How to Answer Comparison Questions

Hongxuan Tang, Yu Hong, Xin Chen, Kaili Wu, Min Zhang
School of Computer Science and Technology, Soochow University
Suzhou, China
{*hxtang,xchen,klwu*}*@stu.suda.edu.cn,tianxianer@gmail.com,minzhang@suda.edu.cn*

***Abstract*—"*Which city has the larger population, Tokyo or New York?*". To answer the question, in general, we necessarily obtain the prior knowledge about the populations of both cities, and accordingly determine the answer by numeric comparison. Using Machine Reading Comprehension (MRC) to answer such a question has become a popular research topic, which is referred to as a task of Comparison Question Answering (CQA). In this paper, we propose a novel neural CQA model which is trained to answer comparison question. The model is designed as a sophisticated neural network which performs inference in a step-by-step pipeline, including the steps of attentive entity detection (e.g., "*city*"), alignment of comparable attributes (e.g., "*population*" of the target "*cities*"), contrast calculation (larger or smaller), as well as binary classification of positive and negative answers. The experimentation on HotpotQA illustrates that the proposed method achieves an average F1 score of 63.09%, outperforming the baseline with about 10% F1 scores. In addition, it performs better than a series of competitive models, including DecompRC, BERT.**

***Keywords*-Machine Reading Comprehension; Comparison Question Answering; Calculation**

I. Introduction

Machine Reading Comprehension (MRC) aims to extract answers from textual data. It is an important task in the field of natural language processing (NLP). After the release of the SQuAD dataset [1], MRC has spurred tremendous interest in the NLP community. CQA is an instance of MRC, questions of which need comparison, counting and arithmetic. It is quite challenging for current MRC models. In this paper, we evaluate our CQA model on comparison type questions in HotpotQA [2] (HotpotCQA). Table 1 shows an example from HotpotCQA.

Table I
An Example from HotpotCQA.

Question: ***Who has more well known films, Quincy Perkins or Bill Forsyth?***
Document α : *Quincy Perkins (born July 16, 1980 in Key West, Florida) is an American director most famous for directing, producing and writing the narrative fiction short film "Swingers Anonymous" which debuted at the (Cannes Film Festival) in 2015.*
Document β : *William David "Bill" Forsyth (born 29 July 1946) is a Scottish film director and writer known for his films "Gregory's Girl" (1981), "Local Hero" (1983), and "Comfort and Joy" (1984).*
Document γ : $\cdots\cdots$
Answer: ***William David "Bill" Forsyth***

As the question shown in Table I, "Who has more well known films, Quincy Perkins or Bill Forsyth?". Two entities, Quincy Perkins and Bill Forsyth, can be focused easily. The golden answer is probably one of them. But it is still hard to determine which is the golden answer. Commonly, we determine the answer as follows: Firstly, figuring out the larger value or smaller value is needed by analyzing the question. We call the questions which are looking for larger values positive questions, otherwise negative questions. Then, we should know the numbers of well known films directed by Quincy Perkins or Bill Forsyth respectively. Actually, we don't need the exact numbers of well-known films directed by each of them. Due to all questions in HotpotCQA are comparing between two entities and a document describe only one entity, thus we only need to know if the corresponding value of entity α (document α contains) is greater than the corresponding value of entity β (document β contains). This is a document polarity problem. If document α contains the larger value, the document polarity is positive, otherwise negative document polarity. For the example in Table 1, the question polarity is positive because of the word "more" in the question, and the document polarity is negative because document β contains the larger value (3 well-known films in document β compared to 1 well-known film in document α). Question polarity and document polarity constitute a semantic-level XOR problem, and it is challenging for current MRC models.

In this paper, we propose a CQA model with question polarity module and document polarity module. The two modules can determine question polarity and document polarity respectively. At last, We get the answer by question polarity XOR document polarity. The way we determine the golden answer is shown in Fig.1. Besides, we jointly train the two modules. Our main contributions can be summarized as follows:

- We discovered a semantic-level XOR problem in the comparison question.
- We propose an objective function to retrieve related documents and a neural CQA model to answer comparison questions. We conduct experiments on HotpotCQA dataset and outperforms state-of-the-art methods.

