Multiple Choice Reading Comprehension with Small Transformer Models

Yilun Wu, Bo Wu, Jiayue Guo
{yilunw, bw1, jiayueg}@andrew.cmu.edu
Language Technology Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Multiple Choice Reading Comprehension is a language comprehension task where given an article and a question about the article, the machine must select the correct answer from a fixed number of options. We re-implement a baseline model suited for English language understanding to answer such multiple choice questions, and explore different options to improve upon the model. Our models and experiments are based on ALBERT, DUMA, and end-to-end memory network, and it is trained and evaluated on the RACE dataset. The code used to obtain our results can be found at `https://github.com/CanYouTeachMeHowToCode/Multiple-Choice-Reading-Comprehension-with-Small-Transformer-Models`.

1 Introduction

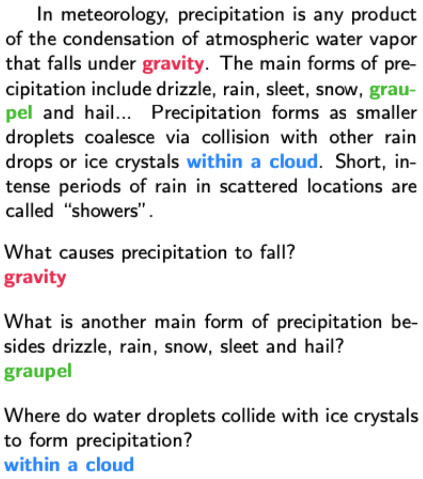
In recent years, machine reading comprehension (MRC) has become an increasingly popular and heated topic in natural language processing (NLP) research. MRC is a fundamental and long-standing goal of QA task which aims to teach machine to answer question automatically according to the content of the input passages/texts [1]. Reading comprehension is a challenging task for machines as it requires both understanding of natural language and knowledge in the context of different situations in our world.[2].

Early research on MRC focused on extracting the correct answers of certain relevant questions based on the content of the sample passage.

Most existing MRC models are trained on the SQuAD dataset, where the span of answer is marked in the passage. Such models could only handle the questions with expected answers in the form of a segment or short consecutive pieces of the corresponding passage. Multiple choice reading comprehension questions like those present in standardized tests such as TOEFL, SAT, and GRE typically require further reasoning or passage summarization, and shows the insufficiency of previous models.

In this paper, we focus on Multiple-choice Machine Reading Comprehension task, the task that utilizes deep neural network models to figure out the correct answer of a reading comprehension question from several different choices based on the passage content. We first discuss the relevant research and models regarding to Multiple-choice MRC, and we are going to re-implemented several models based on ALBERT and compared their performance. More specifically, we tailored End-To-End memory network[3] as a summarizer in our model; we also use cosine similarity to perform hard selection on passage sentences.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.



In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity. The main forms of precipitation include drizzle, rain, sleet, snow, graupel and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals within a cloud. Short, intense periods of rain in scattered locations are called "showers".

What causes precipitation to fall?
gravity

What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?
graupel

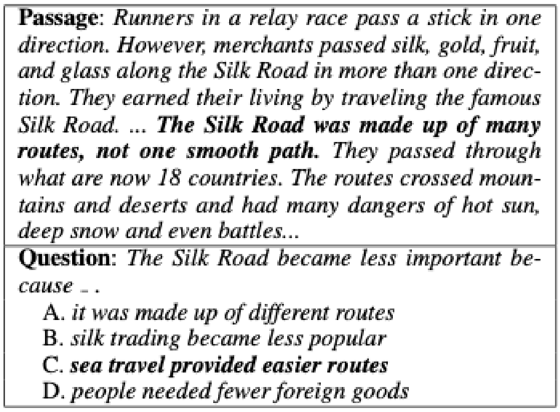
Where do water droplets collide with ice crystals to form precipitation?
within a cloud

Figure 1: Question-answer pairs for a sample passage in the SQuAD dataset [2]

