Sequential Match Network: A New Architecture for Multi-turn Response Selection in Retrieval-based Chatbots

Yu Wu[†*] , **Wei Wu**[‡] , **Zhoujun Li**[†] , **Ming Zhou**[‡]
[†]State Key Lab of Software Development Environment, Beihang University, Beijing, China
[‡] Microsoft Research, Beijing, China
{wuyu,lizj}@buaa.edu.cn {wuwei,mingzhou}@microsoft.com

arXiv:1612.01627v1 [cs.CL] 6 Dec 2016

Abstract

We study response selection for multi-turn conversation in retrieval based chatbots. Existing works either ignores relationships among utterances, or misses important information in context when matching a response with a highly abstract context vector finally. We propose a new session based matching model to address both problems. The model first matches a response with each utterance on multiple granularities, and distills important matching information from each pair as a vector with convolution and pooling operations. The vectors are then accumulated in a chronological order through a recurrent neural network (RNN) which models the relationships among the utterances. The final matching score is calculated with the hidden states of the RNN. Empirical study on two public data sets shows that our model can significantly outperform the state-of-the-art methods for response selection in multi-turn conversation.

1 Introduction

Traditional research in human-computer conversation focused on building task-oriented dialog systems in vertical domains to help people complete specific tasks such as ordering and tutoring etc (Boden, 2006; Wallace, 2009; Young et al., 2010). Recently, with the large amount of conversation data available on Internet, there has been a surge of interest in building non-task-oriented chatbots that can naturally and meaningfully converse with humans on open domain topics (Jafarpour et al., 2010; Ritter et al., 2011). Existing work on building chatbots includes generation based methods and retrieval based methods. In this work, we study retrieval based chatbots, because they select responses from an index of existing conversation and thus can leverage the existing search power and always return fluent responses. While most existing work on retrieval based chatbots studies response selection for single-turn conversation (Wang et al., 2013; Wang et al., 2015), we consider the problem in a multi-turn scenario which is the nature of conversation but has not been well explored yet.

Table 1: An example of multi-turn conversation

Context 1: I am going to **hold a drum class** in Shanghai. Anyone wants to join? The location is near the Bound.
Context 2: Interesting! Do you have coaches who can help me practice **drum**?
Context 3: Of course.
Message: Can I have a free first lesson?

Different from response selection in single-turn conversation in which one only needs to match a response with a single input message, response selection in multi-turn conversation requires matching between a response and a conversation session in which one needs to consider not only the matching between the response and the input message but also the matching between the response and the utterances in previous turns as context. The challenges of the task include (1) how to identify important information (words, phrases, and sentences) in context that is crucial to selecting a proper response for the session and how to leverage the information in matching; and (2) how to model relationships among the utterances. Table 1 illustrates the challenges with an example. First, "hold a drum class" and "drum" in the context are very important. Without them, one may find responses relevant to the message but nonsense in the session (e.g., "what lessons do you

*The work was done when the first author was an intern in Microsoft Research Asia.

want?"). On the other hand, although "Shanghai", "the Bund", and "coaches" are also keywords in their utterances, they are useless and even noise to response selection. It is crucial yet non-trivial to extract the important information from the context and leverage them in matching while circumvent the noise. Second, the message highly depends on Context 1, and the order of the utterances matters in response selection: exchanging Context 2 and the message may lead to different responses. Existing work, however, either ignores relationships among utterances (Lowe et al., 2015; Yan et al., 2016), or loses important information in context in the process of converting the whole session to a vector without enough supervision from responses (Lowe et al., 2015; Zhou et al., 2016).

We propose a new session based matching model which can tackle both challenges in an end-to-end way. One major problem suffered by the existing models is that responses in matching cannot meet the session until the final step, which results in information loss. To overcome this drawback, our model matches a response with each utterance in the session (message and context) at the very beginning. For each utterance-response pair, the model constructs a word-word similarity matrix and a sequence-sequence similarity matrix by the embedding of words and the hidden states of a recurrent neural network with gated unites (GRU) (Chung et al., 2014) respectively. The two matrices capture important matching information in the pair on a word level and a segment level respectively, and the information is distilled and fused as a matching vector through an alternation of convolution and pooling operations on the matrices. By this means, important information in context is recognized under sufficient supervision from the response and carried into matching with minimal loss. The matching vectors are then uploaded to a GRU to form a matching score for the session and the response. The GRU accumulates the pair matching in its hidden states in the chronological order of the utterances in the session. It models the relationships and the dependencies among the utterances in a matching fashion and has the utterance order supervise the accumulation of pair matching. The gate mechanism of the GRU helps select important pairs and filter out noise. The matching degree of the session and the response is computed by a logit model with the hidden states of the GRU. Our model extends the powerful "2D" matching paradigm in text pair matching for single-turn conversation to session based matching for multi-turn conversation, and enjoys the advantage that both important information in utterance-response pairs and relationships among utterances are sufficiently preserved and leveraged in matching.