II. Related Work

We review the related work about MRC in Section II-A. Then we introduce the characteristics of CQA in Section II-B.

978-1-7281-5014-7/19/$31.00 ©2019 IEEE

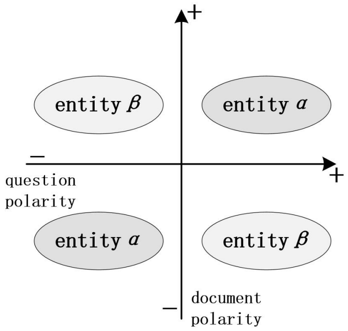

Figure 1. Getting the answer by question polarity and document polarity

A. End-to-end MRC model

Recently, MRC makes significant progress in span-based question task. Common approaches [3], [4], [5] use Recurrent Neural Networks (RNN) [6] to encode textual data, and use sophisticated interactions to capture the high-dimensional information, and use Pointer Network [7] to predict the start and end position of the answer. Pointer Network has been the standard output layer. However, different kinds of attention mechanism are widely used in MRC models to capture the high-dimensional information. Bidirectional attention mechanism [8] is used to capture the interaction among the document words conditioned on question word, self attention mechanism [9] is used to align the sentence representation against itself and multi-head attention [10] is used to capture semantic information hierarchically.

B. Why CQA

In recent years, several datasets and evaluation metrics on MRC have been built [11], [12], [1], [13]. On SQuAD 1.0, SQuAD 2.0 [1] and CoQA [14], neural network models can achieve either super-human or the ceiling performance, but it does not mean machine can do reading comprehension better than human. Machine still makes some simple mistakes which humans hardly make [15], it indicates that sometimes machine does not understand natural language which is one of the aims of MRC. So we should use multi-hop datasets to evaluate the comprehension ability of current neural network models. The comparison question is a kind of multi-hot question because we need to find knowledge about comparison objects respectively. While the knowledge is hidden in different documents, it means more than one document is needed to support the answer. Therefore study on CQA can make contributions to MRC and natural language understanding (NLU).

III. Model

In CQA, the model is given a set of documents and a related question. The model aims to answer the question supported by more than one documents. We call the supporting documents as "golden documents". There are two golden documents in HotpotCQA which describe two comparison objects respectively. In this paper, we mark two golden documents as document α and document β. Our model solves CQA in two steps: 1) Comparison object detection. 2) Answer prediction, comparison and selection. The first step is to retrieve golden documents from the given documents set by an objective function (Section III-A). Then, in the second step, we use a CQA neural network model to extract the candidate answers from two golden documents respectively. Then we calculate the question polarity and document polarity. Lastly, we select one of the candidate answers as the final answer.

The CQA model contains four modules: 1) Answer prediction (Section III-B) extract answers from two golden documents respectively. 2) Question polarity (Section III-C) determine the polarity of the related question, in order to decompose the semantic-level XOR problem. 3) Documents polarity (Section III-D) calculate the degree of document polarity and determines the polarity of documents. 4) Answer selection (Section III-E) considers the results of the previous two modules, and selects the final answer based on conditional probability. The overview of our approach is shown in Fig. 2.

A. Document Selection

In order to retrieve golden documents, we evaluate the documents by related questions, and we use the recall rate to rank the documents. We treat the documents' titles and questions as two bags-of-words, then compute the number of overlaps between titles and questions:

$$\begin{aligned} score_i &= f(question, title_{doc_i}) \\ &= RECALL/TWORDS \end{aligned} \tag{1}$$

where $question$ denotes the bags-of-word of given question, $title_{doc_i}$ denotes the bags-of-word of i-th document's title, *RECALL* denotes the number of words both in $question$ and $title_{doc_i}$, *TWORDS* denotes the number of $title_{doc_i}$.

According to the score of each document, we consider two documents with the highest score as golden documents and the two documents describe two comparison objects mentioned in the question. We mark the two golden documents as document α and document β.

B. Answer Prediction

1) Encoding Layer: We get word-level distributed word representations by mapping each word in question Q and golden documents $\{D_\alpha, D_\beta\}$ to a high-dimensional vector space. Then, we use Convolutional Neural Networks (CNN) [16] to get character-level distributed word representations. Lastly, we concatenate word-level embedding, character-level embedding and three-input EM (TEM) feature as the final embedding.