2 Related Works

2.1 Task Specification

The specific NLP problem that we choose to tackle is Multi-choice Machine Reading Comprehension (MRC), which belongs to text-based QA. Given a passage P, a question Q is proposed with a set of candidate answers $\{A_1, ..., A_i, ..., An\}$ provided. Among all the answers, one is correct and we refer the reference answer as A_r. Our algorithm needs to select the correct answer with the information presented. Here is an example of multiple choice QA:



Passage: *Runners in a relay race pass a stick in one direction. However, merchants passed silk, gold, fruit, and glass along the Silk Road in more than one direction. They earned their living by traveling the famous Silk Road. ...* ***The Silk Road was made up of many routes, not one smooth path.*** *They passed through what are now 18 countries. The routes crossed mountains and deserts and had many dangers of hot sun, deep snow and even battles...*

Question: *The Silk Road became less important because _ .*

A. *it was made up of different routes*
B. *silk trading became less popular*
C. ***sea travel provided easier routes***
D. *people needed fewer foreign goods*

Figure 2: Example of a MC question that requires reasoning [1]

2.2 MRC Model Evolution

As stated in paper[4], before deep learning and neural network have gained popularity, people used rule-based methods such as bag-of-words, Word2Vec[5], and GloVE[6] to extract important information from text as embedding. However, such hand-crafted rule-based model cannot generalize well to large-scale corpus due to the tremendous human-effort it requires.

Due to the limitation of rule-based models and the scarcity of well-formed dataset, early MRC models have poor performance. However, with the appearance of neural networks, especially the appearance of attention mechanism[7] and transformer, MRC models learn to capture more complicated relations between text. Nowadays, the typical workflow for MRC is to first extract the embeddings of the given tokenized inputs, and to extract the features from the embeddings. This is typically done through transformer models such as BERT [8], ALBERT [9], GPT [10], T5 [11], and RoBERTa.[12]

One particular trait for MRC is that it requires the interaction between question and context features, so as to mimic the human reading behavior: when solving questions during reading, people usually refer to passage to find relevant information. Attention mechanism [7] is used in MRC to perform such interaction. Essentially, in MRC, there are two kinds of attention: unidirectional and bidirectional. Unidirectional readers attend passage (context) by question so as to extract the most relevant part of the context given the question. Modules that uses unidirectional attention includes Attentive Reader and Impatient Reader [13], where the module first extract features using LSTM and attend the feature to get the final embedding.

Bidirectional attention for MRC, on the other hand, also attend the question by the context so as to provide complimentary information. Models that rely upon bidirectional attention include Attention-over-Attention Reader [14], where the module sum over the row and column of attention scores in the first iteration, and performs attention for the second time; Bi-DAF[15], where the model calculates context-to-query attention and query-to-context attention. Dual Multi-head Co-Attention (DUMA)[16] separates the sequence into passage and question/answer parts, and do co-attention among the two sequences to produce representations that are aware of the context. Dual Co-Matching Network for Multi-choice Reading Comprehension (DCMN)[1] separates the hidden sequence produced by the pretrained model into hidden-state vectors for passage, question, and option, and then performs pair-wise attention.

According to how many times attention scores between context and query are calculated in the module, we can further categorize MRC models into one-hop interaction models and multi-hop interaction models. The models mentioned above (except Impatient Readers) are one-hop models. Examples of multi-hop models include Multi-stage Multi-task learning framework for Multi-choice reading comprehension (MMM) [17], which employs multi-step attention network (MAN) and End-To-End Memory Network Model[3], which employs recurrent attention.

Researchers are now trying to incorporate knowledge-base into MRC models. Knowledge base such as WordNet[18], Freebase[19], and Wikidata[20] introduces external knowledge that helps MRC models to better understand the semantic meanings of some entities in the passage and boost the performance.

