ECNU at SemEval-2018 Task 11: Using Deep Learning Method to Address Machine Comprehension Task

Yixuan Sheng[1], **Man Lan**[1,2*], **Yuanbin Wu**[1,2]
[1]Department of Computer Science and Technology,
East China Normal University, Shanghai, P.R.China
[2]Shanghai Key Laboratory of Multidimensional Information Processing
51164500026@stu.ecnu.edu.cn, {mlan, ybwu}@cs.ecnu.edu.cn

Abstract

This paper describes the system we submitted to the Task 11 in SemEval 2018, i.e., Machine Comprehension using Commonsense Knowledge. Given a passage and some questions that each have two candidate answers, this task requires the participate system to select out one answer meet the meaning of original text or commonsense knowledge from the candidate answers. For this task, we use a deep learning method to obtain final predict answer by calculating relevance of choices representations and question-aware document representation.

1 Introduction

In recent years, the presentation of challenge and large-scale reading comprehension corpora has driven the development of technology for machine reading comprehension, and most of these machine comprehension datasets do not need commonsense knowledge to answer questions. The purpose of Machine Comprehension using Commonsense Knowledge task in Semeval 2018 is to provide a platform for finding a way for the machine to better understand the text and enable the machine answer questions based on the text, and encourage participants to make use any external resources (e.g., DeScript, narrative chains, Wikipedia, etc) to improve the system performance (Ostermann et al., 2018b). The task 11 is a multiple-choice machine comprehension, which requires a system read a narrative text about everyday activities (Ostermann et al., 2018a) and then answer multiple-choice questions based on this text. Some questions need to be answered according to the original text, and others can be answered by commonsense knowledge. Each question is associated with a set of two answers. Table 1 gives an example of the dataset.

To address this machine comprehension task, we utilized rule-based methods and a deep learning method. Our final submission use Gated-Attention Reader (Dhingra et al., 2016) to fuse question information into document and acquire a question-awared document representation, the degree of interaction between choices and document are regard as the probabilities of choices being returned as an answer. The above two methods do not use additional commonsense knowledge, which may lead to the poor preformance of our system. In future work, we may explore more methods to integrate common knowledge into models.

document: Early this morning I woke up to the sound of my newspaper landing on my driveway. I sat up and wrapped my pink robe around me. I slipped my feet into my slippers and looked at the clock. It was only 7:00 but it was time for me to get my newspaper and drink some coffee. I looked out the window and noticed it was raining quite a bit. I saw the newspaper at the end of my driveway, as far away as it could be. I grabbed my umbrella out of my coat closet and opened my front door enough to stick the umbrella through and open it outside. I stepped out the door and quickly covered my head with the umbrella. Then I ran to the end of my driveway, scooped up the newspaper in its plastic wrapping, and ran back to my front door. I closed my umbrella, took off my slippers, and dried off. Then, I unwrapped my newspaper and sat down to read it. **question**: Do they read the paper daily? **candidate answers**: 0. No they usually watch TV in the mornings. 1. Yes **answer**: 1

Table 1: An example from the SemEval2018 task11 dataset.

The rest of this paper is organized as follows. Section 2 describes our systems. Section 3 describes datasets, experimental setting and analyse results on datasets. Finally, Section 4 concludes this work.

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 1048–1052
New Orleans, Louisiana, June 5–6, 2018.

2 System Description

2.1 Task Description

Formally, this multiple-choice machine comprehension task can be expressed as a quadruple:<**D**, **Q**, **A**, **a**>. Where **D** represents a narrative text about everyday activities, **Q** represents a question for the content of the narrative text, **A** is the candidate answer choice set to the question(this task contains two candidate answers choice a_0 and a_1) and **a** represents the correct answer. The system is expected to select an answer from **A** that best answers **Q** according to the evidences in document D or commonsense knowledge.

2.2 Two Rule-based Baselines

First of all, we implemented a rule-based system proposed in (Richardson et al., 2013), which used the sliding-window (SW) and word distance-based (WD) algorithms to calculate the answer scores according to the rules and return the highest-score answer. We also tried the improved SW and WD algorithms proposed in (Smith et al., 2015), and the system performance has improvement. Sliding-window and Word Distance-based algorithms are are described as follows:

Sliding-Window: Given a data sample <**D**, **Q**, a_0(or a_1), **a**>, firstly, we calculate the inverse word counts of each word in the document **D**. Then we set a window that slides word by word from the beginning of the document to the end. When the window slides to a position, the sum of inverse word counts of all the words that appears in the question **Q** or the candidate choice a_0 (or a_1) is the score of the window at this moment. Until the window slides to the end of the passage, we choose the highest window score as the final score of the candidate choice a_0 (or a_1). Window size is size of union of the question **Q** and the choice a_0 (or a_1), and the window slides over full passage only once. In the improved SW algorithm, the window size is 2-30, and window passes full passage window several times, and increasing the size of the window by one after each sliding over the full passage. The summing up values of all passes is served as improved sliding window score.

