

TripleNet: Triple Attention Network for Multi-Turn Response Selection in Retrieval-based Chatbots

Wentao Ma[†], Yiming Cui^{††}, Nan Shao[†],
Su He[†], Wei-Nan Zhang[‡], Ting Liu[‡], Shijin Wang^{†§}, Guoping Hu[†]

[†]State Key Laboratory of Cognitive Intelligence, iFLYTEK Research, China

[‡]Research Center for Social Computing and Information Retrieval (SCIR),
Harbin Institute of Technology, Harbin, China

[§]iFLYTEK AI Research (Hebei), Langfang, China

^{†§}{wtma, ymcui, nanshao, suhe, sjwang3, gpku}@iflytek.com

[‡]{ymcui, wnzhang, tliu}@ir.hit.edu.cn

Abstract

We consider the importance of different utterances in the context for selecting the response usually depends on the current query.¹ In this paper, we propose the model TripleNet to fully model the task with the triple $\langle context, query, response \rangle$ instead of $\langle context, response \rangle$ in previous works. The heart of TripleNet is a novel attention mechanism named triple attention to model the relationships within the triple at four levels. The new mechanism updates the representation for each element based on the attention with the other two concurrently and symmetrically. We match the triple $\langle C, Q, R \rangle$ centered on the response from char to context level for prediction. Experimental results on two large-scale multi-turn response selection datasets show that the proposed model can significantly outperform the state-of-the-art methods.²

1 Introduction

To establish a human-machine dialogue system is one of the most challenging tasks in Artificial Intelligence (AI). Existing works on building dialogue systems are mainly divided into two categories: retrieval-based method (Yan et al., 2016; Zhou et al., 2016), and generation-based method (Vinyals and Le, 2015). The retrieval-based method retrieves multiple candidate responses from the massive repository and selects the best one as the system’s response, while the generation-based method uses the encoder-decoder framework to generate the response, which is similar to machine translation.

¹In this paper, we define the last message which is waiting for a response as the ‘query,’ the conversation history including the query as ‘context,’ and each message in the context as an ‘utterance.’

² TripleNet source code is available at <https://github.com/wtma/TripleNet>.

A: i downloaded angry ip scanner and now it doesn't work and i can't uninstall it

B: you installed it via package or via some binary installer

A: i installed from ubuntu soft center

B: hm i do n't know what package it is but it should let you remove it the same way

A: ah makes sense then ... hm was it a deb file

True Response: i think it was another format maybe sth starting with r

False Response: thanks i appreciate it try sudo apt-get install libxine-extracodecs

Figure 1: A real example in the Ubuntu Corpus. The upper part is the conversation between speaker A and B. The speaker A want to uninstall the ip scanner and the current query is about the format of the package, so the true response is about the format, but the existing conversation model can be easily misled by the high frequency term ‘install’ as they deal with the query and other utterances in the same way.

In this paper, we are focusing on the retrieval-based method because it is more practical in applications. Selecting a response from a set of candidates is an important and challenging task for the retrieval-based method. Many of the previous approaches are based on Deep Neural Network (DNN) to select the response for single-turn conversation (Lu and Li, 2013). We study multi-turn response selection in this paper, which is rather difficult because it not only requires identification of the important information such as keywords, phrases, and sentences, but also the latent dependencies between the context, query, and candidate response.

Previous works (Zhou et al., 2018; Wu et al., 2017) show that representing the context at different granularities is vital for multi-turn response selection. However, it is not enough for multi-turn response selection. Figure 1 illustrates the problem with a real example in Ubuntu Corpus. As demonstrated, the following two points should be

modeled to solve the problem: (1) the importance of current query should be highlighted, because it has great impact on the importance of different utterances in the context. For example, the query in the case is about the format of the file ('deb file'), which leads the last two utterances (including the query) are more important than the previous ones. If we only match the response with the context, the model may be misled by the high frequency word 'install' and choose the false candidate. (2) the information of different granularities is important, which includes not only the word, utterance, and context level, but also the char level. For example, the different tenses ('install,' 'installed') and the misspelling word ('angry') appear constantly in the conversation. Similar to the role of question for the task of machine reading comprehension (Seo et al., 2016; Cui et al., 2017; Chen et al., 2019), the query in this task is also the key to selecting the response. In this paper, we propose a model named TripleNet to excavate the role of query. The main contributions of our work are listed as follows.

- we use a novel triple attention mechanism to model the relationships within $\langle C, Q, R \rangle$ instead of $\langle C, R \rangle$;
- we propose a hierarchical representation module to fully model the conversation from char to context level;
- The experimental results on Ubuntu and Douban corpus show that TripleNet significantly outperform the state-of-the-art result.

