

Sequential Attention-based Network for Noetic End-to-End Response Selection

Qian Chen, Wen Wang

Speech Lab, DAMO Academy, Alibaba Group
{tanqing.cq, w.wang}@alibaba-inc.com

Abstract

The noetic end-to-end response selection challenge as one track in Dialog System Technology Challenges 7 (DSTC7) aims to push the state of the art of utterance classification for real world goal-oriented dialog systems, for which participants need to select the correct next utterances from a set of candidates for the multi-turn context. This paper describes our systems that are ranked the top on both datasets under this challenge, one focused and small (Advising) and the other more diverse and large (Ubuntu). Previous state-of-the-art models use hierarchy-based (utterance-level and token-level) neural networks to explicitly model the interactions among different turns' utterances for context modeling. In this paper, we investigate a sequential matching model based only on chain sequence for multi-turn response selection. Our results demonstrate that the potentials of sequential matching approaches have not yet been fully exploited in the past for multi-turn response selection. In addition to ranking the top in the challenge, the proposed model outperforms all previous models, including state-of-the-art hierarchy-based models, and achieves new state-of-the-art performances on two large-scale public multi-turn response selection benchmark datasets.

Introduction

Dialogue systems are gaining more and more attention due to their encouraging potentials and commercial values. With the recent success of deep learning models (Serban et al. 2016), building an end-to-end dialogue system became feasible. However, building end-to-end multi-turn dialogue systems is still quite challenging, requiring the system to memorize and comprehend multi-turn conversation context, rather than only considering the current utterance as in single-turn dialogue systems.

Multi-turn dialogue modeling can be divided into generation-based methods (Serban et al. 2016; Zhou et al. 2017) and retrieval-based methods (Lowe et al. 2015; Wu et al. 2017). The latter is the focus of the noetic end-to-end response selection challenge in DSTC7¹ (Yoshino et al. 2018). Retrieval-based methods select the best response from a candidate pool for the multi-turn context, which can

be considered as performing a multi-turn response selection task. The typical approaches for multi-turn response selection mainly consist of sequence-based methods (Lowe et al. 2015; Yan, Song, and Wu 2016) and hierarchy-based methods (Zhou et al. 2016; Wu et al. 2017; Zhang et al. 2018; Wu et al. 2018). Sequence-based methods usually concatenate the context utterances into a long sequence. Hierarchy-based methods normally model each utterance individually and then explicitly model the interactions among the utterances.

Recently, previous work (Wu et al. 2017; Zhang et al. 2018) claims that hierarchy-based methods with complicated networks can achieve significant gains over sequence-based methods. However, in this paper, we investigate the efficacy of a sequence-based method, i.e., Enhanced Sequential Inference Model (ESIM) (Chen et al. 2017a) originally developed for the natural language inference (NLI) task. Our systems are ranked the top on both datasets, i.e., Advising and Ubuntu datasets, under the DSTC7 response selection challenge. In addition, the proposed approach outperforms all previous models, including the previous state-of-the-art hierarchy-based methods, on two large-scale public benchmark datasets, the Lowe's Ubuntu (Lowe et al. 2015) and E-commerce datasets (Zhang et al. 2018).

Hierarchy-based methods usually use extra neural networks to explicitly model the multi-turn utterances' relationship. They also usually need to truncate the utterances in the multi-turn context to make them the same length and shorter than the maximum length. However, the lengths of different turns usually vary significantly in real tasks. When using a large maximum length, we need to add a lot of zero padding in hierarchy-based methods, which will increase computation complexity and memory cost drastically. When using a small maximum length, we may throw away some important information in the multi-turn context. We propose to use a sequence-based model, the ESIM model, in the multi-turn response selection task to effectively address the above mentioned problem encountered by hierarchy-based methods. We concatenate the multi-turn context as a long sequence, and convert the multi-turn response selection task into a sentence pair binary classification task, i.e., whether the next sentence is the response for the current context. There are two major advantages of ESIM over hierarchy-based methods. First, since ESIM does not need to make each utterance

the same length, it has less zero padding and hence could be more computationally efficient than hierarchy-based methods. Second, ESIM models the interactions between utterances in the context implicitly, yet in an effective way as described in the model description section, without using extra complicated networks.

Task Description

DSTC7 is divided into three different tracks, and the proposed approach is developed for the noetic end-to-end response selection track. This track focuses on goal-oriented multi-turn dialogs and the objective is to select the correct response from a set of candidates. Participating systems should not be based on hand-crafted features or rule-based systems. Two datasets are provided, i.e., Ubuntu and Advising, which will be introduced in detail in the experiment section.

The response selection track provided series of subtasks that have similar structures, but vary in the output space and available context. In Table 1, \checkmark indicates that the task is evaluated on the marked dataset, and \times indicates not applicable.

