Capsule Networks for Chinese Opinion Questions Machine Reading Comprehension

Longxiang Ding[1], Zhoujun Li[1,*] , Boyang Wang[1], Yueying He[2]

[1] State Key Laboratory of Software Development Environment, Beihang University, China
[2] National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing, China
{antdlx,lizj,wangboyang}@buaa.edu.cn, hyy@cert.edu.cn

Abstract. In recent years, machine reading comprehension is becoming a more and more popular research topic. Promising results were obtained when the machine reading comprehension task had only two inputs, context and query. In this paper, we propose a capsule networks based model for Chinese opinion machine reading comprehension task which has three inputs: context, query and alternatives. First, we use a bi-directional LSTM to encode the three inputs. Second, model the complex interactions between context and query with a multiway attention layer. In addition to the attention mechanism used in BiDAF, the other two attention functions are designed to match the relationship between inputs. Finally, we present a capsule networks layer to route the right alternative. Specifically, we use two strategies to improve the dynamic routing process to filter noisy capsules, which may contain useless information such as stop words. Our single model achieves competitive results compared to the baseline methods on a Chinese dataset and obtains a significant improvement of 2.45% accuracy.

Keywords: Capsule Networks, Machine Reading Comprehension, Multiway Attention.

1 Introduction

The tasks of machine reading comprehension (MRC) and automated question answering (QA) have gained great popularity. In this paper, we mainly focus on opinion questions MRC task which needs to use the information of multiple sentences in the whole article for comprehensive analysis to get the correct answer. The differences between them can be listed as follows. First, the standard MRC is a QA task, while the opinion questions MRC is essentially an answer selection task. Second, standard MRC generally only has two inputs, context and query. Whereas opinion questions MRC has alternative inputs additionally. To better explain the difference, we present two examples in Table 1. There are two kinds of opinion questions, first kind of these

* Corresponding Author.

are the alternatives which are appeared in context like example-1, the other alternatives are “YES” and “NO” which need models do some reasoning like example-2.

Over the past few years, lots of end-to-end models show promising results on a variety of tasks. One of the latest advancements is BERT[1]. However, Chinese pre-trained BERT only supports 512 characters as inputs. For many Chinese MRC tasks, the input length limitation is so short and the cost is too huge to re-train it. It is thus necessary to design a light-weight and powerful model. In addition, other successful models generally have three key components. First, using an RNN or CNN based embedding layer to process sequential inputs. Second, an attention layer to model the interactions between inputs. Third, a neural based encoder to match specific tasks, such as BiDAF[2]v r-net[3] and Qanet[4]. These standard models basically support only two inputs but opinion questions MRC task has three inputs: context, query and alternatives. Naturally, attention mechanism plays an important role in models, it is meaningful to design a new attention mechanism which could extract interactions as many as possible. Meanwhile, with the growth of complex information extracted, the noise is growing and the encoder of primitive works can’t separate them clearly.

Table 1. Examples of MRC and opinion questions MRC

Task	Data
MRC	Context: Super Bowl 50 was an American football game to determine the champion of ……
	Query: Which NFL team represented the AFC at Super Bowl 50?
	Outputs: start span, end span
Opinion Questions MRC-example1	Context: 科学研究表明，在各种水果中……
	Query: 吃什么水果降血脂？
	Alternatives: 苹果， 葡萄，无法确定
	Outputs: 苹果
Opinion Questions MRC-example2	Context: 有研究表明，长期熬夜会导致……
	Query: 长期熬夜对身体是否有害？
	Alternatives:是， 否，无法确定
	Outputs: 是

In this paper, aiming to match the opinion questions MRC task, we use three Bi-LSTMs[5] to encode the inputs. Subsequently, we design a multiway attention layer including the attention mechanism used in BiDAF, bilinear attention function used in [6] and an element-wise dot product attention function used in [7]. Then we design a capsule networks structure introduced in [8] to filter noise. The main thought behind the design of these networks is the following: convolution network captures the feature of the interactions between query and context, while the dynamic routing can aggregate noise together. Inspired by [9], we add two strategies to eliminate the interference of noisy capsules. Eventually, we encode each alternative to a capsule and

pass them through a gate. By multiplying these capsules with attention mechanism outputs, our model will choose the right answer.

