2021 24th International Conference on Computer and Information Technology (ICCIT), 18-20 December, 2021.

Augmenting BERT with CNN for Multiple Choice Question Answering

Shishir Roy[1]
shishir29@student.sust.edu

Nayeem Ehtesham[1]
theeyeman18@gmail.com

Md. Saiful Islam[1]
saiful-cse@sust.edu

Marium-E-Jannat[1]
jannat-cse@sust.edu

[1]Department of Computer Science and Engineering
Shahjalal University of Science and Technology, Sylhet, Bangladesh

***Abstract*—Multiple Choice Question (MCQ) answering is a strenuous task intended to determine the right answer from a set of given options. It demands a deep semantic understanding of the question, answer, and knowledge. In this article, we present a Convolutional Neural Networks (CNN) based model extended with Bidirectional Encoder Representations from Transformers (BERT) to answer complex multiple-choice questions. Given an article and a MCQ, the model selects the correct option by ranking each question-option tuple. The proposed CNN based model uses question-option tuple as input to perform better than Long Short-Term Memory(LSTM) based baselines by 22.7% for the Textbook Question Answering (TQA) [1] and SciQ [2] datasets.**

***Index Terms*—Question Answering, BERT, CNN, TQA, SciQ**

I. Introduction

MCQ answering task focuses on the comprehension of paragraphs and identifying the correct answer to a question from a set of possible options. In the context of information retrieval techniques, this field of study has been explored extensively. By considering deep neural network approaches, it has gained significant development over the last few years. The answer to a particular question may be spread throughout multiple sentences. So a specific set of diverse inferences and skills are required for models to deduce the solutions as well as perform simple mathematical calculations. It is the demand for skills that makes question-answering such a challenging task.

There has been a notable development in knowledge-based question answering over the last few years. In recent times, in particular, it has been impressive; thanks to the emergence of more challenging datasets and noble neural network architectures. Now, the models can find answers from passages [3], Wikipedia pages [4], tables [5], or even lessons [1]. Large datasets [6], [7] paved the way for experimenting with more complex and sophisticated neural network architectures. Using an attention mechanism [5], the answer can be derived from the text by driving the model for a particular section of the text to attend.

In recent years, one of the significant breakthroughs in Natural Language Processing (NLP) is the development of effective transfer learning and contextual word embedding models. Bidirectional Encoder Representations from Transformers (BERT) [8] is a language modeling neural network pre-trained on large corpora that can be fine-tuned for numerous downstream NLP tasks, including QA [9] and sentiment analysis [10]. The pre-trained language model has proven to be useful in learning the universal language representations.

Long Short-Term Memory (LSTM) [11] has been used for the task of question answering [5]. However, CNN has been used for various NLP works more effectively, such as in the case of sentiment analysis, question classification [12]. TableILP model developed by Khashabi, Daniel, et al [13] converts the MCQ answering system into a graph structure. They deemed the problem as a subgraph selection that requires to be optimized. The model tries to find a tuple (question, response) with the highest similarity with the knowledge base. Chaturvedi, Akshay, Onkar Pandit, and Utpal Garain [14] designed a CNN based system that takes question and option as a tuple and feeds it into CNN. However, this model does not perform well when it comes down to rich semantic relation analysis. Moholkar, Kavita, and S. H. Patil [15] introduced an ensemble approach for MCQ to predict answers using hybrid LSTM and CNN. Nevertheless, this model does not consider the fact of dealing with special options, i.e. *all of the above* or *none of them.*

In this paper, we augmented BERT with CNN for text-based multiple choice question answering with a deep understanding of semantic relations, and we have found that the CNN based model outperforms several LSTM based baselines by 22.7% for the TQA dataset. The model scores the possible options and uses them to find the correct answer. There can be situations where finding the correct answer needs more comprehensive linguistic understanding. For instance, options can be *all of the above* or *none of them* or *both a and b*. Here, the proposed model can better predict these types of questions than previous baselines [14]. The proposed model produces state-of-the-art results due to the following reasons (i) better understanding of underlying semantic relations between question and paragraph, (ii) question-option tuple as input, and (iii) the ability to tackle special options more gently. The main focus of this paper is to incorporate BERT with CNN, elevating multiple layers of representations for better interaction between questions and

978-1-6654-9435-9/21/$31.00 © 2021 IEEE

paragraphs.

II. Method

Consider the question Q and a set of answers $\{A_1, A_2, ..., A_n\}$; the system aims to determine the most suitable answer among these candidate answers. The proposed model comprises two parts. The first part generates a tensor representation for each word in a sentence and feeds it to a CNN model. The second part is to select the correct option based on the probability distribution of each option depending on the output of the first part.