Let $D = \{x_1, x_2, ..., x_d\}$ and $Q = \{q_1, q_2, ..., q_q\}$ represent the words in the document and question. In order to get encoding representations of golden documents and questions, we use a bidirectional Gated Recurrent Unit (BiGRU) [6], [17] with h hidden size.

$$H_Q = BiGRU(Q) \tag{2}$$

$$H_{D_{\alpha,\beta}} = BiGRU(D_{\alpha,\beta}) \tag{3}$$

where $H_{D_\alpha}, H_{D_\beta} \in \mathbb{R}^{d*2h}$ and $H_Q \in \mathbb{R}^{q*2h}$.

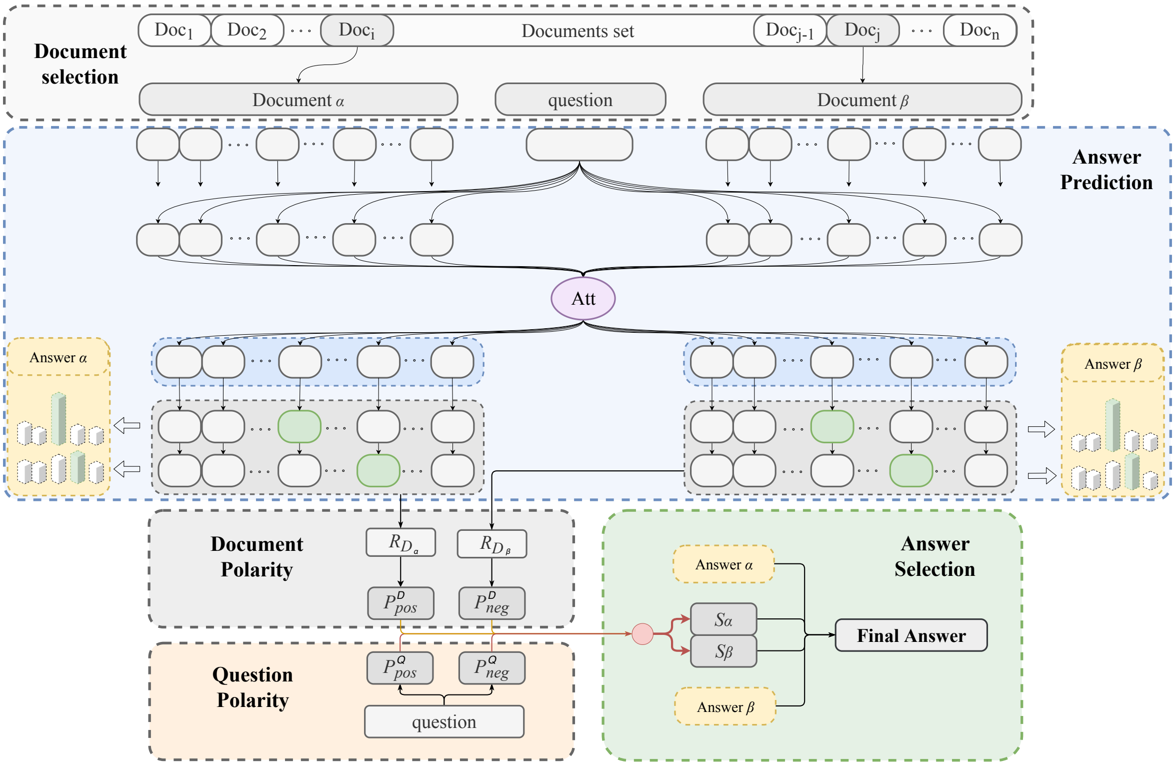

Figure 2. Overview of our approach

2) TEM Feature: EM feature [18] can significantly improve the performance of MRC models. Now we propose a new version EM feature for the three-input model. Different from the traditional EM feature, we use five numbers to mark different words instead of a simple binary number. Each number maps to a high-dimensional vector space. Fig. 3 shows the rule we mark each word and the difference between two versions.

