Multi-hop Selector Network for Multi-turn Response Selection in Retrieval-based Chatbots

Chunyuan Yuan[1,2,†], **Shangwen Lv**[1,2,†], **Mingming Li**[1,2], **Wei Zhou**[1,2,*], **Fuqing Zhu**[2], **Jizhong Han**[2] **and Songlin Hu**[1,2]
[1]School of Cyber Security, University of Chinese Academy of Sciences
[2]Institute of Information Engineering, Chinese Academy of Sciences
{yuanchunyuan, lvshangwen, limingming, zhouwei,
zhufuqing, hanjizhong, husonglin}@iie.ac.cn

Abstract

Multi-turn retrieval-based conversation is an important task for building intelligent dialogue systems. Existing works mainly focus on matching candidate responses with every context utterance on multiple levels of granularity, which ignore the side effect of using excessive context information. Context utterances provide abundant information for extracting more matching features, but it also brings noise signals and unnecessary information.

In this paper, we will analyze the side effect of using too many context utterances and propose a multi-hop selector network (MSN) to alleviate the problem. Specifically, MSN firstly utilizes a multi-hop selector to select the relevant utterances as context. Then, the model matches the filtered context with the candidate response and obtains a matching score. Experimental results show that MSN outperforms some state-of-the-art methods on three public multi-turn dialogue datasets.

1 Introduction

Building a dialogue system that can naturally and consistently converse with humans has drawn increasing research interests in past years. Existing works on building dialogue systems include generation-based and retrieval-based methods. Compared with generation-based methods, retrieval-based methods have advantages in providing fluent and informative responses. Many industrial products have applied retrieval-based dialogue system, e.g., the E-commerce assistant AliMe Assist from Alibaba Group (Li et al., 2017) and the XiaoIce (Shum et al., 2018) from Microsoft.

Early studies (Tan et al., 2015; Yan et al., 2016; Wan et al., 2016) of retrieval-based dialogue system focus on response selection for single-turn conversation. Recently, researchers have begun to pay attention to the multi-turn conversation, aiming at selecting the most related response from a set of candidates given the context utterances of a conversation. Some effective models, such as Sequential Matching Network (SMN) (Wu et al., 2017), Deep Attention Matching network (DAM) (Zhou et al., 2018c), Multi-Representation Fusion Network (MFRN) (Tao et al., 2019), have been proposed to capture the matching features on multiple levels of granularity (words, phrases, sentences, etc.) and short-term and long-term dependencies among words.

Previous works have shown that utilizing multi-turn utterances can further improve the matching performance than only using single-turn utterance (i.e., last utterance). But context utterance is a "double-edged sword", it also provides a lot of noise while providing abundant information, which would influence the performance due to the sensitivity of these matching-based methods.

Table 1: An error case of SMN (Wu et al., 2017), DAM (Zhou et al., 2018c) from E-commerce Corpus. The scores in the table are matching scores predicted by the models.

Turns	Dialogue Text	SMN	DAM
Turn-1	A: Are there any **discounts** activities recently?		
Turn-2	B: No. Our product have been **cheaper** than before.		
Turn-3	A: Oh.		
Turn-4	B: Hum!		
Turn-5	A: I'll buy these nuts. Can you sell me **cheaper**?		
Turn-6	B: You can get some **coupons** on the homepage.		
Turn-7	A: Will you give me some nut **clips**?		
Turn-8	B: Of course **we will**.		
Turn-9	A: **How many clips** will you give?		
Resp-1	**One clip** for every package. (True)	0.832	0.854
Resp-2	OK, **we will** give you a **coupons** worth $1. (False)	0.925	0.947

To illustrate the problem, we show an error case of SMN (Wu et al., 2017) and DAM (Zhou et al., 2018c) from E-commerce Corpus in Table 1. We can see that although "Resp-1" is the right answer for utterance "Turn-9", the SMN and DAM mod-

†Equally contributed.
* Corresponding author.

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 111–120,
Hong Kong, China, November 3–7, 2019.

els still choose "Resp-2". Because it has more words overlap with context utterances, thus accumulating a larger similarity score. We can easily observe that "Resp-2" is relevant to former utterances (Turn-1 to Turn-6), but the topic has changed after "Turn-6". Besides, we can see that "Turn-3" and "Turn-4" do not provide any useful information for selecting candidate response. From this example, irrelevant context utterances may cause the models making simple mistakes that humans would not make. Furthermore, we conduct several adversarial experiments and the results show that these matching-based models are very sensitive to the adversarial samples.

In this paper, we propose a multi-hop selector network to tackle the above problem. Intuitively, the closer the utterance to the response is, the more it reflects the intention of the last dialogue session. Thus, we firstly use the last utterance as key to select context utterances that are relevant to it on the word and sentence level. However, we find that there are many samples whose last utterance is very short and contains very limited information (such as "good", "ok"), which will cause the selectors to lose too much useful context information. Therefore, we propose multi-hop selectors to select more relevant context utterances, yielding k different context. Then, we fuse these selected context utterances and match it with candidate response. During the matching stage, the convolution neural network (CNN) is applied to extract matching features and the gated recurrent unit (GRU) is applied to learn the temporal relationship of utterances.