We test our model on the Ubuntu dialogue corpus (Lowe et al., 2015) which is a large public English data set for research in multi-turn conversation. The results show that our model can significantly outperform the state-of-the-art methods, and improvement to the best baseline model on R_{10}@1 is over 6%. One problem with the Ubuntu data is that negative examples are randomly sampled which might oversimplify the multi-turn problem in a real retrieval based chatbot. To further verify the efficacy of the proposed model in a real situation, we simulate the procedure of a retrieval based chatbot and create a large scale Chinese test set. Instead of negative sampling, labels in the data are generated by 3 human judges. On this data, our model improves the best baseline model over 4% on P@1 (equivalent to R_{10}@1). We published the data at `https://github.com/MarkWuNLP/MultiTurnResponseSelection`.

Our contributions in this paper are three-folds: (1) proposal of a new session based matching model for multi-turn response selection in retrieval based chatbots; (2) empirical verification of the effectiveness of the model on public data sets; (3) publication of a large human labeled data set to research communities.

2 Related Work

Early work (Weizenbaum, 1966) on chatbots exploits hand crafted templates to generate responses, which requires huge human effort and is not scalable. Recently, data driven approaches (Ritter et al., 2011; Higashinaka et al., 2014) have drawn a lot of attention. Existing work along this line includes retrieval based methods and generation based methods. The former selects a proper response from an index based on matching between the response and an input message with or without context (Hu et al., 2014; Ji et al., 2014; Wang et al., 2015; Yan et al., 2016; Wu et al., 2016; Zhou et al., 2016), while the latter employs statistical machine translation techniques (Ritter et al., 2011) or the sequence to sequence framework

(Shang et al., 2015; Serban et al., 2015; Vinyals and Le, 2015; Li et al., 2015; Li et al., 2016; Xing et al., 2016; Serban et al., 2016) to generate responses. Our work belongs to retrieval based methods, and we study response selection with context information.

Early studies of retrieval based chatbots focus on response selection for single-turn conversation (Wang et al., 2013; Ji et al., 2014; Wang et al., 2015; Wu et al., 2016). Recently, researchers begin to pay attention to multi-turn conversation. For example, Lowe et al. (Lowe et al., 2015) match a response with the literal concatenation of context utterances. Yan et al. (Yan et al., 2016) concatenate context utterances with the input message as reformulated queries and perform matching with a deep neural network architecture. Zhou et al. (Zhou et al., 2016) improve multi-turn response selection with a multi-view model including an utterance view and a word view. The stark difference between our model and the existing models is that our model matches a response with each utterance at the very first and matching information instead of sentences is accumulated in a temporal manner through a GRU.

3 Matching Approach

3.1 Problem Formalization

Suppose that we have a data set $\mathcal{D} = \{(y_i, s_i, r_i)\}_{i=1}^N$, where $s_i = \{u_{i,1}, \ldots, u_{i,n_i}\}$ represents a conversation session with $\{u_{i1}, \ldots, u_{i,n_i-1}\}$ utterances in context and u_{i,n_i} an input message. r_i is a response candidate and $y_i \in \{0, 1\}$ denotes a label. $y_i = 1$ means r_i is a proper response for s_i, otherwise $y_i = 0$. Our goal is to learn a matching model $g(\cdot, \cdot)$ with $\mathcal{D}$. For any session-response pair (s, r), $g(s, r)$ measures the matching degree between s and r.

3.2 Model Overview

Figure 1 gives the architecture of our model. The model first decomposes session-response matching into several utterance-response pair matching and then all pair matching are accumulated as a session based matching through a recurrent neural network. Specifically, the model consists of two layers. The first layer matches a response candidate with each utterance (context and message) in the session on a word level and a segment level. An utterance-response pair is transformed to a word-word similarity matrix and a sequence-sequence similarity matrix, and important matching information in the pair is distilled from the two matrices and encoded in a matching vector. The matching vectors are then fed to the second layer where they are accumulated in the hidden states of a recurrent neural network with gated unites (GRU) following the chronological order of the utterances in the session. The matching degree of the session and the response is calculated with the hidden states of the GRU.

Our model enjoys several advantages over the existing models. First, a response candidate can meet each utterance in the session at the very beginning of the whole matching procedure, thus matching information in every utterance-response pair can be sufficiently extracted and carried to the final matching score with minimal loss. Second, information extraction from each utterance is conducted on different granularities and under sufficient supervision from the response, thus semantic structures that are useful to response selection in each utterance can be well identified and extracted. Third, matching and utterance relationships are coupled rather than separately modeled, thus utterance relationships (e.g., order), as a kind of knowledge, can supervise the formation of the matching score.

By taking utterance relationships into consideration, our model extends the "2D" matching which has proven effective in text pair matching for single-turn response selection to sequential "2D" matching for session based matching in response selection for multi-turn conversation. We name our model "Sequential Match Network" (SMN). In the following sections, we will describe details of the two layers.

