# CNN for Text-Based Multiple Choice Question Answering

Akshay Chaturvedi, Onkar Pandit, Utpal Garain

# CNN for Text-Based Multiple Choice Question Answering

**Akshay Chaturvedi**[†] **Onkar Pandit**[‡] **Utpal Garain**[†]
[†]Computer Vision & Pattern Recognition Unit, Indian Statistical Institute,
203, B. T. Road, Kolkata-700108, India
[‡]INRIA,40 Avenue Halley, Villeneuve-d'Ascq 59650, Lille, France
{akshay91.isi,oapandit}@gmail.com, utpal@isical.ac.in

## Abstract

The task of Question Answering is at the very core of machine comprehension. In this paper, we propose a Convolutional Neural Network (CNN) model for text-based multiple choice question answering where questions are based on a particular article. Given an article and a multiple choice question, our model assigns a score to each question-option tuple and chooses the final option accordingly. We test our model on Textbook Question Answering (TQA) and SciQ dataset. Our model outperforms several LSTM-based baseline models on the two datasets.

## 1 Introduction

Answering questions based on a particular text requires a diverse skill set. It requires look-up ability, ability to deduce, ability to perform simple mathematical operations (e.g. to answer questions like how many times did the following word occur?), ability to merge information contained in multiple sentences. This diverse skill set makes question answering a challenging task.

Question Answering (QA) has seen a great surge of more challenging datasets and novel architectures in recent times. Question Answering task may require the system to reason over few sentences (Weston et al., 2015), table (Pasupat and Liang, 2015), Wikipedia passage (Rajpurkar et al., 2016; Yang et al., 2015), lesson (Kembhavi et al., 2017). Increase in the size of the datasets has allowed researchers to explore different neural network architectures (Chen et al., 2016; Cui et al., 2016; Xiong et al., 2016; Trischler et al., 2016) for this task. Given a question based on a text, the model needs to attend to a specific portion of the text in order to answer the question. Hence, the use of attention mechanism (Bahdanau et al., 2014) is common in these architectures.

Convolutional Neural Networks (CNN) have been shown to be effective for various natural language processing tasks such as sentiment analysis, question classification etc. (Kim, 2014). However for the task of question answering, Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) based methods are the most common. In this paper we build a CNN based model for multiple choice question answering[1]. We show the effectiveness of the proposed model by comparing it with several LSTM-based baselines.

The main contributions of this paper are (i) The proposed CNN model performs comparatively or better than LSTM-based baselines on two different datasets. (Kembhavi et al., 2017; Welbl et al., 2017) (ii) Our model takes question-option tuple to generate a score for the concerned option. We argue that this is a better strategy than considering questions and options separately for multiple choice question answering. For example, consider the question "The color of the ball is" with three options: red, green and yellow. If the model generates a vector which is to be compared with the three option embeddings, then this might lead to error since the three option embeddings are close to each other. (iii) We have devised a simple but effective strategy to deal with questions having options like none of the above, two of the above, all of the above, both (a) and (b) etc. which was not done before. (iv) Instead of attending on words present in the text, our model attends at sentence level. This helps the model for answering look-up questions since the necessary information required to answer such questions will often be contained in a single sentence.

[1]The code is available at https://github.com/akshay107/CNN-QA



*Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pages 272–277
Melbourne, Australia, July 15 - 20, 2018. 

## 2 Method

Given a question based on an article, usually a small portion of article is needed to answer the concerned question. Hence it is not fruitful to give the entire article as input to the neural network. To select the most relevant paragraph in the article, we take both the question and the options into consideration instead of taking just the question into account for the same. The rationale behind this approach is to get the most relevant paragraphs in cases where the question is very general in nature. For example, consider that the article is about the topic *carbon* and the question is "Which of the following statements is true about carbon?". In such a scenario, it is not possible to choose the most relevant paragraph by just looking at the question. We select the most relevant paragraph by word2vec based query expansion (Kuzi et al., 2016) followed by tf-idf score (Foundation, 2011).

