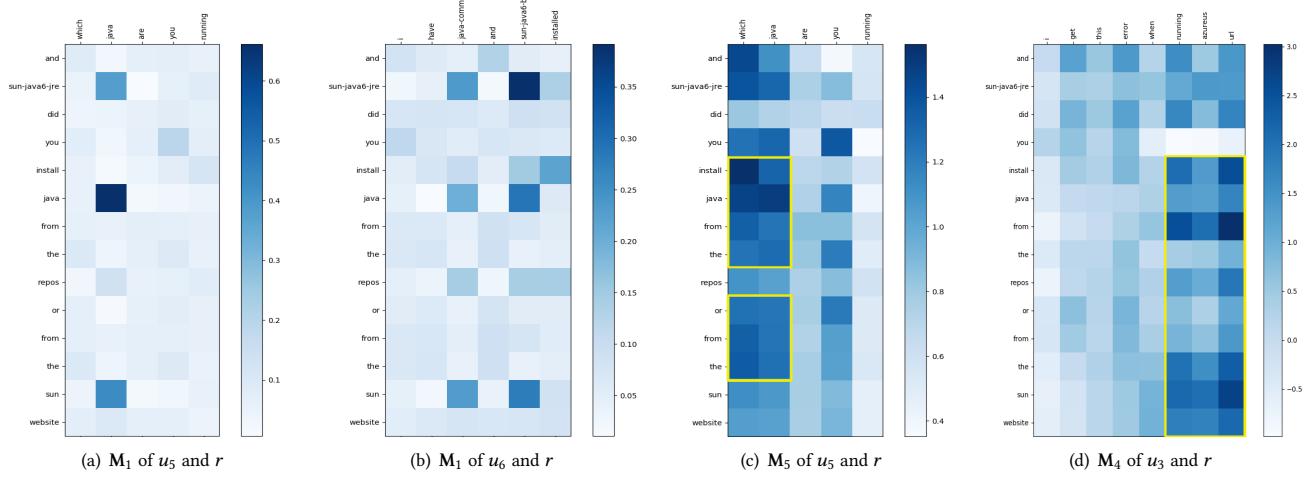


Figure 5: Matching matrices visualization, darker area means larger value.

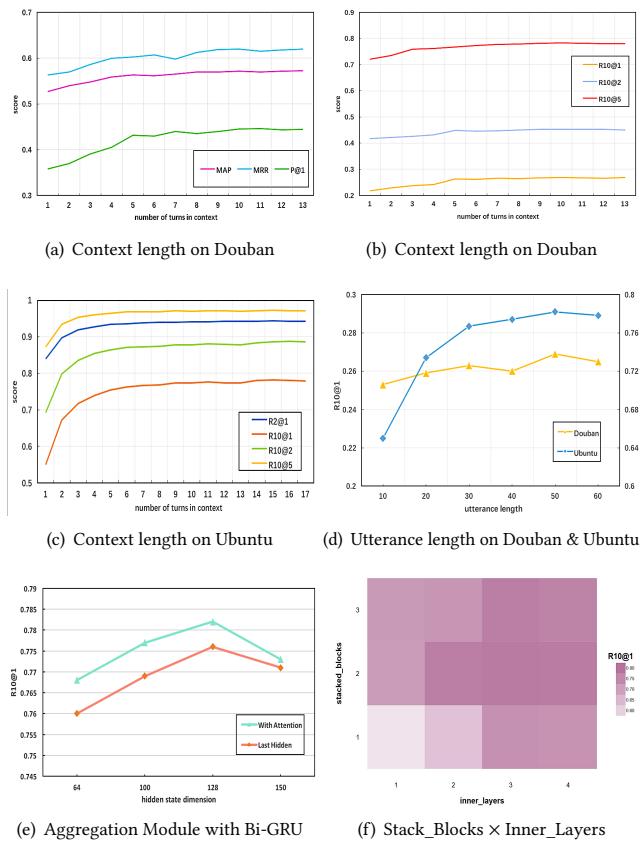


Figure 6: Comparison of variants with different parameters.

length. The performance of input text with only 10 words is obviously lower than the longer ones, as information within each text is limited for semantic and functional matching. Based on hierarchical stacking of the AGDR blocks, IACMN can capture more long

dependency relationship of context utterances and the response. Due to the balance between efficacy and computational cost, we set the utterance length to 50, the number of turns to 15 and 10 for Ubuntu corpus and Douban corpus respectively in our reported experimental results.