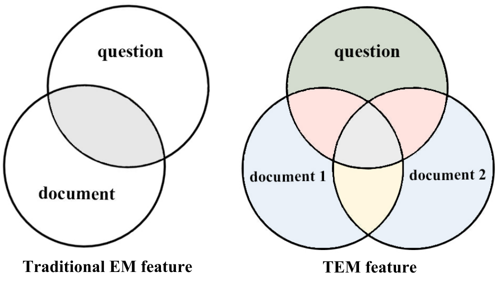

Figure 3. Traditional EM and TEM features. The three circles in TEM figure represent the three inputs of our CQA model. The intersection of the circles represents the same words in inputs. Different colors denote different numbers which be mapped to by words.

3) Attention Layer: We use three kinds of attention mechanisms: question-document, document-document and self attention. After we get the encoding representations H_{D_α}, H_{D_β} and H_Q, we use a bi-directional question-document attention mechanism [8], [2] to update the H_D, in order to capture semantic information between the question and the documents. Then we capture semantic information between two documents. Lastly, self attention [9] is used to align document presentation. We get the encoding representations of two documents G_α and G_β as follows:

$$S^1_{D_{\alpha,\beta}} = W_1 * Att(H_{D_{\alpha,\beta}}, H_Q) + b_1 \tag{4}$$

$$S^2_{D_{\alpha,\beta}} = W_2 * Att(S^1_{D_{\alpha,\beta}}, S^1_{D_{\beta,\alpha}}) + b_2 \tag{5}$$

$$G_{\alpha,\beta} = W_3 * Att(S^2_{D_{\alpha,\beta}}, S^2_{D_{\alpha,\beta}}) + b_3 \tag{6}$$

where $W_1, W_2, W_3 \in \mathbb{R}^{8h*h}$ are trainable weight vectors. $S^1_{D_{\alpha,\beta}} \in \mathbb{R}^{d*h}$ is the representation of golden documents which obtains the question information. $S^2_{D_{\alpha,\beta}} \in \mathbb{R}^{d*h}$ combines the information in D_α and D_β. $G_{\alpha,\beta} \in \mathbb{R}^{d*h}$ aligns the document representation.

4) Decoding Layer: In order to extract the answer span from the documents, we use BiGRU to get the decoding representation of start position $O^{start} \in \mathbb{R}^{d*2h}$. Different from start position, we concatenate the decoding representation of start position O^{start} and document representation G to get the decoding representation of end position $O^{end} \in \mathbb{R}^{d*3h}$. At last, we use Pointer Network [7] to get the probability of each word to be the start $p^1 \in \mathbb{R}^d$ and the end $p^2 \in \mathbb{R}^d$:

$$O^{start}_{\alpha,\beta} = BiGRU(G_{\alpha,\beta}) \tag{7}$$

$$O^{end}_{\alpha,\beta} = BiGRU([O^{start}_{\alpha,\beta} : G_{\alpha,\beta}]) \tag{8}$$

$$p^{1,2}_{\alpha,\beta} = softmax(max(O^{start,end}_{\alpha,\beta})) \tag{9}$$

where subscripts α and β are used to distinguish between two golden documents.

We get Ans_α from document α based on p^1_α and p^2_α. Similarly, we get Ans_β from document β based on p^1_β and p^2_β.

C. Question Polarity

In order to determine the question polarity, we import thirteen comparative words to label the questions in training set. The comparative words include 7 more-like words (more, higher, taller ect.) and 6 less-like words (less, lower ect.). If the question contains any one of more-like words, the polarity of the question is positive, otherwise negative polarity. We label the positive question "pos" and negative question "neg". We calculate the probabilities of positive polarity and negative polarity as follow:

$$P_{pos}^{Q} = sigmoid(W_Q * H_Q + b_Q) \tag{10}$$

$$P_{neg}^{Q} = 1 - P_{pos}^{Q} \tag{11}$$

where P_{pos}^{Q} represents the probability of positive question polarity, P_{neg}^{Q} represents the probability of negative question polarity.