### 2.1 Neural Network Architecture

We use word embeddings (Mikolov et al., 2013) to encode the words present in question, option and the most relevant paragraph. As a result, each word is assigned a fixed $d$-dimensional representation. The proposed model architecture is shown in Figure 1. Let $q, o_i$ denote the word embeddings of words present in the question and the $i^{th}$ option respectively. Thus, $q \in \mathbb{R}^{d \times l_q}$ and $o_i \in \mathbb{R}^{d \times l_o}$ where $l_q$ and $l_o$ represent the number of words in the question and option respectively. The question-option tuple $(q, o_i)$ is embedded using Convolutional Neural Network (CNN) with a convolution layer followed average pooling. The convolution layer has three types of filters of sizes $f_j \times d \; \forall j = 1, 2, 3$ with size of output channel of $k$. Each filter type j produces a feature map of shape $(l_q + l_o - f_j + 1) \times k$ which is average pooled to generate a $k$-dimensional vector. The three $k$-dimensional vectors are concatenated to form $3k$-dimensional vector. Note that Kim (2014) used max pooling but we use average pooling to ensure different embedding for different question-option tuples. Hence,

$$h_i = CNN([q; o_i]) \quad \forall i = 1, 2, .., n_q \tag{1}$$

where $n_q$ is the number of options, $h_i$ is the output of CNN and $[q; o_i]$ denotes the concatenation of q and $o_i$ i.e. $[q; o_i] \in \mathbb{R}^{d \times (l_q + l_0)}$. The sentences in the most relevant paragraph are embedded using the same CNN. Let $s_j$ denote the word embeddings of words present in the $j^{th}$ sentence i.e. $s_j \in \mathbb{R}^{d \times l_s}$ where $l_s$ is the number of words in the sentence. Then,

$$d_j = CNN(s_j) \quad \forall j = 1, 2, .., n_{sents} \tag{2}$$

where $n_{sents}$ is the number of sentences in the most relevant paragraph and $d_j$ is the output of CNN. The rationale behind using the same CNN for embedding question-option tuple and sentences in the most relevant paragraph is to ensure similar embeddings for similar question-option tuple and sentences. Next, we use $h_i$ to attend on the sentence embeddings. Formally,

$$a_{ij} = \frac{h_i \cdot d_j}{||h_i||.||d_j||} \tag{3}$$

$$r_{ij} = \frac{exp(a_{ij})}{\sum_{j=1}^{n_{sents}} exp(a_{ij})} \tag{4}$$

$$m_i = \sum_{j=1}^{n_{sents}} r_{ij} d_j \tag{5}$$

where $||.||$ signifies the $l^2$ norm, $exp(x) = e^x$ and $h_i \cdot d_j$ is the dot product between the two vectors. Since $a_{ij}$ is the cosine similarity between $h_i$ and $d_j$, the attention weights $r_{ij}$ give more weighting to those sentences which are more relevant to the question. The attended vector $m_i$ can be thought of as the *evidence* in favor of the $i^{th}$ option. Hence, to give a score to the $i^{th}$ option, we take the cosine similarity between $h_i$ and $m_i$ i.e.

$$score_i = \frac{h_i \cdot m_i}{||h_i||.||m_i||} \tag{6}$$

Finally, the scores are normalized using softmax to get the final probability distribution.

$$p_i = \frac{exp(score_i)}{\sum_{i=1}^{n_q} exp(score_i)} \tag{7}$$

where $p_i$ denotes the probability for the $i^{th}$ option.