6.1.2 Aggregation Module with Bi-GRU. We further investigate the influence of hidden state dimension of bi-GRU and the effectiveness of the turns-aware attention mechanism in the aggregation module. With annotations of the multi-turn matching vectors $[h_1, \dots, h_{n_T}]$ in bi-GRU obtained from Formula 9, we compare two fusion methods: (1) taking the last hidden state h_{n_T} to calculate the final matching score; (2) combining the annotations into a fused vector under the supervision of the attention mechanism denoted in Formula 10. We examine the score of $\mathbf{R}_{10}@1$ on Ubuntu dataset, experimental results are shown in Figure 6(e). We can see that applying the attention mechanism consistently performs better than using the final state directly. In other words, even though the bi-GRU has the ability to select useful information from the matching sequence with its gate mechanism, the turns-aware attention design can still contribute to improving the performance.

6.1.3 Combination of Stack_Blocks × Inner_Layers. Figure 6(f) shows the performance on Ubuntu dataset across different number of stacked AGDR blocks and intra-block layers, where darker areas represent larger values. As demonstrated, the more blocks we stack, the higher the score of $\mathbf{R}_{10}@1$. Even if only stacking a 2-layer block twice, IACMN can also achieve a competitive result. In terms of the iterated layers within a block, the performance improves significantly when the depth of the layer is less than 3, while the 4-layer structure is not better than the 3-layer one. The reason is might that the perceived field of dilated convolution filters at each layer in a 3-layer block (the dilation rates are [1-2-4] correspondingly) ranges among [3-7-15], which is sufficient to model multiple n-gram features through the hierarchical composition of local interactions, compared to the average length of utterances.

6.2 Case Study (Visualization)

For a better insight into how IACMN captures multi-view interactive information and how it selects important matching vectors in aggregation module, we perform a case study by visualizing the similarity matrices and the turns-aware attention weights. The example is shown in Figure 7, which comes from the test sets of Ubuntu Corpus, and our model successfully selected the best matching reply for it.

6.2.1 Visualization of Matching Matrices. We study the effectiveness of stacking AGDR blocks to construct hierarchical text representations. Figure 5 gives the visualization results of the matching matrices (calculated by formula 5) at different granularities, denoted as M_k . We can see that the important matching information captured by the $1^{st} - level$ matching matrices (M_1 , shown in Figure 5(a) and Figure 5(b)) were mainly lexical relevance. For example, the word “java” in u_5 had higher correlation with the words “java”, “sun” and “sun-java6-jre” in r , which may due to their similar co-occurrence information encoded in word embeddings. Between u_6 and r , associated pairs like “installed”&“install”, “java-common”&“sun-java6-jre”, “sun-java6-bin”&“sun-java6-jre | sun” were also successfully identified. Differently, the function of higher-level matching matrices lies in two aspects:

- (1) The ability to identify more sophisticated semantic structures and latent long-term dependencies. In Figure 5(c), the intersection areas between the segment “which java” in u_5 and the textual structure “install java from ... or from ...” in r significantly got larger matching scores. This is reasonable because both expressions are about “the installed version of java”. Although such semantic relationships are implicit and long-term dependent, IACMN can effectively capture the matching signals.
- (2) The ability to distinguish more important segments of a sentence and ignore the useless parts, from the perspective of global interaction. Figure 5(d) reflects this function. The matching signals were mainly centralized in the latter parts of u_3 and r , while their former parts with less information were given small weights in the matching matrix.

6.2.2 Visualization of Turns-Aware Attention. To examine whether the turns-aware attention mechanism in the aggregation module helps to recognize the different correlations of multiple history utterances for selecting the response, Figure 7 illustrates the normalized weights of the annotations for each utterance-response pair in bi-GRU, calculated by the softmax operation in formula 10. As demonstrated, our model significantly identifies the informative utterances like u_4-u_6 , and chooses to discard u_2 , which is unrelated to the topic and has little information.