D. Documents Polarity

In question polarity module, we get the question polarity. Thus, in documents polarity module we should only make the module able to give higher scores for documents with larger values. We concatenate the two representations, O^{start} and O^{end}, of two documents in decoding layer respectively. Then, we get two new documents representations $R_{D_\alpha}^{S}$ and $R_{D_\beta}^{S}$ of each document. Lastly, we use a linear to calculate the score of two documents, and use a soft-max function to get the probability of positive document polarity and negative document polarity.

$$R_D^S = [O^{start} : O^{end}] \tag{12}$$

$$score_\alpha = W_D * R_{D_\alpha}^{S} + b_D \tag{13}$$

$$score_\beta = W_D * R_{D_\beta}^{S} + b_D \tag{14}$$

$$P_{pos}^{D}, P_{neg}^{D} = softmax(score_\alpha, score_\beta) \tag{15}$$

where $P_{pos}^{D}, P_{pos}^{D} \in \mathbb{R}^1$, $score_\alpha$ and $score_\beta$ represent the degree of document polarity respectively. P_{pos}^{D} represent the probability of positive document polarity, P_{neg}^{D} represent the probability of negative document polarity.

E. Answer Selection

In this module, the result of question polarity and documents polarity will be used to calculate the scores of two candidate answers based on the probability theory. We mark the probability that answer is from the document α as P_α and the probability that answer is from the document β as P_β.

$$P_\alpha = P_{pos}^{Q} * P_{pos}^{D} + P_{neg}^{Q} * P_{neg}^{D} \tag{16}$$

$$P_\beta = P_{pos}^{Q} * P_{neg}^{D} + P_{neg}^{Q} * P_{pos}^{D} \tag{17}$$

where $P_{pos}^{Q} + P_{neg}^{Q} = 1$, $P_{pos}^{D} + P_{neg}^{D} = 1, P_\alpha + P_\beta = 1$.

Generally, we select Ans_α in two conditions: First, we get a positive question and a positive document, such as example 1 shown in Table II. Second, we get a negative question and a negative document, such as example 2 shown in Table II.

Table II
EXAMPLES FOR ANSWER SELECTION.

Example 1 **Question:** ***Which city has larger population, CITY α or CITY β?*** **Document α :** ***CITY α has a population of 2 million.*** **Document β :** ***CITY β has a population of 1 million.***
Example 2 **Question:** ***Which city has less population, CITY α or CITY β?*** **Document α :** ***CITY α has a population of 1 million.*** **Document β :** ***CITY β has a population of 2 million.***
Answer : ***CITY α***

F. Joint Training

We jointly train answer prediction module, question polarity module, document polarity module and answer selection module, the loss of $question_i$ ($loss_i$) contains the loss of answer prediction L_{ans}, the loss of question polarity L_{ques}, the loss of document polarity L_{doc} and the loss of answer selection L_{sele}. Specially, we multiply an indicator function $I(i)$ before L_{ans} as a switch, L_{ans} will be set to zero if answer selection module doesn't make a correct judgment. Besides, to avoid getting a smaller loss when the wrong answer is selected, we multiply a hyperparameter multiplier λ before L_{sele}:

$$loss_i = I(i)L_{ans} + L_{ques} + L_{doc} + \lambda * L_{sele} \tag{18}$$

$$Among \quad I(i) = \begin{cases} 1, & predict = label \\ 0, & otherwise \end{cases} \tag{19}$$

where L_{ans}, L_{ques}, L_{doc} and L_{sele} are cross entropy loss function.

IV. EXPERIMENTS

We evaluate the model on HotpotCQA which is a part of HotpotQA [2]. HotpotCQA contains 18,943 question-answer pairs on 189,430 documents. The dataset not only provides the answer span but also provides sentence-level supporting facts required for reasoning. The solution proposed in this paper can provide document-level supporting facts instead of sentence-level.

A. Implementation Details

We develop our CQA model using Pytorch deep learning framework. We use Glove 300-dimensional word embeddings [19] and 8-dimensional char embeddings. Our encoder hidden size is set to 50 for word-level embedding, and 100 for character-level embedding. We use Adam [20] as our optimizer, and set learning rate to 0.0007. Multiplier λ which is used to keep balance between document selection and answer prediction is set to 1.0.

B. Experimental Results

Following SQuAD [1] and HotpotQA [2], we use EM and F1 scores to evaluate our model. We resplit the dataset evenly and evaluate our model by four-fold cross-validation because the test set of this part is not released. Each fold contains about 16,594 training data, 1,181 dev data, and 1,181 test data. We use the golden documents in training set to test our document selection module, the result shows that over 90% scores of golden

Table III
FOUR-FOLD CROSS-VALIDATION RESULTS ON HOTPOTCQA.