2.3 Model For MRC

Most state-of-the-art models for MRC fine-tune on pretrained transformer-based models like BERT, RoBERTa, T5, and XLNet. There are two ways to fine-tune the model: 1) one is to directly use the pooler output (hidden vector for [CLS] token) produced by the pre-trained model and linearly project the pooler output to learn the probability distribution for each option. In backward training, the objective is set to cross-entropy classification loss; 2) another popular way is to build additional components on top of the pre-trained model. Instead of simply using the pooler output, the extra components usually manipulate the entire sequence of hidden vectors produced by the pre-trained model and produce the probability distribution for each option. For example, Dual Multi-head Co-Attention (DUMA) [16] separates the sequence into passage and question/answer parts, and do co-attention among the two sequences to produce representations that are aware of the context. Dual Co-Matching Network for Multi-choice Reading Comprehension (DCMN) [1] separate the hidden sequence produced by the pretrained model into hidden-state vectors for passage, question, and option, and then performs pair-wise attention. End-To-End memory network [3] performs recurrent attention between passage sentences embeddings and

3 Dataset Selection

In this paper, we choose RACE: Large-scale ReAding Comprehension Dataset From Examinations[1] [21] as our target dataset for model evaluation and analysis.

[1] Note that in this part we are providing a brief demonstration and summary of the RACE dataset, and for more information please refer to `https://arxiv.org/pdf/1704.04683.pdf`

3.1 RACE Dataset Introduction

RACE is a new large dataset for benchmark evaluation of MRC models. The main content of the dataset is the collection of the reading comprehension problems from English exams for middle school and high school Chinese students from 12 to 18 [21]. RACE dataset consists of near 28,000 passages and about 100,000 questions generated by English instructors (representative of human experts on reading comprehension tasks) [21]. Furthermore, this dataset covers a variety of topics range from news, story to ads, which are all carefully designed for evaluating the students' ability in reasoning and comprehension [16][21].

3.2 RACE Dataset Analysis

3.2.1 Dataset Statistics

In China, there usually exists drastic difference between middle school (12-15 years old) students and high school (15-18 years old) students in the reading comprehension abilities and English skills. Therefore, it is necessary to divide the dataset into two subgroups, RACE-M for middle school examinations and RACE-H for high school examinations [21], for further analysis.

Dataset	RACE-M	RACE-H	RACE-all
Passage Length	231.1	353.1	321.9
Question Length	9.0	10.4	10.0
Option Length	3.9	5.8	5.3
Vocabulary size	32,811	125,120	136,629

Table 1: Statistics of RACE dataset [21]

According to Table 1, the passage length and the vocabulary size in the RACE-H are much larger than that of the RACE-M, which indicates the higher difficulty levels of high school examinations comparing to middle school examinations.

3.2.2 Reasoning Types of the Questions

The questions in the RACE dataset are classified into 5 classes as follows with ascending order of difficulty [21]:

1. Word matching: The question exactly matches a span in the article. The answer is self-evident.
2. Paraphrasing: The question is entailed or paraphrased by exactly one sentence in the passage. The answer can be extracted within the sentence.
3. Single-sentence reasoning: The answer could be inferred from a single sentence of the article by recognizing incomplete information or conceptual overlap.
4. Multi-sentence reasoning: The answer must be inferred from synthesizing information distributed across multiple sentences.
5. Insufficient/Ambiguous: The question has no answer or the answer is not unique based on the given passage.

The statistics of the proportion of questions with each reasoning type in RACE, CNN, SQuAD, and NewsQA based on 1000 samples per dataset is summarized in Table 2 below:
According to Table 2, the reasoning questions with higher difficulty level in the RACE dataset has higher proportion than that in other datasets in general. Therefore, RACE is more suitable for the evaluation of the Multiple-choice MRC models comparing to other datasets.