We conduct experiments on the latest Chinese opinion questions MRC dataset† with three competitive models (BiDAF, r-net, Qanet) as baselines. Our single model obtains an accuracy of 74.03% compared to the best baseline result of 71.58% (Qanet) on evaluation set. In summary, the contribution of this paper are as follows:

- We propose a strategy with capsule networks to filter noise in our model. When you concatenate lots of vectors which are come out from attentions or other algorithms, you may get more noise as well. It is meaningful to alleviate the disturbance of noise.
- We design a model based on capsule networks to deal with Chinese Opinions MRC task and get a good result.

2 Related Work

The MRC and QA have gained a significant improvement, an important contributor to the improvement is the high quality datasets, such as SQuAD[10], CNN/Daily News[11], Wikireading[12], Dureader[13] and the AI-Challenger2018 dataset[15] we used. A great number of end-to-end reading comprehension works achieve promising results, including BiDAF, r-net, Qanet, DCN[14], Document Reader[16] and Interactive AoA Reader[17].

Most of the successful models generally use attention mechanism. Basically, the interactions between two sentences can be modeled in sentence level and word level. Sentence level framework decides the relationship between sentences solely based on sentence vectors[20,21,22]. However, this kind of strategy ignores the lower interactions. Therefore, the word level framework proposes matching two sentence at word level. [23] used this strategy to improve LSTM-based network and achieved a better result. Our model uses two widely-used attention mechanisms[6,7] as part of attention layer. Specific to MRC model, attention mechanism can be separated in two distinct ways. The first way is dynamic attention[22]. Given the context, query vectors and previous attention, the attention weights will be updated dynamically. Hermann made some experiments on CNN/Daily News and the result showed that dynamic attention actually made sense. The other way is computing attention weights only one time[23], when given a similarity matrix between query and context[17]. Then we can use the attention in subsequent modeling layers. BiDAF uses a memory-less dynamic attention mechanism which means it is not directly depending on the attention at previous time steps and we use this as part of our attention layer.

For opinion questions MRC, after modeling the interactions between query and context, we can use the attention information and alternative vectors to select the right answer. This challenge can be considered as a text classification task. Primitive algo-

† This dataset was published by AI-Challenger2018 and available at: challenger.ai/competition/oqmrc2018

rithms used typical features such as n-grams, POS tags as inputs. Specifically, Support vector machine(SVM)[24], random forest[25] and xgboost[26] are common used on text classification. In recent years, neural networks, especially LSTMs[5] and CNNs[27], have largely improved the performance of text classification task. Kim migrated CNN from computer vision domain to natural language processing domain on sentence classification[27]. [28] proposed fastText model which could be trained on a huge dataset in a few minutes. However, CNN and RNN have lots of weakness. [29] firstly introduced the concept of capsule to cope with the representational limitations of CNNs and RNNs. Then [8] designed a capsule network on MNIST task. They replaced the CNN scalar outputs with capsule vectors and max-pooling with dynamic routing algorithm. At the same time, [8] designed two methods to filter noisy capsules. Recently, [9] investigated capsule networks with dynamic routing for text classification and the result showed its potential on classification task.

3 Our Model

In this section, we first describe the opinion questions MRC task and then introduce our model in detail.

3.1 Task Definition

The opinion questions MRC task which is investigated in this paper, is defined as follows. Given a context passage with n words $\boldsymbol{C} = \{c_1, c_2, \ldots, c_n\}$, a query with m words $\boldsymbol{Q} = \{q_1, q_2, \ldots, q_m\}$ and a group of k alternatives $\boldsymbol{A} = \{A_1, A_2, \ldots, A_k\}$, $\boldsymbol{A_i}$ is an answer segment with L words $\boldsymbol{A_i} = \{a_1, a_2, \ldots, a_L\}$, output a label y representing the best alternative to answer the query from the context. Specifically, $y \in \{1,2,\ldots,k\}$.

3.2 Model Overview

Our model can be separated to five components in high level structure (Figure 1): an embedding layer, a contextual embedding layer, an attention layer, a model encoder layer and an output layer, it's a common strategy for most MRC models. However, the major differences between ours and other models are as follows: we use multiway attention mechanism and aggerate attentions information for subsequent layers. As a result, our networks model the interactions between query and context better. Otherwise, we use capsule networks in the model encoder layer, this makes our model filters lots of noise and gets a 2.45% accuracy gain in our experiments. Compared to the other MRC tasks, we have three inputs, so we cope with the information from model encoder layer and alternative capsules in output layer.

