Learning to Respond with Deep Neural Networks for Retrieval-Based Human-Computer Conversation System

Rui Yan
Baidu Inc.
No. 10, Xibeiwang East Road,
Beijing 100193, China
yanrui02@baidu.com

Yiping Song
Baidu Inc.
No. 10, Xibeiwang East Road,
Beijing 100193, China
songyiping01@baidu.com

Hua Wu
Baidu Inc.
No. 10, Xibeiwang East Road,
Beijing 100193, China
wu_hua@baidu.com

郁建峰
51174506039
2018/5/3

Outline

1. Background
2. Introduction
3. Approach
4. Experiments
5. Conclusion

Background

- To establish an automatic conversation system between human and computers is regarded as one of the most hardcore problems in computer science, which involves interdisciplinary techniques in **information retrieval**, **natural language processing**, **artificial intelligence**, etc.

- The challenges lie in how to respond so as to maintain a **relevant** and **continuous** conversation with humans.

- In this paper, the author proposes a conversation system through a **deep neural network** framework **driven by web data**.
  - ✓ **Retrieval-based**
  - ✓ **Deep learning-to-respond schema**

Introduction

In this paper, the author proposes a "**deep learning-to-respond**" framework for open-domain conversation systems.

- **Dataset**: Created from Web(zhidao, weibo, etc.), and the crawled data are stored as an **atomic unit of natural conversations**: an **utterance**, namely a **posting**, and its **reply**.

- **Search and retrieval**: For a given **query**, they first apply traditional **keyword-based retrieval methods** and obtain a list of candidate replies.

- **Contextual query reformulation**: Adding its **contexts**, and obtain a set **of reformulated queries** as well as the **original query**.

- **DNN-based scoring, ranking, and ranked list fusion**: a deep neural network (DNN)-based ranker.

Task Modeling

Table 1: An example of the original microblog *posting* and the associated *replies*. Each posting might have more than one reply, e.g., $Reply_1$ and $Reply_2$. To create our database of conversation data, we separate different replies to a same post, and obtain ⟨*post-reply*⟩ pairs. We store two *Posting-Reply* pairs in the conversational dataset, i.e., ⟨$Posting$-$Reply_1$ ⟩ and ⟨$Posting$-$Reply_2$ ⟩. User accounts are anonymized.

Posting: 近视了需要戴眼镜...   (I need a pair of glasses because of the myopia...)
$Reply_1$: 我送你眼镜！   (I will offer the glasses for you!) $Reply_2$: 可以恢复的，别紧张...   (You will be recovered. Don't worry.)

Task Modeling

Table 2: Part (I) indicates a real human (denoted by A) - computer (denoted by B) conversation scenario, while Part (II) indicates our proposed task modeling and formulations. A_2 is the current user-issued query. We have contexts and reformulated queries as listed. '⊞' is the literal concatenation action. Note that the selected response $Reply_1$ is associated with a $Posting$ in the conversational database shown in Table 1.

(I)

Human-Computer Conversation
A_1: 天哪一把年纪的人居然近视了 (OMG I got myopia at such an "old" age) B_1: 真的吗？ (Really?) A_2:嗯哪。求个眼镜做礼物！ (Yeah. Wish a pair of glasses as a gift.)
B_2:我送你眼镜！ (I will offer the glasses for you!)

(II)

Task Formulation
User query: $q_0 = A_2$ **Context information:** $\mathbb{C}=\{c_1=A_1, c_2=B_1\}$ **Reformulated queries:** $q_1=A_2 \boxplus A_1$, $q_2=A_2 \boxplus B_1$ $q_3=A_2 \boxplus A_1 \boxplus B_1, \ldots$
Top-1 ranked response: $r^\star = Reply_1$

Task Modeling

Table 3: Symbols and annotations for problem formulation.

q_0	the current query
r, p	candidate reply with the associated antecedent posting
$\mathbb{C}=\{c_i\}$	contexts (utterances before q_0). c_i is a utterance in $\mathbb{C}$
$\mathcal{Q}=\{q_i\}$	reformulated query: q_0 concatenates with some c_i
$f(.)$	matching metric for *Query-Reply*
$g(.)$	matching metric for *Query-Posting*
$h(.)$	matching metric for *Query-Context*
Input	Conversation repository: $\{\langle p, r\rangle\}$ Query: q_0 Context: $\mathbb{C}$
Output	Selected response: $r^* = \underset{r}{\operatorname{argmax}}\, \mathcal{F}\left(r \mid q_0, \mathbb{C}, \{\langle p, r\rangle\}\right)$

Contextual Reformulation

Generally, context information may be informative (but sometimes may be not) when modeling a query. It is non-trivial to explore different strategies to utilize context information for conversations.

- *No Context*: the simplest reformulation strategy is that no context information will be added, i.e., $\mathcal{Q}_{\text{No Context}} = \{q_0\}$.
- *Whole Context*: we do not distinguish different context sentences and hence incorporate the entire contexts as a whole, i.e., $\mathcal{Q}_{\text{Whole Context}} = \{q_0, q_0 \boxplus \mathbb{C}\}$.
- *Add-One*: we concatenate q_0 with one context sentence, one at a time, i.e., $\mathcal{Q}_{\text{Add-One}} = \{q_0, q_0 \boxplus c_1, \ldots, q_0 \boxplus c_N\}$.
- *Drop-Out*: we concatenate q_0 with the whole context while leave-one-out each context sentence, one at a time, i.e., $\mathcal{Q}_{\text{Drop-Out}} = \{q_0, q_0 \boxplus [\mathbb{C} \backslash c_1], \ldots, q_0 \boxplus [\mathbb{C} \backslash c_N]\}$.
- *Combined*: the combination of all strategies, i.e., $\mathcal{Q} = \mathcal{Q}_{\text{No Context}} \bigcup \mathcal{Q}_{\text{Whole Context}} \bigcup \mathcal{Q}_{\text{Add-One}} \bigcup \mathcal{Q}_{\text{Drop-Out}}$.