Dataset	RACE-M	RACE-H	RACE-all	CNN	SQuAD	NewsQA
Word matching	29.4%	11.3%	15.8%	13.0%	39.8%	32.7%
Paraphrasing	14.8%	20.6%	19.2%	41.0%	34.3%	27.0%
Single-sentence reasoning	31.3%	34.1%	33.4%	19.0%	8.6%	13.2%
Multi-sentence reasoning	22.6%	26.9%	25.8%	2.0%	11.9%	20.7%
Insufficient/Ambiguous	1.8 %	7.1%	5.8%	25.0%	5.4%	6.4%

Table 2: Statistics of Reasoning Type Proportions [21][2][22][23]

4 Baseline Selection

We choose to re-implement ALBERT+DUMA, where ALBERT is used to encode the sequence of inputs and DUMA is used to process the encoded sequence [9][16]. On top of that, a linear classifier serves as a decoder to generate a probability distribution for each option.

4.1 ALBERT Pre-trained Encoder

As mentioned in Section 2, one drawback of transformer-based models is that they have hundred of millions of parameters, which requires tremendous computation power and huge memory. To tackle this problem, ALBERT [9] is proposed, which has the similar structure as BERT. To reduce the number of parameters and make BERT model scale better, it employs the following tricks:

- cross-layer parameter sharing
    - ALBERT enables hyper-parameter sharing across different layers, which significantly reduce the number of parameters. The experiment result shows that although parameter sharing reduces the performance of model, the reduction in performance is not severe and can be easily compensated by increasing the size of the model. Also, parameter sharing can help stabilize training.
- factorized embedding parameterization
    - Traditionally, the dimension of the word embedding (E) is the same as the hidden dimension of the model in the following layers (H). This means that the word embedding matrix will be of size $O(VH)$. Since V is large, the word embedding matrix will be large. ALBERT reduces the number of parameters in word embedding matrix by decoupling word embedding dimension from the hidden dimension (first project the one-hot vector for word into vector of dimension E, and then project it to a vector of dimension H). By doing this, the number of parameters become $O(VE + EH)$.

Apart from reducing the number of parameters, ALBERT changes the next-sentence-prediction(NSP) task of BERT to sentence-order-prediction (SOP). BERT generates negative samples by combining two segments from different corpus, thus confounding topic prediction and coherence prediction, and makes NSP an easy task. In contrast, ALBERT generates negative training samples by drawing two segments from the same corpus with their order switched, explicitly requiring model to focus or the coherence between texts.

4.2 DUMA Layer for Fine-tuning

Besides using ALBERT for encoding, we also choose a DUMA [16] layer to fine-tune the encoded sequences. The specific procedures of DUMA is listed below:

1. Separate the output representation from Encoder to obtain $E^P = [e_1^p, e_2^p, ..., e_{l_p}^p]$ and $E^QA = [e_1^{qa}, e_2^{qa}, ..., e_{l_{qa}}^{qa}]$, where e_i^p, e_j^{qa} denote the *i*-th and *j*-th token representation of passage and question-answer respectively and l_p, l_{qa} are the length.
2. Calculate the attention representations in a bi-directional way, that is, take 1) E^P as *Query*, E^{QA} as *Key* and *Value*, and 2) E^P as *Key* and *Value*, E^{QA} as *Query*. The exact formula is

shown below:

$$
\begin{aligned}
&\text{Attention}(E^P, E^{QA}, E^{QA}) = \text{softmax}\left(\frac{E^P(E^QA)^T}{\sqrt{d_k}}\right)E^{QA} \\
&\text{head}_i = \text{Attention}(E^PW_i^Q, E^{QA}W_i^K, E^{QA}W_i^V) \\
&\text{MHA}(E^P, E^{QA}, E^{QA}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \\
&\text{MHA}_1 = \text{MHA}(E^P, E^{QA}, E^{QA}) \\
&\text{MHA}_2 = \text{MHA}(E^{QA}, E^P, E^P) \\
&\text{DUMA}(E^P, E^{QA}) = \text{Fuse}(\text{MHA}_1, \text{MHA}_2)
\end{aligned}
$$

, where $W_i^Q \in \mathbb{R}^{d_{model}\times d_q}$, $W_i^K \in \mathbb{R}^{d_{model}\times d_k}$, $W_i^V \in \mathbb{R}^{d_{model}\times d_v}$, $W_i^O \in \mathbb{R}^{hd_v} \times d_{model}$ are parameter matrices, d_q, d_k, d_v denote the dimension of *Query* vectors, *Key* vectors and *Value* vectors, h denotes the number of heads, MHA($\cdot$) denotes Multi-head Attention and DUMA($\cdot$) denotes our Dual Multi-head Co-Attention.The Fuse($\cdot$,$\cdot$) function first uses mean pooling to pool the sequence outputs of MHA($\cdot$), and then aggregates the two pooled outputs through a fusing method.

