SI630 Project Proposal: Machine Reading Comprehension on Multiple-Choice Questions

Zihui Liu

Abstract

In this project I work on a Machine Reading Comprehension problem on multiple-choice questions. The dataset I use is the RACE dataset(Lai et al., 2017), a multiple-choice reading comprehension dataset extracted from Chinese middle and high school exams. I tried the bert-base-uncased model(Devlin et al., 2018a) and switch to MiniLM-L6-H384-uncased model(Wang et al., 2020a) for training and fine-tuning the model. The result is evaluated by the accuracy. The best baseline achieves 30.15% and 27.19% accuracy for middle and high school set respectively. My fine-tuned model achieves 66.226% and 55.203% accuracy for middle and high school set respectively.

1 Introduction

Question Answer(QA) refers to the system that could automatically generate answers from the questions and context provided. Nowadays, QA systems are widely applied in online and telephone customer service and search engines. Some common applications of QA system are: chatbots for customer services and fast search result for search engines. One approach to building a QA system is by building up a Machine Reading Comprehension(MRC) model, which could save a huge amount of human effort when coping with basic problems.By reading in a passage, it could derive answers to the questions based on the article. For instance, such model is already applied in Bing's search services to provide answers to some simple style questions based on the context retrieved. A well-developed MRC model will benefit researchers in parsing and extracting information needed.

Previous research on machine reading comprehension mainly focus on locating the position where answers could be extracted from, and some do not work well if understanding of a larger scope is involved. Some others have huge amount of parameters to train on and complex model, which requires computational resource. In this project, I use MiniLM-L6-H384-uncased model(Wang et al., 2020a) to train on this dataset. I fine-tune the model for a better performance. For training, I use the passage, corresponding questions and answers to generate a model. The model should take the input of a passage and its questions(with several choices), then output the correct choice. Particularly, the model should demonstrate some understanding of the context for the whole passage instead of just allocating and searching for a phrase or sentence. BERT model fits this criteria since it is originally proposed to perform Masked Language Modeling(MLM) and Next Sentence Prediction(NSP), which require an understanding of the original context. For MiniLM model,it is a distilled version of large pre-trained transformer-based language model that is compressed to have light weight. Having both the features of BERT(Devlin et al., 2018b) and RoBERTa(Liu et al., 2019), it is also suitable for this task.

Our baseline is based on three aspects: The first is random guess, the second one is all choosing the choice with highest frequency, and the third one is based on comparison of sentence similarity. Among the three methods, the third one performs the best, achieving 30.15% accuracy on middle school set and 27.19% accuracy on high school set. After fine-tuning our model, it achieves 66.226% and 55.203% accuracy on middle and high school test set respectively, which shows much improvement compared with our baseline.

2 Dataset

While most existing MRC datasets focus on questions where answers are accessible by directly extracting information from text spans, like

SQuAD dataset(Rajpurkar et al., 2016) and AdversarialQA(Bartolo et al., 2020), they did not incorporate the ability to generalize and comprehend information based on the passage. Therefore, I aim to use another dataset to address this problem. The dataset I would like to make use of is the RACE dataset(Lai et al., 2017). It contains 27,933 passages and 97,687 questions collected from English tests of middle and high school in China. The answers of these questions are generated by school instructors instead of by crowdsourcing.

The general format of a training example is a text file but similar to json format, consisting of several keys: a unique id of the passage, the passage itself, a string list of questions, a list of the option list for each question, and the correct answer to each question marked by letters in upper cases. For the questions, they are categorized into two types: questions that fill in blanks(containing a "_"), and direct questions that end with a question mark and expect a response at the end. Based on calculation, the number of "fill in blank" questions is larger than that of direct questions for both middle and high school data for the training set.


```
{
    "answers": [
        "C",
        "C"
    ],
    "options": [
        [
            "all men were equal when they were born",
            "the lower classes should be ruled by the upper class",
            "the purpose of man was to seek freedom and wisdom",
            "people should not ask others to do what they did not want to"
        ],
        [
            "family .",
            "potential .",
            "knowledge .",
            "community ."
        ]
    ],
    "questions": [
        "socrates shared with confucius the idea that _ .",
        "what made some people different from others according to confucius ?"
    ],
    "article": "kong zi , also called confucius ( 551-479 b.c ) , and socrates ( 469-399 b . c ) lived
    only a hundred years apart , and during their lifetimes there was no contact between china and
    greece , but it is interesting to look at how the world that each of these great philosophers came
    from shaped their ideas , and how these ideas in turn , shaped their societies . confucius
    suggested the golden rule as a principle for the conduct of life : '' do not do to others what you
    would not want others to do to you . '' he assumed that all men were equal at birth , though some
    bad more potential than others , and that it was knowledge that set men apart . socrates focused on
    the individual , and thought that the greatest purpose of man was to seek wisdom . he believed that
    the superior class should rule the inferior classes . for socrates , the family was of no
    importance , and the community of little concern . for confucius , however , the family was the
    centre of the society , with family relations considered much more important than political
    relations . both men are respected much more today than they were in their lifetimes .",
    "id": "high74.txt"
}
```