3.3 Utterance-Response Matching

At the first layer, given an utterance u in a session s and a response candidate r, the model looks up an embedding table and represents u and r as $\mathbf{U} = [e_{u,1}, \ldots, e_{u,n_u}]$ and $\mathbf{R} = [e_{r,1}, \ldots, e_{r,n_r}]$ respectively, where $e_{u,i}, e_{r,i} \in \mathbb{R}^d$ are the embeddings of the i-th word of u and r respectively. $\mathbf{U} \in \mathbb{R}^{d \times n_u}$ and $\mathbf{R} \in \mathbb{R}^{d \times n_r}$ are then used to construct a word-word similarity matrix $\mathbf{M}_1 \in \mathbb{R}^{n_u \times n_r}$ and a sequence-sequence similarity matrix $\mathbf{M}_2 \in \mathbb{R}^{n_u \times n_r}$ which are two input channels of a convolutional neural network (CNN). The CNN distills important matching information from the matrices and encodes the information into a

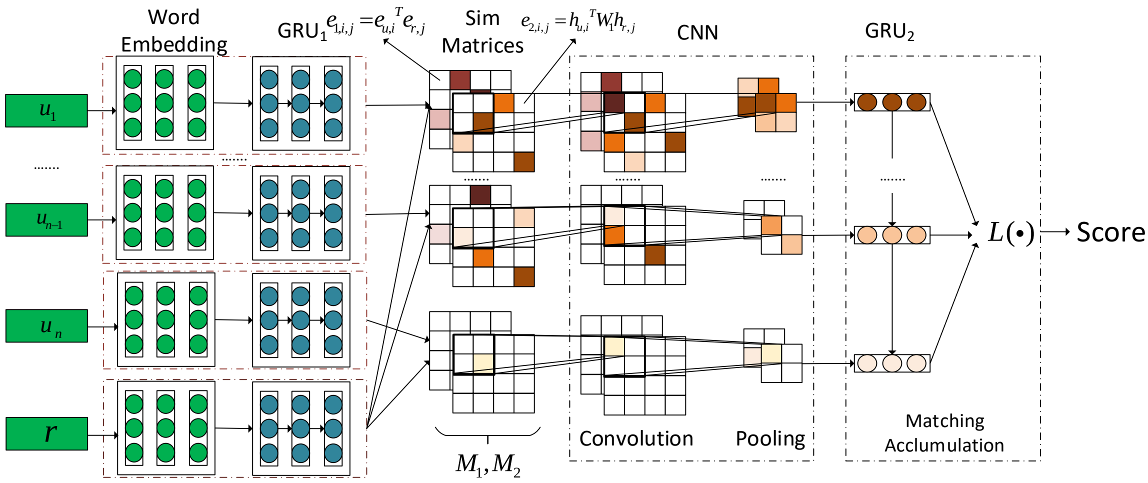


Figure 1: Architecture of SMN

matching vector v.

Specifically, $\forall i, j$, the (i, j)-th element of $\mathbf{M}_1$ is defined by

$$e_{1,i,j} = e_{u,i}^{\top} \cdot e_{r,j}. \tag{1}$$

$\mathbf{M}_1$ models the matching between u and r on a word level.

To construct $\mathbf{M}_2$, we first employ a recurrent neural network with gated units (GRU) (Chung et al., 2014) to transform $\mathbf{U}$ and $\mathbf{R}$ to hidden vectors. Suppose that $\mathbf{H}_u = [h_{u,1}, \ldots, h_{u,n_u}]$ is the hidden vectors of $\mathbf{U}$, then $\forall i$, $h_{u,i} \in \mathbb{R}^m$ is defined by

$$\begin{aligned}
z_i &= \sigma(\mathbf{W_z} e_{u,i} + \mathbf{U_z} h_{u,i-1}) \\
r_i &= \sigma(\mathbf{W_r} e_{u,i} + \mathbf{U_r} h_{u,i-1}) \\
\widetilde{h}_{u,i} &= tanh(\mathbf{W_h} e_{u,i} + \mathbf{U_h}(r_i \odot h_{u,i-1})) \\
h_{u,i} &= z_i \odot \widetilde{h}_{u,i} + (1 - z_i) \odot h_{u,i-1},
\end{aligned} \tag{2}$$

where $h_{u,0} = 0$, z_i and r_i are an update gate and a reset gate respectively, $\sigma(\cdot)$ is a sigmoid function, and $\mathbf{W_z}$, $\mathbf{W_h}$, $\mathbf{W_r}$, $\mathbf{U_z}$, $\mathbf{U_r}$,$\mathbf{U_h}$ are parameters. Similarly, we have $\mathbf{H}_r = [h_{r,1}, \ldots, h_{r,n_r}]$ as the hidden vectors of $\mathbf{R}$. Then, $\forall i, j$, the (i, j)-th element of $\mathbf{M}_2$ is defined by

$$e_{2,i,j} = h_{u,i}^{\top} \mathbf{W_1} h_{r,j}, \tag{3}$$

where $\mathbf{W_1} \in \mathbb{R}^{m \times m}$ is a linear transformation. $\forall i$, GRU models the sequential relationship and the dependency among words up to position i and encodes the text segment until the i-th word to a hidden vector. Therefore, $\mathbf{M}_2$ models the matching between u and r on a segment level.