Model

Sentence Pair Modeling

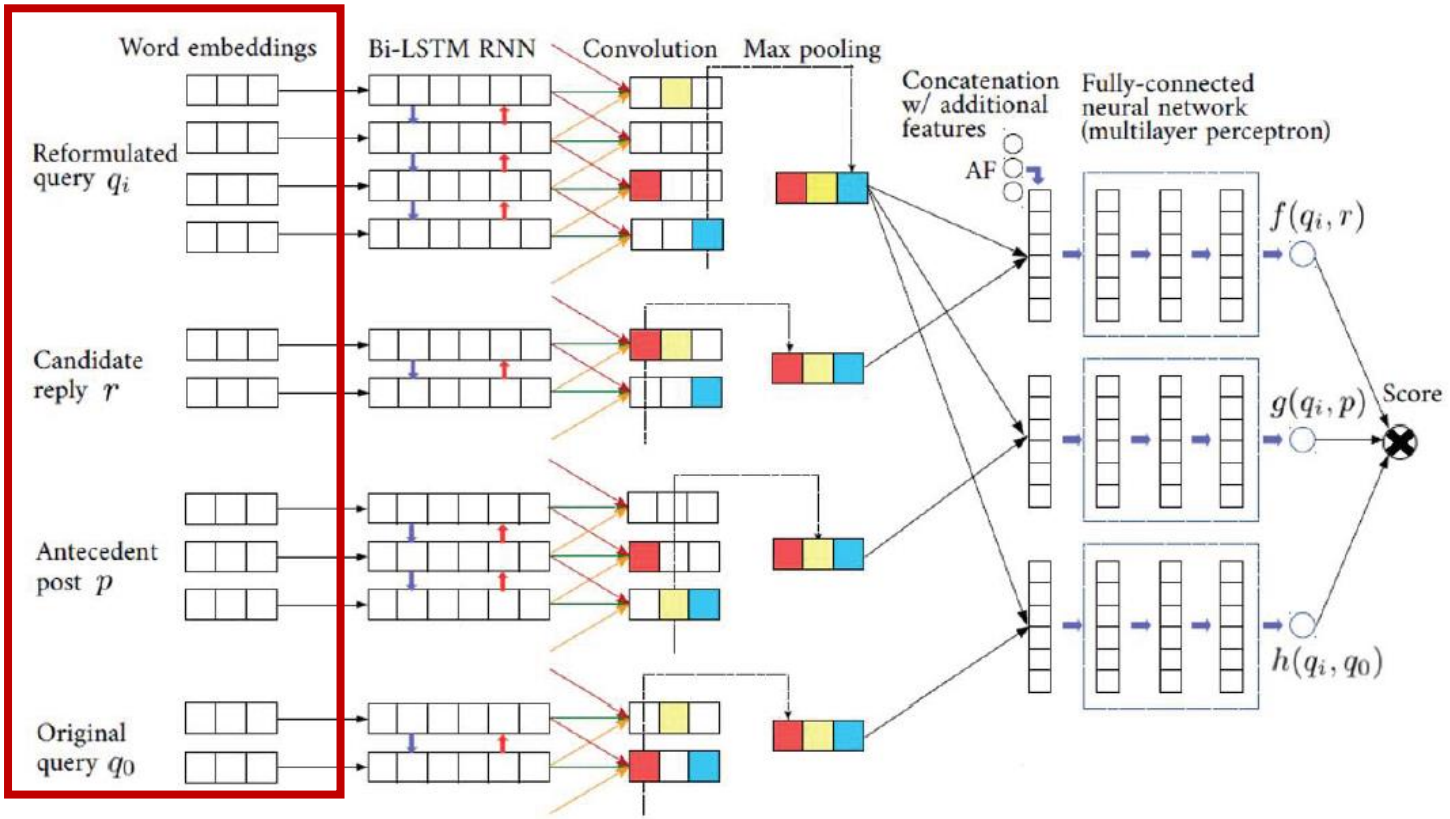
- f(q, r): This scoring function directly judges the relatedness between a reply and the (reformulated) query. A larger score achieved means more relevance of the candidate reply.

- g(q, p): If the associated posting of the reply is similar to the query, its subsequent reply is likely to be an appropriate response.

- h(q, q_0): This scoring function tells how the reformulated query is correlated with the original q_0. A more relevant context, i.e., a larger h(q, q_0), should leads to a more confident ranking result, and the scores from f(q, r) and g(q, p) will be credited with more weights for the final Sum fusion.