Input Embedding Layer. Nowadays, obtaining the embedding of each word by concatenating its word embedding and character embedding is a popular strategy, we directly use this approach to get embeddings. For both word and character embed-

ding, we produce the *dim* = 300 dimension Word2vec[30] word vectors by training data. All the out-of-vocabulary words are mapped to <unk> token which is random initialized and trainable. Specifically, every word can be viewed as the concatenation of each characters. Most of Chinese words have two to four characters, so we truncate or pad each word to the length of four. Following [27], we use a convolutional neural networks and a max-pooling to model the character level embeddings of words, then we can get an embedding X_c. We finally pass the concatenation of X_c and word embedding to a Highway Network[31] like what BiDAF did. Now, we have matrix $\boldsymbol{C} \in \mathbb{R}^{d \times N}$ for the context, where d is the hidden size of highway network, matrix $\boldsymbol{Q} \in \mathbb{R}^{d \times M}$ for the query and K matrices for each alternative $\boldsymbol{A_k} \in \mathbb{R}^{d \times L}$.

Contextual Embedding Layer. We simply apply a bi-directional LSTM on the top of embeddings provided by previous layers. Here, we attain matrix $\boldsymbol{H} \in \mathbb{R}^{2d \times N}$ from $\boldsymbol{C}$, matrix $\boldsymbol{U} \in \mathbb{R}^{2d \times M}$ from $\boldsymbol{Q}$ and K matrices $P_k \in \mathbb{R}^{2d \times L}$ for each alternative. By concatenating both directions of LSTM outputs, the first dimension of matrices is *2d* now.

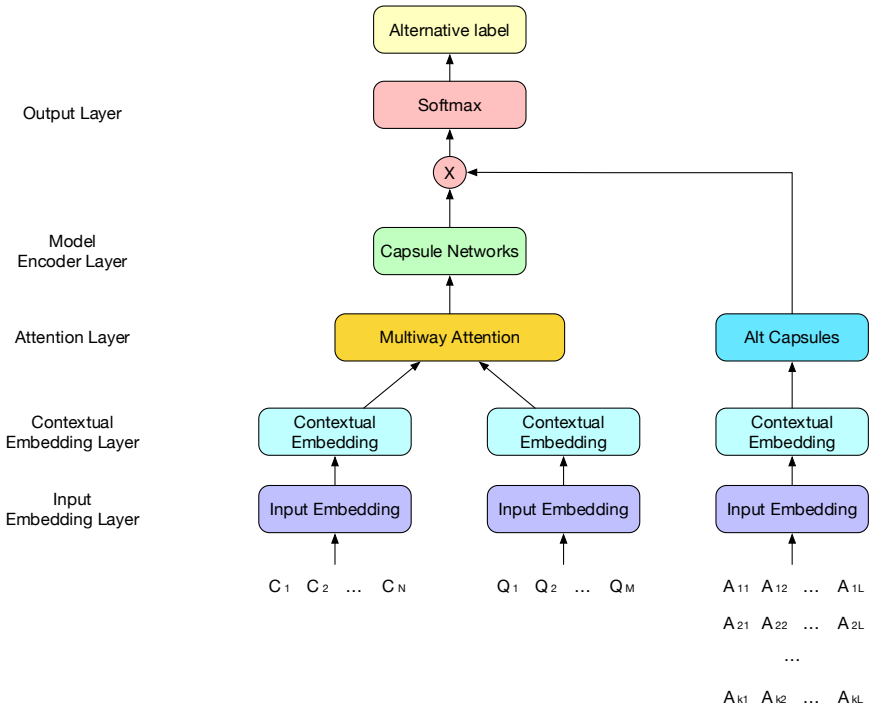

Figure. 1. Overview of our model

Attention Layer. This component is standard in almost every previous MRC models. BiDAF has made a significant improvement on attention mechanism, we directly use its structure as part of this layer. BiDAF attention uses $\boldsymbol{H}$ and $\boldsymbol{U}$ as inputs and

outputs $\boldsymbol{B} \in \mathbb{R}^{8d \times N}$ which represents the complex interactions between context and query. To extract more interactions, we also use another two attention mechanisms which will be introduced below. Therefore, we obtain a vector $\hat{G} \in \mathbb{R}^{12d \times N}$ by concatenating these vectors together. Finally, the layer uses a Bi-LSTM to fuse multiway attention interactions $\hat{G}$ and then outputs $G \in \mathbb{R}^{2d \times N}$.