The contributions of this paper are summarized as follows:

- We find the noises in context utterances could influence the matching performance and design adversarial experiments to verify it.
- We propose a unified network MSN to select relevant context utterances from word and utterance level and fuse the selected context to generate a better context representation.
- Experimental results on three public datasets achieve significant improvement, which shows the effectiveness of MSN.

The outline of the paper is as follows. Section 2 introduces related works. Section 3 describes adversarial experiment to check how sensitivity of previous models to the context utterances. Section 4 describes every component of MSN model. Section 5 discusses the experiments and corresponding results. Section 6 discusses some experiments to explore the influence of hyper-parameters on performance. We conclude our work in Section 7.

2 Related Work

With the development of natural language processing, building intelligent chatbots with data-driven approaches has drawn increasing attention in recent years. Existing works can be generally categorized into retrieval-based methods (Wan et al., 2016; Wu et al., 2017; Zhang et al., 2018; Tao et al., 2019) and generation-based methods (Shang et al., 2015; Serban et al., 2016; Xing et al., 2017; Wu et al., 2018; Zhou et al., 2018a,b). In this work, we focus on retrieval-based method and study context-based response selection.

Early retrieval-based chatbots are devoted to response selection for single-turn conversation (Wang et al., 2013; Tan et al., 2015; Yan et al., 2016). Recently, researchers have begun to turn to the multi-turn conversation. Lowe et al. (2015) use RNN to read context and response, use the last hidden states to represent context and response as two semantic vectors to measure their relevance. Zhou et al. (2016) perform context-response matching with a multi-view model on both word and utterance levels. Considering concatenating utterances in context may lose relationships among utterances or important contextual information, Wu et al. (2017) separately match the response with each utterance based on a convolutional neural network. This paradigm is applied in many subsequent works. Zhou et al. (2018c) consider the dependency relation among utterances based on the attention mechanism. Tao et al. (2019) fuse words, n-grams, and sub-sequences of utterances representations and capture both short-term and long-term dependencies among words.

Different from previous works, (i) we study the influence of using excessive context utterances, (ii) we explore how to filter out irrelevant context to improve the robustness of matching-based methods.

3 Adversarial experiments

To study how sensitive of the previous models (Wu et al., 2017; Zhang et al., 2018; Zhou et al., 2018c; Tao et al., 2019) to the context utterances, we conduct several adversarial experiments inspired

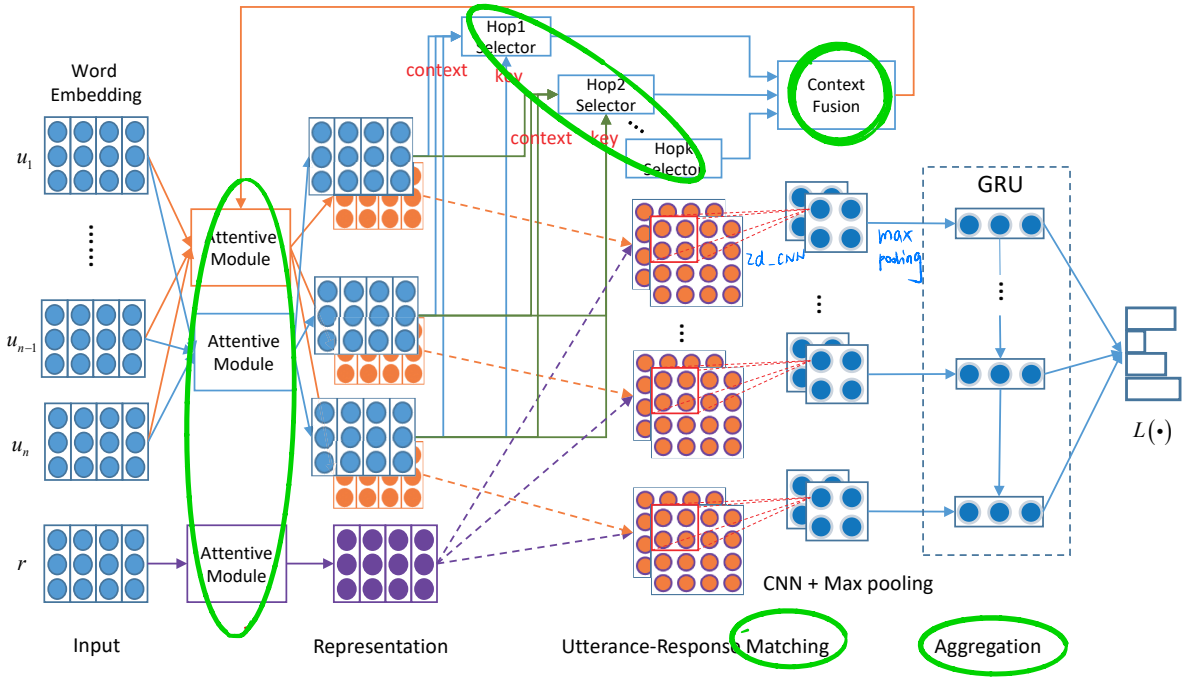


Figure 1: Architecture of multi-hop selector network.

by (Jia and Liang, 2017). We keep the training set unchanged and add some noises to the original test set. To be specific, we randomly sample 1∼3 words from context utterances and append them on every candidate response. In this way, we can obtain 3 different adversarial test sets: adversarial_set1, adversarial_set2, adversarial_set3.