### 2.2 Dealing with forbidden options

We refer to options like none of the above, two of the above, all of the above, both (a) and (b) as *forbidden options*. During training, the questions

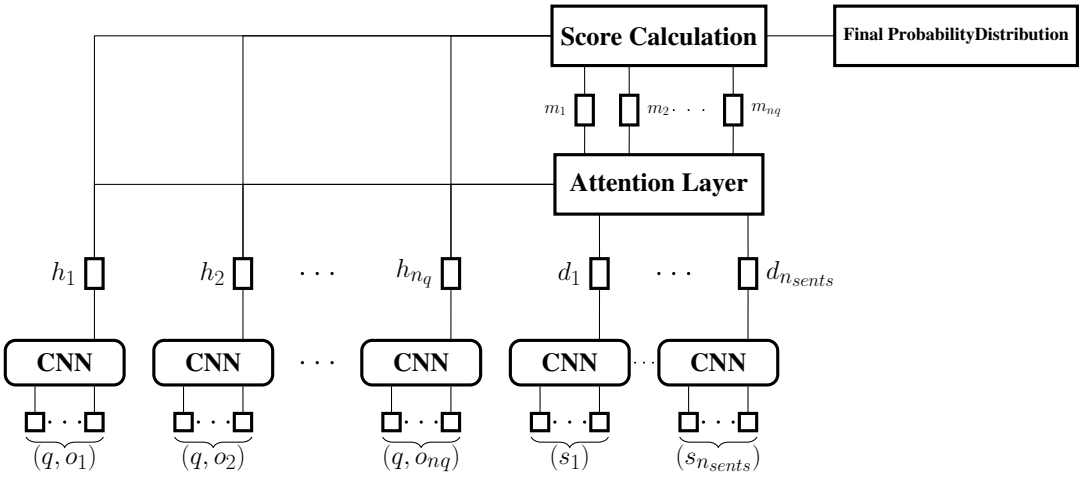


Figure 1: Architecture of our proposed model. Attention layer attends on sentence embeddings $d_j$'s using question-option tuple embeddings $h_i$'s. Score Calculation layer calculates the cosine similarity between $m_i$ and $h_i$ which is passed through softmax to get the final probability distribution.

having a forbidden option as the correct option were not considered. Furthermore, if a question had a forbidden option, that particular question-option tuple was not taken into consideration. Let $S = [score_i \; \forall i \mid i^{th}$ option not in forbidden options] and $|S| = k$. During prediction, the questions having one of the forbidden options as an option are dealt with as follows:

1. **Questions with none of the above/ all of the above option**: If the $max(S) - min(S) < threshold$ then the final option is the concerned forbidden option. Else, the final option is $argmax(p_i)$.

2. **Questions with two of the above option**: If the $S_{(k)} - S_{(k-1)} < threshold$ where $S_{(n)}$ denotes the $n^{th}$ order statistic, then the final option is the concerned forbidden option. Else, the final option is $argmax(p_i)$.

3. **Questions with both (a) and (b) type option**: For these type of questions, let the corresponding scores for the two options be $score_{i_1}$ and $score_{i_2}$. If the $|score_{i_1} - score_{i_2}| < threshold$ then the final option is the concerned forbidden option. Else, the final option is $argmax(p_i)$.

4. **Questions with any of the above option**: Very few questions had this option. In this case, we always choose the concerned forbidden option.

We tried different $threshold$ values ranging from 0.0 to 1.0. Finally, the $threshold$ was set to a value gave the highest accuracy on the training set for these kind of questions.

### 2.3 Training Details

We tried two different CNN models, one having $f_j$'s equal to 3,4,5 and other having $f_j$'s equal to 2,3,4. We refer to two models as $CNN_{3,4,5}$ and $CNN_{2,3,4}$ respectively. The values of hyperparameters used are: $d = 300, k = 100$. The other hyperparamters vary from dataset to dataset. Since the number of options vary from question to question, our model generates the probability distribution over the set of available options. Similarly, the number of sentences in the most relevant paragraph can vary from question to question, so we set $a_{ij} = -\infty$ whenever $d_j$ was a zero vector. Cross entropy loss function was minimized during training.