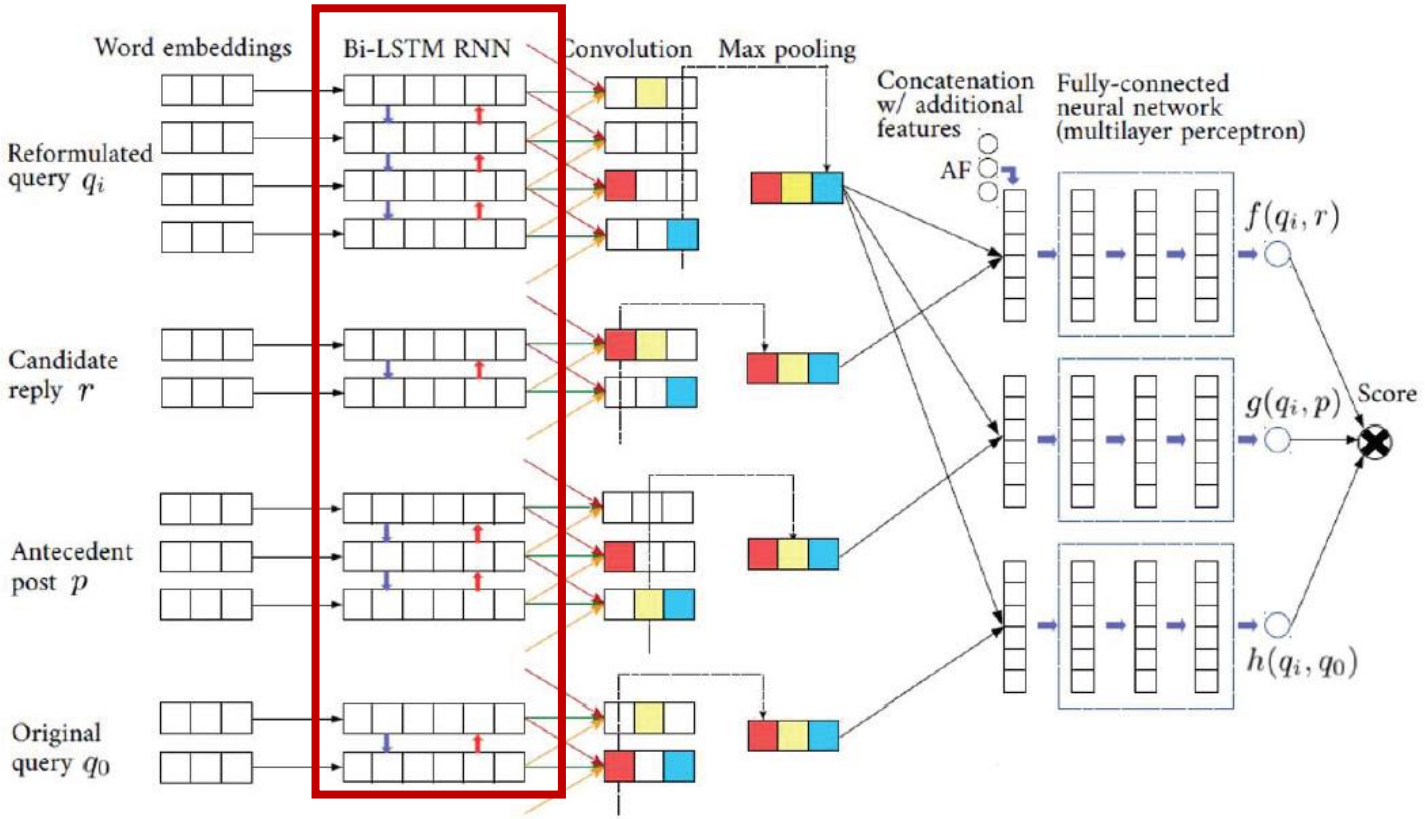
Model

Word Embedding: Word embeddings are initialized randomly, and then tuned during training as part of model parameters.



Model

Bi-LSTM: Using a Bi-LSTM recurrent network to propagate information along the word sequence.



Model

Bi-LSTM: Using a Bi-LSTM recurrent network to propagate information along the word sequence.

$$S = \{x_0, x_1, \ldots, x_T\}$$

$$\begin{bmatrix} i_t \\ f_t \\ o_t \\ l_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} W \cdot \begin{bmatrix} h_{t-1} \\ e_t \end{bmatrix}$$