2 Related Works

Earlier works on building the conversation systems are generally based on rules or templates (Walker et al., 2001), which are designed for the specific domain and need much human effort to collect the rules and domain knowledge. As the portability and coverage of such systems are far from satisfaction, people pay more attention to the data-driven approaches for the open-domain conversation system (Ritter et al., 2011; Higashinaka et al., 2014). The main challenge for open-domain conversation is to produce a corresponding response based on the current context. As mentioned previously, the retrieval-based and generation-based methods are the mainstream approaches for conversational response generation.

In this paper, we focus on the task response selection which belongs to retrieval-based approach.

The early studies of response selection generally focus on the single-turn conversation, which use only the current query to select the response (Lu and Li, 2013; Ji et al., 2014; Wang et al., 2015). Since it is hard to get the topic and intention of the conversation by single-turn, researchers turn their attention to multi-turn conversation and model the context instead of the current query to predict the response. First, Lowe et al. (2015) released the Ubuntu Dialogue dataset and proposed a neural model which matches the context and response with corresponding representations via RNNs and LSTMs. Kadlec et al. (2015) evaluate the performances of various models on the dataset, such as LSTMs, Bi-LSTMs, and CNNs. Later, Yan et al. (2016) concatenated utterances with the reformulated query and various features in a deep neural network. Baudiš et al. (2016) regarded the task as sentence pair scoring and implemented an RNN-CNN neural network model with attention. Zhou et al. (2016) proposed a multi-view model with CNN and RNN, modeling the context in both word and utterance view. Further, Xu et al. (2017) proposed a deep neural network to incorporate background knowledge for conversation by LSTM with a specially designed recall gate. Wu et al. (2017) proposed matching the context and response by their word and phrase representations, which had significant improvement from previous work. Zhang et al. (2018) introduced a self-matching attention to route the vital information in each utterance, and used RNN to fuse the matching result. Zhou et al. (2018) used self-attention and cross-attention to construct the representations at different granularities, achieving a state-of-the-art result.

Our model is different from the previous methods: first we model the task with the triple $\langle C, Q, R \rangle$ instead of $\langle C, R \rangle$ in the early works, and use a novel triple attention matching mechanism to model the relationships within the triple. Then we represent the context from low (character) to high (context) level, which constructs the representations for the context more comprehensively.

3 Model

In this section, we will give a detailed introduction of the proposed model TripleNet. We first formal-