Word Distance-based: Given a data sample <**D**, **Q**, a_0(or a_1), **a**>, firstly, we define two collections, set_{dq} and set_{dc}, set_{dq} represents the intersection of the question words and the document words, and set_{dc} represents the intersection of the words in the choice and the words of the document. If neither set_{dq} nor set_{dc} is empty set, we calculate the shortest distance between words of set_{dp} and words of set_{dc} in the document, denote the shortest distance as d_{min}, and the word distance score of the choice is $\frac{d_{min}+1}{|D|-1}$, otherwise, the word distance score of the choice is zero.

The sliding-window score minus the word distance-based score is the final score of the choice. We separately calculated the scores of the two choices for the question and then selected the choice with higher score as the answer to the question.

2.3 Deep Learning model

Both of the above unsupervised methods score the overlap that between each answer and the document by making a sliding-window passes over the document. Therefore, we roughly count the proportion of words in correct answers appear in the document[1], and we find that the proportion of correct answers whose words appear entirely in the article is not high in all correct answers. The proportion show that there is a limit to using the above method to improve system performance. Hence we used a deep learning approach to passage representations modeling. Inspired by (Lai et al., 2017), we use the state-of-art Gated-Attention Reader which performs well on several datasets. When a sample data <**D**, **Q**, **A**, **a**> is given, the steps of the model processing this data sample are described below, Figure 1 shows the system.

2.3.1 Passage, Question and Choice Encoder

First, each word in **D**, **Q**, and choices (two choices in candidate answer set **A**) is mapped to d-dimendional vector. The 300-dim GloVe embedding (Pennington et al., 2014) is used. For the input word vectors of **D**, we also include a 5-dim binary feature to indicates the overlap between the ducument and the question(or choices) which inspired by (Chen et al., 2017). Each dimension of the 5-dim binary match feature represent whether the word present in the query, in the choice a_0, in the choice a_1, in both question and

[1]We use the following equations to estimate how many answers appear entirely in the document: if $|answer\ word \cap document\ word|/|answer\ word| = 1$, it means the answer appears entirely in the document, where $|A|$ means size of set A. Then we calculate $|ans_{ce}|/|ans_c|$, where ans_{ce} means correct answers which entirely appeared in document, and ans_c means correct answers. The percentage of the correct answers entirely appeared in document is about 24%.

choice a_0 , in both question and choice a_1, respectively. Take passage as an example, we have document $\mathbf{D}$: $x_1^{\mathbf{D}}, x_2^{\mathbf{D}}, ..., x_m^{\mathbf{D}} \in R^{|D|*dim}$, and next we use bi-directional GRU to encode each document word embedding $x_i^{\mathbf{D}}$,

$$\overrightarrow{h_i^{\mathbf{D}}} = biGRU(\overrightarrow{h_{i-1}^{\mathbf{D}}}, x_1^{\mathbf{D}}), i = 1, 2, ..., m$$

$$\overleftarrow{h_i^{\mathbf{D}}} = biGRU(\overleftarrow{h_{i+1}^{\mathbf{D}}}, x_1^{\mathbf{D}}), i = m, m-1, ..., 1$$

we define $\boldsymbol{h}_i^{\mathbf{D}} \in R^{2d}$ is concatenation of the $\overrightarrow{h_i^{\mathbf{D}}}$ and $\overleftarrow{h_i^{\mathbf{D}}}$, where d is hidden size. At this time, we get the encoded document representation $\mathbf{D}^e = \{ \boldsymbol{h}_1^{\mathbf{D}}, \boldsymbol{h}_2^{\mathbf{D}}, ... , \boldsymbol{h}_m^{\mathbf{D}} \}$. Meanwhile, we use separate bi-directional GRU to form representation for question, we denote these representations as $\mathbf{Q}^e = \{ \boldsymbol{h}_1^{\mathbf{Q}}, \boldsymbol{h}_2^{\mathbf{Q}}, ... , \boldsymbol{h}_n^{\mathbf{Q}} \}$. As for choices, we concat $\overrightarrow{h_n^{\mathbf{C}}}$ and $\overleftarrow{h_1^{\mathbf{C}}}$ to make up a vector represent a choice, so we get $\mathbf{C}_0 \in R^{2d}$ and $\mathbf{C}_1 \in R^{2d}$.