$\mathbf{M}_1$ and $\mathbf{M}_2$ are then processed by a CNN to form v. $\forall i = 1, 2$, CNN regards $\mathbf{M}_i$ as an input channel, and alternates convolution and max-pooling operations. Suppose that $z^{(l,f)} = \left[z_{i,j}^{(l,f)}\right]_{I^{(l,f)} \times J^{(l,f)}}$ denotes the output of feature maps of type-f on layer-l, where $z^{(0,f)} = \mathbf{M}_f$, $\forall f = 1, 2$. On the convolution layer, we employ a 2D convolution operation with a window size $r_w^{(l,f)} \times r_h^{(l,f)}$, and define $z_{i,j}^{(l,f)}$ as

$$z_{i,j}^{(l,f)} = \sigma(\sum_{f'=0}^{F_{l-1}} \sum_{s=0}^{r_w^{(l,f)}} \sum_{t=0}^{r_h^{(l,f)}} \mathbf{W}_{s,t}^{(l,f)} \cdot z_{i+s,j+t}^{(l-1,f')} + \mathbf{b}^{l,k}), \tag{4}$$

where $\sigma(\cdot)$ is a ReLU, $\mathbf{W}^{(l,f)} \in \mathbb{R}^{r_w^{(l,f)} \times r_h^{(l,f)}}$ and $\mathbf{b}^{l,k}$ are parameters, and F_{l-1} is the number of feature maps on the $(l-1)$-th layer. An max pooling operation follows a convolution operation and can be formulated as

$$z_{i,j}^{(l,f)} = \max_{p_w^{(l,f)} > s \geq 0} \max_{p_h^{(l,f)} > t \geq 0} z_{i+s,j+t}, \tag{5}$$

where $p_w^{(l,f)}$ and $p_h^{(l,f)}$ are the width and the height of the 2D pooling respectively. The output of the final feature maps are concatenated and mapped to a low dimensional space with a linear transformation as the matching vector $v \in \mathbb{R}^q$.

From Equation (1), (3), (4), and (5), we can see that by learning word embedding and parameters of GRU from training data, words or segments in an utterance that are useful to recognize the appropriateness of a response may have high similarity with some words or segments in the response and result in high value areas in the similarity matrices. These areas will be transformed and selected by convolution and pooling operations and carry the important information in the utterance to the matching vector. This is how our model identifies important information in context and leverage it in matching under the supervision of the response. We consider multiple channels because we want to capture important matching information on multiple granularities of text.

3.4 Matching Accumulation

Suppose that $[v_1, \ldots, v_n]$ is the output of the first layer (corresponding to n pairs), at the second layer, a GRU takes $[v_1, \ldots, v_n]$ as an input and encodes the matching sequence into its hidden states $H_m = [h_1, \ldots, h_n] \in \mathbb{R}^{q \times n}$ with a detailed parameterization similar to Equation (2). This layer has two functions: (1) it models the dependency and the temporal relationship of utterances in the session; (2) it leverages the temporal relationship to supervise the accumulation of the pair matching as a session based matching. Moreover, from Equation (2), we can see that the reset gate (i.e., r_i) and the update gate (i.e., z_i) control how much information from the previous hidden state and the current input flows to the current hidden state, thus important matching vectors (corresponding to important utterances) can be accumulated while noise in the vectors can be filtered out.

With H_m, we define $g(s, r)$ as

$$g(s, r) = softmax(\mathbf{W_2} L[h_1, \ldots, h_n] + \mathbf{b_2}), \tag{6}$$

where $\mathbf{W_2}$ and $\mathbf{b_2}$ are parameters. We consider three parameterizations for $L[h_1, \ldots, h_n]$: (1) only the last hidden state is used. Then $L[h_1, \ldots, h_n] = h_n$. (2) the hidden states are linearly combined. Then, $L[h_1, \ldots, h_n] = \sum_{i=1}^{n} w_i h_i$, where $w_i \in \mathbb{R}$. (3) we follow (Yang et al., 2016) and employ an attention mechanism to combine the hidden states. Then, $L[h_1, \ldots, h_n]$ is defined as

$$\begin{aligned} t_i &= tanh(\mathbf{W_3} h_i + \mathbf{b_3}), \\ \alpha_i &= \frac{exp(t_i^\top t_s)}{\sum_i (exp(t_i^\top t_s))}, \\ L[h_1, \ldots, h_n] &= \sum_{i=1}^{n} \alpha_i h_i, \end{aligned}$$

where $\mathbf{W_3} \in \mathbb{R}^{q \times q}$ and $\mathbf{b_3} \in \mathbb{R}^q$ are parameters. $t_s \in \mathbb{R}^q$ is a high level virtual context vector which is randomly initialized and jointly learned in training.