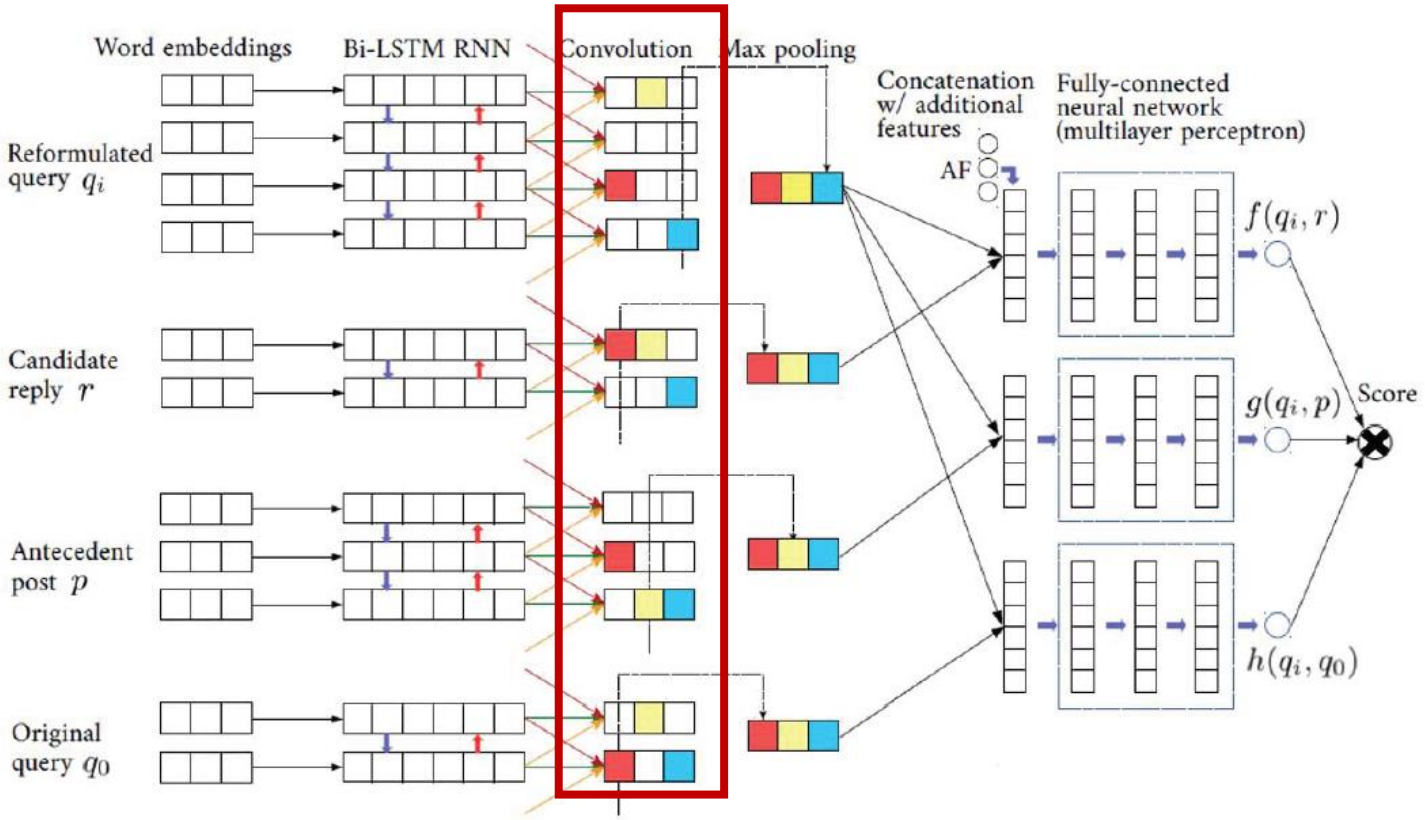
$$\tilde{h}_t = f_t \cdot \tilde{h}_{t-1} + i_t \cdot l_t$$

$$h_t^s = o_t \cdot \tilde{h}_t$$

$$h_t = \left[\overrightarrow{h_t}; \overleftarrow{h_t}\right]$$

Model

CNN: The paper further applies a CNN to extract local neighboring features of successive words.



Model

CNN: The paper further applies a CNN to extract local neighboring features of successive words.

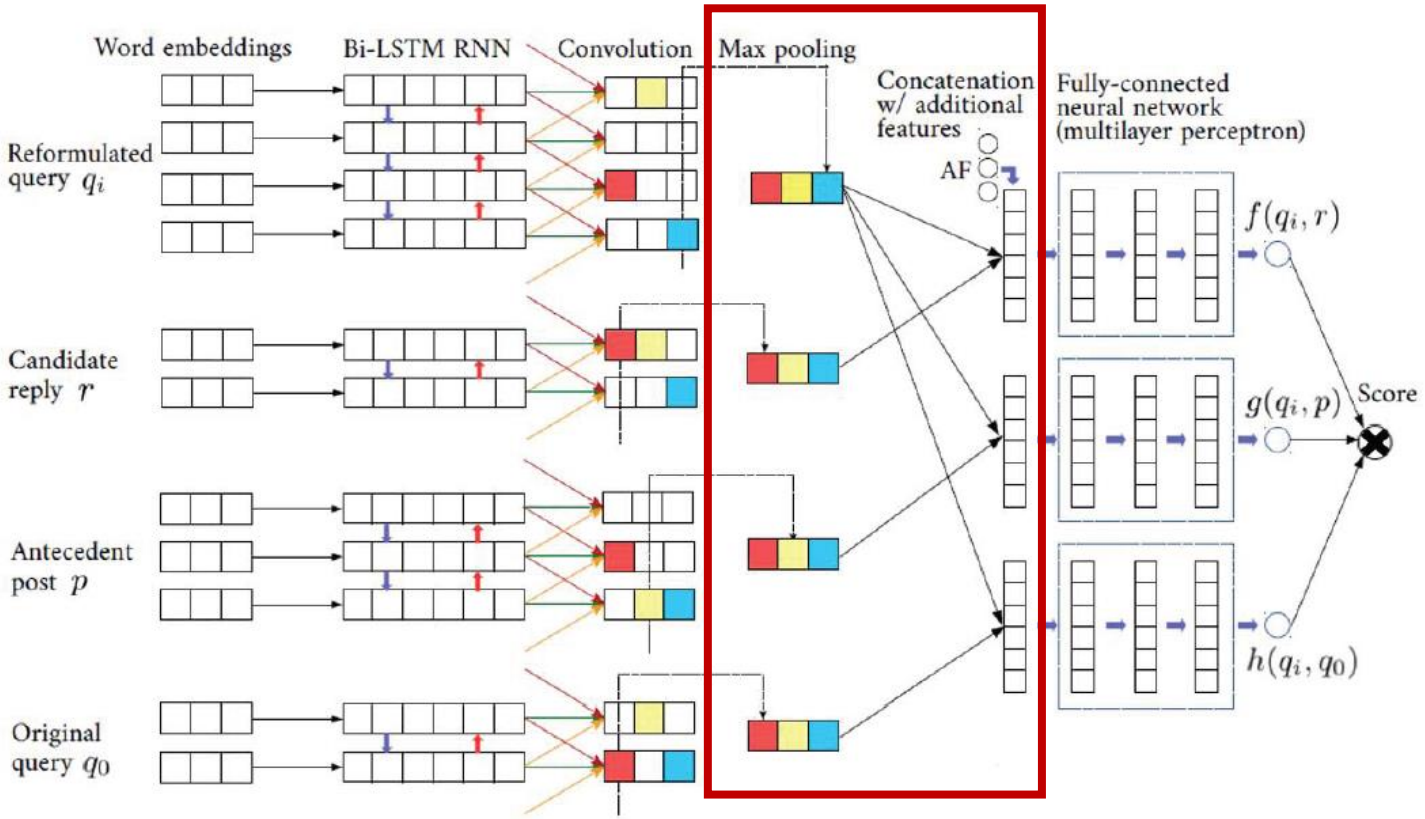
$$(H_t)_m = [h_t, h_{t+1}, \cdots, h_{t+m-1}]$$