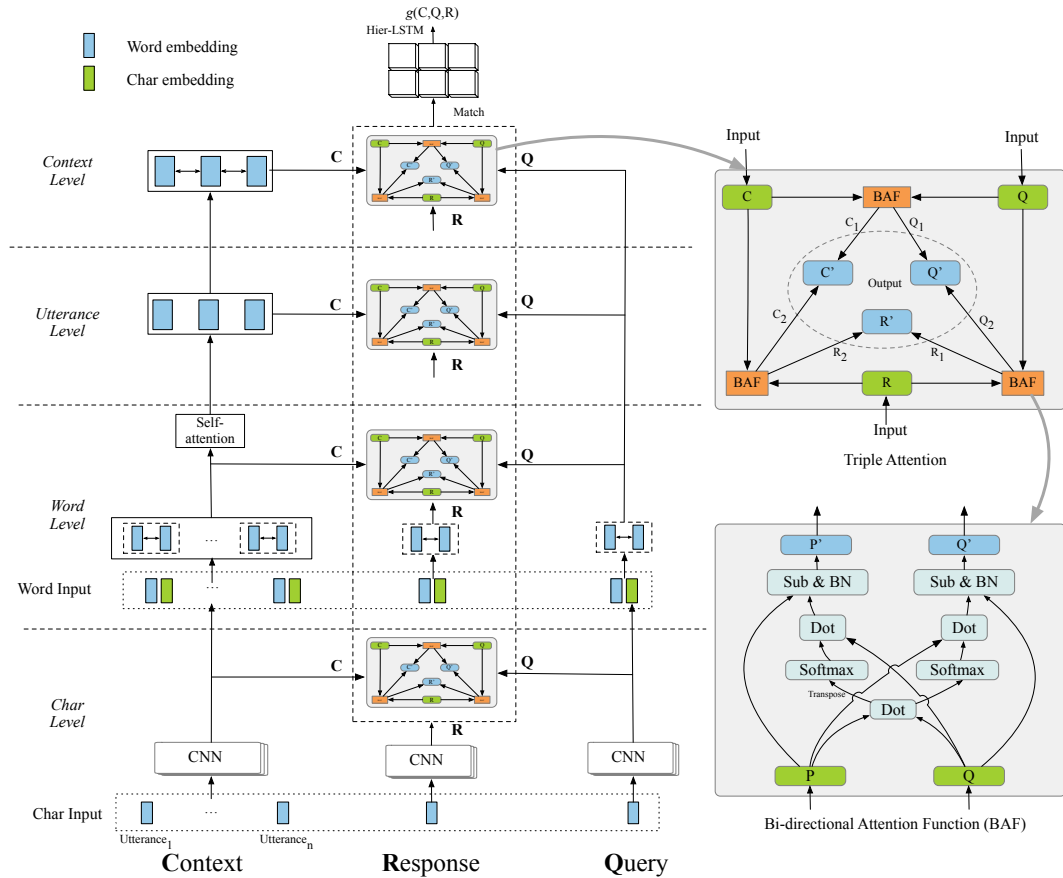


Figure 2: The neural architecture of the model TripleNet. (best viewed in color)

ize the problem of the response selection for multi-turn conversation. Then we briefly introduce the overall architecture of the proposed model. Finally, the details of each part of our model will be illustrated.

3.1 Task Definition

For the response selection, we define the task as given the context C , current query Q and candidate response R , which is different from almost all the previous works (Zhou et al., 2018; Wu et al., 2017). We aim to build a model function $g(C, Q, R)$ to predict the possibility of the candidate response to be the correct response.

$$score = g(C, Q, R) \quad (1)$$

The information in context is composed of four levels: context, utterances, words and characters, which can be formulated as $C = (u_1, u_2, \dots, u_i, \dots, u_n)$, where u_i represents the i th utterance, and n is the maximum utterance number. The last utterance in the context is query $Q = U_n$; we still use query as the end of context to maintain the integrity of the information in

context. Each utterance can be formulated as $u_i = (w_1, \dots, w_j, \dots, w_m)$, where w_j is the j th word in the utterance and m is the maximum word number in the utterance. Each word can be represented by multiply characters $w_j = (ch_1, \dots, ch_k, \dots, ch_l)$, where ch_k is the k th char and l is the length of the word in char-level. The latter two levels are similar in the query and response.

3.2 Model Overview

The overall architecture of the model TripleNet is displayed in Figure 2. The model has a bottom-up architecture that organizes the calculation from char to context level. In each level, we first uses the hierarchical representation module to construct the representations of context, response and query. Then the triple attention mechanism is applied to update the representations. At last, the model matches them while focused on the response and fuses the result for prediction.

In the hierarchical representation module, we represent the conversation in four perspectives including char, word, utterance, and context. In the char-level, a convolutional neural network (CNN)

is applied to the embedding matrix of each word and produces the embedding of the word by convolution and maxpooling operations as the char-level representation. In word-level, we use a shared LSTM layer to obtain the word-level embedding for each word. After that, we use self-attention to encode the representation of each utterance into a vector which is the utterance-level representation. At last, the utterance-level representation of each utterance is fed into another LSTM layer to further model the information among different utterances, forming the context-level representation.

The structure of the triple attention mechanism can be seen in the right part of Figure 2. We first design a bi-directional attention function (BAF) to calculate the attention between two sequences and output their new representations. To model the relationship of the triple $\langle C, Q, R \rangle$, we apply BAF to each pair within the triple and get two new representations for each one element, and then we add them together as its final attention-based representation. In the triple attention mechanism, we can update the representation of each one based on the attention result with the other two simultaneously, and each element participates in the whole calculation in the same way.

3.3 Hierarchical Representation

Char-level Representation. At first, we embed the characters in each word into fixed size vectors and use a CNN followed by max-pooling to get character-derived embeddings for each word, which can be formulated by

$$ch_{j,t} = \tanh(W_1^j * x_{t:t+s_j-1} + b_1^j) \quad (2)$$

$$ch_j = \text{MaxPooling}_{t=0}^L[ch_{j,t}] \quad (3)$$

where W_1^j , b_1^j are parameters, $x_{t:t+s_j-1}$ refers to the concatenation of the embedding of $(x_t, \dots, x_{t+s_j-1})$, s_j is the window size of j th filter, and the ch is the representation of the word in char-level.