Model Description

The multi-turn response selection task is to select the next utterance from a candidate pool, given a multi-turn context. We convert the problem into a binary classification task, similar to the previous work (Lowe et al. 2015; Wu et al. 2017). Given a multi-turn context and a candidate response, our model needs to determine whether or not the candidate response is the correct next utterance. In this section, we will introduce our model, Enhanced Sequential Inference Model (ESIM) (Chen et al. 2017a) originally developed for natural language inference. The model consists of three main components, i.e., input encoding, local matching, and matching composition, as shown in Figure 1(b).

Input Encoding

Input encoding encodes the context information and represents tokens in their contextual meanings. Instead of encoding the context information through complicated hierarchical structures as in hierarchy-based methods, ESIM encodes the context information simply as follows. The multi-turn context is concatenated as a long sequence, which is denoted as $\mathbf{c} = (c_1, \dots, c_m)$. The candidate response is denoted as $\mathbf{r} = (r_1, \dots, r_n)$. Pre-trained word embedding $\mathbf{E} \in \mathbb{R}^{d_e \times |V|}$ is then used to convert \mathbf{c} and \mathbf{r} to two vector sequences $[\mathbf{E}(c_1), \dots, \mathbf{E}(c_m)]$ and $[\mathbf{E}(r_1), \dots, \mathbf{E}(r_n)]$, where $|V|$ is the vocabulary size and d_e is the dimension of the word embedding. There are many kinds of pre-trained word embeddings available, such as GloVe (Pennington, Socher, and Manning 2014) and fastText (Mikolov et al. 2018). We propose a method to exploit multiple embeddings. Given k kinds of pre-trained word embeddings $\mathbf{E}_1, \dots, \mathbf{E}_k$, we concatenate all embeddings for the word i , i.e., $\mathbf{E}(c_i) = [\mathbf{E}_1(c_i); \dots; \mathbf{E}_k(c_i)]$. Then we use a feed-forward layer with ReLU to reduce dimension from $(d_{e_1} + \dots + d_{e_k})$ to d_h .

To represent tokens in their contextual meanings, the context and the response are fed into BiLSTM encoders to ob-

tain context-dependent hidden states \mathbf{c}^s and \mathbf{r}^s :

$$\mathbf{c}_i^s = \text{BiLSTM}_1(\mathbf{E}(\mathbf{c}), i), \quad (1)$$

$$\mathbf{r}_j^s = \text{BiLSTM}_1(\mathbf{E}(\mathbf{r}), j), \quad (2)$$

where i and j indicate the i -th token in the context and the j -th token in the response, respectively.

Local Matching

Modeling the local semantic relation between a context and a response is the critical component for determining whether the response is the proper next utterance. For instance, a proper response usually relates to some keywords in the context, which can be obtained by modeling the local semantic relation. Instead of directly encoding the context and the response as two dense vectors, we use the cross-attention mechanism to align the tokens from the context and response, and then calculate the semantic relation at the token level. The attention weight is calculated as:

$$e_{ij} = (\mathbf{c}_i^s)^T \mathbf{r}_j^s. \quad (3)$$

Soft alignment is used to obtain the local relevance between the context and the response, which is calculated by the attention matrix $\mathbf{e} \in \mathbb{R}^{m \times n}$ in Equation (3). Then for the hidden state of the i -th token in the context, i.e., \mathbf{c}_i^s (already encoding the token itself and its contextual meaning), the relevant semantics in the candidate response is identified as a vector \mathbf{c}_i^d , called dual vector here, which is a weighted combination of all the response's states, more specifically as shown in Equation (4).

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^n \exp(e_{ik})}, \quad \mathbf{c}_i^d = \sum_{j=1}^n \alpha_{ij} \mathbf{r}_j^s, \quad (4)$$

$$\beta_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^m \exp(e_{kj})}, \quad \mathbf{r}_j^d = \sum_{i=1}^m \beta_{ij} \mathbf{c}_i^s, \quad (5)$$

where $\alpha \in \mathbb{R}^{m \times n}$ and $\beta \in \mathbb{R}^{m \times n}$ are the normalized attention weight matrices with respect to the 2-axis and 1-axis. The similar calculation is performed for the hidden state of each token in the response, i.e., \mathbf{r}_j^s , as in Equation (5) to obtain the dual vector \mathbf{r}_j^d .