3. (Decoder part) Take the output of DUMA and computes the probability distribution over answer options. Denote A_i as the *i*-th answer option, $O_i \in \mathbb{R}^l$ as the output of *i*-th $< P, Q, A_i >$ triplet, and A_r as the correct answer option, then the loss function is computed as:

$$
\begin{aligned}
O_i &= DUMA(E^P, E^{QA_i}) \\
L(A_r|P, Q) &= -\log \frac{\exp(W^TO_r)}{\sum_{i=1}^{s}\exp(W^TO_i)}
\end{aligned}
$$

, where $W \in \mathbb{R}^l$ is a learnable parameter and s denotes the number of candidate answer options.

4.3 End-To-End Memory Network for Summarizetion

One significant drawback of transformer model is that there is a fixed length limit on the input sequence during the tokenization process. One way to deal with this problem is to separate the passage into sentences and perform summarization over all the sentence embeddings.

In End-To-End Memory Network paper [??? cite], the author proposes a recurrent attention network and several variations that will be able to perform soft sentence selection and passage summarization. The model performs attention in a recurrent manner and add residual connections among hops. The exact procedures is listed below:

1. Given a set of sentences $\{x_1, x_2, ..., x_i\}$, the model first converts the tokenized sentences to key vectors $\{k_i\}$ using a learnable embedding matrix A. It also uses another embedding matrix C to produce a set of value vectors $\{v_i\}$ for the sentences.
2. For each question, an embedding matrix B is used to convert the tokenized question into question embedding q, which serves as query.
3. The attention score p_i is calculated as following:

$$p_i = Softmax(q^Tk_i)$$

We then take a weighted sum over value vectors $\{v_i\}$ to get a new query embedding q:

$$q_{new} = q_{old} + \sum_i p_i * v_i$$

4. The procedure above is called a "hop," in every iteration, we recurrently performs attention to get an internal embedding, and we also add residual connection from the original query embedding to the internal embedding to produce the new query embedding for next iteration. The weights for all embedding matrices are shared across hops.

The End-To-End Memory Network module is ideal for summarizing the whole passage, since the attention mechanism instructs it to focus on specific sentences. The recurrent hierarchical structure produces a more fine-grained summarizing vector for the passage.

5 Experiment

5.1 Evaluation Metrics

We choose to evaluate the performance of our model by classification accuracy. We model each option as a separate class. For example, if there are four options $\{A_1, A_2, A_3, A_4\}$ for *i*-th question and there are N questions in total, the accuracy is calculated by $N_{correct}/N$, where $N_{correct} = 1$ only when model select the correct answer and 0 otherwise.

5.2 Experimental Setting

We use albert-base-v2 as our encoder due to the computation limitation and add one DUMA layer on top of the ALBERT encoder. DUMA layer operates on the entire encoded sequence of hidden vectors instead of solely on the pooler output (hidden vector for [CLS] token). On the top of the model, five linear layers that share parameters are used in parallel, each followed by a dropout layer with dropout probability 0.5. We average the logits produced by the five linear classifiers to get the final logits. Cross-entropy loss is then applied on the logits.

We also set up two comparison experiments where DUMA layer is not added. In one setting, we simply mean-pool the sequence of hidden vectors produces by ALBERT and feed it to the classifier. In the other setting, we feed the hidden vector of [CLS] token to the classifier instead of manipulating on the hidden sequence. We experiment with the maximum sequence length for ALBERT encoder to see its impact on the accuracy.

