|  |
| --- |
| **Multiple Choice Question-Answering Using Information Retrieval and BERT** |
|  |
| [**https://github.com/mscheriger/mscheriger\_text\_analytics\_project**](https://github.com/mscheriger/mscheriger_text_analytics_project)**/tree/master**  **Michael Scheriger** |
|  |
|  |
|  |
|  |

Abstract

|  |
| --- |
| 000  001  002  003  004  005  006  007  008  009  010  011  012  013  014  015  016  017  018  019  020  021  022  023  024  025  026  027  028  029  030  031  032  033  034  035  036  037  038  039  040  041  042  043  044  045  046  047  048  049 |

In this paper I display results from attempting to answer multiple choice questions using information retrieval and a BERT transformer. In addition to the ARC data containing 7,787 multiple choice questions and four options for each question, I also used the ARC corpus that contains 14 million sentences related to science. Despite attempting to use a BERT transformer, the best model was one that used the highest score using Elasticsearch when searching for both the question and answer within the ARC corpus.

Introduction

BERT and other transformers over the past few years have become effective at several NLP tasks, including question answering. Because of their self-attention mechanisms, transformers are capable of digesting questions and finding the answers to those questions within another body of text. For the task in this paper, I attempted to use these same transformers to answer multiple choice questions rather than open questions with nothing to choose from. Since the model only needs to choose between four options, my original hypothesis was that the model would perform better relative to the task of answering open-ended questions. However, the difficult part of this task is to identify the correct context for BERT to search through. While the model does have access to over 14 million sentences which may contain the answer, asking a transformer to find the answer to a question within such a vast amount of data is infeasible.

In order to find the context for BERT to search through, I utilized Elasticsearch. By searching for a specific question, I could take the top search results, append them together, and create a context that would hopefully contain the answer to the question. Unfortunately, this does not guarantee that the answer will be contained within the returned context. Ultimately, this is what led to poor results from the BERT model.

Related Work

When scientists first began experimenting with question-answering problems, there was a lack of datasets available to train and test models. To fix this, researchers developed synthetic datasets of triplets, comprised of context, questions, and answers1. However, models created from these datasets typically fail when transitioning to real-world data. Hermann et al. addressed this by creating triples from CNN and Daily Mail articles and their summaries; the articles themselves were the context, and multiple questions were created from the summaries by selecting an entity within the summary as the answer, and the remaining text became the question. Since the summaries were not direct copies of the text within the articles, the ability to answer these questions would demonstrate a form of reading comprehension as opposed to recognizing semantic or syntactic similarities. The researchers developed LSTM and transformer models to attempt to answer the questions in their new test dataset, which yielded accuracies ranging from 60% - 70%.

Models developed in this way excel at answering questions with answers that are explicitly stated but fail to perform well on more complex questions2. Welbl et al. sought to address this by creating a set of questions that could only be answered by referencing two or more articles, thereby eliminating questions that could be answered by reading a single sentence3. They did so by creating the “WikiHop” dataset. This is like the news dataset from Hermann et al. in its structure as triples of context, questions, and answers; however, rather than a single article, the context in the WikiHop dataset is comprised of multiple Wikipedia articles. This makes the question-answering problem much harder since the model may not know where to locate the answer. To be successful, the model would need to hop from one article to another when searching for the answer. The researchers tested multiple models, but their most effective model was again a form of LSTM neural network. This model had an accuracy of 54.5%. Given that human accuracy on this problem is typically around 85%, there is an opportunity to improve upon this result. Additionally, Clark et al. point out that questions are usually binary predicates and can often be answered without having to “hop” to another article.

There have also been attempts to create models answering standardized test questions similar to the task this paper will attempt. However, past attempts have only been able to use small datasets, leading simpler models to perform better than deeper, more complex models4. For example, Clark et al. attempted a similar task with a training set of only 108 questions (compared to the training set to be used in this paper of size 7,787 questions). The models which performed the best on their test set were basic information retrieval methods or even correlation. While this paper attempted to use BERT to tackle this problem, I also found that a basic information retrieval method performed better than deeper models.

Dataset

The ARC multiple choice dataset is comprised of 7,787 multiple choice questions, which have already been split into a training set and a test set. The questions are also divided into easy and difficult questions. Clark et al. categorized questions as easy if a prior information retrieval method was able to correctly answer the question. Each question comes with four potential answers.

In addition to the multiple-choice questions, Clark et al. also provide the ARC corpus, which contains 14 million sentences related to science in some way. According to Clark et al., the ARC Corpus has enough information to answer 95% of the multiple-choice questions. However, the difficulty of the problem lies in being able to extract the relevant information from the ARC Corpus.