A. Preparation

The pre-trained features from intermediate layers of BERT are more transferable [16], [17], [18], [19] or applicable to new tasks compared to later layers [20], which change more after fine-tuning [21], [22]. For fine-tuning BERT, we initialized all layers except one output layer with pre-trained weights. This is motivated by object detection transfer learning results which show that lower pre-trained layers learn more general features while higher layers close to output layers is more task specific [23]. BERT takes a sequence of special input representations of words as input and gives the encoded representation for each token as output. We get *d*-dimensional word embedding vector corresponding to each word.

A relevant paragraph to a question and options comprises several sentences. Let s_i denote the embedding representation of i^{th} sentence i.e $s_i = \mathbb{R}^{d \times n_i}$ n_i be the number of words present in i^{th} sentence.

$$u_i = CNN(s_i) \qquad \forall i = 1, 2, 3, ..., n_s \tag{1}$$

where n_s and u_i denote the number of sentences in the relevant paragraph and CNN's output. Let q, o_j denote the embeddings of words present in the question and the j^{th} option respectively. Thus, $q \in R^{d \times n_q}$ and $o_j \in R^{d \times n_o}$ where n_o and n_q refers to the number of words in the option and question. The question q and option tuple o_j is concatenated, and fed into CNN followed by average pooling.

$$v_j = CNN([q \oplus o_j]) \qquad \forall j = 1, 2, 3, ..., n_{option} \tag{2}$$

where v_j, n_{option} denotes the output of CNN, the number of options and $[q \oplus o_j]$ refers to the concatenation of question q and option o_j. The convolution layer has three types of filter of size f_j $\times d$ where $\forall_j$ = 1, 2, 3, with the size of the output channel of *k*.

B. Attention Layer

We determine the attention matrix e_{ij} of sentences u_i and question-option tuple v_j. Looking back at the purpose of attention, we try to consider local text substructure significance between the sentences u_i and question-option tuple v_j. Normally, weights are placed on each token in u_i according to their similarity with v_j. The related sub-phrases are obtained by a weighted combination of all tokens according to the following Eq. (3).

$$e_{ij} = \frac{u_i \cdot v_j}{\|u_i\| \cdot \|v_j\|} \tag{3}$$

$$r_{ij} = \frac{\exp\left(e_{ij}\right)}{\sum_{j=1}^{n_s} \exp\left(e_{ij}\right)} \tag{4}$$

$$m_i = \sum_{j=1}^{n_s} u_i r_{ij} \tag{5}$$

where $||{\cdot}||$ denotes the l^2 norm, *exp(x)* = e^x and u_i $\cdot v_j$ is the dot product between the two vectors. e_{ij} signifies the cosine similarity between u_i and v_j , the attention weights r_{ij} give more weightage to the more relevant sentences of the question.

C. Rank of Options

The attended vector m_i stands for the affirmation of the i^{th} option. We considered the cosine similarity between m_i and v_i while assessing the score for the i^{th} option

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

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

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

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

D. Dealing With Special Options

Special options refer to the options such as *all of the above, none of the above, any of the above, both a and b.* These special options were out of consideration during training. Let G = [$score_i$ $\forall i$ | i^{th} option not in forbidden options]. In the following, we experimented with different values of *threshold* ranging from 0·0 to 1·0. Finally we set the *threshold* that gave the highest accuracy for these kind of questions. We tackled forbidden options in four ways. During prediction, the questions having any type of the forbidden options as an option are dealt with as follows:

Algorithm 1: Determining the correct option while one is special

```
Result: Return the correct option
$H_{(n)}$ denotes the $n^{th}$ order statistic
if type is Both (a) and (b) then
    $score_{ia}$, $score_{ib}$ = Corresponding scores for $a$ and $b$ options
    if $|score_{ia}$ - $score_{ib}$ $| < threshold$ then
        correct option ← Both (a) and (b)
else if type is Two of the above then
    if H(k) - H(k-1) $<$ threshold then
        correct option ← Two of the above
else if type is None of the above or All of the above then
    if max(H) - min(H) $<$ threshold then
        correct option ← None of the above or All of the above
else if type is Any of the above then
    correct option ← Any of the above
else
    correct option ← argmax($P_i$)
end
```

III. Results And Discussion

A. Dataset Details

We have used two standard multiple choice questions answering dataset i.e Textbook Question Answering (TQA) [1] and SciQ [2] dataset, whose detailed statistics are shown in Table I.

Dataset	Split	Questions
TQA	TRAIN	15756
	DEV	5252
	TEST	5252
SciQ	TRAIN	8207
	DEV	2735
	TEST	2737

TABLE I: Statistics of TQA and SciQ datasets.