Figure 1: Example of RACE Dataset

Furthermore, we count the distribution of choices for the training set as well. It could be seen that "C" have a larger proportion than other choices for both middle and high school train and development sets(Fig.2,Fig.3). For the test set, "B" is slightly more than "C" for the middle school set(Fig.4).

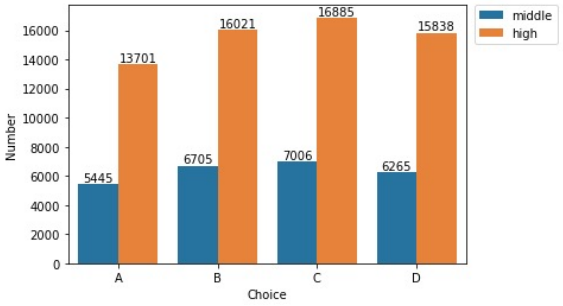

Figure 2: Distribution of Choices for the Training Set

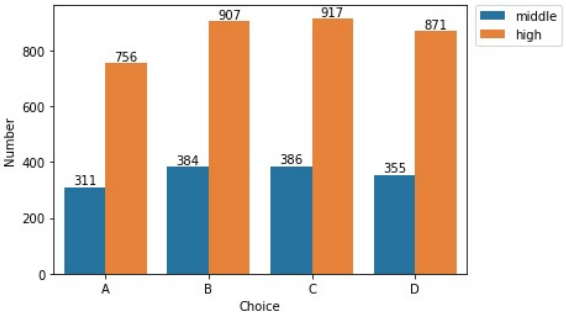

Figure 3: Distribution of Choices for the Development Set

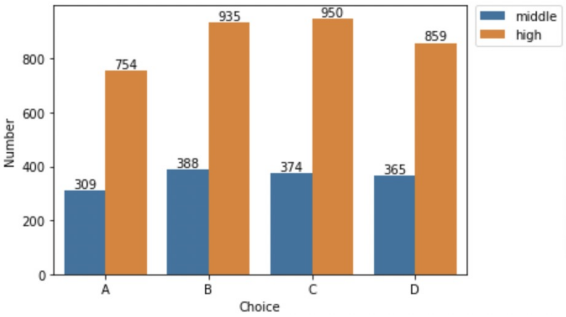

Figure 4: Distribution of Choices for the Test Set

3 Previous Work

Several studies concerning machine reading comprehension problem have been proposed by researchers.

Bi-Directional Attention Flow(BIDAF) network is introduced in *Bi-Directional Attention Flow for Machine Comprehension*(Seo et al., 2016), which uses a hierarchical multi-stage architecture to model context paragraphs. BIDAF includes character, word and contextual level embeddings, and computes attention at every step and previous layers. Therefore, the information is able to flow through several layers and the information loss is minimized. However, this method failed to identify

	train		dev		test	
	blank	direct	blank	direct	blank	direct
middle	14178	11243	805	631	792	644
high	31897	30548	1770	1681	1848	1650

Table 1: Question Type Statistics.

the boundaries for some question precisely.

Inspired by human's transitional thinking process, a DUal Multi-head Co-Attention(DUMA) model(Zhu et al., 2020) is proposed to aid pretrained language models encoded. On the basis of a multi-head attention model(Vaswani et al., 2017), it separates the passage and question-answer pair as two separate encodings. Then it calculates the attention using both encodings in a bi-directional way. Its co-directional attention architecture performs much better than soft attention since it could extract more matching information.