6.3 Error Analysis

Although IACMN significantly outperforms the baseline methods on the two data sets in all evaluation metrics, there are still several problems that cannot be handled perfectly. In order to understand the limitations of IACMN and where the future improvements might lie, we further analyze the cases that failed to correctly select the best response candidate. We find three major problems that need further investigation:

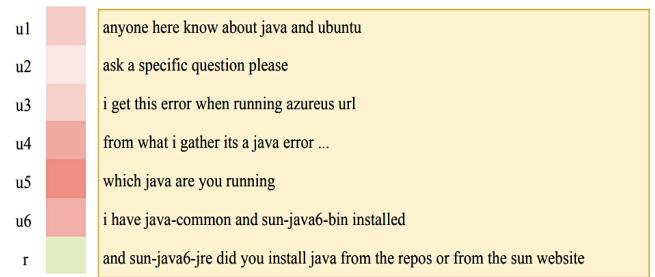


Figure 7: Turns-aware attention weights distribution in Bi-GRU. u_1-u_6 are the utterances in the context, and the last line is the response. The color bar on the left represents the attention weight of each turn, darker area means larger value.

- **Confusing Candidate.** Some candidates are similar to the real response and have reasonable correlations with the conversation context. This increases the difficulty of properly ranking replies in the candidate pool, especially for more stringent evaluation metrics, such as $R_{10}@1$ and $P@1$.
- **Universal Reply.** Universal expressions that are short and meaningless such as “hmm”, “thanks”, “I don’t know” can sometimes be regarded as general replies, although not relevant to a particular topic. This will increase the diversity of possible choices.
- **Logical Consistency.** IACMN has the advantage of modeling multi-view interactions between the context and response, including lexical, semantic and functional matching information, but is limited in understanding the logical consistency. For example, given a context { *A: Wouldn’t you get fat if ate this? B: You look so thin in the picture. A: I used to weigh 40 kilos, believe it? Now I’m almost 50 kilos.* }, the response candidate { *how did you get thinner?* } is logically contradictory to the previous conversation, but it is easy to confuse the model due to the strong lexical similarity. It illustrates that logical consistency is another important clue for multi-turn context modeling and matching.

7 CONCLUSION AND FUTURE WORK

In this paper, we investigate a new deep matching network for multi-turn response selection in retrieval-based chatbots. We propose AGDR, a refined combination of gated dilated-convolution and self-attention to iteratively construct multi-grained representations of the response candidate and its multi-turn history context. The architecture is superior to the recurrent units in parallelization and the entirely attention-based approach in reducing token-pair memory consumption. The interactive information between each utterance-response pair is extracted and integrated from different views, and then accumulated into a fused vector to calculate the final matching score. Experiments on two large-scale public datasets demonstrate that our model significantly outperforms the state-of-the-art methods in terms of all metrics. To better understand the contribution of different components of the model, we provide thorough ablation tests, quantity and visualization analysis as well as case studies. In the future, we would like to explore how to improve the modeling of complicated contextual dependencies such as logical consistency, and how to design models that can be better applied to practical applications.