Word-level Representation. Furthermore, we embed word x by pre-trained word vectors, and we also introduce a word matching (MF) feature to the embedding to make the model more sensitive to concurrent words. If the word appears in the response and context or query simultaneously, we set the feature to 1, otherwise to 0.

$$e(x) = [W_e \cdot x; ch(x); MF] \quad (4)$$

where $e(x)$ to denotes the embedding representation, W_e is the pre-trained word embedding, and $ch(x)$ is the character embedding function. We use a shared bi-directional LSTM to get contextual word representations in each utterance, query, and the response. The representation of each word is formed by concatenating the forward and backward LSTM hidden output.

$$\overleftarrow{h}(x) = \overleftarrow{\text{LSTM}}(e(x)) \quad (5)$$

$$\overrightarrow{h}(x) = \overrightarrow{\text{LSTM}}(e(x)) \quad (6)$$

$$h(x) = [\overleftarrow{h}(x); \overrightarrow{h}(x)] \quad (7)$$

where $h(x)$ is the representation of the word. We denote the word-level representation of the context as $h_u \in \mathbb{R}^{m*d_w}$ and the response as $h_r \in \mathbb{R}^{m*d_w}$, where d_w is the dimension of Bi-LSTMs. Until now, we have constructed the representations of context, query, and response in char and word level, and we only represent the latter two in these two levels because they don't have such rich contextual information as the context.

Utterance-level Representation. Given the k th utterance $u_k = [h_{u_k}^i]_{i=1}^m$, we construct the utterance-level representation by self-attention (Lin et al., 2017):

$$\alpha_i^k = \text{softmax}(W_3 \tanh(W_2 h_{u_k}(i)^T)) \quad (8)$$

$$u_k = \sum_{i=1}^m h_{u_k}^i \alpha_i^k \quad (9)$$

where $W_2 \in \mathbb{R}^{d*d_w}$, $W_3 \in \mathbb{R}^d$ are trainable weights, d is a hyperparameter, u_k is the utterance-level representation, and α_i^k is the attention weight for the i th word in the k th utterance, which signifies the importance of the word in the utterance.

Context-level Representation. To further model the continuity and contextual information among the utterances, we fed the utterance-level representations into another bi-directional LSTM layer to obtain the representation for each utterance in context perspective.

$$c_k = \text{Bi-LSTM}([u_k]_{k=1}^n) \quad (10)$$

where $c_k \in \mathbb{R}^{d_c}$ is the context-level representation for the k th utterance in the context and d_c is the output size of the Bi-LSTM.

3.4 Triple Attention

In this part, we update the representations of context, query, and response in each level

by triple attention, the motivation of which is to model the latent relationships within $\langle context, query, response \rangle$.

Given the triple $\langle C, Q, R \rangle$, we fed each of its pairs into bi-directional attention function (BAF).

$$C_1, Q_1 = BAF(C, Q) \quad (11)$$

$$C_2, R_1 = BAF(C, R) \quad (12)$$

$$Q_2, R_2 = BAF(Q, R) \quad (13)$$

$$C' = BN(C_1 + C_2) \quad (14)$$

$$Q' = BN(Q_1 + Q_2) \quad (15)$$

$$R' = BN(R_1 + R_2) \quad (16)$$

where BN denotes the batch normalization layer (Ioffe and Szegedy, 2015) which is conducive to preventing vanishing or exploding of gradients. BAF produces the new representations for two sequences (P, Q) by the attention from two directions, which is inspired by Seo et al. (2016). We can formulate it by

$$M_{pq} = P^T \tanh(W_3 Q) \quad (17)$$

$$Att_{pq} = \text{softmax}(M_{pq}) \quad (18)$$

$$Att_{qp} = \text{softmax}(M_{pq}^T) \quad (19)$$

$$P' = P - \tilde{Q}; \quad \tilde{Q} = Q Att_{pq}; \quad (20)$$

$$Q' = Q - \tilde{P}; \quad \tilde{P} = P Att_{qp}; \quad (21)$$

where Att_{pq} , Att_{qp} are the attention between P and Q in two directions, P' , Q' are the new representations the two sequences (P, Q), and we apply a batch normalization layer upon them too.

We find that the triple attention has some interesting features: (1) triple, the representation for each element in the triple $\langle C, Q, R \rangle$ is updated based on the attention to the other two concurrently; (2) symmetrical, which means each element in the triple plays the same role in the structure because their contents are similar in the whole conversation; (3) unchanged dimension, all the outputs of triple attention has the same dimensions as the inputs, so we can stack multiple layers as needed.

3.5 Triple Matching and Prediction

Triple Matching. We match the triple $\langle C, Q, R \rangle$ in each level with the cosine distance using new representations produced by triple attention. This process focuses on the response because it is our target. For example, in the char-level, we match

the triple by

$$\tilde{M}_{rc}^1(i, k, j) = \text{cosine}(ch'_r(i), ch'_{u_k}(j)) \quad (22)$$

$$M_{rc}^1(i, k) = \max_{0 < j < m} \tilde{M}_1(i, j, k) \quad (23)$$

$$M_{rq}^1(i, j) = \text{cosine}(ch'_r(i), ch'_q(j)) \quad (24)$$

$$M_1 = [M_{rc}^1(i, k); M_{rq}^1(i, j)] \quad (25)$$

where ch' is the representation updated by triple attention, $M_1 \in \mathbb{R}^{m \times (n+m)}$ is the char-level matching result, the word-level matches the triple in the same way, and the utterance and the context level match the triple without the maxpooling operation. We use M_2, M_3, M_4 as the matching results in the word, utterance and context levels.