Both (2) and (3) aim to learn weights for $\{h_1, \ldots, h_n\}$ from training data and dynamically highlight the effect of important matching vectors in the final matching score. The difference is that weights in (2) are nonparametric and unnormalized, while in (3) they are parametric and normalized. We denote our model with the three parameterizations of $L[h_1, \ldots, h_n]$ as SMN$_{last}$, SMN$_{non-para}$, and SMN$_{para}$ respectively, and empirically compare them in experiments.

We learn $g(\cdot, \cdot)$ by minimizing cross entropy with $\mathcal{D}$. Let Θ denote the parameters of our model, then the objective function $\mathcal{L}(\mathcal{D}, \Theta)$ of learning can be formulated as

$$-\sum_{i=1}^{N} \left[y_i log(g(s_i, r_i)) + (1 - y_i) log(1 - g(s_i, r_i)) \right], \tag{7}$$

where N in the number of instances in $\mathcal{D}$.

4 Response Candidate Retrieval

In practice of a retrieval based chatbot, to apply the matching approach to response selection, one needs to retrieve a bunch of response candidates from an index beforehand. While candidate retrieval is not the focus of the paper, it is an important step in a real system. In this work, we exploit a heuristic method to obtain response candidates from index. Given a message u_n with $\{u_1, \ldots, u_{n-1}\}$ utterances in its previous turns, we extract top 5 keywords based on their tf-idf values[1] and expand u_n with the keywords. Then we send the expanded message to the index and retrieve response candidates using the inline retrieval algorithm of the index. Finally, we use $g(s, r)$ to re-rank the candidates and return the top one as a response to the session.

5 Experiment

We tested our model on a public English data set and a Chinese data set we publish with this paper.

5.1 Experiment setup

The English data set is the Ubuntu Corpus (Lowe et al., 2015) which contains large scale multi-turn dialogues collected from chat logs of Ubuntu Forum. The data set consists of 1 million session-response pairs for training, 0.5 million pairs for validation, and 0.5 million pairs for test. Positive responses are true responses from human, and

[1] Tf is word frequency in the session, while idf is calculated using the entire index.

negative ones are randomly sampled. The ratio of the positive and the negative is 1:1 in training, and 1:9 in validation and test. We used the copy shared by Xu et al. (2016)[2] in which numbers, urls, and paths are replaced by special placeholders.

One problem with the Ubuntu data is that negative examples is much easier to identify than those in a real chatbot, because they are randomly sampled and most of them are far from the semantics of the context. A better data set that can simulate the real scenario of a retrieval based chatbot must have responses generated following the procedure of information retrieval and labels annotated by humans. As far as we know, however, there are no such data sets publicly available. To test our model in a setting closer to the real case and facilitate the research of multi-turn response selection, we created a new data set and publish it to research communities with the paper. We crawled 15 million post-reply pairs from Sina Weibo[3] which is the largest microblogging service in China and indexed the pairs with an open source Lucene[4]. We then crawled 1.1 million dyadic dialogues (conversation between two people) longer than 2 turns from Douban group[5] which is a popular forum in China. From the data, we randomly sampled 0.5 million dialogues for creating a training set, 25 thousand dialouges for creating a validation set, and $1,000$ dialogues for creating a test set, and made sure that there is no overlap among the three sets. For each dialogue in training and validation, we took the last turn as a positive response for the previous turns as a session and randomly sampled another response from the 1.1 million data as a negative response. In total, there are 1 million session-response pairs in the training set and 50 thousand pairs in the validation set. To create the test set, we took the last turn of each dialogue as a message, retrieved 10 response candidates from the index following the method in Section 4, and finally formed a test set with $10,000$ session-response pairs. We recruited three labelers to judge if a candidate is a proper response to the session. A proper response means the response can naturally reply to the message given the context. Each pair received three labels and the majority of the labels was taken as the final decision.

[2]https://www.dropbox.com/s/2fdn26rj6h9bpvl/ubuntudata.zip?dl=0

[3]http://weibo.com/

[4]https://lucenenet.apache.org/

[5]https://www.douban.com/group

Table 2: Statistics of the Chinese data set

	train	val	test
# session-response pairs	1M	50k	10k
# response candidates per session	2	2	10
Avg. # positive responses per session	1	1	1.18
Min. # turns per session	3	3	3
Max. # turns per session	53	40	44
Avg. # turns per session	6.03	5.81	5.95
Avg. # words per utterance	16.75	17.22	17.17

Table 2 gives the statistics of the three sets. Note that the Fleiss' kappa (Fleiss, 1971) of the labeling is 0.41, which indicates that the three labelers reached a relatively high agreement.

On the Ubuntu data, we followed (Lowe et al., 2015) and employed recall at position k in n candidates ($R_n@k$) as evaluation metrics, and on the human labeled data, we followed the convention of information retrieval and employed mean average precision (MAP) (Baeza-Yates et al., 1999), mean reciprocal rank (MRR) (Voorhees and others, 1999), and precision at position 1 (P@1) as metrics. Note that when using the labeled set, we removed sessions with all negative responses or all positive responses, as models make no difference on them. After that there are $6,670$ session-response pairs left in test.