For solving end-to-end machine comprehension problems, Wang and Jiang proposed a new architecture for machine comprehension based on match-LSTM and Pointer Net(Vinyals et al., 2015). A match-LSTM model is used for textual entailment, where one sentence being the premise of another. The premise, or hypothesis is weighed by an attention matrix. The weighed premise after adding attention is fed into an LSTM and final predictions are made based on that LSTM. A pointer network also employs attention mechanism to choose a position in the input sequence as the output symbol. Combining these two techniques, researchers are able to identify a subsequence in the original text containing the target of the question. This approach works well with problems whose sources could be directly extracted from original text, but for questions involving understanding of a larger scope of context, this method fails to work well.

In the paper *Gated Self-Matching Networks for Reading Comprehension and Question Answering*,Wang et al. proposed a gated self-matching networks for reading comprehension problems. By combining attention-based gated recurrent network with self-matching layers, it achieves higher accuracy in locating and emphasizing question-passage relationship, and works well for questions with longer context.

For solving multiple-choice machine reading comprehension problem, Jiang et al. states that multiple-choice questions could be transformed into single-choice ones as a binary classifier. Every [Problem, Question, Option] pair could be vectorized and encoded as a sequence of token. For the output, it is either 0(wrong answer) and 1(correct answer), and thus transformed from a multi-choice model into a binary classifier. For the predicted output, it makes the final choice by comparing confidence scores of these binary classifiers and taking the highest one. The model is constructed with ALBERT-xxlarge model, and parameters are tuned with AutoML strategy(Elshawi et al., 2019). Compared with its counterparts on RACE leaderboard, this method achieves an accuracy as high as 91.4%. However, since this method is based on AutoML that requires huge amount of pretrained parameters, it is not suitable for our setting in this project. However, the idea of concatenating input as one entity but not as several pairs respectively is similar to my method.

4 Methodology

In this project, I first implement RACE dataset with the Bidirectional Encoder Representations from Transformers(BERT) model(Devlin et al., 2018b). Since BERT model is too large in practice, I also make use of MiniLM-L12-H384-uncased(Wang et al., 2020a) on hugging face to test the performance of the dataset. Then I tune parameters for the second model and analyze the performance.

4.1 Primary Setup of the model

BERT is based on the idea of multiple layers of transformer architecture. It is pretrained based on two tasks. One is masked language modeling(MLM) that covers tokens in sentences randomly with [MASK] token or random token. The other task is Next Sentence Prediction(NSP), predicting the possibility of sentence B being the next sentence of A(Devlin et al., 2018b). The main idea is taking a sequence of input and true output pair and train base on it. More detailedly, by taking two sequences of tokens $x_1, ..., x_N$ and $y_1, ..., y_N$ as input, they are concatenate into a single input

sequence as

$$[CLS], x_1, ...x_N, [SEP], y_1, ..., y_N[EOS]$$

Since RACE dataset is not the same as MLM or NSP task, the input sequence should be adjusted to fit it. The article part is regarded as x and the question-answer combination is y(Si et al., 2019). Therefore, the input becomes:

$$[CLS], article, [SEP], QuestionAnswer, [EOS]$$

Since the length of context is different for each reading, I use zero-padding to ensure the input lengths are the same. The label of each input sequence is from 0 to 3, where 0 represents A and 3 indicates D. Apart from encoding the input sequence for the original text, there are two other inputs. One is whether the input should be masked. In this case, none of the tokens in the input are masked ones, meaning that their ids should be 1. Another is an input term separating the article and the question-answer pair. For positions corresponding to the article, they should be encoded as 0, and the question-answer pair should be encoded as one. For input sequence for text, separation and masks, they are all of the same length.So zero-padding is needed for separation and mask ids as well.
For the output after training, the BERT model will output its predicted label. By comparing if the predicted value matches the actual one, I could evaluate the performance and tune parameters.

4.2 Preprocessing of the Dataset

I first retrieve the data from the given text files in each directory, then get the article, questions, options and golden label to each question from the files and save each (article, question, options, label) item by using a self-defined class type. Then there are two options for processing the QuestionAnswer part as the second input. One is directly concatenating a question and 4 corresponding answers one by one. However, since there are some cloze-like questions with underlines to fill in, and if the underlined parts are not substituted by the real choices, chances are that the trainer won't be able to perform well due to the discontinuity in information provided. Therefore, I also tried another way: for cloze-like questions, I substitute the choices into the underline parts to make a coherent sentence. Later on in the evaluation part, I will compare the performance of these two options.