During training, the initial learning rate is $2\text{e}{-5}$ and the batch size is set to 4. We train the model for 10 epochs in 6 hours. We use warmup schedulaer with warmup proportion 0.1 to stabilize the training in the initial few epochs. We use FP16 mixed-precision training to speed up the process.

5.3 Results

Table 5 shows the result of our re-implementation. Due to the computation limitation, we only choose to implement the base version of ALBERT. Our re-implementation results matched the ones reported in the original ALBERT [9] and DUMA[16] paper.

Model	max-seq-length	test acc(RACE-all)	test acc(reported)
ALBERT$_{base}$+DUMA	128	54.41	NA
ALBERT$_{base}$ (hidden sequence)	128	53.94	NA
ALBERT$_{base}$+DUMA	256	62.73	NA
ALBERT$_{base}$ (hidden sequence)	256	61.21	NA
ALBERT$_{base}$+DUMA	384	67.93	67.58
ALBERT$_{base}$ (hidden sequence)	384	65.84	NA
ALBERT$_{base}$ ([CLS] pooler output)	N/A	66.53	63.30
ALBERT$_{base}$ ([CLS] pooler output)	512	71.86	70.96

Table 3: Result

5.4 Error Analysis

Although our implementations mostly matched the accuracies reported by previous papers, they are still a decent chunk behind the state-of-the-art accuracies at around 90%, and even state-of-the-art methods suffer from inaccuracies. We here analyze 2 possible reasons for this error.

Model	max-seq-length	test acc(RACE-all)
ALBERT$_{base}$+DUMA	128	54.41
ALBERT$_{base}$ (hidden sequence)	128	53.94
ALBERT$_{base}$+DUMA	256	62.73
ALBERT$_{base}$ (hidden sequence)	256	61.21
ALBERT$_{base}$+DUMA	384	67.93
ALBERT$_{base}$ (hidden sequence)	384	65.84
ALBERT$_{base}$ ([CLS] pooler output)	512	70.96

Table 4: Result

Model	Test acc(RACE-all)	Notes
Baseline ALBERT$_{base}$	64.0	result from [1]
ALBERT$_{base}$+Sentence Selection	63.70	worse than baseline
ALBERT$_{base}$ (hidden sequence)	65.84	max-seq-len=384
ALBERT$_{base}$+DUMA	67.93	max-seq-len=384
ALBERT$_{base}$+DUMA	69.74	max-seq-len=512
ALBERT$_{base}$+HP Tuning	70.96	

Table 5: Results

Firstly, the limitations in the computational resources is significant compared to those used by state-of-the-art implementations. DUMA [16] uses 8 NVIDIA P40 GPUs, DCMN [1] uses 8 NVIDIA GTX 1080 Ti GPUs, and the current top entry on the leaderboard [24] uses 8 NVIDIA V100 GPUs. Each of these were trained for 2 days. In some cases, the authors used even more resources to speed up the training during the experiment phase, including uses of state-of-the-art auto tuning and resource management systems. In comparison, the only machine we had full access to had one TITAN RTX and one 1080 Ti only, which gives at least a 5 times difference in compute power. There was another machine we could borrow with good compute power, but we could only do so for a few hours. With few resources, it becomes much harder to tune the network since each experiment may take up to one whole day. To mitigate this, we may have to rent cloud resources using class coupons. We could also perform relevant experiments on the smallest models available as we did here, and then finally use those results on larger networks like ALBERT$_{xxlarge}$.