$$F = [F(0), \ldots, F(m-1)]$$

$$o_F = \tanh\left[\sum_{i=0}^{m-1} h(t+i) * F(i)\right]$$

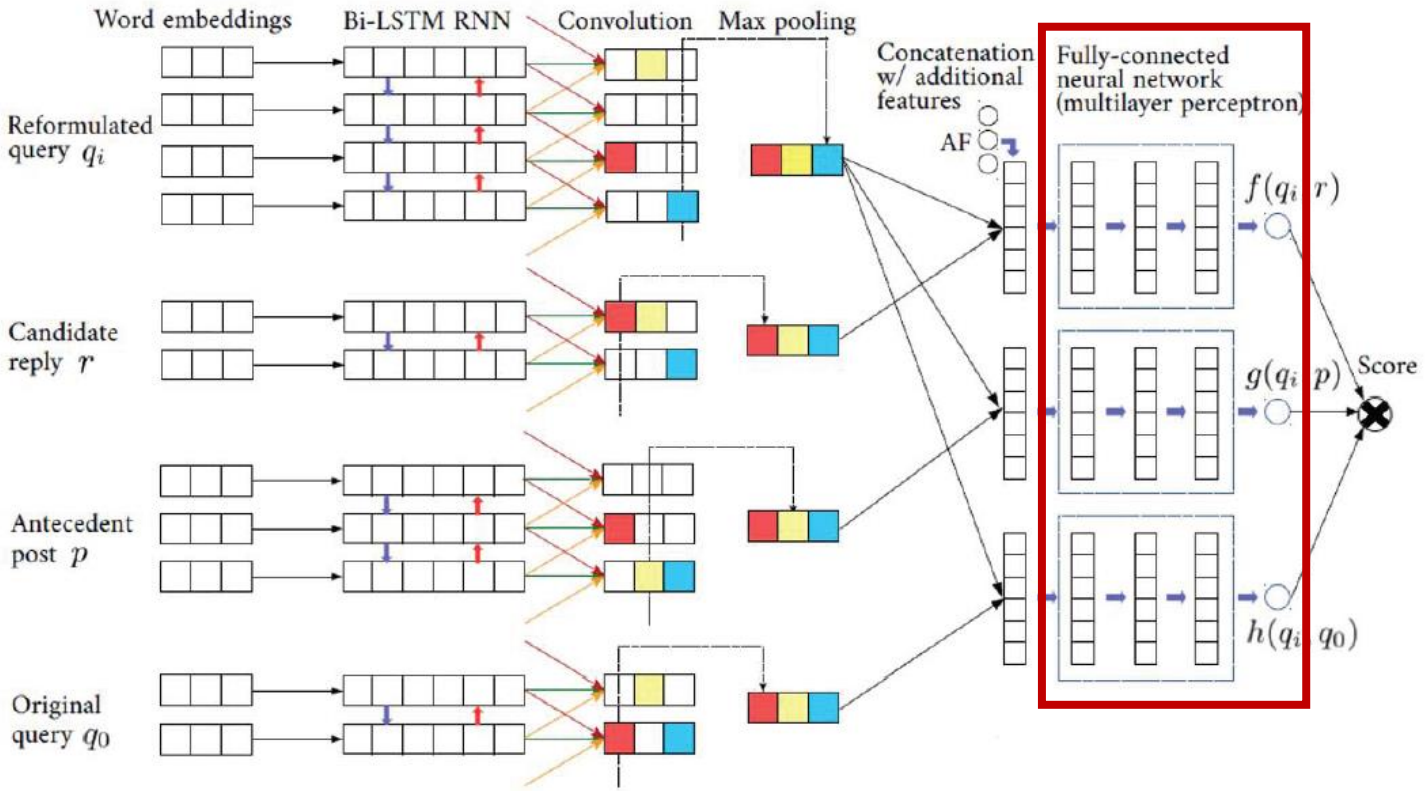
Model

Pooling & Concatenation: <Query-Reply>, <Query-Post>, <Query-Context>



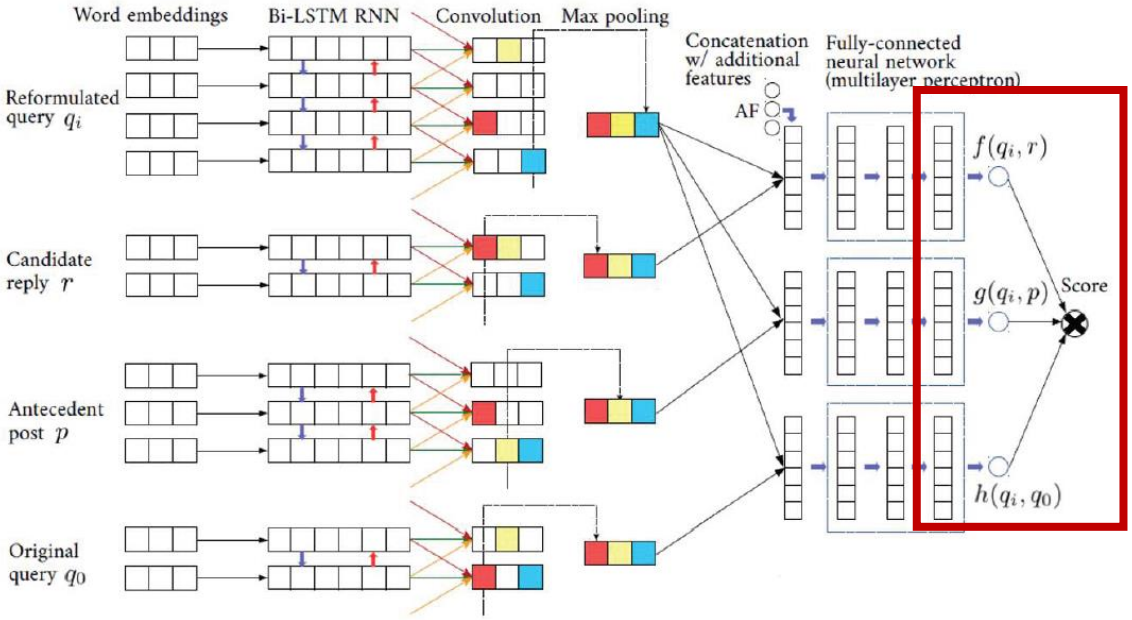
Model