Fusion. After obtaining the four-level matching matrix, we use hierarchical RNN to get highly abstract features. Firstly, we concatenate the four matrices to form a 3D cube $M \in \mathbb{R}^{m \times (n+m) \times 4}$ and we use m as one of the matrix in M , which denotes the matching result for one word in response in four levels.

$$M = [M_1; M_2; M_3; M_4] \quad (26)$$

$$\tilde{m} = \text{MaxPooling}_{i=0}^{n+m} [\text{Bi-LSTM}(m_i)] \quad (27)$$

$$v = \text{MaxPooling}_{j=0}^m [\text{Bi-LSTM}(\tilde{m}_j)] \quad (28)$$

Where m_i and \tilde{m}_j are the i th, j th row in the matrix m and \tilde{m} . We merge the results from different time steps in the outputs of LSTM by max-pooling operation. Until now, we encode the matching result into a single feature vector v .

Final Prediction. For the final prediction, we fed the vector V into a full-connected layer with sigmoid output activation.

$$g(C, Q, R) = \text{sigmoid}(W_4 \cdot v + b_4) \quad (29)$$

where W_4, b_4 are trainable weights. Our purpose is to predict the matching score between the context, query and candidate response, which can be seen as a binary classification task. To train our model, we minimize the cross entropy loss between the prediction and ground truth.

4 Experiments

4.1 Dataset

We first evaluate our model on Ubuntu Dialogue Corpus (Lowe et al., 2015) because it is the largest public multi-turn dialogue corpus which consists of about one million conversations in the specific domain. To reduce the number of unknown words,

	Ubuntu Dialogue 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	90.1	63.8	78.4	94.9	48.5	52.7	32.0	18.7	34.3	72.0
MV-LSTM	90.6	65.3	80.4	94.6	49.8	53.8	34.8	20.2	35.1	71.6
Match-LSTM	90.4	65.3	80.4	94.6	49.8	53.8	34.8	20.2	34.8	71.0
DL2R	89.9	62.6	78.3	94.4	48.8	52.7	33.0	19.3	34.2	70.5
Multi-View	90.8	66.2	80.1	95.1	50.5	54.3	34.2	20.2	35.0	72.9
SMN	92.6	72.6	84.7	96.1	52.9	56.9	39.7	23.3	39.6	72.4
RNN-CNN	91.1	67.2	80.9	95.6	-	-	-	-	-	-
DUA	-	75.2	86.8	96.2	<i>55.1</i>	59.9	42.1	24.3	<i>42.1</i>	<i>78.0</i>
DAM	93.8	76.7	<i>87.4</i>	96.9	55.0	<i>60.1</i>	42.7	25.4	41.0	75.7
TripleNet	94.3	79.0	88.5	97.0	56.4	61.8	44.7	26.8	42.6	77.8
TripleNet _{elmo}	95.1	80.5	89.7	97.6	60.9	65.0	47.0	27.8	48.7	81.4
TripleNet _{ensemble}	95.6	82.1	90.9	98.0	63.2	67.8	51.5	31.3	49.4	83.2

Table 1: Experimental results on two public dialogue datasets. The table is segmented into three sections: Non-Attention models, Attention-based models and our models. The italics denotes the previous best results, and the scores in bold express the new state-of-the-art result of single model without any pre-training layer.

we use the shared copy of the Ubuntu corpus by Xu et al. (2017) which replaces the numbers, paths, and URLs by specific symbols.³ Furthermore, to verify the generalization of our model, we also carry out experiments on Douban Conversation Corpus (Wu et al., 2017), which shares similar format with the Ubuntu corpus but is open-domain and in the Chinese language.

For the Ubuntu corpus, we use the recall at position k in n candidate responses ($R_n@k$) as evaluation metrics, and we use MAP (Mean Average Precision), MRR (Mean Reciprocal Rank), and Precision-at-one as the additional metrics for Douban corpus, following the previous work (Wu et al., 2017).