5.2 Baseline

We considered the following baselines:

Basic models: models in (Lowe et al., 2015) and (Kadlec et al., 2015) including TF-IDF, RNN, CNN, LSTM and BiLSTM.

Multi-view: the model proposed by Zhou et al. (2016) who utilize a hierarchical recurrent neural network to model utterance relationships.

Deep learning to respond (DL2R): the model proposed by Yan et al. (2016).

Advanced single-turn matching models: since LSTM and BiLSTM do not represent the state-of-the-art matching models, we concatenated the utterances in a session and matched the long text with a response candidate using more powerful models including MV-LSTM (Wan et al., 2016), Match-LSTM (Wang and Jiang, 2015), and Multi-Channel which is described in Section 3.3. Multi-Channel is a simple version of our model without considering utterance relationships.

5.3 Parameter Tuning

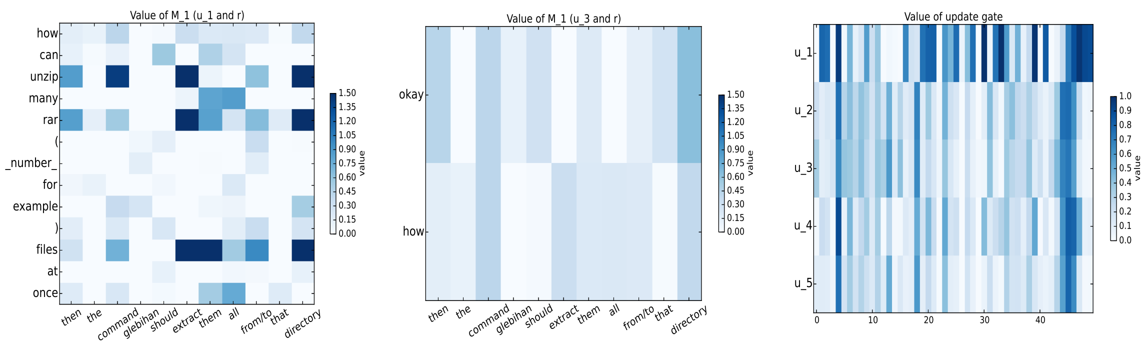
For baseline models, if their results are available in the existing literatures (e.g., those on Ubuntu Corpus), we just copied the numbers, otherwise

Table 3: Evaluation results on the two data sets

	Ubuntu data				Chinese data		
	$R_2@1$	$R_{10}@1$	$R_{10}@2$	$R_{10}@5$	MAP	MRR	P@1
TF-IDF (Lowe et al., 2015)	0.659	0.410	0.545	0.708	0.331	0.359	0.179
RNN (Lowe et al., 2015)	0.768	0.403	0.547	0.819	0.390	0.422	0.208
CNN (Kadlec et al., 2015)	0.848	0.549	0.684	0.896	0.417	0.440	0.226
LSTM (Kadlec et al., 2015)	0.901	0.638	0.784	0.949	0.485	0.527	0.320
BiLSTM (Kadlec et al., 2015)	0.895	0.630	0.780	0.944	0.479	0.514	0.313
Multi-View (Zhou et al., 2016)	0.908	0.662	0.801	0.951	0.505	0.543	0.342
DL2R (Yan et al., 2016)	0.899	0.626	0.783	0.944	0.488	0.527	0.330
MV-LSTM (Wan et al., 2016)	0.906	0.653	0.804	0.946	0.498	0.538	0.348
Match-LSTM (Wang and Jiang, 2015)	0.904	0.653	0.799	0.944	0.500	0.537	0.345
Multi-Channel	0.904	0.656	0.809	0.942	0.506	0.543	0.349
SMN_{last}	**0.923**	**0.723**	**0.842**	**0.956**	**0.526**	**0.571**	**0.392**
$\text{SMN}_{non-para}$	**0.927**	**0.725**	**0.838**	**0.962**	**0.523**	**0.572**	**0.387**
SMN_{para}	**0.926**	**0.726**	**0.848**	**0.960**	**0.517**	**0.561**	**0.372**

Table 4: Evaluation results of model ablation

	Ubuntu data				Chinese data		
	$R_2@1$	$R_{10}@1$	$R_{10}@2$	$R_{10}@5$	MAP	MRR	P@1
Replace_M	0.905	0.661	0.799	0.950	0.503	0.541	0.343
Replace_S	0.918	0.716	0.832	0.954	0.522	0.565	0.376
Only M_1	0.919	0.704	0.832	0.955	0.518	0.562	0.370
Only M_2	0.921	0.715	0.836	0.956	0.521	0.565	0.382
SMN_{last}	0.923	0.723	0.842	0.956	0.526	0.571	0.392