## 3 Results and Discussion

The accuracy of our proposed model on validation set of TQA and SciQ dataset (Kembhavi et al., 2017; Welbl et al., 2017) is given in Table 1 and Table 2. $GRU_{bl}$ refers to the model where CNN is replaced by Gated Recurrent Unit (GRU) (Cho et al., 2014) to embed question-option tuples and the sentences. The size of GRU cell was 100.

For SciQ dataset, we used the associated passage provided with the question. AS Reader (Kadlec et al., 2016) which models the question and the paragraph using GRU followed by attention mechanism got 74.1% accuracy on the SciQ test set. However, for a question, they used a different corpus to extract the text passage. Hence it is not judicious to compare the two models. As

| Model | True-False (Correct/Total) | Multiple Choice (Correct/Total) |
|---|---|---|
| $GRU_{bl}$ | 536/994 (53.9%) | 529/1530 (34.6%) |
| $CNN_{3,4,5}$ | 531/994 (52.4%) | 531/1530 (34.7%) |
| $CNN_{2,3,4}$ | 537/994 (54.0%) | 543/1530 (35.5%) |

Table 1: Accuracy for true-false and multiple choice questions on validation set of TQA dataset.

can be seen from the Tables 1 and 2, $CNN_{2,3,4}$ gives the best performance on the validation set of both the datasets so we evaluate it on the test sets. Note that $GRU_{bl}$ highly overfits on the SciQ dataset which shows that CNN-based models work better for those datasets where long-term dependency is not a major concern. This rationale is also supported by the fact that $CNN_{2,3,4}$ performed better than $CNN_{3,4,5}$ on the two datasets.

| Model | Accuracy |
|---|---|
| $GRU_{bl}$ | 68.2% |
| $CNN_{3,4,5}$ | 87.1% |
| $CNN_{2,3,4}$ | 87.8% |
| $CNN_{2,3,4}$ | 84.7% (test-set) |

Table 2: Accuracy of the models on SciQ dataset. The first three accuarcies are on validation set. The last accuracy is of $CNN_{2,3,4}$ model on the test set.

**Baselines for TQA dataset:** Three baselines models are mentioned in Kembhavi (2017) . These baseline models rely on word-level attention and encoding question and options separately. The baseline models are random model, Text-Only model and BIDAF Model (Seo et al., 2016). Text-Only model is a variant of Memory network (Weston et al., 2014) where the paragraph, question and options are embedded separately using LSTM followed by attention mechanism. In BIDAF Model, character and word level embedding is used to encode the question and the text followed by bidirectional attention mechanism. This model predicts the subtext within the text containing the answer. Hence, the predicted subtext is compared with each of the options to select the final option.

Note that the result of the baseline models given in Kembhavi (2017) were on test set but the authors had used a different data split than the publicly released split. As per the suggestion of the authors, we evaluate $CNN_{2,3,4}$ model by combining validation and test set. The comparison with the baseline models is given in Table 3.

| Model | True-False | Multiple Choice |
|---|---|---|
| Random* | 50.0 | 22.7 |
| Text-Only* | 50.2 | 32.9 |
| BIDAF* | 50.4 | 32.2 |
| $CNN_{2,3,4}$ | **53.7** | **35.8** |

Table 3: Accuracy of different models for true-false and multiple choice questions. Results marked with (*) are taken from Kembhavi (2017) and are on test set obtained using a different data split. Result of our proposed model is on publicly released validation and test set combined.

As can be seen from Table 3, $CNN_{2,3,4}$ model shows significant improvement over the baseline models. We argue that our proposed model outperforms the Text-Only model because of three reasons (i) sentence level attention, (ii) question-option tuple as input, and (iii) ability to tackle forbidden options. Sentence level attention leads to better attention weights, especially in cases where a single sentence suffices to answer the question. If question is given as input to the model, then the model has to extract the embedding of the answer whereas giving question-option tuple as input simplifies the task to comparison between the two embeddings.