Document Selection										
Model	**1st**		**2nd**		**3rd**		**4th**		**Avg**	
	ACC	**F1**	**ACC**	**F1**	**ACC**	**F1**	**ACC**	**F1**	**ACC**	**F1**
ours	88.63	94.14	88.80	94.23	88.70	94.01	88.91	94.28	88.76	94.17
Final Results										
Model	**1st**		**2nd**		**3rd**		**4th**		**Avg**	
	EM	**F1**	**EM**	**F1**	**EM**	**F1**	**EM**	**F1**	**EM**	**F1**
baseline-cp	47.20	53.68	44.65	51.74	47.35	53.79	42.25	48.95	45.36	52.04
baseline-total	48.66	55.38	43.63	51.01	46.49	52.66	43.94	50.02	45.68	52.27
ours	**56.24**	**63.16**	**54.84**	**62.56**	**57.36**	**64.76**	**55.80**	**61.87**	**56.06**	**63.09**

Table IV
EXPERIMENTAL RESULTS ON THE DEV SET OF HOTPOTCQA.

setting	EM	F1
ours	**58.43**	**66.22**
DecompRC	-	62.78
BERT	-	57.81
baseline	48.55	55.05

documents are ranged from 0.8 to 1.0. The results show that the objective function we proposed can retrieve golden documents effectively. The score distribution is shown in Fig.4.

We train the baseline model in two ways, using total HotpotQA datasets and using HotpotCQA only. Four-fold cross-validation results show that our approach reaches an average performance of 63.09% F1 and 56.06% EM, including over 10% F1 score gains. The four results in document selection are similar, it shows the proposed objective function is reliable. The result of four-fold cross-validation is shown in Table III.

Table IV compares the results of our CQA model with other proposed models on the HotpotCQA development set. The results show that our CQA model outperforms BERT [3] 8.41% F1 score and outperforms DecompRC [21] 3.44% F1 score.

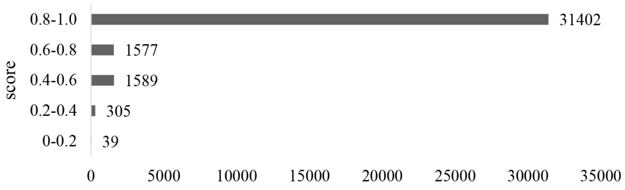

Figure 4. Score distribution of the golden documents

C. Interpretability Experiments

In order to prove the interpretability of our model, we test the model by manual data. We modify only one of the words in the question to structure an opposite question. For example, original question *"Who has more well known films, Quincy Perkins or Bill Forsyth?"* becomes *"Who has less well known films, Quincy Perkins or Bill Forsyth?"*. We use our model to extract answers from the same documents set relating these two opposite questions. And the result shows in Table V.

For further research, the intermediate parameters of each module are shown in Table VI. Two opposite samples got similar document polarity score, but got different question polarity score. According to the Eq. 16 and Eq. 17 given in section III-E, we get $P_\alpha = 0.16\%$ and $P_\beta = 0.84\%$ in original sample, but $P_\alpha = 0.81\%$ and $P_\beta = 0.19\%$ in opposite sample. Our CQA model answers both two opposite questions correctly, and the result shows that question polarity module distinguishes the difference between two questions and judges the question polarity correctly.

Table V
RESULT OF INTERPRET ABILITY EXPERIMENT.

Document α : *Quincy Perkins (born July 16, 1980 in Key West, Florida) is an American director most famous for directing, producing and writing the narrative fiction short film "Swingers Anonymous" which debuted at the [Cannes Film Festival] in 2015.*
Document β : *William David "Bill" Forsyth (born 29 July 1946) is a Scottish film director and writer known for his films "Gregory's Girl" (1981), "Local Hero" (1983), and "Comfort and Joy" (1984).*
Original Question: *Who has more well known films, Quincy Perkins or Bill Forsyth?* **Answer:** *William David "Bill" Forsyth*
Opposite Question: *Who has less well known films, Quincy Perkins or Bill Forsyth?* **Answer:** *Quincy Perkins*