By comparing vector pair $\langle \mathbf{c}_i^s, \mathbf{c}_i^d \rangle$, we can model the token-level semantic relation between aligned token pairs. The similar calculation is also applied for vector pair $\langle \mathbf{r}_j^s, \mathbf{r}_j^d \rangle$. We collect local matching information as follows:

$$\mathbf{c}_i^l = F([\mathbf{c}_i^s; \mathbf{c}_i^d; \mathbf{c}_i^s - \mathbf{c}_i^d; \mathbf{c}_i^s \odot \mathbf{c}_i^d]), \quad (6)$$

$$\mathbf{r}_j^l = F([\mathbf{r}_j^s; \mathbf{r}_j^d; \mathbf{r}_j^s - \mathbf{r}_j^d; \mathbf{r}_j^s \odot \mathbf{r}_j^d]), \quad (7)$$

where a heuristic matching approach (Mou et al. 2016) with difference and element-wise product is used here to obtain local matching vectors \mathbf{c}_i^l and \mathbf{r}_j^l for the context and response, respectively. F is a one-layer feed-forward neural network with ReLU to reduce the dimension.

Matching Composition

Matching composition is realized as follows. To determine whether the response is the next utterance for the current

Sub	Description	Ubuntu	Advising
1	Select the next utterance from a candidate pool of 100 sentences	✓	✓
2	Select the next utterance from a candidate pool of 120000 sentences	✓	✗
3	Select the next utterance and its paraphrases from a candidate pool of 100 sentences	✗	✓
4	Select the next utterance from a candidate pool of 100 which might not contain the correct next utterance	✓	✓
5	Select the next utterance from a candidate pool of 100 incorporating the external knowledge	✓	✓

Table 1: Task description.

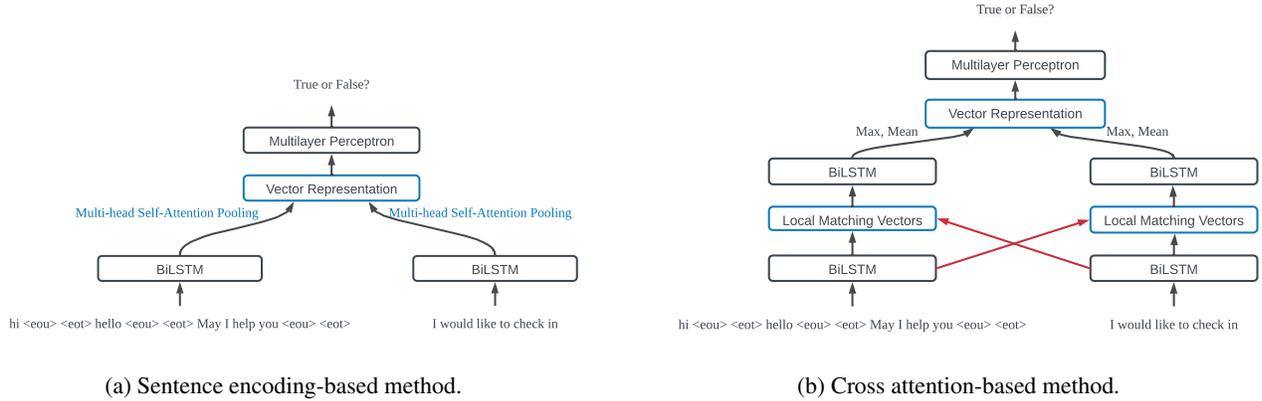


Figure 1: Two kinds of neural network-based methods for sentence pair classification.

context, we explore a composition layer to compose the local matching vectors (c^l and r^l) collected above:

$$c_i^v = \text{BiLSTM}_2(c^l, i), \quad (8)$$

$$r_j^v = \text{BiLSTM}_2(r^l, j). \quad (9)$$

Again we use BiLSTMs as building blocks for the composition layer, but the role of BiLSTMs here is completely different from that in the input encoding layer. The BiLSTMs here read local matching vectors (c^l and r^l) and learn to discriminate critical local matching vectors for the overall utterance-level relationship.

The output hidden vectors of BiLSTM_2 are converted to fixed-length vectors through pooling operations and fed to the final classifier to determine the overall relationship. Max and mean poolings are used and concatenated altogether to obtain a fixed-length vector. Then the final vector is fed to the multi-layer perceptron (MLP) classifier, with one hidden layer, \tanh activation, and softmax output layer. The entire ESIM model is trained via minimizing the cross-entropy loss in an end-to-end manner.

$$y = \text{MLP}([c_{max}^v; c_{mean}^v; r_{max}^v; r_{mean}^v]). \quad (10)$$

Sentence-encoding based methods

For subtask 2 on the Ubuntu dataset, we need to select the next utterance from a candidate pool of 120000 sentences. If we use the cross-attention based ESIM model directly, the computation cost is unacceptable. Instead, we first use a sentence-encoding based method to select the top 100 candidates from 120000 sentences and then rerank them using ESIM. Sentence-encoding based models use the Siamese architecture (Bromley et al. 1993; Chen et al. 2017b) shown in

Figure 1 (a). Parameter-tied neural networks are applied to encode both the context and the response. Then a neural network classifier is applied to decide the relationship between the two sentences. Here, we use BiLSTMs with multi-head self-attention pooling to encode sentences (Lin et al. 2017; Chen, Ling, and Zhu 2018), and an MLP to classify.