Then, we evaluate the models again to see how much will the performance change. To ensure the fairness of the experiments, we use the results from their papers for the original test set. Moreover, we use their open source code for adversarial experiments. We employ recall at position k in n candidates ($R_n@k$) as the evaluation metric, which is the same as previous works.

Table 2: Adversarial experimental results on Ubuntu Dialogue Corpus. The results of SMN (Wu et al., 2017), DUA (Zhang et al., 2018), DAM (Zhou et al., 2018c), MFRN (Tao et al., 2019) on original test set are cited from their papers.

Models	original test set		adversarial_set1		adversarial_set2		adversarial_set3	
	$R_{10}@1$	$R_{10}@2$	$R_{10}@1$	$R_{10}@2$	$R_{10}@1$	$R_{10}@2$	$R_{10}@1$	$R_{10}@2$
SMN	72.6	84.7	66.2	82.1	63.8	79.4	57.1	75.0
DUA	75.2	86.8	64.0	80.4	58.0	75.6	52.7	70.8
DAM	76.7	87.4	67.5	82.3	61.2	76.8	54.3	71.6
MRFN	78.6	88.6	65.4	81.7	65.1	76.4	58.2	72.3
MSN	80.0	89.9	70.7	84.6	66.2	81.4	64.6	79.9

The experimental results are shown in Table 2. From the table, we can observe that the one-word noise will bring about 7% ∼ 13% absolute decrease on $R_{10}@1$ and the three-word noise brings about 20% $R_{10}@1$ decrease. Thus, we can see that matching-based models (Wu et al., 2017; Zhang et al., 2018; Zhou et al., 2018c; Tao et al., 2019) are very sensitive to small noises of the dataset. Moreover, using too many context utterances will greatly increase the probability of introducing noise. The results of MSN also show that filtering irrelevant utterances can effectively alleviate this problem and improve the robustness of matching-based models.

4 Model

4.1 Problem Formalization

Suppose that we have a data set $D = \{(U_i, r_i, y_i)\}_{i=1}^{N}$, where $U_i = \{u_{i1}, u_{i2}, \ldots, u_{iL}\}$ represents a conversation context with L utterances and every utterance u_{ij} contains T words. 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 U_i, otherwise $y_i = 0$. Our goal is to learn a matching model $g(\cdot, \cdot)$ with D. For any context-response pair (U_i, r_i), $g(U_i, r_i)$ measures the matching degree between U_i and r_i.

To this end, we need to address two problems: (1) how to select proper context utterances from U_i; and (2) how to fuse these selected utterances together for a better representation.

4.2 Model Overview

We propose a multi-hop selector network (MSN) to model $g(\cdot, \cdot)$. Figure 1 gives the architecture, which generally follows the representation-matching-aggregation framework (Wu et al., 2017; Zhang et al., 2018; Zhou et al., 2018c; Tao et al., 2019) to match response with multi-turn context.

Different from previous works, we add a selection process before the above framework. MSN first constructs semantic representations at word level by an Attentive Module. Then, each utterance are packed as context or key and sent to the "Hopk Selector" to calculate relevance scores. The scores of k different selectors are fused together by a Context Fusion module. Finally, the fused scores are performed over original context utterances to filter out irrelevant context. The rest context utterances are applied for response matching.

4.3 Attentive Module

We use the Attentive Module to learn the context information for word representation. Attentive Module is proposed in DAM (Zhou et al., 2018c) and it is a variant of Multi-head Attention (Vaswani et al., 2017). Figure 2 shows its structure.

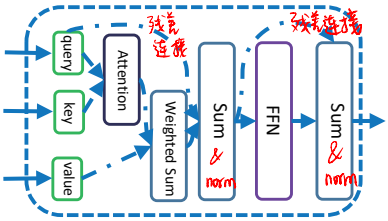


Figure 2: Architecture of Attentive Module.

The $\mathbf{AttentiveModule}(Q, K, V)$ has three input sentences: the query sentence, the key sentence and the value sentence, namely $Q \in \mathbb{R}^{n_q \times d}$, $K \in \mathbb{R}^{n_k \times d}$, and $V \in \mathbb{R}^{n_v \times d}$ respectively, where n_q, n_k, and n_v denote the number of words in each sentence, and d is the dimension of the embedding.

The Attentive Module first takes each word in the query sentence to attend to words in the key sentence via Scaled Dot-Product Attention (Vaswani et al., 2017), and then applies those attention weights upon the value sentence:

$$\mathbf{V}_{att} = softmax\left(\frac{QK^T}{\sqrt{d}}\right)V\,. \quad (1)$$

Then, a feed-forward network (FFN) with RELU (LeCun et al., 2015) activation is applied upon the normalization result, to further process the fused embeddings:

$$\mathbf{FFN}(x) = \max(0, x\mathbf{W}_1 + b_1)\mathbf{W}_2 + b_2\,, \quad (2)$$

where x is a 2D matrix in the same shape of query sentence Q and $\mathbf{W}_1$, b_1,$\mathbf{W}_2$, b_2 are learnt parameters. The result $\mathbf{FFN}(x)$ is a 2D matrix that has the same shape as x, $\mathbf{FFN}(x)$ is then residually added to x, and the fusion result is then normalized as the final outputs.