4.2 Experiment Setup

We implement our model by Keras (Chollet et al., 2015) with TensorFlow backend. In the Embedding Layer, the word embeddings are pre-trained using the training set via GloVe (Pennington et al., 2014), the weights of which are trainable. For char embedding, we set the kernel shape as 3 and filter number as 200 in the CNN layer. For all the Bi-directional LSTM layers, we set their hidden size to 200. We use Adamax (Kingma and Ba, 2014) for weight updating with an initial learning rate of 0.002. For ensemble models, we generate 6 models for each corpus using different random seeds and merge the result by voting.

For better comparison with the baseline models, the main super parameters in TripleNet, such

³https://www.dropbox.com/s/2fdn26rj6h9bpv1/ubuntu_data.zip

as the embedding size, max length of each turn, and the vocabularies, are the same as those of the baseline models. The maximum number of conversation turns, which changes with the models, is 12 in our model, 9 in DAM (Wu et al., 2017), and 10 in SMN (Wu et al., 2017).

4.3 Baseline Models

We basically divided baseline models into two categories for comparisons.

Non-Attention Models. The majority of the previous works on this task are designed without attention mechanisms, including the Sequential Matching Network (SMN) (Wu et al., 2017), Multi-View model (Zhou et al., 2016), Deep Learning to Respond (DL2R) (Yan et al., 2016), Match-LSTM (Wang and Jiang, 2016), MV-LSTM (Wan et al., 2016), and DualEncoder (Lowe et al., 2015).

Attention-based Models. The attention-based models typically match the context and the candidate response based on the attention among them, including DAM (Zhou et al., 2018), DUA (Zhang et al., 2018), and RNN-CNN (Baudiš et al., 2016).

4.4 Overall Results

The overall results on two datasets are depicted in Table 1. Our results are obviously better on the two datasets compared with recently attention-based model DAM, which exceeds 2.3% in $R_{10}@1$ of Ubuntu and 2.6% in $P@1$ of Douban. Furthermore, our score is significantly exceeding in almost all metrics except the $R_{10}@5$ in Douban when compared with DUA, which may be be-

cause the metric is not very stable as the test set in Douban is very small (1000).

To further improve the performance, we utilize pre-trained ELMo (Peters et al., 2018) and fine-tune it on the training set in the Ubuntu condition while we train ELMo from scratch using the Douban training set. As the baseline of Douban corpus is relatively lower, we observe much bigger improvements in the corpus using ELMo. The model ensemble has further improvements based on the single model with ELMo; the score of $R_{10}@1$ in Ubuntu is close to the average performance of human experts at 83.8 (Lowe et al., 2016).

Compared to non-attention models such as the SMN and Multi-view, which match the context and response at two levels, TripleNet shows substantial improvements. On $R_{10}@1$ for Ubuntu corpus, there is a 6.3% absolute improvement from SMN and 12.8% from Multi-view, showing the effectiveness of triple attention.

4.5 Model Ablation

To better demonstrate the effectiveness of TripleNet, we conduct the ablations on the model under the Ubuntu corpus for its larger data size.

We first remove the triple attention and matching parts (-TAM); the result shows a marked decline (2.4% in $R_{10}@1$), which is in the second part of Table 2. The performance of the model is similar to the baseline model DAM. This indicates that our four-level hierarchical representation may play a similar role to the five stacks Transformer in DAM. We then remove the triple attention part, which means we match the pairs $\langle C, R \rangle$ and $\langle Q, R \rangle$ with their original representation in each level; the score of $R_{10}@1$ drops 1.4%, which shows the effect of triple attention. We also have tried to remove all the parts related to the query (-Query). That means the attention and matching parts are only calculated within the pair $\langle C, R \rangle$. It is worth mentioning that the information of the query is still contained at the end of the context. The performance also has a marked drop (1.6% in $R_{10}@1$), which shows that it is necessary to model the query separately. To find out which subsection in those parts is more important, we remove each one of them.

Triple attention matching ablation. As we can see in the third part of Table 2, when attention between context and response is removed (- A_{CR}),

	$R_2@1$	$R_{10}@1$	$R_{10}@2$	$R_{10}@5$
TripleNet	94.3	79.0	88.5	97.0
-TAM	93.5	76.6	86.8	96.6
- A_{tri}	93.8	77.6	87.6	96.9
-Query	93.8	77.4	87.3	96.6
- A_{CR}	94.1	78.4	87.9	97.0
- A_{QR}	94.1	78.5	88.1	97.0
- A_{CQ}	94.3	78.7	88.3	97.0
- M_{CR}	93.7	76.9	87.0	96.7
- M_{QR}	94.4	78.5	88.1	97.1
-char	94.1	78.3	88.0	97.1
-word	94.3	78.5	88.2	97.0
-utterance	94.1	78.6	88.1	97.1
-context	94.0	78.4	88.0	97.0

Table 2: Ablation studies on Ubuntu Dialogue Corpus. The letter ‘A’ stands for the subsection in triple attention, and ‘M’ the is triple matching part.

the largest decrease (0.6% in $R_{10}@1$) appears, which indicates that the relationship between context and response is most important in the triple. The attentions in the other two pairs $\langle C, Q \rangle$ and $\langle Q, R \rangle$ all lead to a slight performance drop (0.3 and 0.5 in $R_{10}@1$), which may be because they overlap with each other for updating the representation of the triple.