4.3 BERT Implementation of RACE Dataset

For my first trail, I ran my model with bert-base-uncased model(Devlin et al., 2018a). Being trained on BookCorpus and English Wikipedia, it is a very large and comprehensive model. However, I soon find that the model is too large even for training on Great Lakes cluster. If I implement it with batch size that exceeds 1, there will be memory issues. Therefore, after discussing with the professor, I switch to a distilled version of the BERT model for my implementation.

4.4 MiniLM Implementation of RACE Dataset

Since bert-base-uncased model is too large and will cause memory problem when the batch size exceeds 1 on Great Lakes cluster, I turn to implement with MiniLM-L6-H384-uncased(Wang et al., 2020a) and fine tune the parameters. Being a distilled model from BERT and RoBERTa, it could perform masked language prediction tasks and works similar to BERT model. In other words,the input format is similar to that of BERT, therefore no changes are needed in encoding the input. The optimizer I use is BertAdam in pytorch, being a similar implementation to the optimizer in Tensorflow implementation of BERT(Hug).

I also employ a warmup linear learning rate instead of a constant learning rate, where the learning rate gradually increases for several steps before reaching the final learning rate. This is because if some data with strongly observable features are shuffled together at the initial state, the model's initial training state could skew badly towards those features, and this may leads to training extra epochs before converging. I further decrease the warmup linear proportion to a suitable value(0.05) and eventually discard it, which achieves the best performance.

Due to the storage limit of the server, I found larger batch size(>8) are not applicable in my training. To test with the result of larger batch size, I searched online and implement a gradient accumulation step to imitate the effect of larger batch size. This is done by not updating but saving the gradient values for several steps, accumulate the gradient for some batches and update them together.

5 Evaluation

For the machine reading comprehension model for multiple-choice questions I generated, I plan

to evaluate it mainly based on accuracy, that is, the proportion of questions answered correctly. Though there are other metrices as F1 score or precision available, accuracy is still the most direct and effective way to evaluate for a multiple choice problem. Besides the overall accuracy, I also analyze the accuracy for the two types of questions(cloze and direct) respectively. This could be helpful to understand where the model needs to be improved.

5.1 Baseline

The baseline of my project is based on three aspects.

The first one is based on random selection. That is, for each question, select A,B,C or D randomly. This method achieves 25.63% accuracy on middle school set and 25.67% accuracy on high school set.

One is always stick to the more frequent choice, which is C(since B is slightly more than C only in one part of the whole dataset, we still regard C as the most frequent choice). This baseline will have 26.04% accuracy on middle school problems and 27.16% accuracy on high school problems.

Another baseline I tried is based on the idea of sentence similarity. First, for each article, I split it into sentences with nltk's tokenize function. For each question-answer pair, I join them together. Then I use pre-trained GloVe(Pennington et al., 2014) vector representation to encode each sentence and question-answer pair. By comparing the cosine similarity of each sentence and the question-answer pair, I select the question-answer pair that has the highest cosine-similarity with one sentence of the article, and set that as my predicted choice. This method achieves 30.15% accuracy on middle school questions and 27.19% accuracy on high school problems for the test set, which has higher accuracy than that of the first method.

Figure 5: Accuracy of Three Baselines

5.2 Evaluation Results

I tried two ways of merging the question answer pair, and fine-tuned the batch size, the learning rate and the warmup proportion for the learning rate. I fix the gradient accumulation step to be 8 at first and tune the learning rate or batch size.

It could be seen that my model is more effective than the baseline in predicting the choice according to the context. There are no obvious difference in the two methods for handling Question-Answer pair.

I also evaluated the performance of direct questions and cloze questions for the two methods. Though the replacement method could raise the accuracy for cloze questions a bit, it does not improve too much and instead performs worse in middle school dataset.