SciQ dataset didn't have any questions with forbidden options. However, in the validation set of TQA, 433 out of 1530 multiple choice questions had forbidden options. Using the proposed threshold strategy for tackling forbidden options, $CNN_{2,3,4}$ gets 188 out of 433 questions correct. Without using this strategy and giving every question-option tuple as input, $CNN_{2,3,4}$ gets 109 out of 433 questions correct.

## 4 Conclusions and Future Work

In this paper, we proposed a CNN based model for multiple choice question answering and showed its effectiveness in comparison with several LSTM-based baselines. We also proposed a strategy for dealing with forbidden options. Using question-option tuple as input gave significant advantage to our model. However, there is a lot of scope for future work. Our proposed model doesn't work well in cases where complex deductive reasoning is needed to answer the question. For example, suppose the question is "How much percent of parent isotope remains after two half-lives?" and the

lesson is on carbon dating which contains the definition of *half-life*. Answering this question using the definition requires understanding the definition and transforming the question into a numerical problem. Our proposed model lacks such skills and will have near random performance for such questions.

## References


Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR* abs/1409.0473. http://arxiv.org/abs/1409.0473.

Danqi Chen, Jason Bolton, and Christopher D. Manning. 2016. A thorough examination of the cnn/daily mail reading comprehension task. In *Association for Computational Linguistics (ACL)*. https://www.aclweb.org/anthology/P16-1223.

Kyunghyun Cho, Bart van Merriënboer, Çalar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, pages 1724–1734. http://www.aclweb.org/anthology/D14-1179.

Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. 2016. Attention-over-attention neural networks for reading comprehension. *CoRR* abs/1607.04423. http://arxiv.org/abs/1607.04423.

Apache Software Foundation. 2011. Apache lucene - scoring. Letzter Zugriff: 20. Oktober 2011. https://lucene.apache.org/core/.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780. http://www.bioinf.jku.at/publications/older/2604.pdf.

Rudolf Kadlec, Martin Schmid, Ondrej Bajgar, and Jan Kleindienst. 2016. Text understanding with the attention sum reader network. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. http://aclweb.org/anthology/P/P16/P16-1086.pdf.

Aniruddha Kembhavi, Minjoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. pages 1746–1751. http://aclweb.org/anthology/D/D14/D14-1181.pdf.

Saar Kuzi, Anna Shtok, and Oren Kurland. 2016. Query expansion using word embeddings. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, New York, NY, USA, CIKM '16, pages 1929–1932. https://doi.org/10.1145/2983323.2983876.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR* abs/1301.3781. http://arxiv.org/abs/1301.3781.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*. pages 1470–1480. http://aclweb.org/anthology/P/P15/P15-1142.pdf.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*. pages 2383–2392. http://aclweb.org/anthology/D/D16/D16-1264.pdf.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *CoRR* abs/1611.01603. http://arxiv.org/abs/1611.01603.

Adam Trischler, Zheng Ye, Xingdi Yuan, Philip Bachman, Alessandro Sordoni, and Kaheer Suleman. 2016. Natural language comprehension with the epireader. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*. pages 128–137. http://aclweb.org/anthology/D/D16/D16-1013.pdf.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. *CoRR* abs/1707.06209. http://arxiv.org/abs/1707.06209.

Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *CoRR* abs/1502.05698. http://arxiv.org/abs/1502.05698.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. *CoRR* abs/1410.3916. http://arxiv.org/abs/1410.3916.

Caiming Xiong, Victor Zhong, and Richard Socher. 2016. Dynamic coattention networks for question answering. *CoRR* abs/1611.01604. http://arxiv.org/abs/1611.01604.

Yi Yang, Scott Wen-tau Yih, and Chris Meek. 2015. Wikiqa: A challenge dataset for open-domain question answering. ACL Association for Computational Linguistics. https://www.microsoft.com/en-us/research/publication/wikiqa-a-challenge-dataset-for-open-domain-question-answering/.