(a) $\mathbf{M}_1$ of u_1 and r (b) $\mathbf{M}_1$ of u_3 and r (c) Update gate

Figure 2: Model visualization

we implemented the models following the settings in the literatures. All models were implemented using Theano (Theano Development Team, 2016). Word embeddings were initialized by the results of word2vec [6] run on the training data, and the dimension of word vectors is 200. For Multi-Channel and layer one of our model, we set the dimension of the hidden states of GRU as 200. We tuned the window size in convolution and pooling in $\{(2,2),(3,3)(4,4)\}$ and chose $(3,3)$ finally. The number of feature maps is 8. In layer two, we set the dimensions of matching vectors and the hidden states of GRU as 50. We optimized the objective function using back-propagation and the parameters were updated by stochastic gradient descent with Adam algorithm (Kingma and Ba, 2014) on a single Tesla K80 GPU. The initial learning rate is 0.001, and the parameters of Adam, β_1 and β_2, that control exponential decay are 0.9 and 0.999 respectively. We employed early-stopping (Lawrence and Giles, 2000) as a regularization strategy. Models were trained in mini-batches with a batch size 200, and we padded zeros if the length of an utterance exceeds 50.

[6] `https://code.google.com/archive/p/word2vec/`

5.4 Evaluation Results

Table 3 shows the evaluation results on the two data sets. Our models outperform baselines greatly in terms of all metrics on both data sets, and the improvements are statistically significant (t-test with p-value $\leq$ 0.01). Even the state-of-the-art single-turn matching models perform much worse than our models. The results demonstrate that one cannot neglect utterance relationships and simply perform multi-turn response selection by transforming it to a single-turn problem. Our models achieve significant improvements over Multi-View, which justified our "matching first" strategy. DL2R is also worse than our models, indicating that utterance reformulation with heuristic rules is not a good method to utilize context information. Numbers on the Ubuntu data are much higher than those on the Chinese data (R_{10}@1 and P@1 are equivalent). The results showed the merit of our new data and supported our claim that the Ubuntu data oversimplified the problem of multi-turn response selection. There is no significant difference among our three models. The reason might be GRU has already selected useful signals from the matching sequence and accumulated them in the final state with its gate mechanism, especially when the sequence is not long, and there is no need to equip another attention mechanism on top of it.

5.5 Further Analysis

Visualization: we visualize the similarity matrices and the gates of GRU in layer two using an example from the Ubuntu Corpus to further clarify how our model identifies important information in context and how it selects important matching vectors with the gate mechanism of GRU as described in Section 3.3 and Section 3.4. The example is $\{u_1$: how can unzip many rar (*_number_* for example) files at once; u_2: sure you can do that in bash; u_3: okay how? u_4: are the files all in the same directory? u_5: yes they all are; r: then the command glebihan should extract them all from/to that directory$\}$. It is from the test set and our model successfully ranked the correct response to the top position. Due to space limitation, we only visualized $\mathbf{M}_1$ of u_1 and r in Figure 2(a), $\mathbf{M}_1$ of u_3 and r in Figure 2(b), and the update gate (i.e. z) in Figure 2(c). They are already enough to support our analysis. In all pictures, darker areas mean larger values. We can see that in u_1 important words including "unzip", "rar", "files" are recognized and carried to matching by "command", "extract", and "directory" in r, while u_3 is almost useless and thus little information is extracted from it. u_1 is crucial to response selection and nearly all information from u_1 and r flows to the hidden state of GRU, while other utterances are less informative and the corresponding gates are almost "closed" to keep the information from u_1 and r until the final state.

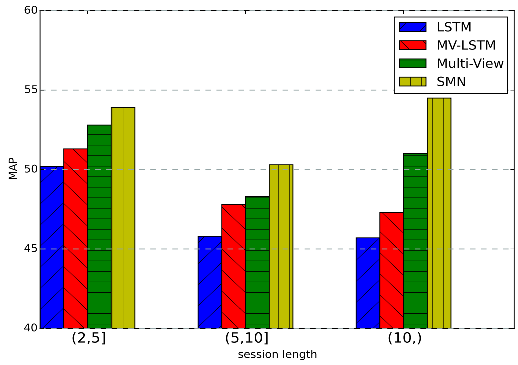


Figure 3: Comparison across session length

Model ablation: we investigate the effect of different parts of our model by removing them one by one. Table 4 reports the results. First, replacing the multi-channel "2D" matching with a neural tensor network (NTN) (Socher et al., 2013) (denoted as Replace_M) makes the performance drop dramatically. This is because NTN only matches a pair by an utterance vector and a response vector and misses important information in the pair. Together with the visualization, we can conclude that "2D" matching plays a key role in the "matching first" strategy as it can capture the important matching information in each pair with minimal loss. Second, the performance slightly drops when replacing the GRU for matching accumulation with a multi-layer perceptron (denoted as Replace_S). This indicates that utterance relationships are also useful. Finally, we left only one channel in matching and found that $\mathbf{M}_2$ is a little more powerful than $\mathbf{M}_1$ and we can achieve the best results with both of them.