B. Experimental Setup

We fine-tuned the uncased, 24-layer $BERT_{Large}$[1] [8] with batch size 32, dropout 0.1, and peak learning rate 2×10^{-5} for three epochs. We evaluated with two separate CNN models (One's $f_{js} \approx$ 2,3,4 and other's $f_{js} \approx$ 3,4,5). Used hyperparameter values are as follows : k = 100, d = 300. The rest of the hyperparameters differ depending on the dataset. Our model produces the uniform distribution of probability over available options volume to tackle the fact that the option's quantity can differ among questions. Moreover, the sentence quantity can differ from question to question in the most relevant paragraph.

C. Result Analysis

Table II and Table III show the accuracy of our model on TQA and SciQ datasets on the validation sets, respectively. First, we experimented with replacing CNN by Gated Recurrent Unit GRU_{bl} [24] to embed the sentences and question-option tuples. The size of the GRU cell was set to 100. In Table II and III, validation metrics reveal that the GRU_{bl} model does not perform well on SciQ and multiple-choice questions of TQA. Next, we explore with $CNN_{2,3,4}$ and $CNN_{3,4,5}$ [14] for both datasets. Performance on the SciQ dataset increased by more than 19%.

We fine-tuned BERT with the previous baseline and created self-attention patterns to emphasize linguistic encoding features that give high attention weights to specific tokens from all other tokens in a question-option tuple. Token-to-token attention [25] of BERT increases the model's capacity to integrate semantic information about subject-verb and noun-pronoun relation. The model achieves a new state-of-the-art performance by reaching 58.4% on TQA and 89.7% accuracy on SciQ, surpassing the previous state-of-the-art set by Chaturvedi, Akshay, Onkar Pandit, and Utpal Garain [14].

Model	**True-False**	**Multiple Choice**
GRU_{bl}	53.9%	34.6%
$\text{CNN}_{3,4,5}$	52.4%	34.7%
$\text{CNN}_{3,4,5} + BERT$	58.0%	57.7%
$\text{CNN}_{2,3,4}$	54.0%	35.5%
$\text{CNN}_{2,3,4} + BERT$	58.4%	58.2%

TABLE II: Performance comparison of the models on the test set of TQA dataset for true-false and MCQ.

Model	Accuracy
GRU_{bl}	68.2%
$\text{CNN}_{3,4,5}$	87.1%
$\text{CNN}_{3,4,5} + BERT$	89.6%
$\text{CNN}_{2,3,4}$	87.8%
$\text{CNN}_{2,3,4} + BERT$	89.7%

TABLE III: Accuracy of the models on SciQ dataset.

For SciQ dataset, we have experimented with Attention Sum Reader. (AS Reader, a GRU with a pointer-attention mechanism; Kadlec, Rudolf, et al [26]) and Gated Attention Reader (GA Reader, an AS Reader with additional gated attention layers; Dhingra, Bhuwan, et al [27]). AS Reader scores 74.1% accuracy on the SciQ test set by using GRU followed by attention mechanism. However, for extracting the text passage they used different corpus, which makes the comparison of the two models difficult. Another fact to consider that GRU_{bl} model overfits on SciQ dataset, which indicates that CNN-based models work better if long-term dependency is not a major concern [14].

[1] https://github.com/huggingface/transformers

Baselines for TQA dataset:

Model	True-False	Multiple Choice
Random$^{\alpha}$	50.0	22.7
Text-Only$^{\alpha}$	50.2	32.9
BiDAF$^{\alpha}$	50.4	32.2
CNN$_{2,3,4}$	54.0	35.5
CNN$_{2,3,4}$ $+ BERT$	58.4	58.2

TABLE IV: Models performance comparison in terms of accuracy for True-False and MCQ of TQA dataset. Results labelled with α are are on test set obtained using a different data split and procured from Kembhavi, Aniruddha, et al. [1]. We evaluated the proposed model on publicly released validation and test set.

Kembhavi, Aniruddha, et al. [1] introduced three baseline models, i.e. random model, Text-Only model, and BiDAF Model [28]. They [1] modified the output layer of BiDAF to answer MCQ. Specifically, each answer option is juxtaposing to the predicted answer span, and the one with the maximum similarity is selected as the ultimate answer. These models don't perform well for both True/False and MCQ because for answering them correctly; it requires paraphrasing, multiple sentences reasoning. These baseline models focus on attention at word and sentence level, encoding questions and options individually and jointly. The baseline model results in [1] were on a test set, but the authors used a different data split than the publicly released split. we evaluated the proposed $CNN_{2,3,4} + BERT$ model combining the validation and test set as authors' suggestion. Apart from that, this model can make reasoning over multiple sentences and attends on sentence-level attention which reduces the unnecessary complexity of finding a question that has its answer in a single sentence. As shown by Table IV, our model reveals substantial improvement over other baseline models and achieves state-of-the-art results for multiple-choice question answering.