Matching: The joint vector is then passed through a 3-layer, fully-connected, feed-forward neural network.



Model

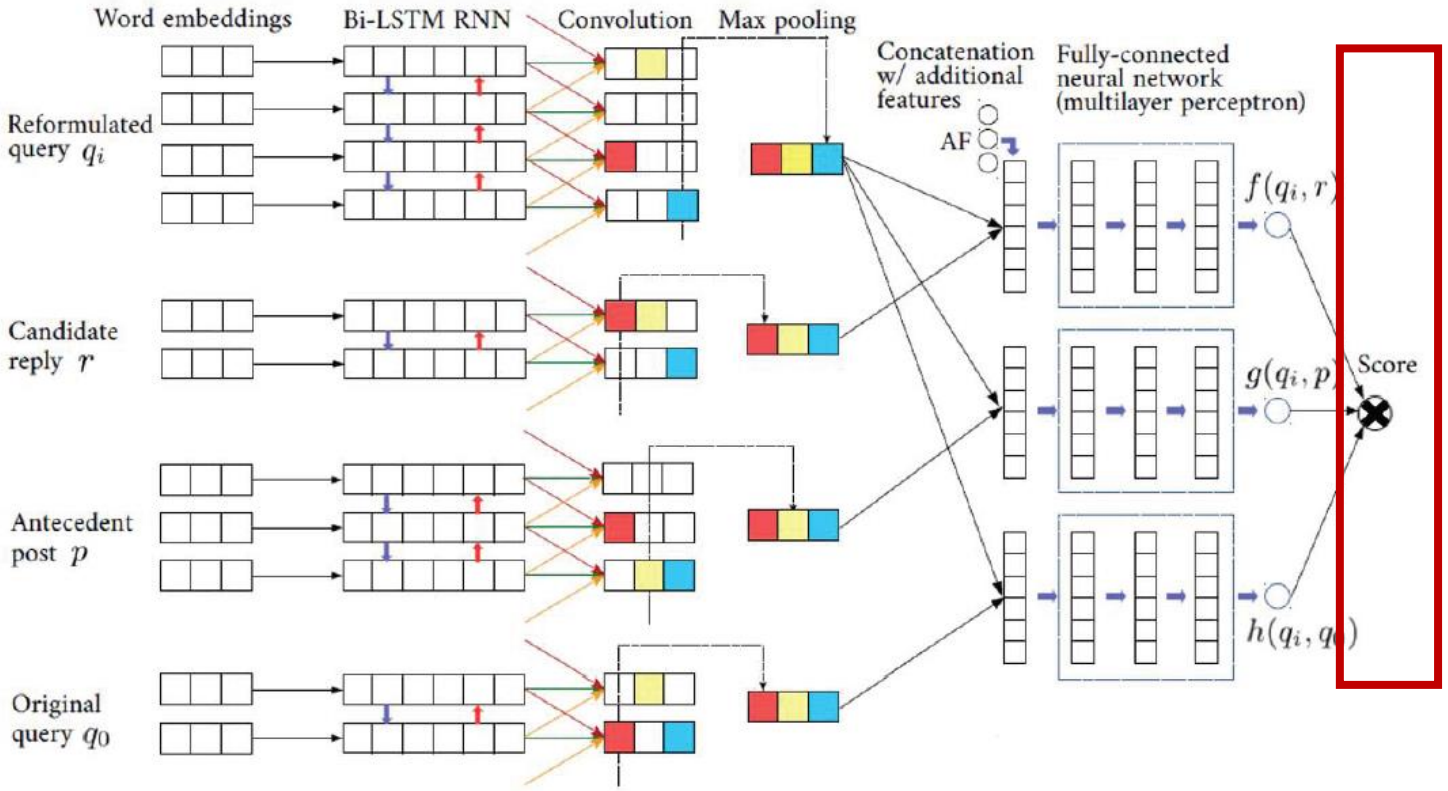
Matching: The joint vector is then passed through a 3-layer, fully-connected, feed-forward neural network.