Session length: we finally study how our model (SMN_{last}) performs with respect to the length of sessions. Figure 3 shows the comparison on MAP in different length intervals on the Chinese data. We can see that our model consistently performs better than the baselines, and when sessions become longer, the gap becomes larger. The results demonstrate that our model can well capture the dependencies, especially long dependencies, among utterances in sessions.

6 Conclusion

We present a new model for multi-turn response selection in retrieval-based chatbots. Experiment results on public data sets show that the model can significantly outperform the state-of-the-art methods.

References

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

[Boden2006] Margaret Ann Boden. 2006. *Mind as machine: A history of cognitive science*. Clarendon Press.

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

[Fleiss1971] Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

[Higashinaka et al.2014] Ryuichiro Higashinaka, Kenji Imamura, Toyomi Meguro, Chiaki Miyazaki, Nozomi Kobayashi, Hiroaki Sugiyama, Toru Hirano, Toshiro Makino, and Yoshihiro Matsuo. 2014. Towards an open-domain conversational system fully based on natural language processing. In *COLING*, pages 928–939.

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

[Jafarpour et al.2010] Sina Jafarpour, Christopher JC Burges, and Alan Ritter. 2010. Filter, rank, and transfer the knowledge: Learning to chat. *Advances in Ranking*, 10.

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

[Kadlec et al.2015] Rudolf Kadlec, Martin Schmid, and Jan Kleindienst. 2015. Improved deep learning baselines for ubuntu corpus dialogs. *arXiv preprint arXiv:1510.03753*.

[Kingma and Ba2014] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[Lawrence and Giles2000] Steve Lawrence and C Lee Giles. 2000. Overfitting and neural networks: conjugate gradient and backpropagation. In *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*, volume 1, pages 114–119. IEEE.

[Li et al.2015] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

[Li et al.2016] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A persona-based neural conversation model. *arXiv preprint arXiv:1603.06155*.

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

[Ritter et al.2011] Alan Ritter, Colin Cherry, and William B Dolan. 2011. Data-driven response generation in social media. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 583–593. Association for Computational Linguistics.

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

[Serban et al.2016] Iulian Vlad Serban, Tim Klinger, Gerald Tesauro, Kartik Talamadupula, Bowen Zhou, Yoshua Bengio, and Aaron Courville. 2016. Multiresolution recurrent neural networks: An application to dialogue response generation. *arXiv preprint arXiv:1606.00776*.

[Shang et al.2015] Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1577–1586.

[Socher et al.2013] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*, pages 926–934.

[Theano Development Team2016] Theano Development Team. 2016. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May.

[Vinyals and Le2015] Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.

[Voorhees and others1999] Ellen M Voorhees et al. 1999. The trec-8 question answering track report. In *Trec*, volume 99, pages 77–82.

[Wallace2009] Richard S Wallace. 2009. *The anatomy of ALICE*. Springer.

[Wan et al.2016] Shengxian Wan, Yanyan Lan, Jun Xu, Jiafeng Guo, Liang Pang, and Xueqi Cheng. 2016. Match-srnn: Modeling the recursive matching structure with spatial rnn. *arXiv preprint arXiv:1604.04378*.

[Wang and Jiang2015] Shuohang Wang and Jing Jiang. 2015. Learning natural language inference with lstm. *arXiv preprint arXiv:1512.08849*.

[Wang et al.2013] Hao Wang, Zhengdong Lu, Hang Li, and Enhong Chen. 2013. A dataset for research on short-text conversations. In *EMNLP*, pages 935–945.

[Wang et al.2015] Mingxuan Wang, Zhengdong Lu, Hang Li, and Qun Liu. 2015. Syntax-based deep matching of short texts. *arXiv preprint arXiv:1503.02427*.

[Weizenbaum1966] Joseph Weizenbaum. 1966. Eliza?a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45.

[Wu et al.2016] Yu Wu, Wei Wu, Zhoujun Li, and Ming Zhou. 2016. Topic augmented neural network for short text conversation. *CoRR*, abs/1605.00090.

[Xing et al.2016] Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2016. Topic augmented neural response generation with a joint attention mechanism. *arXiv preprint arXiv:1606.08340*.

[Xu et al.2016] Zhen Xu, Bingquan Liu, Baoxun Wang, Chengjie Sun, and Xiaolong Wang. 2016. Incorporating loose-structured knowledge into lstm with recall gate for conversation modeling. *arXiv preprint arXiv:1605.05110*.

[Yan et al.2016] Rui Yan, Yiping Song, and Hua Wu. 2016. Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016*, pages 55–64.

[Yang et al.2016] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

[Young et al.2010] Steve Young, Milica Gašić, Simon Keizer, François Mairesse, Jost Schatzmann, Blaise Thomson, and Kai Yu. 2010. The hidden information state model: A practical framework for pomdp-based spoken dialogue management. *Computer Speech & Language*, 24(2):150–174.

[Zhou et al.2016] Xiangyang Zhou, Daxiang Dong, Hua Wu, Shiqi Zhao, R Yan, D Yu, Xuan Liu, and H Tian. 2016. Multi-view response selection for human-computer conversation. *EMNLP16*.