We use h_t represents the *t*-th vector in $\boldsymbol{H}$, u_i represents the *i*-th vector in $\boldsymbol{U}$ and W represents the parameters. The major functions of bilinear and dot attention are listed as follows.

Bilinear Attention:

$$s_i^t = Relu(u_i^T W_b h_t)$$

$$a_j^t = Softmax(s_j^t)$$

$$b\_att_t = a^t u$$

Dot Attention:

$$s_i^t = Relu(W_d(u_i \odot h_t))$$

$$a_j^t = Softmax(s_j^t)$$

$$d\_att_t = a^t u$$

where s_i^t represents the similarity of u_i and h_t, then we attain a_j^t by adopting softmax used to scale s_i^t. Finally, our functions output the attention results b_att_t and d_att_t.

Model Encoder Layer. Inspired by [8], we design a capsule networks layer to encode the attention information. It consists of 3 elements: convolutional layer, primary capsule layer and dynamic routing. The structure can be illustrated in Figure 2.

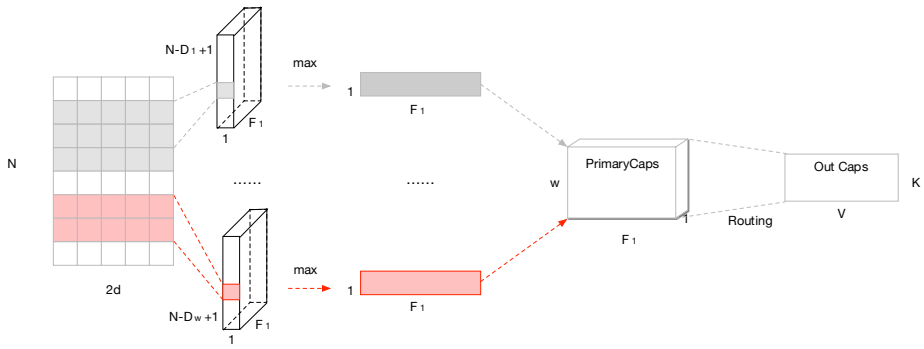

Figure. 2. Capsule networks structure

Convolutional Layer. This layer is just a standard convolutional layer aimed to extract n-gram features from $\boldsymbol{G}$. Suppose $W_b \in \mathrm{R}^{\mathrm{D}\times 2\mathrm{d}}$ denotes one filter where the n-gram size is D, we can produce a feature vector $m_i \in \mathrm{R}^{(\mathrm{N}-\mathrm{D}+1)\times 1}$ with stride of 1 for each filter. When we have F_1 filters and w kinds of filter sizes, the outputs are $\boldsymbol{Cov} \in \mathbb{R}^{(\mathrm{N}-\mathrm{D}+1)\times 1\times \mathrm{F}_1\times \mathrm{w}}$.

Primary Capsule Layer. This is a layer to produce our capsules called "primary capsule" from convolution outputs. Capsule vectors contain more information than scalars, such as position and semantic information of texts. We take the maximum of first dimension to represent outputs of one filter. For F_1 filters of each filter size, we can adopt $\mathrm{cov} \in \mathbb{R}^{1\times \mathrm{F}_1}$. Therefore, this layer outputs primary capsules $\boldsymbol{Pri} \in \mathbb{R}^{1\times \mathrm{w}\times \mathrm{F}_1}$ by concatenating w kinds of filter size covs.

Dynamic Routing. As argued in [8], dynamic routing is a more effective method than primitive routing strategies. It allows the networks to automatically learn relationships between the child capsule i to parent capsule j. First, generate prediction vectors (votes) by computing:

$$\hat{\mu}_{j|i} = W_{ij}^{y}\mu_i + \hat{b}_{j|i}$$

where $\hat{b}_{j|i}$ is bias item, and W_{ij}^{y} is the weights between two capsule layers. Then algorithm in an iterative manner would route each vote to an appropriate parent in the subsequent layer, meanwhile, noisy capsules can also be routed together. Initially, the votes routed to each parent and the networks can increase or decrease the weights by dynamic routing. Inspired by [9], we use two approaches to eliminate the interference of noisy capsules. First is the orphan category. To separate the noise such as stop words, we add an orphan category when computing softmax in dynamic routing and this would help our networks model the relationships between two capsule layers more efficiently. Second is using leaky-softmax which was implemented by Sara[32] to take the place of standard softmax. This function adds extra dimension to routing logits. When active capsule is not a good fit for any of the capsules in layer above, they will be routed to the extra dimension. The algorithm can be depicted in Figure 3.