4.4 Context Selector

Given $\mathbf{U}_i = [\mathbf{u}_{i1}, \ldots, \mathbf{u}_{ij}, \ldots, \mathbf{u}_{iL}]$, the word-level embedding representations for utterance $\mathbf{u}_{ij} \in \mathbb{R}^{T \times d}$, where d is the dimension of word vector, we use the Attentive Module to reconstruct the word representations of each utterance to encode the context and dependency information into word, which is formulated as:

$$\mathbf{u}'_{ij} = \mathbf{AttentiveModule}(\mathbf{u}_{ij}, \mathbf{u}_{ij}, \mathbf{u}_{ij})\,, \quad (3)$$

where $\mathbf{u}'_{ij} \in \mathbb{R}^{T \times d}$. $\mathbf{U}'_i = [\mathbf{u}'_{i1}, \mathbf{u}'_{i2}, \ldots, \mathbf{u}'_{iL}] \in \mathbb{R}^{L \times T \times d}$.

We first discuss how to construct "Hop1 Selector", which consists of word and utterance selector. To capture matching features at multiple levels of granularity, we leverage word and utterance level matching features to select relevant context.

4.4.1 Word Selector

At word level, we utilize cross attention to obtain a matching feature map for each context utterance $\mathbf{u}'_{ij}$ and key $\mathbf{K}_1 = \mathbf{u}'_{iL}$, which is formulated as:

$$\mathbf{A} = \mathbf{v}^T \tanh(\mathbf{K}_1^T \mathbf{W}\mathbf{U}'_i + \mathbf{b})\,, \quad (4)$$

where $\mathbf{W} \in \mathbb{R}^{d \times d \times h}$, $\mathbf{b} \in \mathbb{R}^h$ and $\mathbf{v} \in \mathbb{R}^{h \times 1}$. And we get a word alignment matrix $\mathbf{A} \in \mathbb{R}^{L \times T \times T}$.

Then, we extract the most prominent matching features from $\mathbf{A}$ by max pooling over row and column. Then, they are concatenated together:

$$m_1(\mathbf{K}_1, \mathbf{U}'_i) = [\max_{dim=2} \mathbf{A}; \max_{dim=3} \mathbf{A}]\,, \quad (5)$$

where $m_1(\mathbf{K}_1, \mathbf{U}'_i) \in \mathbb{R}^{L \times 2T}$, which reflects which words have identical or similar meaning between utterances $\mathbf{u}_{ij}$ and key $\mathbf{u}_{iL}$. The matching features are transformed to the relevance score by a linear layer:

$$\mathbf{s}_1 = \mathbf{softmax}(m_1(\mathbf{K}_1, \mathbf{U}'_i)\mathbf{c} + b)\,, \quad (6)$$

where $\mathbf{c} \in \mathbb{R}^{2T\times 1}$ and $b \in \mathbb{R}^{L\times 1}$.

The word selector can only capture word-level relevance between key and utterances. It can not reflect whether key and context are compatible on the overall semantic level. Thus, we continue to evaluate the relevance on the utterance level.

4.4.2 Utterance Selector

Firstly, the word-level representations $\mathbf{U}'_i$ are transformed to utterance-level representations by mean pooling over word dimension:

$$\widetilde{\mathbf{U}}_i = \mathbf{mean}(\mathbf{U}'_i)\,, \tag{7}$$

where $\widetilde{\mathbf{U}}_i \in \mathbb{R}^{L\times d}$.

We use cosine similarity to measure the relevance between key $\mathbf{K}_2 = \widetilde{\mathbf{U}}_{iL}$ and context utterances $\widetilde{\mathbf{U}}_i$, which is formulated as:

$$s_2 = \frac{\widetilde{\mathbf{U}}_i \mathbf{K}_2^T}{||\widetilde{\mathbf{U}}_i||_2 \, ||\mathbf{K}_2||_2}\,, \tag{8}$$

where $\mathbf{s}_2 \in \mathbb{R}^{L\times 1}$ is the relevance score at utterance level.

Both the scores of word selector and utterance selector are important to measure the relevance of last utterance and context. In order to make full use of word and utterance selectors, we design a combined strategy to fuse two scores. Specifically, we use the weighted sum of two scores for selection:

$$\mathbf{s}^{(1)} = \alpha * \mathbf{s}_1 + (1-\alpha) * \mathbf{s}_2\,, \tag{9}$$

where α is a hyper-parameter and $\mathbf{s}^{(1)}$ is the final score that hop1 selector produces. The default value of α is set to 0.5.

4.4.3 Hopk Selector

Although "Hop1 Selector" can choose proper context utterances that are related to the last dialogue session, we find that there are many samples whose last utterance contains very little information (such as "good", "ok"), which will cause the selector lose too much useful context information. Thus, we combine it with $\widetilde{u}_{i,L-1}$, $\widetilde{u}_{i,L-2}$, ..., $\widetilde{u}_{i,L-k}$ by mean pooling. Then, we treat them as key to conduct the same process as "Hop1 Selector" for context selection. In this way, we can get k different selectors, yielding k different scores $\mathbf{S} = [\mathbf{s}^{(1)}, \mathbf{s}^{(2)}, \ldots, \mathbf{s}^{(k)}] \in \mathbb{R}^{L\times k}$.