We use the same input encoding process as ESIM. To transform a variable length sentence into a fixed length vector representation, we use a weighted summation of all BiLSTM hidden vectors (\mathbf{H}):

$$\mathbf{A} = \text{softmax}(\mathbf{W}_2 \text{ReLU}(\mathbf{W}_1 \mathbf{H}^T + \mathbf{b}_1) + \mathbf{b}_2)^T, \quad (11)$$

where $\mathbf{W}_1 \in \mathbb{R}^{d_a \times 2d_h}$ and $\mathbf{W}_2 \in \mathbb{R}^{d_m \times d_a}$ are weight matrices; $\mathbf{b}_1 \in \mathbb{R}^{d_a}$ and $\mathbf{b}_2 \in \mathbb{R}^{d_m}$ are bias; d_a is the dimension of the attention network and d_h is the dimension of BiLSTMs. $\mathbf{H} \in \mathbb{R}^{T \times 2d_h}$ are the hidden vectors of BiLSTMs, where T denotes the length of the sequence. $\mathbf{A} \in \mathbb{R}^{T \times d_m}$ is the multi-head attention weight matrix, where d_m is a hyper-parameter of the head number that needs to be tuned using the held-out set. Instead of using max pooling or mean pooling, we sum up the BiLSTM hidden states \mathbf{H} according to the weight matrix \mathbf{A} to get a vector representation of the input sentence.

$$\mathbf{V} = \mathbf{A}^T \mathbf{H}, \quad (12)$$

where the matrix $\mathbf{V} \in \mathbb{R}^{d_m \times 2d_h}$ can be flattened into a vector representation $\mathbf{v} \in \mathbb{R}^{2d_h d_m}$. To enhance the relationship between sentence pairs, similarly to ESIM, we concatenate the embeddings of two sentences and their absolute difference and element-wise product (Mou et al. 2016) as the input to the MLP classifier:

$$y = \text{MLP}([\mathbf{v}_c; \mathbf{v}_r; |\mathbf{v}_c - \mathbf{v}_r|; \mathbf{v}_c \odot \mathbf{v}_r]). \quad (13)$$

The MLP has two hidden layers with *ReLU* activation, shortcut connections, and *softmax* output layer. The entire model is trained end-to-end through minimizing the cross-entropy loss.

Experiments

Datasets

We evaluated our model on both datasets of the DSTC7 response selection track, i.e., the Ubuntu and Advising datasets. In addition, to compare with previous methods, we also evaluated our model on two large-scale public multi-turn response selection benchmarks, i.e., the Lowe’s Ubuntu dataset (Lowe et al. 2015) and E-commerce dataset (Zhang et al. 2018).

Ubuntu dataset The Ubuntu dataset includes two party conversations from Ubuntu Internet Relay Chat (IRC) channel (Kummerfeld et al. 2018). Under this challenge, the context of each dialog contains more than 3 turns and the system is asked to select the next turn from the given set of candidate sentences. Linux manual pages are also provided as external knowledge. We used a similar data augmentation strategy as in (Lowe et al. 2015), i.e., we considered each utterance (starting at the second one) as a potential response, with the previous utterances as its context. Hence a dialogue of length 10 yields 9 training examples. To train a binary classifier, we need to sample negative responses from the candidate pool. Initially, we used a 1:1 ratio between positive and negative responses for balancing the samples. Later, we found using more negative responses improved the results, such as 1:4 or 1:9. Considering efficiency, we chose 1:4 in the final configuration for all subtasks except 1:1 for subtask 2.

Advising dataset The advising dataset includes two-party dialogs that simulate a discussion between a student and an academic advisor. Structured information is provided as a database including course information and personas. The data also includes paraphrases of the sentences and the target responses. We used a similar data augmentation strategy as for the Ubuntu dataset based on original dialogs and their paraphrases. The ratio between positive and negative responses is 1:4.33.

Lowe’s Ubuntu dataset This dataset is similar to the DSTC7 Ubuntu data. The training set contains one million context-response pairs and the ratio between positive and negative responses is 1:1. On both development and test sets, each context is associated with one positive response and 9 negative responses.

E-commerce dataset The E-commerce dataset (Zhang et al. 2018) is collected from real-word conversations between customers and customer service staff from Taobao², the largest e-commerce platform in China. The ratio between positive and negative responses is 1:1 in both training and development sets, and 1:9 in the test set.