2.3.2 Summarize Question-aware Passage Representation

The interaction layer of Gated-Attention Reader is a *l*-layers multi-hop architecture with gated-attention units. Each multi-hop layer contain a bi-GRU and a gated-attention unit. As shown in Figure 1, we sent $\mathbf{Q}^e \in R^{|Q|*2d}$ and $\mathbf{D}^e \in R^{|D|*2d}$ into a gated-attention unit. Gated-attention unit fuses information from question to each document tokens and generates a set of vectors $\mathbf{D}_l^{GA} = \{ \mathbf{d}_1^{(l)}, \mathbf{d}_2^{(l)}, ..., \mathbf{d}_m^{(l)} \}$, where superscript (l) denote *l*-th multi-hop layer. To generates $\mathbf{D}_l^{GA}$, firstly, the question soft attention to each document word to obtain attention weight α_i, and then we use α_i to calculate a weighted question representation $\overline{q}_i$ for *i*-th word in $\mathbf{D}$, finally, the weighted question $\overline{q}_i$ representation is element-wise multiplied by $\boldsymbol{h}_i$ makes $\mathbf{d}_i$. The specific calculation steps of a gated-attention unit are as follows.

$$\alpha_i = softmax(\mathbf{Q}^e \mathbf{h}_i) \quad (1)$$

$$\overline{q}_i = \mathbf{Q}^{eT} \alpha_i \quad (2)$$

$$\mathbf{d}_i = \mathbf{h}_i \cdot \overline{q}_i \quad (3)$$

After obtaining the current layer question-aware document representation, we put this representation into next hop layer, until after *l* layers multi-hops, we generate the a set of question-aware vectors $\mathbf{D}_l^{GA}$ for document. Finally, we sent $\mathbf{D}_l^{GA}$ into a layer biGRU and concat the last outputs of each direction ($\overrightarrow{h_{l+1}^{GA}}$ and $\overleftarrow{h_{l+1}^{GA}}$) to get a ultimate question-aware document representation vector $\widetilde{\mathbf{D}} \in R^{2d}$

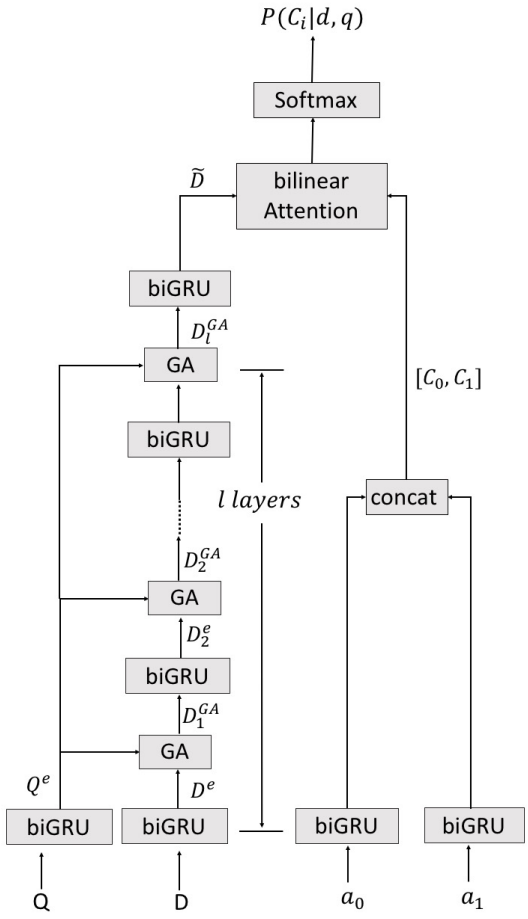

Figure 1: Architecture of our system.

2.3.3 Answer Selection

Now, we have a question-aware representation $\widetilde{\mathbf{D}}$, two choice representations $\mathbf{C}_0$ and $\mathbf{C}_1$. We estimate the probability that the choice selected as the correct answer by equation (4), and the choice with a higher-probability is returned as the predict answer.

$$[p_0, p_1] = softmax([\mathbf{C}_0, \mathbf{C}_1] W \widetilde{\mathbf{D}}) \quad (4)$$

3 Experiments

3.1 Datasets and Evaluation Metric

Table 2 shows the statistics of articles and questions in training, development, test data sets of this task. Here "#text" and "#commonsense" represent the question types, which are unknown during test and officially provided by organizors after test. Therefore, we do not use the class information of questions for system construction. Clearly, around 70% questions are from text and 30% are from commonsense. Without the aid of additional commonsense knowledge base, these questions from commonsense makes this task a huge challenge.

Dataset	Articles	Questions		
		#text	#commonsense	#total
train	1,470	7,032	2,699	9,731
dev	219	1,006	405	1,411
test	429	2,074	723	2,797

Table 2: The statistics of data sets in training, development and test data.

To evaluate the system performance, the official evaluation criterion is *accuracy*.