4.5 Context Fusion

Then we fuse the similarity scores from different selectors and apply it to select relevant context utterances for matching.

Firstly, we combine the similarity scores $\mathbf{S} \in \mathbb{R}^{L\times k}$ to form the final scores for each context utterances and filter out irrelevant context, which is formulated as:

$$\begin{aligned} \mathbf{s}' &= \mathbf{S}\mathbf{W}^T \\ \widetilde{\mathbf{s}} &= \mathbf{s}' \odot (\mathbf{sigmoid}(\mathbf{s}') \geq \gamma) \end{aligned}\,, \tag{10}$$

where $\mathbf{W} \in \mathbb{R}^{1\times k}$ is a dynamic weight vector and will be tuned by the gradient. γ is the threshold and will be tuned according to the dataset. The default value of γ can be set to 0.5. The utterances whose scores are below γ will be allocated lower weights or filtered out.

Then, we multiply the mask weight $\widetilde{\mathbf{s}}$ and context utterances to filter irrelevant context:

$$\hat{\mathbf{U}}_i = \widetilde{\mathbf{s}} \odot \mathbf{U}_i\,, \tag{11}$$

and generate $\hat{\mathbf{U}}_i \in \mathbb{R}^{L\times T\times d}$, where $\mathbf{U}_i \in \mathbb{R}^{L\times T\times d}$ is the original utterances tensor.

4.6 Utterance-Response Matching

Similar to DAM (Zhou et al., 2018c), we utilize the self and cross matching paradigm to construct better matching feature maps.

4.6.1 Origin Matching

Given the filtered utterances $\hat{\mathbf{U}}_i = [\hat{\mathbf{u}}_{i1}, \ldots, \hat{\mathbf{u}}_{ij}, \ldots, \hat{\mathbf{u}}_{iL}]$ and candidate response $\mathbf{r}_i \in \mathbb{R}^{T\times d}$, they are then used to construct a word-word similarity matrix $\mathbf{M_1} \in \mathbb{R}^{L\times 2\times T\times T}$ by dot product and cosine similarity. Both of them are stacked together as the channel dimension. The process can be formulated as:

$$\mathbf{M_1} = [\hat{\mathbf{U}}_i \mathbf{A_1} \mathbf{r}_i^T; \mathbf{cos}(\hat{\mathbf{U}}_i, \mathbf{r}_i)]\,. \tag{12}$$

where $\mathbf{A_1} \in \mathbb{R}^{d\times d}$ is a linear transformation matrix.

4.6.2 Self Matching

Then, we use the Attentive Module over word dimension to construct multi-grained representations, which is formulated as:

$$\begin{aligned} \hat{\mathbf{U}}_i^{self} &= \mathbf{AttentiveModule}(\hat{\mathbf{U}}_i, \hat{\mathbf{U}}_i, \hat{\mathbf{U}}_i) \\ \mathbf{r}_i^{self} &= \mathbf{AttentiveModule}(\mathbf{r}_i, \mathbf{r}_i, \mathbf{r}_i)\,. \end{aligned} \tag{13}$$

By this means, the words in each utterance or candidate response are connected together repeatedly to

combine more and more overall characterizations. Different from DAM (Zhou et al., 2018c), we do not stack many Attentive Module layers because it will drastically increase the computational expense.

Then, we use them to construct $\mathbf{M_2} \in \mathbb{R}^{L\times 2\times T\times T}$, whose element is

$$\mathbf{M}_2 = [\hat{\mathbf{U}}_i^{self}\mathbf{A_2}(\mathbf{r}_i^{self})^T; \mathbf{cos}(\hat{\mathbf{U}}_i^{self}, \mathbf{r}_i^{self})], \quad (14)$$

where $\mathbf{A_2} \in \mathbb{R}^{d\times d}$ is a linear transformation matrix.

4.6.3 Cross Matching

Similarly, we build the semantic association between every utterance and response by the attentive module:

$$\begin{aligned}\hat{\mathbf{U}}_i^{cross} &= \mathbf{AttentiveModule}(\hat{\mathbf{U}}_i, \mathbf{r}_i, \mathbf{r}_i), \\ \mathbf{r}_i^{cross} &= \mathbf{AttentiveModule}(\mathbf{r}_i, \hat{\mathbf{U}}_i, \hat{\mathbf{U}}_i).\end{aligned} \quad (15)$$

In this way, we can make the inter-dependent segment pairs close to each other, and aliment scores between those latently inter-dependent pairs could get increased, which will better encode the dependency relation into representation.

Finally, we use $\hat{\mathbf{U}}_i^{cross}$ and $\mathbf{r}_i^{cross}$ to construct $\mathbf{M}_3 \in \mathbb{R}^{L\times 2\times T\times T}$, whose element is

$$\mathbf{M}_3 = [\hat{\mathbf{U}}_i^{cross}\mathbf{A_3}(\mathbf{r}_i^{cross})^T; \mathbf{cos}(\hat{\mathbf{U}}_i^{cross}, \mathbf{r}_i^{cross})], \quad (16)$$

where $\mathbf{A_3} \in \mathbb{R}^{d\times d}$ is a linear transformation.