²<https://www.taobao.com>

Training Details

We used spaCy³ to tokenize text for the two DSTC7 datasets, and used original tokenized text without any further pre-processing for the two public datasets. The multi-turn context was concatenated and two special tokens, `__eou__` and `__eot__`, were inserted, where `__eou__` denotes end-of-utterance and `__eot__` denotes end-of-turn.

The hyperparameters were tuned based on the development set. We used GloVe (Pennington, Socher, and Manning 2014) and fastText (Mikolov et al. 2018) as pre-trained word embeddings. For subtask 5 of the Ubuntu dataset, we also used word2vec (Mikolov et al. 2013) to train word embedding from the provided Linux manual pages. The detail is shown in Table 2. Note that for subtask 5 of the Advising dataset, we tried using the suggested course information as external knowledge but didn’t observe any improvement. Hence, we submitted the results for the Advising dataset without using any external knowledge. For Lowe’s Ubuntu and E-commerce datasets, we used pre-trained word embedding on the training data by word2vec (Mikolov et al. 2013). The pre-trained embeddings were fixed during the training procedure for the two DSTC7 datasets, but fine-tuned for Lowe’s Ubuntu and E-commerce datasets.

Adam (Kingma and Ba 2014) was used for optimization with an initial learning rate of 0.0002 for Lowe’s Ubuntu dataset, and 0.0004 for the rest. The mini-batch size was set to 128 for DSTC7 datasets, 16 for the Lowe’s Ubuntu dataset, and 32 for the E-commerce dataset. The hidden size of BiLSTMs and MLP was set to 300.

To make the sequences shorter than the maximum length, we cut off last tokens for the response but did the cut-off in the reverse direction for the context, as we hypothesized that the last few utterances in the context is more important than the first few utterances. For the Lowe’s Ubuntu dataset, the maximum lengths of the context and response were set to 400 and 150, respectively; for the E-commerce dataset, 300 and 50; for the rest datasets, 300 and 30.

More specially, for subtask 2 of DSTC7 Ubuntu, we used BiLSTM hidden size 400 and 4 heads for sentence-encoding methods. For subtask 4, the candidate pool may not contain the correct next utterance, so we need to choose a threshold. When the probability of positive labels is smaller than the threshold, we predict that candidate pool doesn’t contain the correct next utterance. The threshold was selected from the range [0.50, 0.51, ..., 0.99] based on the development set.

Embedding	Training corpus	#Words
glove.6B.300d	Wikipedia + Gigaword	0.4M
glove.840B.300d	Common Crawl	2.2M
glove.twitter.27B.200d	Twitter	1.2M
wiki-news-300d-1M.vec	Wikipedia + UMBC	1.0M
crawl-300d-2M.vec	Common Crawl	2.0M
word2vec.300d	Linux manual pages	0.3M

Table 2: Statistics of pre-trained word embeddings. The 1-3 rows are from GloVe; 4-5 rows are from fastText; 6 is from word2vec.

³<https://spacy.io/>

Subtask	Measure	Ubuntu	Advising1	Advising2
Subtask1	Recall@1	0.645	0.398	0.214
	Recall@10	0.902	0.844	0.630
	Recall@50	0.994	0.986	0.948
	MRR	0.735	0.5408	0.339
Subtask2	Recall@1	0.067	NA	
	Recall@10	0.185		
	Recall@50	0.266		
	MRR	0.1056		
Subtask3	Recall@1		0.476	0.290
	Recall@10		0.906	0.750
	Recall@50	NA	0.996	0.978
	MRR		0.6238	0.4341
	MAP		0.7794	0.5327
Subtask4	Recall@1	0.624	0.372	0.232
	Recall@10	0.941	0.886	0.692
	Recall@50	0.997	0.990	0.938
	MRR	0.742	0.5409	0.3826
Subtask5	Recall@1	0.653	0.398	0.214
	Recall@10	0.905	0.844	0.630
	Recall@50	0.995	0.986	0.948
	MRR	0.7399	0.5408	0.339

Table 3: The submission results on the hidden test sets for the DSTC7 response selection challenge. NA - not applicable. In total, there are 8 test conditions.

Results

Our results on all DSTC7 response selection subtasks were summarized in Table 3. The challenge ranking considers the average of Recall@10 and Mean Reciprocal Rank (MRR). On the Advising dataset, the test case 2 (Advising2) results were considered for ranking, because test case 1 (Advising1) has some dependency on the training dataset. Our results rank first on 7 subtasks, rank second on subtask 2 of Ubuntu, and overall rank first on both datasets of the DSTC7 response selection challenge⁴. Subtask 3 may contain multiple correct responses, so Mean Average Precision (MAP) is considered as an extra metric.