3.2 Preprocessing and Experimental Setting

For rule-based baselines, we first converted words into their lowercase and then performed tokenization and stemming using Stanford CoreNLP[2]. For deep learning system, we use 300-D pretrained word vectors provided by GloVe[3] as initial word embedding, which are fine-tuned during training. The encoding layer use one layer biGRU with 128-dims hidden size to encoder texts. Learning rate is 0.3, droprate is 0.5, epoch is 100, and num of multi-hops is 2. We use cross entropy and vanilla stochastic gradient descent (SGD) to train our models.

3.3 Experiments on Training Data

Table 3 shows the results of Task 11 with different methods on dev dataset, where "GA(biGRU)" denotes the final system we submit, "GA(biLSTM)" represents the experiment that we replace all biGRU units in the system with biLSTM units, "GA $-f_{match}$" represents the system without 5-dim match feature, "#text" and "#commonsens" represent the accuracy under different question types, respectively.

Methods	Accuracy		
	#text	#commonsense	#total
SW + WD	62.62%	45.92%	57.83%
improved SW+WD	65.01%	47.65%	60.02%
GA(biGRU)	77.33%	78.51%	77.63%
GA(biLSTM)	76.41%	77.53%	76.76%
GA $-f_{match}$	76.34%	78.02%	76.82%

Table 3: The results on dev.

Based on above experimental results, we find that the performance of GA system is much better than rule-based approaches, this is because the multi-hop structure merges the information of the question and the document repeatedly which is helpful to select final answer, unlike the rule-based approach that considers only word matching within a window-size distance. Furthermore, we find that the improved SW + WD algorithm is better than SW + WD algorithm, because the improved SW + WD algorithm considers the degree of word matching at different distances. From the GA system results, we find the performance of using biGRU units is better than that of biLSTM units and matching features also improves the system performance. Compare the accuracy of different types of questions under different methods, we find that the rule-based approaches considers only the word-matched features lead to lower accuracy on the commonsense type questions. GA systems perform better than rule-based systems on both types of questions, because the GA system takes into account the semantic similarity of the question-aware document and choices. Further, there are some commonsense types questions which the document content does not clearly indicate the correct answer but clearly does not meet the meaning of wrong answer. This may be the reason why we did not use external resources but the accuracy of the commonsense type question predicted by GA system is improved.

3.4 Results on Test Data

Table 4 shows the our result and official results of top-ranked teams on SemEval 2018 Task 11 test set.

Teamname	**Rank**	**Accuracy**(total)
ECNU	10	0.7311
iFLYTEK & HIT (HFL)	1	0.8413
Yuanfudao	2	0.8395
MITRE	3	0.8227

Table 4: Our result and the top three results on test sets.

The final result we submitted is generated by GA system used biGRU units, the specific configuration of which is mentioned in Section 3.2. Compared with the top ranked systems, there is much room for improvement in our work. In addition, the use of external knowledge resources by the system also have an impact on system performance because there are about 26% commonsense type questions in the dataset. This is where our

[2]https://stanfordnlp.github.io/CoreNLP/

[3]http://nlp.stanford.edu/data/wordvecs/glove.6B.zip

system lacks.

4 Conclusion

In this paper, we implement rule-based and deep learning approaches to address Machine Comprehension Using Commonsense Knowledge task in SemEval 2018. We explored two rule-based algorithm i.e., sliding window and word distance-based algorithm. We also utilized a deep learning method which use a multi-hop architecture (Gated-attention Reader). The above two methods do not use additional commonsense knowledge, this is a point that we need to improve.

Acknowledgements

This work is is supported by the Science and Technology Commission of Shanghai Municipality Grant (No.15ZR1410700) and the open project of Shanghai Key Laboratory of Trustworthy Computing (No.07dz22304201604).

References

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*.

Bhuwan Dhingra, Hanxiao Liu, Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. 2016. Gated-attention readers for text comprehension.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. In *EMNLP*.

Simon Ostermann, Ashutosh Modi, Michael Roth, Stefan Thater, and Manfred Pinkal. 2018a. MCScript: A Novel Dataset for Assessing Machine Comprehension Using Script Knowledge. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan.

Simon Ostermann, Michael Roth, Ashutosh Modi, Stefan Thater, and Manfred Pinkal. 2018b. SemEval-2018 Task 11: Machine Comprehension using Commonsense Knowledge. In *Proceedings of International Workshop on Semantic Evaluation (SemEval-2018)*, New Orleans, LA, USA.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew Richardson, Christopher J. C. Burges, and Erin Renshaw. 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *EMNLP*, pages 193–203. ACL.

Ellery Smith, Nicola Greco, Matko Bosnjak, and Andreas Vlachos. 2015. A strong lexical matching method for the machine comprehension test. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1693–1698, Lisbon, Portugal. Association for Computational Linguistics.