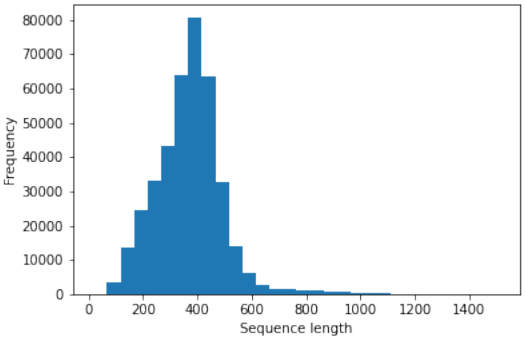


Figure 3: Lengths of the input sequences in RACE dataset

Secondly, maximum sequence length seems to affect performance greatly. In Table 5, we see that the models with higher maximum length produced better results under all conditions. One easy explanation is that longer sequences simply contain more information that the model can use to its advantage. We propose that not only is this the case, shorter sequence lengths can sometimes discard critical information that makes it impossible to predict the answer at all. Consider the length of the tokenized inputs to the model. The average input length is 370, and Figure 3 shows the histogram of the lengths of all input sequences in the RACE dataset. The percentage of sequences that are longer than 128, 256, 384, 512 are 98%, 83%, 46%, 8%, respectively. Consider any passage with 256 or more tokens, of which only 128 are used due to truncation. With more than half of the entire passage deleted, the information required to answer the question might not even be present, as many reading comprehension questions require information from a few key sentences in the passage. Since 83% of the data fall under this category, it is surprising that the model can even achieve a 54% accuracy with a max-seq-length of 128. This explains why the state-of-the-art models typically set the maximum to 512, as very few articles exceed the length so much as to truncate the important information. To mitigate this, we could also switch to a max length of 512, which requires more computing power but is doable with enough time. A better method is utilized in DCMN+ [1], where only a few sentences from the whole passage are passed into the transformer. The authors note that 87% of the questions can be answered using only two sentences from the passage, which gives a logical basis. The top K sentences relevant to the question are selected according to a similarity metric, and are then used as inputs to the transformer. This ensures that key information is retained, while reducing computational costs by a big margin. We will likely try this in our future experiments.

References

[1] Shuailiang Zhang, Hai Zhao, Yuwei Wu, Zhuosheng Zhang, Xi Zhou, and Xiang Zhou. DCMN+: dual co-matching network for multi-choice reading comprehension. *CoRR*, abs/1908.11511, 2019.

[2] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250, 2016.

[3] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. Weakly supervised memory networks. *CoRR*, abs/1503.08895, 2015.

[4] Kaixuan Li, Xiujuan Xian, Jiafu Wang, and Niannian Yu. First-principle study on honeycomb fluorated-inte monolayer with large rashba spin splitting and direct bandgap. *Applied Surface Science*, 471:18–22, Mar 2019.

[5] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[6] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[9] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942, 2019.

[10] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020.

[11] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2020.

[12] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[13] Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. *CoRR*, abs/1506.03340, 2015.

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

[15] Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *CoRR*, abs/1611.01603, 2016.

[16] Pengfei Zhu, Hai Zhao, and Xiaoguang Li. Dual multi-head co-attention for multi-choice reading comprehension. *CoRR*, abs/2001.09415, 2020.

[17] Di Jin, Shuyang Gao, Jiun-Yu Kao, Tagyoung Chung, and Dilek Hakkani-Tür. MMM: multi-stage multi-task learning for multi-choice reading comprehension. *CoRR*, abs/1910.00458, 2019.

[18] George A. Miller. WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*, 1994.

[19] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, page 1247–1250, New York, NY, USA, 2008. Association for Computing Machinery.

[20] Denny Vrandečić and Markus Krötzsch. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85, sep 2014.

[21] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard H. Hovy. RACE: large-scale reading comprehension dataset from examinations. *CoRR*, abs/1704.04683, 2017.

[22] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset. *CoRR*, abs/1611.09830, 2016.

[23] Danqi Chen, Jason Bolton, and Christopher D. Manning. A thorough examination of the cnn/daily mail reading comprehension task. *CoRR*, abs/1606.02858, 2016.

[24] Yufan Jiang, Shuangzhi Wu, Jing Gong, Yahui Cheng, Peng Meng, Weiliang Lin, Zhibo Chen, and Mu Li. Improving machine reading comprehension with single-choice decision and transfer learning. *CoRR*, abs/2011.03292, 2020.