Ablation Analysis

Ablation analysis is shown in Table 4 and 5 for the Ubuntu and Advising datasets, respectively. For Ubuntu subtask 1, ESIM achieved 0.854 R@10 and 0.6401 MRR. If we removed context’s local matching and matching composition to accelerate the training process (“-CtxDec”), R@10 and MRR dropped to 0.845 and 0.6210. Further discarding the last words instead of the preceding words for the context (“-CtxDec & -Rev”) degraded R@10 and MRR to 0.840 and 0.6174. Ensembling the above three models (“Ensemble”) achieved 0.887 R@10 and 0.6790 MRR. Ensembling was performed by averaging output from models trained with different parameter initializations and different structures.

For Ubuntu subtask 2, the sentence-encoding based methods (“Sent-based”) achieved 0.082 R@10 and 0.0416 MRR. After ensembling several models with different parameter

⁴The official evaluation allows up to 3 different settings, but we only submitted one setting.

Sub	Models	R@1	R@10	R@50	MRR
1	ESIM	0.534	0.854	0.985	0.6401
	-CtxDec	0.508	0.845	0.982	0.6210
	-CtxDec & -Rev	0.504	0.840	0.982	0.6174
	Ensemble	0.573	0.887	0.989	0.6790
2	Sent-based	0.021	0.082	0.159	0.0416
	Ensemble1	0.023	0.091	0.168	0.0475
	ESIM	0.043	0.125	0.191	0.0713
	-CtxDec	0.034	0.117	0.191	0.0620
4	Ensemble2	0.048	0.134	0.194	0.0770
	ESIM	0.515	0.887	0.988	0.6434
	-CtxDec	0.492	0.877	0.987	0.6277
	-CtxDec & -Rev	0.490	0.875	0.986	0.6212
5	Ensemble	0.551	0.909	0.992	0.6771
	ESIM	0.534	0.854	0.985	0.6401
	+W2V	0.530	0.858	0.986	0.6394
	Ensemble	0.575	0.890	0.989	0.6817

Table 4: Ablation analysis on the development set for the DSTC7 Ubuntu dataset.

Sub	Models	R@1	R@10	R@50	MRR
1	-CtxDec	0.222	0.656	0.954	0.3572
	-CtxDec & -Rev	0.214	0.658	0.942	0.3518
	Ensemble	0.252	0.720	0.960	0.4010
3	-CtxDec	0.320	0.792	0.978	0.4704
	-CtxDec & -Rev	0.310	0.788	0.978	0.4550
	Ensemble	0.332	0.818	0.984	0.4848
4	-CtxDec	0.248	0.706	0.970	0.3955
	-CtxDec & -Rev	0.226	0.714	0.946	0.3872
	Ensemble	0.246	0.760	0.970	0.4110

Table 5: Ablation analysis on the development set for the DSTC7 Advising dataset.

initializations (“Ensemble1”), R@10 and MRR were increased to 0.091 and 0.0475. Using ESIM to rerank the top 100 candidates predicted by “Ensemble1” achieved 0.125 R@10 and 0.0713 MRR. Removing context’s local matching and matching composition (“-CtxDec”) degraded R@10 and MRR to 0.117 and 0.0620. Ensembling the above two kinds of ESIM methods (“Ensemble2”) achieved 0.134 R@10 and 0.0770 MRR.

For Ubuntu subtask 4, we observed similar trend with subtask 1. ESIM achieved 0.887 R@10 and 0.6434 MRR, “-CtxDec” degraded performance to 0.877 R@10 and 0.6277 MRR, and “-CtxDec & -Rev” further degraded to 0.875 R@10 and 0.6212 MRR. Ensembling the above three models (“Ensemble”) achieved 0.909 R@10 and 0.6771 MRR.

For Ubuntu subtask 5, the dataset is the same as subtask 1 except for using the external knowledge of Linux manual pages. Adding pre-trained word embeddings derived from Linux manual pages (“+W2V”) resulted in 0.858 R@10 and 0.6394 MRR, comparable with ESIM without exploring the external knowledge. Ensembling the ensemble model for subtask 1 (0.887 R@10 and 0.6790 MRR) and the “+W2V” model brought further gain, reaching 0.890 R@10 and 0.6817 MRR.