Table VI
THE INTERMEDIATE PARAMETER OF EACH MODULE.

parameters	original	opposite
P^Q_{pos}	99.41%	1.29%
P^Q_{neg}	0.59%	98.71%
P^D_{pos}	15.46%	18.32%
P^D_{neg}	84.54%	81.68%

D. Ablation Study

To show the effects of our model, we conduct an ablation study on the development set of HotpotCQA. The results of ablation study are shown in Table VII. We find that self-attention contributes little to the performance. One possible reason is that the length of the textual data is short enough after document selection. Self-attention makes more contributions when the textual data is long enough. In contrast, the TEM feature contributes remarkably to the performance. To make a comparison, we add EM feature to baseline model, and the result shows that TEM feature has obvious advantages over EM feature. Because of the characteristic of CQA, TEM can highlight the relation between questions and documents, and the

Table VII
RESULT OF ABLATION STUDY.

setting	EM	F1
ours	**58.43**	**66.22**
$baseline + EM$	49.42	56.04
$baseline$	48.55	55.05
$-TEMfeature$	53.19	60.19
$-selfAttention$	58.17	65.19
$-DocAttention$	58.10	64.82
$-L_{ques}$	54.80	61.52

relation between each document. Besides, we set L_{ques} to zero to simulate the situation without distinguishing question polarity. And the result shows that question polarity module provides about 4.7% F1 scores.

V. Conclusion

In this paper, we discovered a semantic-level XOR in the comparison question and propose a method to decompose this XOR problem. We also propose an objective function to retrieve documents and a CQA model to answer comparison questions with considering question polarity and document polarity. Our approach achieve the state-of-the-art performance on HotpotCQA. Moreover, our method of decomposition semantic-level XOR problem to improve the answer location ability is effective.

VI. Acknowledgments

This research work is supported by National Natural Science Foundation of China (Grants No.61672368, No.61703293, No.2017YFB1002104). The authors would like to thank the anonymous reviewers for their insightful comments and suggestions. Yu Hong, Professor in Soochow University, is the corresponding author of the paper, whose email address is tianxianer@gmail.com.

References

[1] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," *arXiv preprint arXiv:1606.05250*, 2016.

[2] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, and C. D. Manning, "Hotpotqa: A dataset for diverse, explainable multi-hop question answering," *arXiv preprint arXiv:1809.09600*, 2018.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[4] M. Hu, Y. Peng, Z. Huang, X. Qiu, F. Wei, and M. Zhou, "Reinforced mnemonic reader for machine reading comprehension," *arXiv preprint arXiv:1705.02798*, 2017.

[5] H. Y. Huang, E. Choi, and W. T. Yih, "Flowqa: Grasping flow in history for conversational machine comprehension," 2018.

[6] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Eleventh annual conference of the international speech communication association*, 2010.

[7] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 2692–2700.

[8] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, "Bidirectional attention flow for machine comprehension," *arXiv preprint arXiv:1611.01603*, 2016.

[9] S. Wang and J. Jiang, "Machine comprehension using match-lstm and answer pointer," *arXiv preprint arXiv:1608.07905*, 2016.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[11] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," in *Advances in neural information processing systems*, 2015, pp. 1693–1701.

[12] F. Hill, A. Bordes, S. Chopra, and J. Weston, "The goldilocks principle: Reading children's books with explicit memory representations," *Computer Science*, 2015.

[13] M. Richardson, C. J. Burges, and E. Renshaw, "Mctest: A challenge dataset for the open-domain machine comprehension of text," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 193–203.

[14] S. Reddy, D. Chen, and C. D. Manning, "Coqa: A conversational question answering challenge," *arXiv preprint arXiv:1808.07042*, 2018.

[15] D. Chen, "Neural reading comprehension and beyond," https://cs.stanford.edu/~danqi/papers/thesis.pdf.

[16] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[17] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[18] D. Chen, A. Fisch, J. Weston, and A. Bordes, "Reading wikipedia to answer open-domain questions," *arXiv preprint arXiv:1704.00051*, 2017.

[19] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[21] S. Min, V. Zhong, L. Zettlemoyer, and H. Hajishirzi, "Multi-hop reading comprehension through question decomposition and rescoring," *arXiv preprint arXiv:1906.02916*, 2019.