Finally, a single neuron outputs the matching score of two sentences. As mentioned, $f(q, r)$ is in $\mathbb{R}$; hence the final scoring neuron is essentially a linear regression. For $g(q, p), h(q, q_0) \in (0, 1)$, we apply a sigmoid/logistic function given by $\sigma(\cdot) = \frac{1}{1+e^{-\cdot}}$.

Model

Merging: $\mathcal{F}(q_0, r) = \sum_{i=0}^{|\mathcal{Q}|} \left(h(q_0, q_i) \sum_{p} \left(f(q_i, r) \cdot g(q_i, p) \right) \right)$



Model

Merging: $\mathcal{F}(q_0, r) = \sum_{i=0}^{|\mathcal{Q}|} \left(h(q_0, q_i) \sum_{p} \left(f(q_i, r) \cdot g(q_i, p) \right) \right)$

Hinge loss:

$$\underset{\Omega}{\text{minimize}} \sum_{q_0, r^+} \max\left\{0, \Delta + \mathcal{F}(q_0, r^+) - \mathcal{F}(q_0, r^-)\right\} + \lambda \|\Omega\|_2^2$$

Given a triple $F(q_0,r+)$ in the training set, we randomly sample a negative instance r−. The objective is to maximize the scores of positive samples while minimizing that of the negative samples.

Experiments

The objectives of our experiments are to

1) evaluate the effectiveness of our proposed deep learning-to-respond schema and
2) evaluate contextual reformulation strategies and components of multidimension of ranking evidences for the conversational task.

The author constructed the dataset of 1,606,583 samples to train the deep neural networks, 357,018 for validation, and 11,097 for testing.

Experiments

Table 5: Retrieval performance against baselines with our proposed adaption of contextual reformulation. '⋆' indicates that we accept the improvement hypothesis of DL2R over the best baseline by Wilcoxon test at a significance level of 0.01. Performance of both generative methods and retrieval methods. For generative methods, they generate one response given each query. Hence the p@1 in fact refers to accuracy. Other metrics are not applicable.

Model	p@1	MAP	nDCG@5	nDCG@10	nDCG@20	MRR
SMT (Ritter et al., [26])	0.363					
LSTM-RNN (Sutskever et al., [32])	0.441					
NRM (Shang et al., [29])	0.465					
Random Match	0.266	0.246	0.247	0.289	0.353	0.083
Okapi BM25	0.272	0.253	0.337	0.302	0.368	0.169
DeepMatch (Lu and Li, [17])	0.457	0.317	0.419	0.454	0.508	0.275
LSTM-RNN (Palangi et al., [25])	0.338	0.283	0.330	0.371	0.431	0.228
ARC (Hu et al., [7])	0.394	0.294	0.397	0.421	0.477	0.232
DeepMatch w/ context adaption	0.603	0.378	0.555	0.584	0.628	**0.349**
LSTM-RNN w/ context adaption	0.362	0.296	0.354	0.395	0.453	0.237
ARC w/ context adaption	0.400	0.309	0.383	0.422	0.480	0.319
Deep Learning-to-Respond (DL2R)	**0.731***	**0.416***	**0.663***	**0.682***	**0.717***	0.333

Experiments

Table 5: Retrieval performance against baselines with our proposed adaption of contextual reformulation. '⋆' indicates that we accept the improvement hypothesis of DL2R over the best baseline by Wilcoxon test at a significance level of 0.01. Performance of both generative methods and retrieval methods. For generative methods, they generate one response given each query. Hence the p@1 in fact refers to accuracy. Other metrics are not applicable.

Model	p@1	MAP	nDCG@5	nDCG@10	nDCG@20	MRR
SMT (Ritter et al., [26])	0.363					
LSTM-RNN (Sutskever et al., [32])	0.441					
NRM (Shang et al., [29])	0.465					
Random Match	0.266	0.246	0.247	0.289	0.353	0.083
Okapi BM25	0.272	0.253	0.337	0.302	0.368	0.169
DeepMatch (Lu and Li, [17])	0.457	0.317	0.419	0.454	0.508	0.275
LSTM-RNN (Palangi et al., [25])	0.338	0.283	0.330	0.371	0.431	0.228
ARC (Hu et al., [7])	0.394	0.294	0.397	0.421	0.477	0.232
DeepMatch w/ context adaption	0.603	0.378	0.555	0.584	0.628	**0.349**
LSTM-RNN w/ context adaption	0.362	0.296	0.354	0.395	0.453	0.237
ARC w/ context adaption	0.400	0.309	0.383	0.422	0.480	0.319
Deep Learning-to-Respond (DL2R)	**0.731***	**0.416***	**0.663***	**0.682***	**0.717***	0.333

Experiments

Table 5: Retrieval performance against baselines with our proposed adaption of contextual reformulation. '⋆' indicates that we accept the improvement hypothesis of DL2R over the best baseline by Wilcoxon test at a significance level of 0.01. Performance of both generative methods and retrieval methods. For generative methods, they generate one response given each query. Hence the p@1 in fact refers to accuracy. Other metrics are not applicable.