Table 5 showed the ablation analysis on the development

Models	Ubuntu			E-commerce		
	R@1	R@2	R@5	R@1	R@2	R@5
TF-IDF (Lowe et al. 2015)	0.410	0.545	0.708	0.159	0.256	0.477
RNN (Lowe et al. 2015)	0.403	0.547	0.819	0.325	0.463	0.775
CNN (Kadlec, Schmid, and Kleindienst 2015)	0.549	0.684	0.896	0.328	0.515	0.792
LSTM (Kadlec, Schmid, and Kleindienst 2015)	0.638	0.784	0.949	0.365	0.536	0.828
BiLSTM (Kadlec, Schmid, and Kleindienst 2015)	0.630	0.780	0.944	0.355	0.525	0.825
MV-LSTM (Wan et al. 2016)	0.653	0.804	0.946	0.412	0.591	0.857
Match-LSTM (Wang and Jiang 2016)	0.653	0.799	0.944	0.410	0.590	0.858
Attentive-LSTM (Tan, Xiang, and Zhou 2015)	0.633	0.789	0.943	0.401	0.581	0.849
Multi-Channel (Wu et al. 2017)	0.656	0.809	0.942	0.422	0.609	0.871
Multi-View (Zhou et al. 2016)	0.662	0.801	0.951	0.421	0.601	0.861
DL2R (Yan, Song, and Wu 2016)	0.626	0.783	0.944	0.399	0.571	0.842
SMN (Wu et al. 2017)	0.726	0.847	0.961	0.453	0.654	0.886
DUA (Zhang et al. 2018)	0.752	0.868	0.962	0.501	0.700	0.921
DAM (Wu et al. 2018)	0.767	0.874	0.969	-	-	-
Our ESIM	0.796	0.894	0.975	0.570	0.767	0.948

Table 6: Comparison of different models on two large-scale public benchmark datasets. All the results except ours are cited from previous work (Zhang et al. 2018; Wu et al. 2018).

set for the Advising dataset. We used ESIM without context’s local matching and matching composition for computational efficiency. We observed similar trends as on the Ubuntu data set. “-CtxDec & -Rev” degraded R@10 and MRR over “-CtxDec”, yet the ensemble of the two models always produced significant gains over individual models.

Comparison with Previous Work

The results on two public benchmarks were summarized in Table 6. The first group of models includes sentence-encoding based methods. They use hand-craft features or neural network features to encode both context and response, then a cosine classifier or MLP classifier was applied to decide the relationship between the two sequences. Previous work used TF-IDF, RNN (Lowe et al. 2015) and CNN, LSTM, BiLSTM (Kadlec, Schmid, and Kleindienst 2015) to encode the context and the response.

The second group of models consists of sequence-based matching models, which usually use the attention mechanism, including MV-LSTM (Wan et al. 2016), Matching-LSTM (Wang and Jiang 2016), Attentive-LSTM (Tan, Xiang, and Zhou 2015), and Multi-Channels (Wu et al. 2017). These models compared the token-level relationship between the context and the response, rather than comparing the two dense vectors directly as in sentence-encoding based methods. These kinds of models achieved significantly better performance than the first group of models.

The third group of models includes more complicated hierarchy-based models, which usually model the token-level and utterance-level information explicitly. Multi-View (Zhou et al. 2016) model utilized utterance relationships from the word sequence view and utterance sequence view. DL2R model (Yan, Song, and Wu 2016) employed neural networks to reformulate the last utterance with other utterances in the context. SMN model (Wu et al. 2017) used CNN and attention to match a response with each utterance in the context. DUA (Zhang et al. 2018) and DAM (Wu et al. 2018) applied a similar framework as SMN (Wu et al.

2017), where one improved with gated self attention and the other improved with the Transformer structure (Vaswani et al. 2017).

Although the previous hierarchy-based work claimed that they achieved the state-of-the-art performance by using the hierarchical structure of multi-turn context, our ESIM sequential matching model outperformed all previous models, including hierarchy-based models. On the Lowe’s Ubuntu dataset, the ESIM model brought significant gains on performance over the previous best results from the DAM model, up to 79.6% (from 76.7%) R@1, 89.4% (from 87.4%) R@2 and 97.5% (from 96.9%) R@5. For the E-commerce dataset, the ESIM model also accomplished substantial improvement over the previous state of the art by the DUA model, up to 57.0% (from 50.1%) R@1, 76.7% (from 70.0%) R@2 and 94.8% (from 92.1%) R@5. These results demonstrated the effectiveness of the ESIM model, a sequential matching method, for multi-turn response selection.

Conclusion

Previous state-of-the-art multi-turn response selection models used hierarchy-based (utterance-level and token-level) neural networks to explicitly model the interactions among the different turns’ utterances for context modeling. In this paper, we demonstrated that a sequential matching model based only on chain sequence can outperform all previous models, including hierarchy-based methods, suggesting that the potentials of such sequential matching approaches have not been fully exploited in the past. Specially, the proposed model achieved top one results on both datasets under the noetic end-to-end response selection challenge in DSTC7, and yielded new state-of-the-art performances on two large-scale public multi-turn response selection benchmarks. Future work on multi-turn response selection includes exploring the efficacy of external knowledge (Chen et al. 2018), such as knowledge graph and user profile.

