NUANS: Miniproject
Machine Reading Comprehension in Multiple Choice Question Answering

Alessandro Garbetta
Sapienza University of Rome
garbetta.1785139@studenti.uniroma1.it
October 20, 2023

Abstract

In this work, I am analyzing and experimenting with various approaches for extracting context for Machine Reading Comprehension in Multiple-Choice Question Answering. Two different datasets are considered: MCTest-500 and QuALITY.
Several metrics and embeddings are employed to extract contexts based on specific questions, including BM25, ROUGE, Cosine Similarity, and different variations of BERT are used to train the model. Accuracy is the model metric.

1 Introduction

Machine Reading Comprehension (MRC) focuses on the ability of computer systems to read natural language text and respond to questions posed about that text. I have chose to work with different texts and multiple choice question answer.
The first step is to extrapolate the most interesting context from the entire story relatively to the specific question using different types of metric and embedding. Finally I fine-tuned natural language models as BERT, RoBERTa and ALBERT to achieve the prediction. In the last section we show the results and the problems that I have encountered during the evaluation of the models.

2 Dataset

Two datasets are being examined: MCTest-500 and QuALITY, to compare the challenges encountered in different types of writing. Both datasets have four answer options.
Once downloaded, the datasets are being processed to create CSV files with the necessary information.

MCTest-500

The dataset MCTest[1] (Richardson et al., 2013) consists of 500 short fictional stories composed of 150-300 words per text and a total of 2000 questions. The dataset is open-domain but constrained to concepts and words that a 7-year-old is expected to understand.

QuALITY

This second dataset[2] (Pang et al., 2021) is more complex, composed of 1.5k-6k words, with Project Gutenberg fiction stories, and other nonfiction articles. The dataset consists of approximately 4668 questions with human accuracy of 93.5% in identifying the correct answers.

3 Contexts

One of the main problems is that in a lengthy text, there may be information that is not relevant to answering a specific question. I use different approaches to extract the context (in entire phrases) from the original text, taking into account the specific question.
In our case, I extract contexts of: 1, 3, 5, 10, and 15 sentences, and I create a CSV file with context, story, question, answer options and numeric label.

3.1 Cosine similarity

Cosine Similarity is a metric that quantifies the similarity between two tokens (words) as two non-zero vectors defined in an inner product space.
To measure cosine similarity, an embedding function must be employed. I experimented with various types of embeddings: TF-IDF, fastText, GloVe and DPR, to assess the differences and observe how the final results are affected.

3.2 ROUGE

Recall-Oriented Understudy for Gisting Evaluation, is a set of metrics that quantify the longest common subsequence between two texts by counting the longest sequence of tokens shared by both. The

[1] https://mattr1.github.io/mctest/data.html

[2] https://github.com/nyu-mll/quality/tree/main/data/v1.0.1

basic idea is that a longer shared sequence suggest an higher degree of similarity.

3.3 BM25

BM25 is a bag-of-words retrieval function that ranks a set of documents based on the presence of the query terms in each document, without considering their proximity within the document.

4 Fine-Tuning of BERT and its variants

First of all, I need to process the data for fine-tuning the model. I repeat the context four times, once for each answer option and I concatenate at each context the question plus the answer. And the end the result is:

- BERT : [CLS] context [SEP] question + answer_opt(i)[PAD]

- ROBERTA : <s> context </s></s> question + answer_opt(i)</s><pad>

- ALBERT: [CLS] context [SEP] question + answer_opt(i)<pad>

This new input is then passed to the tokenizer and it returns the corresponding input_ids and attention_mask. I work with language model using classes and function that are present in Huggingface Transformers.