4.7 Aggregation

MSN aggregates all the matching matrices together $\mathbf{M} = [\mathbf{M}_1; \mathbf{M}_2; \mathbf{M}_3] \in \mathbb{R}^{L\times 6\times T\times T}$ and applies 2D CNN and max pooling for matching feature extraction and use GRU to model the temporal relationship of utterances in the context, which is the same as SMN (Wu et al., 2017).

Then we compute matching score $g(U_i, r_i)$ based on the matching features. Specifically, we use the final state of GRU output h_L as features and apply a single-layer perceptron to obtain score:

$$g(U_i, r_i) = \sigma(\mathbf{W}h_L + b), \quad (17)$$

where $\mathbf{W}$ and b are learnt parameters, $\sigma(\cdot)$ is sigmoid activation function.

Finally, the negative log-likelihood is used as a loss function to optimize the training process.

5 Experiment

5.1 Dataset

We test MSN on three widely used multi-turn response selection datasets, the Ubuntu Corpus (Lowe et al., 2015), the Douban Corpus (Wu et al., 2017) and the E-commerce Corpus (Zhang et al., 2018). Data statistics are in Table 3.

Table 3: Data statistics for Ubuntu, Douban and E-commerce datasets.

Models	Ubuntu			Douban			E-commerce		
	Train	Val	Test	Train	Val	Test	Train	Val	Test
#context-response pairs	1M	500K	500K	1M	50K	50K	1M	10K	10K
#candidates per context	2	10	10	2	2	10	2	2	10
Avg #turns per context	10.13	10.11	10.11	6.69	6.75	6.45	5.51	5.48	5.64
Avg #words per utterance	11.35	11.34	11.37	18.56	18.50	20.74	7.02	6.99	7.11

Ubuntu Corpus consists of English multi-turn conversations about technical support collected from chat logs of the Ubuntu forum.

Douban Corpus contains dyadic dialogs (conversation between two persons) longer than 2 turns from the Douban group [1] which is a popular social networking service in China.

E-commerce Corpus is collected from real-world conversations between customers and customer service staff from Taobao[2], the largest e-commerce platform in China. The dataset contains diverse types of conversations (e.g. commodity consultation, logistics express, recommendation, and chitchat) concerning various commodities.

5.2 Evaluation Metric

Following the previous works (Wu et al., 2017; Zhang et al., 2018; Chaudhuri et al., 2018; Tao et al., 2019), we employ recall at position k in n candidates ($R_n@k$) as evaluation metrics. Apart from $R_n@k$, we use MAP (Mean Average Precision), MRR (Mean Reciprocal Rank), and Precision-at-one P@1 especially for Douban corpus, which is the same as previous works (Wu et al., 2017; Tao et al., 2019). For some dialogues in Douban corpus have more than one true candidate response.

5.3 Baseline Models

Single-turn matching models: Basic models in (Lowe et al., 2015; Kadlec et al., 2015) including RNN, CNN are used in early works. Some advanced single-turn matching models, such as DL2R (Yan et al., 2016), Atten-LSTM (Tan et al.,

[1] https://www.douban.com/group
[2] https://www.taobao.com

Table 4: Experimental results on Ubuntu, Douban and E-commerce datasets. MRFN is the state-of-the-art model until this submission.

Models	Ubuntu Corpus			Douban Corpus						E-commerce Corpus		
	R_{10}@1	R_{10}@2	R_{10}@5	MAP	MRR	P@1	R_{10}@1	R_{10}@2	R_{10}@5	R_{10}@1	R_{10}@2	R_{10}@5
TF-IDF (Lowe et al., 2015)	41.0	54.5	70.8	33.1	35.9	18.0	9.6	17.2	40.5	15.9	25.6	47.7
RNN (Lowe et al., 2015)	40.3	54.7	81.9	39.0	42.2	20.8	11.8	22.3	58.9	32.5	46.3	77.5
CNN (Kadlec et al., 2015)	54.9	68.4	89.6	41.7	44.0	22.6	12.1	25.2	64.7	32.8	51.5	79.2
LSTM (Kadlec et al., 2015)	63.8	78.4	94.9	48.5	53.7	32.0	18.7	34.3	72.0	36.5	53.6	82.8
BiLSTM (Kadlec et al., 2015)	63.0	78.0	94.4	47.9	51.4	31.3	18.4	33.0	71.6	35.5	52.5	82.5
DL2R (Yan et al., 2016)	62.6	78.3	94.4	48.8	52.7	33.0	19.3	34.2	70.5	39.9	57.1	84.2
Atten-LSTM (Tan et al., 2015)	63.3	78.9	94.3	49.5	52.3	33.1	19.2	32.8	71.8	40.1	58.1	84.9
MV-LSTM (Wan et al., 2016)	65.3	80.4	94.6	49.8	53.8	34.8	20.2	35.1	71.0	41.2	59.1	85.7
Match-LSTM (Wang and Jiang, 2016)	65.3	79.9	94.4	50.0	53.7	34.5	20.2	34.8	72.0	41.0	59.0	85.8
Multi-View (Zhou et al., 2016)	66.2	80.1	95.1	50.5	54.3	34.2	20.2	35.0	72.9	42.1	60.1	86.1
SMN (Wu et al., 2017)	72.6	84.7	96.1	52.9	56.9	39.7	23.3	39.6	72.4	45.3	65.4	88.6
DUA (Zhang et al., 2018)	75.2	86.8	96.2	55.1	59.9	42.1	24.3	42.1	78.0	50.1	70.0	92.1
DAM (Zhou et al., 2018c)	76.7	87.4	96.9	55.0	60.1	42.7	25.4	41.0	75.7	-	-	-
MRFN (Tao et al., 2019)	78.6	88.6	97.6	57.1	61.7	44.8	27.6	43.5	78.3	-	-	-
MSN	**80.0**	**89.9**	**97.8**	**58.7**	**63.2**	**47.0**	**29.5**	**45.2**	**78.8**	**60.6**	**77.0**	**93.7**