References

- Bromley, J.; Guyon, I.; LeCun, Y.; Säckinger, E.; and Shah, R. 1993. Signature verification using a siamese time delay neural network. In *Advances in Neural Information Processing Systems 6*, 737–744.
- Chen, Q.; Zhu, X.; Ling, Z.; Wei, S.; Jiang, H.; and Inkpen, D. 2017a. Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017*, 1657–1668.
- Chen, Q.; Zhu, X.; Ling, Z.; Wei, S.; Jiang, H.; and Inkpen, D. 2017b. Recurrent neural network-based sentence encoder with gated attention for natural language inference. In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP, RepEval@EMNLP 2017*, 36–40.
- Chen, Q.; Zhu, X.; Ling, Z.; Inkpen, D.; and Wei, S. 2018. Neural natural language inference models enhanced with external knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018*, 2406–2417.
- Chen, Q.; Ling, Z.; and Zhu, X. 2018. Enhancing sentence embedding with generalized pooling. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018*, 1815–1826.
- Kadlec, R.; Schmid, M.; and Kleindienst, J. 2015. Improved deep learning baselines for ubuntu corpus dialogs. *CoRR* abs/1510.03753.
- Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980.
- Kummerfeld, J. K.; Gouravajhala, S. R.; Peper, J.; Athreya, V.; Gunasekara, C.; Ganhotra, J.; Patel, S. S.; Polymenakos, L.; and Lasecki, W. S. 2018. Analyzing assumptions in conversation disentanglement research through the lens of a new dataset and model. *arXiv preprint arXiv:1810.11118*.
- Lin, Z.; Feng, M.; dos Santos, C. N.; Yu, M.; Xiang, B.; Zhou, B.; and Bengio, Y. 2017. A structured self-attentive sentence embedding. *CoRR* abs/1703.03130.
- Lowe, R.; Pow, N.; Serban, I.; and Pineau, J. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the SIGDIAL 2015 Conference*, 285–294.
- Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *27th Annual Conference on Neural Information Processing Systems 2013.*, 3111–3119.
- Mikolov, T.; Grave, E.; Bojanowski, P.; Puhrsch, C.; and Joulin, A. 2018. Advances in pre-training distributed word representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018*.
- Mou, L.; Men, R.; Li, G.; Xu, Y.; Zhang, L.; Yan, R.; and Jin, Z. 2016. Natural language inference by tree-based convolution and heuristic matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016*.
- Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*, 1532–1543.
- Serban, I. V.; Sordoni, A.; Bengio, Y.; Courville, A. C.; and Pineau, J. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 3776–3784.
- Tan, M.; Xiang, B.; and Zhou, B. 2015. Lstm-based deep learning models for non-factoid answer selection. *CoRR* abs/1511.04108.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. In *Annual Conference on Neural Information Processing Systems 2017*, 6000–6010.
- Wan, S.; Lan, Y.; Xu, J.; Guo, J.; Pang, L.; and Cheng, X. 2016. Match-srnn: Modeling the recursive matching structure with spatial RNN. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016*, 2922–2928.
- Wang, S., and Jiang, J. 2016. Learning natural language inference with LSTM. In *Proceedings of NAACL HLT 2016*, 1442–1451.
- Wu, Y.; Wu, W.; Xing, C.; Zhou, M.; and Li, Z. 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, ACL 2017*, 496–505.
- Wu, H.; Liu, Y.; Chen, Y.; Zhao, W. X.; Dong, D.; Yu, D.; Zhou, X.; and Li, L. 2018. Multi-turn response selection for chatbots with deep attention matching network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018*, 1118–1127.
- Yan, R.; Song, Y.; and Wu, H. 2016. Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In *Proceedings of SIGIR 2016*, 55–64.
- Yoshino, K.; Hori, C.; Perez, J.; D’Haro, L. F.; Polymenakos, L.; Gunasekara, C.; Lasecki, W. S.; Kummerfeld, J.; Galley, M.; Brockett, C.; Gao, J.; Dolan, B.; Gao, S.; Marks, T. K.; Parikh, D.; and Batra, D. 2018. The 7th dialog system technology challenge. *arXiv preprint*.
- Zhang, Z.; Li, J.; Zhu, P.; Zhao, H.; and Liu, G. 2018. Modeling multi-turn conversation with deep utterance aggregation. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018*, 3740–3752.
- Zhou, X.; Dong, D.; Wu, H.; Zhao, S.; Yu, D.; Tian, H.; Liu, X.; and Yan, R. 2016. Multi-view response selection for human-computer conversation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016*, 372–381.
- Zhou, G.; Luo, P.; Cao, R.; Lin, F.; Chen, B.; and He, Q. 2017. Mechanism-aware neural machine for dialogue response generation. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 3400–3407.