In this work I use BERT (Devlin et al., 2019) and its different variants models, RoBERTa (Liu et al., 2019) and ALBERT(Lan et al., 2020), to evaluate the answers. The BERT model is a bidirectional transformer pretrained using a combination of masked language modeling objective and next sentence prediction. The model learns an inner representation of the language that can then be used to extract features useful for downstream tasks.
The RoBERTa model modifies key hyperparameters and training with much larger mini-batches and learning rates. RoBERTa has the same architecture as BERT, but uses a byte-level BPE as a tokenizer (same as GPT-2) and uses a different pretraining scheme.
ALBERT model presents two parameter-reduction techniques to lower memory consumption (e.g layers are split in groups that share parameters) and increase the training speed of BERT. Computational cost remains similar to a BERT-like architecture.

5 Results and Issues

A series of tests were conducted to assess the performance of various models. The networks is trained for each context type developed with different types of embeddings. The best results were achieved with a learning rate of 1e-5, and the batch size could not exceed 4/8 due to memory constraints.
First fine-tuning was performed on BERT using the MCTest-500 dataset, and the results were consistently lower on average compared to those obtained with RoBERTa.
Upon completing the training phase on the first dataset with RoBERTa, it was observed that the best performance was achieved for longer contexts (reaching a maximum of 70% accuracy). This suggests that individual sentences did not contain sufficient information to generate a correct response.
For the second dataset were conducted the same tests. Only with the ALBERT model, the training accuracy changes and increases. All the experiments with BERT and RoBERTa had lower and stable values.
In this case, the accuracy values during training were much lower compared to the previous dataset, not exceeding 38%. In contrast to the first dataset, in the second, higher accuracy values were achieved during training even for contexts with fewer sentences. Additionally, for longer contexts of 10 and 15 sentences, input truncation was applied as it exceeded the maximum acceptable number of tokens.
After the training phase, the weights with the highest accuracy values, calculated for each type of embedding used to extract the contexts, were selected. During the testing phase, these weights were used to assess the contexts extracted with all types of embeddings. Table 1 shows the values for the tests with the highest accuracy. During the testing phase, the combination of DPR contexts with weights calculated through DPR context extraction achieved a final accuracy of 74%, while DPR and FastText weights with ROUGE and DPR contexts respectively reached 73%. On average, for the MCTest-500 Dataset, the worst results were obtained with GloVe embeddings. The results of QuALITY dataset are much lower than the other, but the relative best results, about 40%, are obtained with weights found with GloVe, ROUGE, TF-IDF, and BM25.
Finally, two additional tests were conducted using the weights calculated with the MCTest-500 dataset

MCTest-500			
Embedding	**Weight**	**Context**	**Accuracy**
DPR	DPR	15	0.74
ROUGE		15	0.73
DPR	FastText	15	0.73

QuALITY			
Embedding	**Weight**	**Context**	**Accuracy**
BM25	TF-IDF	1	0.41
GloVe		1	0.41
TF-IDF		3	0.41
BM25	GloVe	15	0.41
GloVe		15	0.41

Table 1: In this table, there are the weights, the embeddings used for context extraction, their length, and the max testing accuracy.

on the contexts and questions of the QuALITY dataset, and vice versa. The results were consistent with the values obtained in various tests on the test sets, with no significant differences observed. Tests on MCTest-500 with QuALITY weights have accuracy about 35%-45% and tests on QuALITY with MCTest-500 weights have accuracy about 30%. Following this link there are all the values and experiment done. Each sheet contain the test done with the best weights on all contexts.

6 Conclusion

The results obtained for the first dataset are quite satisfactory, achieving an accuracy of 75%. These results, when compared to those of the second dataset, can be explained by the fact that the texts in the first dataset were written in a very simple manner, with a highly linear sentence structure. Furthermore, the questions were designed to be easily answered and identifiable within individual sentences. This is not the case for the second dataset, where the texts are much longer and more complex. Additionally, the extraction of contexts does not allow for a clear reference to the subject of a single sentence. Possible improvements could involve using coreference resolution to always have the subject explicitly stated and provide a clear reference.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations.

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.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, and Samuel R. Bowman. 2021. Quality: Question answering with long input texts, yes! *CoRR*, abs/2112.08608.

Matthew Richardson, Christopher J. C. Burges, and Erin Renshaw. 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Conference on Empirical Methods in Natural Language Processing*.