When we remove the matching between context and response, we find that the performance of the model has a marked drop (2.1 in $R_{10}@1$), which shows that the relationship within $\langle C, R \rangle$ is the base for selecting the response. The query and response matching part also leads to a significant decline. This shows that we should pay more attention to query within the whole context.

Hierarchical representation ablation. To find out the calculation of which level is most important, we also tried to remove each level calculation from the hierarchical representation module, which can be seen in the fourth part of Table 2. To our surprise, when we remove char (-char) and context level calculation (-context), we observe that the reduction (0.5 in $R_{10}@1$) is more significant than the other two, indicating that we should pay more attention to the lowest and highest level information. Also by removing the other two levels, there is also a significant reduction from TripleNet, which means each level of the three is indispensable for our TripleNet.

From the experiments in this part, we find that each subsection of the hierarchical representation module only leads to a slight performance drop. Maybe it’s because the representation from each

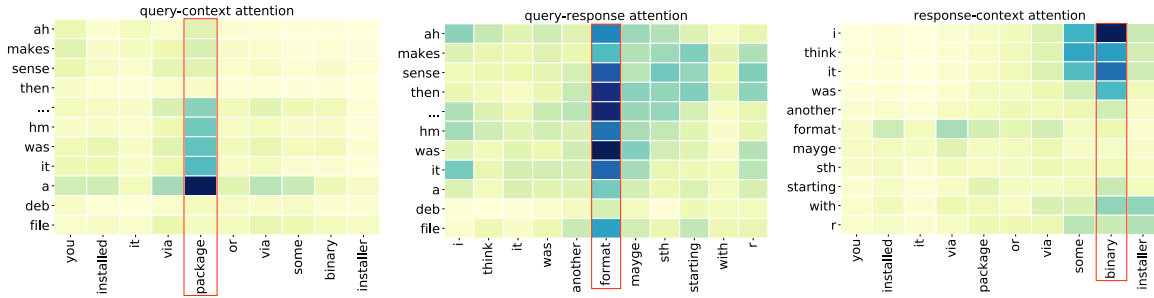


Figure 3: The attention visualization among the query, context, and response in word-level.

level represent the conversation from a unique and indispensable perspective, and the information conveyed by different representations may have some overlap.

5 Analysis and Discussion

5.1 Visualization

By decoding our model for the case in Figure 1, we find that our model TripleNet can choose the true response. To analyze in detail how triple attention works, we get the attention in word-level as the example and visualize it in Figure 3. As there are so many words in the context, we only use the second utterance in the upper part of Figure 1 for its relatively rich semantics.

In the query-context attention, the query mainly pays attention to the keyword ‘package.’ This is helpful to get the topic of the conversation. While the attention of context focuses on the word ‘a’ which is near the key phrase ‘deb file,’ which may be because the representation of the word catches some information from the words nearby by Bi-LSTM. In the query-response attention, the result shows that the attention of the query mainly focuses on the word ‘format,’ which is the most important word in the response. But we can also find that the response does not catch the important words in the query. In the response-context attention, the response pays more attention to the word ‘binary,’ which is another important word in the context.

From the three maps, we find that each attention can catch some important information but miss some useful information too. If we join the information in query-context and response-context attention, we can catch the most important information in the context. Furthermore, the query-response attention can help us catch the most important word in the response. So it is natural for TripleNet

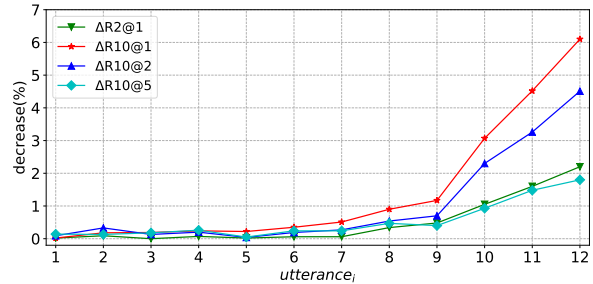


Figure 4: The decrease of the performance when the $utterance_i$ is removed in Ubuntu Corpus.

to select the right response because the model can integrate the three attentions together.