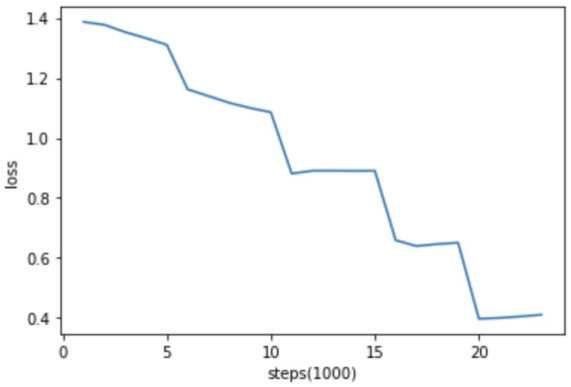

Figure 6: Loss on Training Set every 1000 Steps

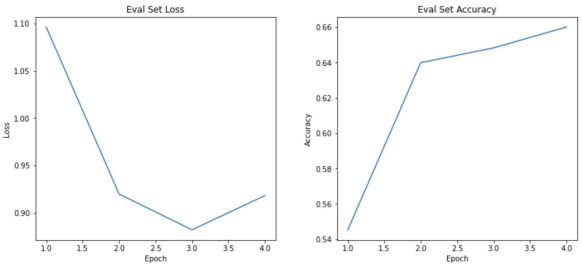

Figure 7: Loss and Accuracy on Evaluation Set Every Epoch

However, inspired by the fact that the training model overfits on the number of epochs I originally assigned(5 epochs), I recognized that the warmup learning rate might actually unnecessary in this case.(see Fig.6 and Fig.7) Therefore I drop the warmup proportion of learning rate and eventually got a result that improves a little. Table 2 and 3 is the overall result.

	Middle	High
Baseline (sentence similarity)	30.15%	27.19%
Direct Concatenation	65.111%	53.631%
Replace "_"	62.883%	54.889%
Replace "_" no warmup proportion	**66.226%**	**55.203%**

Table 2: Accuracy on Middle and High School Datasets

	Cloze	Direct
Direct Concatenation	58.220%	55.536%
Replace "_"	58.864%	55.493%
Replace "_" no warmup proportion	**59.735%**	**56.888%**

Table 3: Accuracy on Different Question Types

The parameters I used for these results are: maximum sequence length=360, batch size=4, gradient accumulation step=5
For the direct concatenation, the learning rate is 2.6e-5,and the warmup proportion of learning rate is 0.05. For the replacing underlines method, the learning rate is 2.7e-5,and the warmup proportion of learning rate is 0.05. For replacing underlines with no gradient accumulation step, the learning rate is 2.5e-5, and there is no the warmup proportion of learning rate.

6 Discussion

Overall, my result improves much compared with the baseline, achieving more than 50% accuracy on both middle school and high school test sets. Therefore, the could demonstrate some understanding of the context in the article, rather than just looking for partial similar sentences. Furthermore, I explore the usage of warmup learning rate and gradient accumulation step on training process.

However, it could be seen that the model is far from satisfaction. For end-users such as teachers trying to get a sample answer set, this accuracy couldn't be applied in practice. Judging from the performance of the evaluation set, the model's evaluation accuracy improves very slightly since the second epoch(all over 63% accuracy) for the middle school set. Though I tried to tune the learning rate, batch size, warmup proportion and gradient accumulation step by controlling the other variables, the final result does not have a very large improvement. The training loss and evaluation result already shows a sign of convergence around the 4th epoch. Therefore, I believe that my model is not well-qualified for RACE dataset. MiniLM model is a distilled version of BERT model. Though its key framework is still a transformer, about two-thirds of the parameters are dropped. Therefore, the correlation of encoded contents is simplified due to reduced parameters and layers in the overall structure, and the text span that could focus on is relatively small. Actually, in the original paper of MiniLM(Wang et al., 2020b), the model could work extremely well on extractive question answering task such as SQuAD(Rajpurkar et al., 2016). This could explain why the accuracy on middle school test set is higher than that of the high school test set by about 10%, since the articles are longer and many questions often involves an understanding of the whole article.

Moreover, another reason is perhaps the domain knowledge that the model is trained on. For BERT model, it is trained on wikipedia and bookcorpus. But for most middle and high school readings for Chinese students, they are few professional settings and are usually of more daily context with fewer vocabulary and relatively simplified sentence structure. Furthermore, many of them are written or revised by Chinese in their own cultural settings and some structures of expression or logistic may therefore differ. One way to resolve this problem is to train our model on more data in a similar context. For example, we could gather more cloze questions in reading comprehensions(which are a common type of question in Chinese English Exams) to adjust the knowledge base of the model to some extent.