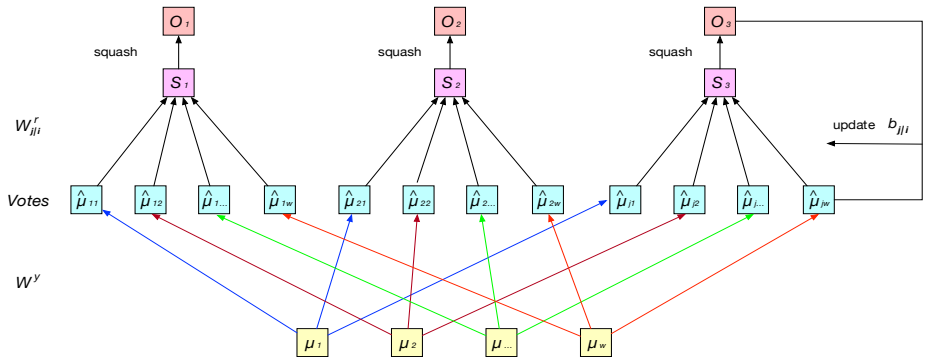

Figure. 3. The dynamic routing algorithm

Given each vote $\hat{\mu}_{j|i}$ and its existence probability $\left|O_j\right|$, the connection weights can be updated by:

Procedure 1: Dynamic Routing Algorithm

1: **procedure** Routing ($\hat{\mu}_{j|i}$, *r, l*)
2: for all capsule *i* in layer *l* and capsule *j* in layer (*l*+*1*), initialize $b_{j|i} = 0$
3: **for** *r* iterations **do**:
4: for all capsule *i* in layer *l*: $W_{j|i}^{r} = \left|O_j\right| \cdot leaky\_softmax(b_{j|i})$
5: for all capsule *j* in layer *l* +1: $O_j = squash(\sum W_{j|i}^{r} \hat{\mu}_{j|i})$
6: for all capsule *i* in layer *l* and capsule *j* in layer *l* +1: $b_{j|i} = b_{j|i} + \hat{\mu}_{j|i} O_j$
7: **return** O_j

$$W_{j|i}^{r} = \left|O_j\right| \cdot leaky\_softmax(b_{j|i})$$

Then the output capsules are computed by:

$$O_j = squash(\sum W_{j|i}^{r} \hat{\mu}_{j|i})$$

We view $\left|O_j\right|$ as the existence probability of *j*-th label. In detail, squash is a non-linear function to ensure that short vectors get shrunk to almost zero length and long vectors get shrunk to a length slightly below 1[8]. Next, we can produce the coefficients used to update W^r by:

$$b_{j|i} = b_{j|i} + \hat{\mu}_{j|i} O_j$$

Finally, a new round of iteration will be started by the steps mentioned above. And the dynamic routing algorithm is summarized in Procedure 1.

Output Layer. This layer is application-specific. Formally, in opinion questions MRC task, we concatenate alternative capsules together to get ***alt*** capsules and initialize a ***aux*** capsule which has the same shape with alternative capsules. Then, a gate is designed to compute the ***alt*** and ***aux*** by:

$$\boldsymbol{gated} = gate \cdot \boldsymbol{alt} + (1 - gate) \cdot \boldsymbol{aux}$$

Next, we use ***gated*** and ***O*** to obtain the similarity matrix:

$$\boldsymbol{Sim} = \boldsymbol{gate} \cdot \boldsymbol{O}$$

To get the probability distribution of the alternative label, we add a softmax layer on the top of ***Sim***. Finally, the objective function is:

$$L(W) = -\frac{1}{N} \sum_{N}^{i} log\ (softmax(\boldsymbol{Sim}_i))$$

4 Experiments

4.1 Dataset and Implementation Details

In order to evaluate our model, we conduct a series of experiments on AI-Challenger2018 MRC dataset. It contains 250K training data and 30K evaluation data. Each data has a context, a query and three alternatives.

In the experiments, we produce Word2vec vectors in 300 dimension by training data to initialize embedding vectors. The batch size is 64 and we use Adam optimization. Meanwhile, we adopt a cosine decay restart algorithm[33] to transform the learning rate. The context sequence length is limited to 300 words, the query sequence length is limited to 50 words and each alternative sequence length is limited to 2 words.