5.2 Discussion

In this section, we will discuss the importance of different utterances in the context. To find out the importance of different utterances in the context, we conduct an experiment by removing each one of them with the model (-Query) in the ablation experiment part because the model deals all the utterances include the query in the same way. For each experiment in this part, we remove the i th ($0 < i < 13$ and $Q = U_{12}$) utterance in the context both in training and evaluation processes and report the decrease of performance in Figure 4. We find that the removing of the query leads the most significant decline (more than 6% in $R_{10}@1$), that indicates the query is much more important than any other utterances. Furthermore, the decrease is stable before the 9th utterances and raises rapidly in the last 3 utterances. We can deduce that the last three utterances are more important than the other ones.

From the whole result, we can conclude that it’s better to model the query separately than deal all of the utterances in the same way for their significantly different importance; we also find that we should pay more attention to the utterances near

the query because they are more important.

6 Conclusion

In this paper, we propose a model TripleNet for multi-turn response selection. We model the context from low (character) to high (context) level, update the representation by triple attention within $\langle C, Q, R \rangle$, match the triple focused on response, and fuse the matching results with hierarchical LSTM for prediction. Experimental results show that the proposed model achieves state-of-the-art results on both Ubuntu and Douban corpus, which ranges from a specific domain to open domain, and English to Chinese language, demonstrating the effectiveness and generalization of our model. In the future, we will apply the proposed triple attention mechanism to other NLP tasks to further testify its extensibility.

Acknowledgement

We would like to thank all anonymous reviewers for their hard work on reviewing and providing valuable comments on our paper. We also would like to thank Yunyi Anderson for proofreading our paper thoroughly. This work is supported by National Key R&D Program of China via grant 2018YFC0832100.

References

- Petr Baudiš, Jan Pichl, Tomáš Vyskočil, and Jan Šedivý. 2016. Sentence pair scoring: Towards unified framework for text comprehension. *arXiv preprint arXiv:1603.06127*.
- Zhipeng Chen, Yiming Cui, Wentao Ma, Shijin Wang, and Guoping Hu. 2019. Convolutional spatial attention model for reading comprehension with multiple-choice questions. *Proceedings of the AAAI, Honolulu, HI*.
- François Chollet et al. 2015. Keras. <https://keras.io>.
- Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. 2017. Attention-over-attention neural networks for reading comprehension. In *Proceedings of the 55th Annual Meeting of ACL (Volume 1: Long Papers)*, pages 593–602. Association for Computational Linguistics.
- 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 *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 928–939. Dublin City University and Association for Computational Linguistics.
- Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456.
- Zongcheng Ji, Zhengdong Lu, and Hang Li. 2014. An information retrieval approach to short text conversation. *arXiv preprint arXiv:1408.6988*.
- Rudolf Kadlec, Martin Schmid, and Jan Kleindienst. 2015. Improved deep learning baselines for ubuntu corpus dialogs. *arXiv preprint arXiv:1510.03753*.
- Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.
- 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. Association for Computational Linguistics.
- Ryan Lowe, Iulian V Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. On the evaluation of dialogue systems with next utterance classification. *arXiv preprint arXiv:1605.05414*.
- Zhengdong Lu and Hang Li. 2013. A deep architecture for matching short texts. In *International Conference on Neural Information Processing Systems*, pages 1367–1375.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP 2014*, pages 1532–1543. Association for Computational Linguistics.
- Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT-2018*, pages 2227–2237. Association for Computational Linguistics.
- Alan Ritter, Colin Cherry, and William B. Dolan. 2011. Data-driven response generation in social media. In *Proceedings of EMNLP 2011*, pages 583–593. Association for Computational Linguistics.
- Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

- Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.
- Marilyn A Walker, Rebecca Passonneau, and Julie E Boland. 2001. Quantitative and qualitative evaluation of darpa communicator spoken dialogue systems. In *Proceedings of ACL 2001*, pages 515–522. Association for Computational Linguistics.
- 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*.
- Mingxuan Wang, Zhengdong Lu, Hang Li, and Qun Liu. 2015. Syntax-based deep matching of short texts. *arXiv preprint arXiv:1503.02427*.
- 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 ACL 2017*, pages 496–505. Association for Computational Linguistics.
- Zhen Xu, Bingquan Liu, Baoxun Wang, Chengjie Sun, and Xiaolong Wang. 2017. Incorporating loose-structured knowledge into conversation modeling via recall-gate lstm. In *International Joint Conference on Neural Networks*, pages 3506–3513.
- Rui Yan, Yiping Song, and Hua Wu. 2016. Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 55–64.
- 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.
- 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 EMNLP 2016*, pages 372–381. Association for Computational Linguistics.
- 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 *Proceedings of ACL 2018*, volume 1, pages 1118–1127.