REFERENCES

- [1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *CoRR abs/1607.06450* (2016).
- [2] Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. 1999. *Modern Information Retrieval*. ACM Press / Addison-Wesley.
- [3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).
- [4] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR abs/1412.3555* (2014).
- [5] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *ICML (Proceedings of Machine Learning Research)*, Vol. 70. PMLR, 933–941.
- [6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*. IEEE Computer Society, 770–778.
- [7] 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*. 2042–2050.
- [8] Zongcheng Ji, Zhengdong Lu, and Hang Li. 2014. An information retrieval approach to short text conversation. *arXiv preprint arXiv:1408.6988* (2014).
- [9] Rudolf Kadlec, Martin Schmid, and Jan Kleinmuntz. 2015. Improved deep learning baselines for ubuntu corpus dialogs. *arXiv preprint arXiv:1510.03753* (2015).
- [10] Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.
- [11] Feng-Lin Li, Minghui Qiu, Haiqing Chen, Xiongwei Wang, Xing Gao, Jun Huang, Juwei Ren, Zhongzhou Zhao, Weipeng Zhao, Lei Wang, Guwei Jin, and Wei Chu. 2017. *AliMe Assist : An Intelligent Assistant for Creating an Innovative E-commerce Experience*. In *CIKM*. ACM, 2495–2498.
- [12] Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. 2016. Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541* (2016).
- [13] Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547* (2017).
- [14] 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 Conference*. The Association for Computer Linguistics, 285–294.
- [15] Ryan Thomas Lowe, Nissan Pow, Iulian Vlad Serban, Laurent Charlin, Chia-Wei Liu, and Joelle Pineau. 2017. Training end-to-end dialogue systems with the ubuntu dialogue corpus. *Dialogue & Discourse* 8, 1 (2017), 31–65.
- [16] Zhengdong Lu and Hang Li. 2013. A deep architecture for matching short texts. In *Advances in neural information processing systems*. 1367–1375.
- [17] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*. 3111–3119.
- [18] Lili Mou, Yiping Song, Rui Yan, Ge Li, Lu Zhang, and Zhi Jin. 2016. Sequence to backward and forward sequences: A content-introducing approach to generative short-text conversation. *arXiv preprint arXiv:1607.00970* (2016).
- [19] Iulian Vlad Serban, Tim Klinger, Gerald Tesauro, Kartik Talamadupula, Bowen Zhou, Yoshua Bengio, and Aaron Courville. 2017. Multiresolution recurrent neural networks: An application to dialogue response generation. In *AAAI*. AAAI Press, 3288–3294.
- [20] Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Thirtyeth AAAI Conference on Artificial Intelligence*.
- [21] Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *AAAI*. AAAI Press, 3295–3301.
- [22] Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. *arXiv preprint arXiv:1503.02364* (2015).
- [23] Heung-Yeung Shum, Xiaodong He, and Di Li. 2018. From Eliza to XiaoIce: challenges and opportunities with social chatbots. *Frontiers of IT & EE* 19, 1 (2018), 10–26.
- [24] Yan Song, Shuming Shi, Jing Li, and Haisong Zhang. 2018. Directional Skip-Gram Explicitly Distinguishing Left and Right Context for Word Embeddings. In *NAACL-HLT (2)*. Association for Computational Linguistics, 175–180.
- [25] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.
- [26] Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. 2017. Fast and accurate entity recognition with iterated dilated convolutions. In *EMNLP*. Association for Computational Linguistics, 2670–2680.
- [27] Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2015. LSTM-based deep learning models for non-factoid answer selection. *CoRR abs/1511.04108* (2015).
- [28] A Turing. 1950. Computing machinery and intelligence. *Mind* 1 (236): 433–460. *Reprinted as* (1950), 40–66.
- [29] 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 *NIPS*. 6000–6010.
- [30] Ellen M Voorhees et al. 1999. The TREC-8 question answering track report. In *Trec*, Vol. 99. Citeseer, 77–82.
- [31] Shengxian Wan, Yanyan Lan, Jun Xu, Jiafeng Guo, Liang Pang, and Xueqi Cheng. 2016. Match-srnn: Modeling the recursive matching structure with spatial rnn. In *IJCAI*. IJCAI/AAAI Press, 2922–2928.
- [32] Hao Wang, Zhengdong Lu, Hang Li, and Enhong Chen. 2013. A dataset for research on short-text conversations. In *EMNLP*. ACL, 935–945.
- [33] Shuhong Wang and Jing Jiang. 2016. Learning natural language inference with LSTM. In *HLT-NAACL*. The Association for Computational Linguistics, 1442–1451.
- [34] Felix Wu, Ni Lao, John Blitzer, Guandao Yang, and Kilian Q. Weinberger. 2017. Fast Reading Comprehension with ConvNets. *CoRR abs/1711.04352* (2017).
- [35] 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 (1)*. Association for Computational Linguistics, 496–505.
- [36] Yu Wu, Wei Wu, Dejian Yang, Can Xu, and Zhoujun Li. 2018. Neural response generation with dynamic vocabularies. In *AAAI*. AAAI Press, 5594–5601.
- [37] Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2017. Topic aware neural response generation. In *Thirty-First AAAI Conference on Artificial Intelligence*.
- [38] Zhen Xu, Bingquan Liu, Baoxun Wang, SUN Chengjie, Xiaolong Wang, Zhuoran Wang, and Chao Qi. 2017. Neural response generation via gan with an approximate embedding layer. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 617–626.
- [39] Zhen Xu, Bingquan Liu, Baoxun Wang, Chengjie Sun, and Xiaolong Wang. 2016. Incorporating loose-structured knowledge into lstm with recall gate for conversation modeling. *CoRR abs/1605.05110* (2016).
- [40] 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.
- [41] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *HLT-NAACL*. The Association for Computational Linguistics, 1480–1489.
- [42] Fisher Yu and Vladlen Koltun. 2016. Multi-Scale Context Aggregation by Dilated Convolutions. In *ICLR*.
- [43] Zhuosheng Zhang, Jiangtong Li, Pengfei Zhu, Hai Zhao, and Gongshen Liu. 2018. Modeling multi-turn conversation with deep utterance aggregation. In *COLING*. Association for Computational Linguistics, 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*. The Association for Computational Linguistics, 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 (1)*. Association for Computational Linguistics, 1118–1127.