4.2 Baselines

We choose three strong MRC models as baselines: QANet[4], BiDAF[2] and r-net[3]. For each baseline, we use their own embedding layer to encode the alternatives, then we add a softmax behind the product of alternative embeddings and attentions. All the experiments use the same word2vec vectors and learning rate transform function.

QANet. We find an open source code[‡] from NLPLearn's github. The batch size is 64, learning rate is 0.001, dropout is 0.1 and epochs is 10.

BiDAF. DuReader had released an BiDAF baseline[§] on github and we just add alternatives embedding behind that. The batch size is 64, learning rate is 0.001, dropout is 0.05 and epochs is 8.

R-net. We still used NLPLearn's code[**] from github. The batch size is 64, learning rate is 0.001, dropout is 0.2 and epochs is 10.

In this paper, we prefer to focus on the performance of existing data and be limited by the inputs length, we do not choose pretrained model BERT[1] as a baseline.

4.3 Results and Discussion

Evaluation. In all of our experiments, the evaluation metric is accuracy. We summarize the results in Table 2. The *r-net*, *BiDAF* and *Qanet* represent the results of three baselines respectively. Moreover, we replace the attention layer with our multiway

---

[‡] This github url is: `https://github.com/NLPLearn/QANet`

[§] This release can be found at: `https://github.com/baidu/ DuReader/tree/master/tensorflow`

[**] This github url is: `https://github.com/NLPLearn/R-net`

attention mechanism on *BiDAF-relu*. The *Capsule-O* model only consists the standard structure without any dynamic routing optimization methods and multiway attention mechanism. Based on *Capsule-O*, *Capsule-relu* adds bilinear and dot attentions. In addition, *Capsule-leaky* uses leaky-softmax to take the place of the standard one in dynamic routing on the base of *Capsule-relu*. Compared to *Capsule-leaky*, *Capsule-orphan* only optimize dynamic routing by add an orphan category. Finally, we use *Capsule-final* to test all the methods together.

Table 2. Results of our models and baselines

Model	Accuracy
r-net	70.22%
BiDAF	71.15%
BiDAF-relu	70.94%
Qanet	71.58%
Capsule-O	73.10%
Capsule-relu	73.47%
Capsule-leaky	73.67%
Capsule-orphan	73.92%
Capsule-final	**74.03%**

Discussion. From the table, we have shown capsule networks without any optimization achieves an impressive 1.52% gain than the best baseline, which verifies the effectiveness of it. Applying multiway attention mechanism and capsule networks to study opinion questions MRC is reliable. We find that multiway attention mechanism extracts more useful information but also noise than single attention mechanism. Furthermore, our results demonstrate that capsule networks filter noise better than traditional methods.

Comparing with *BiDAF* and *BiDAF-relu*, there is a slight decrease in accuracy after we used the multiway attention mechanism. This result means that multiway attention mechanism extracts more noise than single attention mechanism and the primitive model encoder can't filter noise well. Thus, the accuracy is reduced. Furthermore, results of *Capsule-relu* prove that capsule networks are better at noise filtering. We can also make a conclusion that multiway attention mechanism can give more useful interactions than noise comparing the results of *Capsule-relu* with *Capsule-O*. Moreover, both leaky-softmax and orphan category which are used in dynamic routing to eliminate the interference of noisy capsules get significant improvement. This result tells us that routing child capsules to appropriate parent layer and filtering the noisy capsules are both important to opinion questions MRC task.

5 Conclusion and Future Work

In this paper, we propose a new end-to-end model for opinion reading comprehension task. Specifically, we explore a multiway attention mechanism to extract more useful interactions between context and query, two methods for dynamic routing process to eliminate the interference of noisy capsules. In addition, we investigate a strategy to cope with more than two inputs MRC tasks.

One direction of future work is using more powerful embedding vectors to enhance word representation such as ELMO[34]. Otherwise, RNN-based networks are often slow, maybe we can adopt other structure to replace it in contextual embedding layer such as what Qanet did.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (Grand Nos. U1636211, 61672081, 61370126), and the National Key R&D Program of China (No. 2016QY04W0802).

We would like to thank lixinsu, sarasra, freefuiiismyname and andyweizhao. Their open source projects on github reduce our work on coding, thus we can take more time to focus on studying.