7 Conclusion

In this project, I work on a multiple choice machine reading dataset – RACE, which is based on English reading comprehension problems for Chinese tests. I switch from BERT to MiniLM to train and fine-tune the model. The evaluation metrice is the accuracy of prediction. Three baselines are set up, in which the one based on sentence similarity achieves the highest accuracy. My model with replacement of underlines in cloze question and no warmup learning rate achieves the best performance: 66.226% and 55.203% accuracy respectively on middle and high school test set, 59.735% and 56.88% accuracy on cloze and direct questions. Finally, possible reasons for the drawback

of my implementation is discussed. The github repo to this project is https://github.com/OkabeRintarouBeta/si630.

8 Other things tried

I have also tried to use a XLNet(Yang et al., 2019) model to train the data as a sentence prediction task. For each of the 4 options, I choose the one with the highest score by the prediction result. However, this method's accuracy on the evaluation set is very low(around 20% to 25%), and therefore it is not suitable for this task containing such long previous context.

9 What I would do differently

If time permits, the first idea I would like to try is to enlarge the training dataset. This could be done by hand scraping reading comprehensions in Chinese middle and high school English tests(e.g. readings used for College Entrance Exams and other open tests). Another is looking for existing datasets available. I searched and found CLOTH dataset(Xie et al., 2017), which is a cloze question dataset under similar setting for Chinese students' English tests. By getting more training data of similar context, the performance using MiniLM could possibly be improved.

The second idea is trying a different network structure. I am particularly interested in the DUal Multi-head Co-Attention(DUMA) model(Zhu et al., 2020) that could relate question and answer to the article in a transpose way. Since implementing it involves manually writing many layers of neural networks, it is extremely time-consuming to debug for me. Furthermore, there are existing implementations of this network by other students on Github(iamNCJ, 2021). But if time permits, it is still quite fun to write this attention network and test its performance on RACE dataset, since it is claimed that the DUMA model performs well on RACE.

References

Pytorch pretrained bert: The big extending repository of pretrained transformers. https://github.com/huggingface/transformers/blob/694e2117f33d752ae89542e70b84533c52cb9142/README.md#optimizers. Accessed: 2022-03-10.

Max Bartolo, Alastair Roberts, Johannes Welbl, Sebastian Riedel, and Pontus Stenetorp. 2020. Beat the ai: Investigating adversarial human annotation for reading comprehension. *Transactions of the Association for Computational Linguistics*, 8:662–678.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018b. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Radwa Elshawi, Mohamed Maher, and Sherif Sakr. 2019. Automated machine learning: State-of-the-art and open challenges. *arXiv preprint arXiv:1906.02287*.

iamNCJ. 2021. DUMA-pytorch-lightning. https://github.com/iamNCJ/DUMA-pytorch-lightning. [Online; accessed 10-April-2022].

Yufan Jiang, Shuangzhi Wu, Jing Gong, Yahui Cheng, Peng Meng, Weiliang Lin, Zhibo Chen, et al. 2020. Improving machine reading comprehension with single-choice decision and transfer learning. *arXiv preprint arXiv:2011.03292*.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

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

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

Chenglei Si, Shuohang Wang, Min-Yen Kan, and Jing Jiang. 2019. What does bert learn from multiple-choice reading comprehension datasets? *arXiv preprint arXiv:1910.12391*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. *Advances in neural information processing systems*, 28.

Shuohang Wang and Jing Jiang. 2016. Machine comprehension using match-lstm and answer pointer. *arXiv preprint arXiv:1608.07905*.

Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. 2020a. Minilmv2: Multi-head self-attention relation distillation for compressing pretrained transformers. *arXiv preprint arXiv:2012.15828*.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020b. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers.

Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 189–198.

Qizhe Xie, Guokun Lai, Zihang Dai, and Eduard Hovy. 2017. Large-scale cloze test dataset created by teachers. *arXiv preprint arXiv:1711.03225*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Pengfei Zhu, Hai Zhao, and Xiaoguang Li. 2020. Duma: Reading comprehension with transposition thinking. *arXiv preprint arXiv:2001.09415*.