2015), and MV-LSTM (Wan et al., 2016) are also explored in this work. These models concatenate all context utterances together to match a response.

Multi-turn matching models: Multi-view (Zhou et al., 2016) models utterances from word level view and utterance level view; DL2R model (Yan et al., 2016) reformulates the message with other utterances in the context; SMN (Wu et al., 2017) matches a response with each utterance in the context; DUA (Zhang et al., 2018) formulates previous utterances into context using a proposed deep utterance aggregation model; DAM (Zhou et al., 2018c) constructs representations of utterances in the context and the response with stacked self-attention and cross attention; MRFN (Tao et al., 2019) fuses multiple types of representations with a multi-representation fusion network for response matching.

5.4 Model Training

Our model was implemented by PyTorch (Paszke et al., 2017). Word embeddings were initialized by the results of word2vec (Mikolov et al., 2013) which ran on the dataset, and the dimensionality of word vectors is 200. The hyper-parameter k of selectors is set to 3. We use three convolution layers to extract matching features. The 1st convolution layer has 16 [3,3] filters with [1,1] stride, and its max-pooling size is [2,2] with [2,2] stride. The 2nd convolution layer has 32 [3,3] filters with [1,1] stride, and its max pooling size is also [2,2] with [2,2] stride. The 3rd convolution layer has 64 [3,3] filters with [1,1] stride, and its max pooling size is also [3,3] with [3,3] stride. We set the dimension of the hidden states of GRU as 300. The parameters were updated by Adam algorithm (Kingma and Ba, 2014) and the parameters of Adam, β_1 and β_2 are 0.9 and 0.999 respectively. The learning rate is initialized as 1e-3 and gradually decreased during training. Same as previous works (Wu et al., 2017; Zhang et al., 2018), the maximum utterance length is 50 and the maximum context length (i.e., number of utterances) as 10.

5.5 Experiment Result

Table 4 shows the results of MSN and all baseline models on the datasets. All the experimental results are cited from previous works (Zhang et al., 2018; Chaudhuri et al., 2018; Tao et al., 2019).

Referring to the table, MSN significantly outperforms all other models in terms of most of the metrics on the three datasets, including MRFN, which is the state-of-the-art model until this submission. MSN extends from SMN (Wu et al., 2017) and DAM (Zhou et al., 2018c), and it achieves more than 3% absolute improvement on R_{10}@1 compared with SMN and DAM. The improvement also shows the importance of filtering irrelevant context before matching.

6 Further Analysis

6.1 Ablation Study

We perform a series of ablation experiments over the different parts of the model to investigate their relative importance. Firstly, we use the complete

MSN as the baseline. Then, we gradually remove its modules as follows:

- **w/o Word Selector**: A model that is trained using the utterance selector but without the word selector.
- **w/o Utterance Selector**: A model which is trained without the utterance selector.
- **Only Hop1 (Hop2, Hop3) Selector**: A model which is trained only with hop1 or hop2 or hop3 selector.
- **w/o Selector**: Removing all selector modules and only use the attention module for matching.

Table 5: Ablation study on E-commerce corpus.

Model	$R_{10}@1$	$R_{10}@2$	$R_{10}@5$
MSN$_{base}$	60.6	77.0	93.7
w/o Selector	55.4	74.2	92.5
w/o Word Selector	59.3	76.5	92.4
w/o Utterance Selector	58.6	75.3	92.8
Only Hop1 Selector	58.3	74.9	93.3
Only Hop2 Selector	56.8	76.7	94.6
Only Hop3 Selector	56.6	74.7	94.0

From experimental results in Table 5, we can observe that:

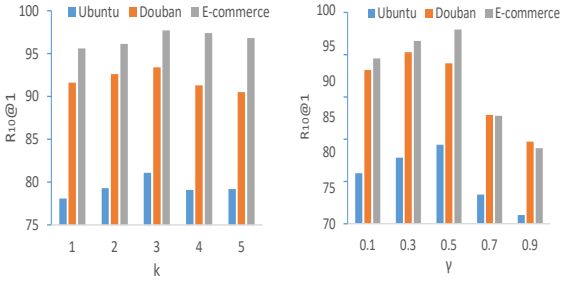
(1) Compared with MSN$_{base}$, removing selectors leads to performance degradation, which shows that the multi-hop selectors are indeed help to improve the selection performance.