References

1. Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).
2. Seo, Minjoon, et al. "Bidirectional attention flow for machine comprehension." arXiv preprint arXiv:1611.01603 (2016).
3. Wang, W., et al. "R-NET: Machine reading comprehension with self-matching networks." Natural Lang. Comput. Group, Microsoft Res. Asia, Beijing, China, Tech. Rep 5 (2017).
4. Yu, Adams Wei, et al. "Qanet: Combining local convolution with global self-attention for reading comprehension." arXiv preprint arXiv:1804.09541 (2018).
5. Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8: 1735-1780 (1997).
6. Chen, Danqi, Jason Bolton, and Christopher D. Manning. "A thorough examination of the cnn/daily mail reading comprehension task." arXiv preprint arXiv:1606.02858 (2016).
7. Wang, Shuohang, and Jing Jiang. "A compare-aggregate model for matching text sequences." arXiv preprint arXiv:1611.01747 (2016).
8. Sabour, Sara, Nicholas Frosst, and Geoffrey E. Hinton. "Dynamic routing between capsules." Advances in neural information processing systems. 2017.
9. Zhao, Wei, et al. "Investigating capsule networks with dynamic routing for text classification." arXiv preprint arXiv:1804.00538 (2018).
10. Rajpurkar, Pranav, et al. "Squad: 100,000+ questions for machine comprehension of text." arXiv preprint arXiv:1606.05250 (2016).
11. Hermann, Karl Moritz, et al. "Teaching machines to read and comprehend." Advances in neural information processing systems (2015).
12. Hewlett, Daniel, et al. "Wikireading: A novel large-scale language understanding task over wikipedia." arXiv preprint arXiv:1608.03542 (2016).

13. He, Wei, et al. "Dureader: a chinese machine reading comprehension dataset from real-world applications." arXiv preprint arXiv:1711.05073 (2017).
14. Xiong, Caiming, Victor Zhong, and Richard Socher. "Dynamic coattention networks for question answering." arXiv preprint arXiv:1611.01604 (2016).
15. AI-Challenger2018 Homepage, https://challenger.ai/competition/oqmrc2018, last accessed 2019/05/17.
16. Chen, Danqi, et al. "Reading wikipedia to answer open-domain questions." arXiv preprint arXiv:1704.00051 (2017).
17. Cui, Yiming, et al. "Attention-over-attention neural networks for reading comprehension." arXiv preprint arXiv:1607.04423 (2016).
18. Bowman, Samuel R., et al. "A large annotated corpus for learning natural language inference." arXiv preprint arXiv:1508.05326 (2015).
19. Yang, Yi, Wen-tau Yih, and Christopher Meek. "Wikiqa: A challenge dataset for open-domain question answering." Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (2015).
20. Tan, Ming, et al. "Improved representation learning for question answer matching." Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Vol. 1. (2016).
21. Rocktäschel, Tim, et al. "Reasoning about entailment with neural attention." arXiv preprint arXiv:1509.06664 (2015).
22. Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).
23. Kadlec, Rudolf, et al. "Text understanding with the attention sum reader network." arXiv preprint arXiv:1603.01547 (2016).
24. Joachims, Thorsten. "Text categorization with support vector machines: Learning with many relevant features." European conference on machine learning. Springer, Berlin, Heidelberg, (1998).
25. Liaw, Andy, and Matthew Wiener. "Classification and regression by randomForest." R news 2.3, 18-22 (2002).
26. Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. ACM, (2016).
27. Kim, Yoon. "Convolutional neural networks for sentence classification." arXiv preprint arXiv:1408.5882 (2014).
28. Joulin, Armand, et al. "Bag of tricks for efficient text classification." arXiv preprint arXiv:1607.01759 (2016).
29. Hinton, Geoffrey E., Alex Krizhevsky, and Sida D. Wang. "Transforming auto-encoders." International Conference on Artificial Neural Networks. Springer, Berlin, Heidelberg, (2011).
30. Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).
31. Srivastava, Rupesh Kumar, Klaus Greff, and Jürgen Schmidhuber. "Highway networks." arXiv preprint arXiv:1505.00387 (2015).
32. Sara Github, https://github.com/Sarasra/models/tree/master/research/capsules, last accessed 2019/5/20.
33. Loshchilov, Ilya, and Frank Hutter. "Sgdr: Stochastic gradient descent with warm restarts." arXiv preprint arXiv:1608.03983 (2016).
34. Peters, Matthew E., et al. "Deep contextualized word representations." arXiv preprint arXiv:1802.05365 (2018)..