Model	p@1	MAP	nDCG@5	nDCG@10	nDCG@20	MRR
SMT (Ritter et al., [26])	0.363					
LSTM-RNN (Sutskever et al., [32])	0.441					
NRM (Shang et al., [29])	0.465					
Random Match	0.266	0.246	0.247	0.289	0.353	0.083
Okapi BM25	0.272	0.253	0.337	0.302	0.368	0.169
DeepMatch (Lu and Li, [17])	0.457	0.317	0.419	0.454	0.508	0.275
LSTM-RNN (Palangi et al., [25])	0.338	0.283	0.330	0.371	0.431	0.228
ARC (Hu et al., [7])	0.394	0.294	0.397	0.421	0.477	0.232
DeepMatch w/ context adaption	0.603	0.378	0.555	0.584	0.628	**0.349**
LSTM-RNN w/ context adaption	0.362	0.296	0.354	0.395	0.453	0.237
ARC w/ context adaption	0.400	0.309	0.383	0.422	0.480	0.319
Deep Learning-to-Respond (DL2R)	**0.731***	**0.416***	**0.663***	**0.682***	**0.717***	0.333

Experiments

Table 5: Retrieval performance against baselines with our proposed adaption of contextual reformulation. '⋆' indicates that we accept the improvement hypothesis of DL2R over the best baseline by Wilcoxon test at a significance level of 0.01. Performance of both generative methods and retrieval methods. For generative methods, they generate one response given each query. Hence the p@1 in fact refers to accuracy. Other metrics are not applicable.

Model	p@1	MAP	nDCG@5	nDCG@10	nDCG@20	MRR
SMT (Ritter et al., [26])	0.363					
LSTM-RNN (Sutskever et al., [32])	0.441					
NRM (Shang et al., [29])	0.465					
Random Match	0.266	0.246	0.247	0.289	0.353	0.083
Okapi BM25	0.272	0.253	0.337	0.302	0.368	0.169
DeepMatch (Lu and Li, [17])	0.457	0.317	0.419	0.454	0.508	0.275
LSTM-RNN (Palangi et al., [25])	0.338	0.283	0.330	0.371	0.431	0.228
ARC (Hu et al., [7])	0.394	0.294	0.397	0.421	0.477	0.232
DeepMatch w/ context adaption	0.603	0.378	0.555	0.584	0.628	**0.349**
LSTM-RNN w/ context adaption	0.362	0.296	0.354	0.395	0.453	0.237
ARC w/ context adaption	0.400	0.309	0.383	0.422	0.480	0.319
Deep Learning-to-Respond (DL2R)	**0.731***	**0.416***	**0.663***	**0.682***	**0.717***	0.333

Experiments

Table 5: Retrieval performance against baselines with our proposed adaption of contextual reformulation. '⋆' indicates that we accept the improvement hypothesis of DL2R over the best baseline by Wilcoxon test at a significance level of 0.01. Performance of both generative methods and retrieval methods. For generative methods, they generate one response given each query. Hence the p@1 in fact refers to accuracy. Other metrics are not applicable.

Model	p@1	MAP	nDCG@5	nDCG@10	nDCG@20	MRR
SMT (Ritter et al., [26])	0.363					
LSTM-RNN (Sutskever et al., [32])	0.441					
NRM (Shang et al., [29])	0.465					
Random Match	0.266	0.246	0.247	0.289	0.353	0.083
Okapi BM25	0.272	0.253	0.337	0.302	0.368	0.169
DeepMatch (Lu and Li, [17])	0.457	0.317	0.419	0.454	0.508	0.275
LSTM-RNN (Palangi et al., [25])	0.338	0.283	0.330	0.371	0.431	0.228
ARC (Hu et al., [7])	0.394	0.294	0.397	0.421	0.477	0.232
DeepMatch w/ context adaption	0.603	0.378	0.555	0.584	0.628	**0.349**
LSTM-RNN w/ context adaption	0.362	0.296	0.354	0.395	0.453	0.237
ARC w/ context adaption	0.400	0.309	0.383	0.422	0.480	0.319
Deep Learning-to-Respond (DL2R)	**0.731***	**0.416***	**0.663***	**0.682***	**0.717***	0.333

Experiments

Table 6: Performance evaluations of different contextual query reformulation strategies.

	p@1	MAP	nDCG@5	nDCG@10	nDCG@20	MRR
No Context	0.522	0.340	0.476	0.509	0.559	0.296
Whole Context	0.698	0.404	0.635	0.657	0.696	0.327
Add-One	0.716	0.411	0.650	0.670	0.706	0.322
Drop-Out	0.720	0.413	0.656	0.675	0.711	0.328
Combined	0.731	0.416	0.663	0.682	0.717	0.333

Table 7: Performance evaluations of different components with multi-dimension of ranking evidences.

	p@1	MAP	nDCG@5	nDCG@10	nDCG@20	MRR
Query-Reply w/o Query-Context	0.522	0.340	0.476	0.509	0.559	0.296
Query-Posting w/o Query-Context	0.510	0.302	0.404	0.425	0.489	0.285
Query-Reply w/ Query-Context	0.596	0.366	0.528	0.561	0.603	0.327
Query-Posting w/ Query-Context	0.563	0.362	0.483	0.516	0.568	0.316
Full Combination	0.731	0.416	0.663	0.682	0.717	0.333

Conclusion

In this paper, we propose to establish an automatic conversation system between humans and computers.

Contributions:

1) Propose a contextual query reformulation framework with ranking fusions for the conversation task.
2) Integrate multi-dimension of ranking evidences, i.e., queries, postings, replies and contexts.
3) Establish the deep neural network architecture featured with above strategies and components.

THANKS