IV. Conclusion

In this work, we proposed a CNN based model extended with BERT for multiple-choice question answering and showed its efficacy compared to numerous baselines. Extensive experiments carried out on Textbook Question Answering (TQA) and SciQ datasets provided by Kembhavi, Aniruddha, et al. [1] and Welbl, Johannes, Nelson F. Liu, and Matt Gardner. [2] respectively showed that the model is capable of handling special options effectively while improving the overall accuracy. We have concluded that augmenting BERT with CNN and question-option tuple as input has significantly improved the performance of our model. The model requires further improvements and fine-tuning in cases where answering a question requires complex mathematical deductive reasoning. Future work will continue to explore this research direction, focusing on understanding the relevant paragraph better and turning the question into a numerical problem.

References

[1] A. Kembhavi, M. Seo, D. Schwenk, J. Choi, A. Farhadi, and H. Hajishirzi, "Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4999–5007.

[2] J. Welbl, N. F. Liu, and M. Gardner, "Crowdsourcing multiple choice science questions," *arXiv preprint arXiv:1707.06209*, 2017.

[3] J. Weston, S. Chopra, and A. Bordes, "Memory networks," *arXiv preprint arXiv:1410.3916*, 2014.

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

[5] P. Pasupat and P. Liang, "Compositional semantic parsing on semi-structured tables," *arXiv preprint arXiv:1508.00305*, 2015.

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

[7] D. Chen, J. Bolton, and C. D. Manning, "A thorough examination of the cnn/daily mail reading comprehension task," *arXiv preprint arXiv:1606.02858*, 2016.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://www.aclweb.org/anthology/N19-1423

[9] S. G. K. G. P. Sharma, R. S. Z. Lan, and M. Chen, "Albert: A lite bert for self-supervised learning of language representations," in *Submitted to International Conference on Learning Representations. https://openreview.net/forum*, 2020.

[10] Y. Song, J. Wang, Z. Liang, Z. Liu, and T. Jiang, "Utilizing bert intermediate layers for aspect based sentiment analysis and natural language inference," *arXiv preprint arXiv:2002.04815*, 2020.

[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

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

[13] D. Khashabi, T. Khot, A. Sabharwal, P. Clark, O. Etzioni, and D. Roth, "Question answering via integer programming over semi-structured knowledge," *arXiv preprint arXiv:1604.06076*, 2016.

[14] A. Chaturvedi, O. Pandit, and U. Garain, "Cnn for text-based multiple choice question answering," 2020.

[15] K. Moholkar and S. Patil, "Multiple choice question answer system using ensemble deep neural network," in *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*. IEEE, 2020, pp. 762–766.

[16] I. Tenney, D. Das, and E. Pavlick, "Bert rediscovers the classical nlp pipeline," *arXiv preprint arXiv:1905.05950*, 2019.

[17] J. Hewitt and P. Liang, "Designing and interpreting probes with control tasks," *arXiv preprint arXiv:1909.03368*, 2019.

[18] J. Hewitt and C. D. Manning, "A structural probe for finding syntax in word representations," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 4129–4138.

[19] N. F. Liu, M. Gardner, Y. Belinkov, M. E. Peters, and N. A. Smith, "Linguistic knowledge and transferability of contextual representations," *arXiv preprint arXiv:1903.08855*, 2019.

[20] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," *arXiv preprint arXiv:1904.09675*, 2019.

[21] A. Merchant, E. Rahimtoroghi, E. Pavlick, and I. Tenney, "What happens to bert embeddings during fine-tuning?" *arXiv preprint arXiv:2004.14448*, 2020.

[22] M. E. Peters, S. Ruder, and N. A. Smith, "To tune or not to tune? adapting pretrained representations to diverse tasks," *arXiv preprint arXiv:1903.05987*, 2019.

[23] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in neural information processing systems*, 2014, pp. 3320–3328.

[24] 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.

[25] O. Kovaleva, A. Romanov, A. Rogers, and A. Rumshisky, "Revealing the dark secrets of bert," *arXiv preprint arXiv:1908.08593*, 2019.

[26] R. Kadlec, M. Schmid, O. Bajgar, and J. Kleindienst, "Text understanding with the attention sum reader network," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 908–918. [Online]. Available: https://www.aclweb.org/anthology/P16-1086

[27] B. Dhingra, H. Liu, Z. Yang, W. W. Cohen, and R. Salakhutdinov, "Gated-attention readers for text comprehension," *arXiv preprint arXiv:1606.01549*, 2016.

[28] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, "Bidirectional attention flow for machine comprehension iclr," 2017.