(2) The performances decay a large margin when the word selector and utterance selector are removed, which proves that both word selector and utterance selector play an important role in selecting relevant context utterances.

(3) For E-commerce dataset, the context selected by Hop1 selector is more important than other selectors. We think the main reason is that the dialogs in E-commerce corpus happen between buyers and sellers on the Taobao platform. The intent of the dialogue is very clear and the dialogue is mainly in the form of one question and one answer. So the last dialogue session has little dependency on the very far context. However, the fusion of these hop selectors' results still brings more performance improvement.

6.2 Parameter Sensitivity

The choices of k for selectors and threshold γ in formula (10) may influence the performance. Thus, we conduct a series of sensitivity analysis experiments on the development dataset to study how different choices of parameters influence the performance of the model.



(a) Effects of k. (b) Effects of threshold γ.

Figure 3: Parameter sensitivity analysis on the development datasets of Ubuntu, Douban, and E-commerce Corpus.

The k decides how many selectors that MSN uses to select relevant context utterances. Referring to Figure 3 (a), only using hop1 selector is not better than using multiple selectors. However, the performance does not increase when $k > 3$. It is easy to see that when k is too large, the key will contain too many noises and cannot reflect the intention of the last dialogue session.

Figure 3 (b) shows the performance with different threshold γ. Intuitively, when γ is too large, the selectors will filter out too much context, which may hurt performance. However, when γ is too small, the selectors do not work very well. We can observe that MSN achieves the best performance when $\gamma = 0.3$ or 0.5.

7 Conclusion and Future Work

In this paper, we analyze the side effect of using unnecessary context utterances and verify matching-based models are very sensitive to the context. We propose a multi-hop selector network to alleviate this problem. Empirical results on three large-scale datasets demonstrate the effectiveness of the model in multi-turn response selection and yield new state-of-the-art results at the same time.

In the future, we will study how to solve the logical consistency problem between utterances and candidate responses to improve selection performance.

8 Acknowledgement

We gratefully thank the anonymous reviewers for their insightful comments. This research is supported in part by the Beijing Municipal Science and Technology Project under Grant Z191100007119008 and Z181100002718004, the National Key Research and Development Program of China under Grant 2018YFC0806900 and 2017YFB1010000.

References

Debanjan Chaudhuri, Agustinus Kristiadi, Jens Lehmann, and Asja Fischer. 2018. Improving response selection in multi-turn dialogue systems by incorporating domain knowledge. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 497–507.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Diana Inkpen, and Si Wei. 2018. Neural natural language inference models enhanced with external knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2406–2417.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031.

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

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

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

Feng-Lin Li, Minghui Qiu, Haiqing Chen, Xiongwei Wang, Xing Gao, Jun Huang, Juwei Ren, Zhongzhou Zhao, Weipeng Zhao, Lei Wang, et al. 2017. Alime assist: an intelligent assistant for creating an innovative e-commerce experience. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2495–2498. ACM.

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 *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 285–294.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. 2016. Natural language inference by tree-based convolution and heuristic matching. In *The 54th Annual Meeting of the Association for Computational Linguistics*, page 130.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *NIPS-W*.

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 *Thirtieth AAAI Conference on Artificial Intelligence*.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1577–1586.

Heung-Yeung Shum, Xiao-dong He, and Di Li. 2018. From eliza to xiaoice: challenges and opportunities with social chatbots. *Frontiers of Information Technology & Electronic Engineering*, 19(1):10–26.

Ming Tan, Cicero Dos Santos, Bing Xiang, and Bowen Zhou. 2015. Lstm-based deep learning models for nonfactoid answer selection. In *Proceedings of the International Conference on Learning Representations*.

Chongyang Tao, Wei Wu, Can Xu, Wenpeng Hu, Dongyan Zhao, and Rui Yan. 2019. Multi-representation fusion network for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 267–275. ACM.

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

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 *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 2922–2928. AAAI Press.

Hao Wang, Zhengdong Lu, Hang Li, and Enhong Chen. 2013. A dataset for research on short-text conversations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 935–945.

Shuohang Wang and Jing Jiang. 2016. Learning natural language inference with lstm. In *Proceedings of NAACL-HLT*, pages 1442–1451.

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 *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 496–505.

Yu Wu, Wei Wu, Dejian Yang, Can Xu, and Zhoujun Li. 2018. Neural response generation with dynamic vocabularies. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

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*.

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*, pages 55–64. ACM.

Zhuosheng Zhang, Jiangtong Li, Pengfei Zhu, Hai Zhao, and Gongshen Liu. 2018. Modeling multi-turn conversation with deep utterance aggregation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3740–3752.

Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. 2018a. Emotional chatting machine: Emotional conversation generation with internal and external memory. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018b. Commonsense knowledge aware conversation generation with graph attention. In *IJCAI*, pages 4623–4629.

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 *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 372–381.

Xiangyang Zhou, Lu Li, Daxiang Dong, Yi Liu, Ying Chen, Wayne Xin Zhao, Dianhai Yu, and Hua Wu. 2018c. Multi-turn response selection for chatbots with deep attention matching network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1118–1127.