Methods

## **Information Retrieval**

The first method attempted was an information retrieval method. First, I created an Elasticsearch database with the ARC Corpus. After preprocessing the corpus, I loaded all 14 million sentences into the Elasticsearch database so that I could easily and quickly query each question. The database was then used in all experiments.

With the database created, the information retrieval method is relatively simple. For each question, I made four queries using Elasticsearch, where each query is the concatenation of the question and each potential answer. The model then selects the answer that returned the highest score according to Elasticsearch. Intuitively, a correct answer to a question should return a higher score when searched relative to a query that uses an incorrect answer. While this method performed decently on what are considered easy question, it performed worse than randomly guessing for the difficult questions.

## **BERT**

With a baseline information retrieval model established, I attempted to use a BERT transformer to answer the questions. For this method, I used the BERT model that is pre-trained on the Squad dataset. In order to use the model to answer each question, I needed a corresponding context for each question within which BERT could search for the answer. To create the context, I used each question as a query using Elasticsearch, and returned the top five answers as my context. BERT could then identify where within the context it believed the answer was, and I could compare this answer to the multiple-choice options. The model then selected the option that was most similar to the answer returned by BERT.

Unfortunately, the context returned by querying Elasticsearch rarely contained the answer, even for the easy questions. In an attempt to remedy this, I tried increasing the number of answers returned from five to ten, and I experimented with different similarity measures (cosine similarity and rouge scores). However, these experiments did little to increase the accuracy. As a result, the simpler information retrieval method proved more successful than BERT.

Results

Results from each experiment can be seen in the table below.

|  |  |  |
| --- | --- | --- |
| **Method** | **Easy Acc.** | **Hard Acc.** |
| IR Method Train | 51.7% | 24.3% |
| IR Method Test | 51.6% | 25.1% |
| BERT 5 line context Cosine Similarity Train | 32.1% | 22.2% |
| BERT 5 line context Cosine Similarity Test | 31.9% | 23.2% |
| BERT 10 line context Cosine Similarity Train | 32.3% | 22.6% |
| BERT 10 line context Cosine Similarity Test | 31.7% | 24.4% |
| BERT 5 line context Rouge Train | 31.7% | 23.3% |
| BERT 5 line context Rouge Test | 31.4% | 23.5% |
| BERT 10 line context Rouge Train | 32.3% | 23.4% |
| BERT 10 line context Rouge Test | 31.1% | 24.7% |

All methods clearly have an easier time identifying the correct answer for the easy questions. For the hard questions, all methods do worse than randomly guessing. While disappointing, this result is not surprising – all results from the methods tried by Clark et al. were in the same range.

While the information retrieval method does fairly well on the easy questions, the BERT methods do only slightly better than a random method. There seems to be no difference between the experiments with similarity scores and context size. This is likely due to the inability of the information retrieval method using Elasticsearch to retrieve the appropriate context given the question.

Discussion

While the BERT methods underperformed, there are opportunities to improve. The primary concern is the inability of the retrieval method to produce a context that BERT can use to find the answer. This paper attempted to improve the Elasticsearch method by expanding the size of the context queried, as well as experimenting with different similarity scores. While these did not work, there are other experiments that future papers could try.

Other papers could attempt to improve the Elasticsearch method by using the answers in the query, similar to the IR method. Perhaps taking the top two results by querying the concatenation of the question and each multiple-choice answer would yield a context that has the true answer that BERT could then identify. Other papers could also abandon Elasticsearch altogether and try using other information retrieval techniques or Telescoping multiple techniques together.

With better context, BERT would surely perform better on the easy questions and possible on the hard questions. However, with better context I could have also fine-tuned the pre-trained model, which could potentially improve results even further. Even with these additional improvements, it is unlikely that this model would reach human performance given that the best method in this paper was 51.7% accurate on the easy questions and 25.1% accurate on the difficult questions. The task of answering multiple-choice questions using a large knowledge-base remains a difficult problem to solve.

References

1. K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt,W. Kay, M. Suleyman, and P. Blunsom. *Teaching machines to read and comprehend*. In Advances in Neural Information Processing Systems, pp. 1693–1701, 2015.

2. Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, Oyvind Tafjord *Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge*

3. J. Welbl, P. Stenetorp, and S. Riedel. *Constructing datasets for multi-hop reading comprehension across documents.* arXiv preprint arXiv:1710.06481, 2017b.

4. P. Clark, O. Etzioni, T. Khot, A. Sabharwal, O. Tafjord, P. D. Turney, and D. Khashabi. *Combining retrieval, statistics, and inference to answer elementary science questions*. In AAAI, pp. 2580–2586, 2016.