Question Answering on Long Context

Group Members: Xinli Gu, Di He, Congyun Jin
Mentors: Jocelyn Beauchesne, Najoung Kim

Abstract

Language model pretraining has led to significant performance on question answering purpose but answering questions when the context is the whole document is challenging. Our project is to answer the question given an unseen wikipedia article. We also enhance model automation by finding out an appropriate confidence threshold.

Introduction

Introduction: Question Answering system is a hot topic in NLP field, but most of the QA systems are based on small contexts, which has limited usage in real life. Our objective is that given a wikipedia full article and a question, answer the question from a span of the article or recognize that the question is impossible to answer in the given article . Furthermore, once a prediction has been made, we should obtain confidence thresholds for a given accuracy target. The goal is to answer as many questions as possible, under the accuracy constraints.

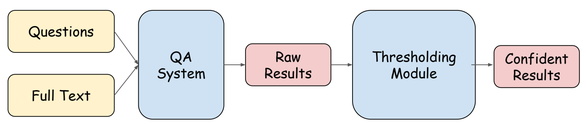

Figure 1. Workflow

Data: We built the dataset based on SQuAD 2.0 (The Stanford Question Answering Dataset), which is a popular benchmark dataset for past question answering works. The original dataset consists of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading passage, or the question might be unanswerable. In this work, our mentor matched the Wikipedia full articles for the specific questions in order to create very long context for QA systems.

Question: The New York Giants and the New York Jets play at which stadium in NYC ?
Context: The city is represented in the National Football League by the New York Giants and the New York Jets , although both teams play their home games at MetLife Stadium in nearby East Rutherford , New Jersey , which hosted Super Bowl XLVIII in 2014 . (Training example 29,883)

Figure 2. A training example from the SQuAD dataset, consisting of a question, context paragraph, and answer span (in green). In this project, we match the context with the full text in Wikipedia for training and prediction

	Train	Validation	Test
Total Articles	394	29	32
Answerable Questions	78,703	4,589	5,406
Unanswerable Questions	38,630	2,901	5,506

Table 1. Data Overview

Methods and Models

Method 1 Sliding Window: The whole article is separated into several overlapping passages based on windows size 320 (384-64) and stride 128. The query and every candidate passage will be passed to the reader to extract possible answers and its confidence scores based on start/end position, also the confidence score for impossible to answer. The final result is given by the one with highest confidence score.

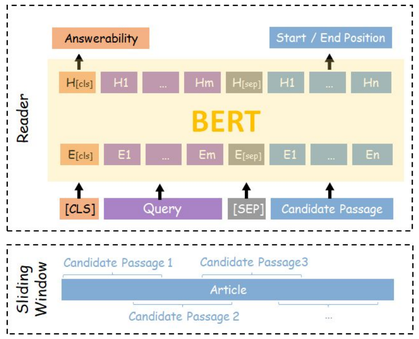

Figure 3. Sliding Window Method

Method 2 Retriever-Reader: The Reader, same as the one of sliding window method, perform the core task of question answering: extract answer based on the query and candidate passages. The Retriever assists the Reader by reducing the number of passages that the Reader has to process.

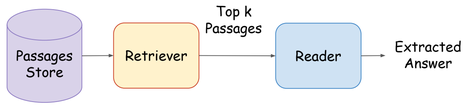

Figure 4. Retriever-Reader Pipeline

Reader: We fine-tuned the BERT and RoBERTa model using both the original SQuAD 2.0 dataset and our enriched dataset by introducing more unanswerable cases.

- BERT is a bi-directional transformer pre-training over unlabeled textual data that can be used to fine-tune for our question answering tasks.
- RoBERTa is a retraining of BERT with improved training methodology.

Lexical Retriever: Lexical retriever looks for literal matches of the query words in passages. And the method we used was BM25. BM25 is a variant of TF-IDF:

- Saturates TF after a set number of occurrences of the given term in the document.
- Normalises by document length so that short documents are favoured over long documents if they have the same amount of word overlap with the query

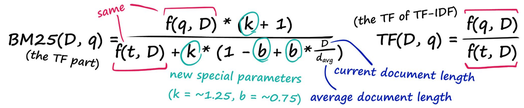

Figure 5. BM25 Formula

Semantic Retriever: Semantic search (or dense retrieval) encodes the query and passages into vectors and retrieves the top k passages which are most similar with the query in vector space.

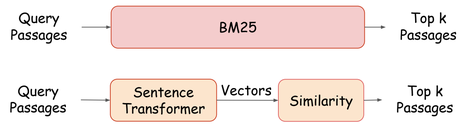

Figure 5. Retrievers

Results

Retriever Performance: The figure shows the matching score between the retrieved top k passages and the right answer context.

- We find BM25 is better than sentence transformer. Therefore, BM25 was chosen as our retriever to do Retriever-Reader experiments.
- The performance increasement came from higher top k has a diminishing effect. Hence BM25 for top 5 (BM25@5) was chosen as the passage retriever.

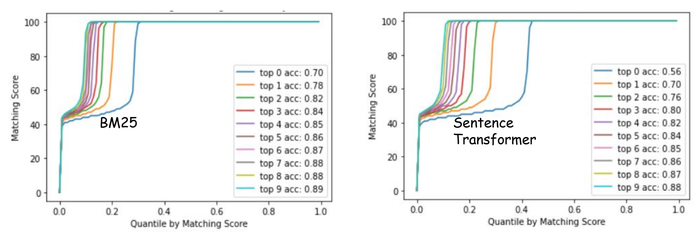

Figure 7. Retriever Performance

Overall Performance: The bolden results are based on original SQuAD 2.0 dataset, which can be viewed as the "performance ceiling" of the reader. We can compare them with performances on article context.

	Models			Total		Has Ans.		No Ans.	
Context	**Method**	**Reader**	**Data**	**EM**	**F1**	**EM**	**F1**	**EM**	**F1**
Paragraph	**NA**	**Fine-tuned BERT**	**NA**	**72.87**	**76.20**	**72.09**	**78.81**	**73.64**	**73.64**
Paragraph	**NA**	**Fine-tuned RoBERTa**	**NA**	**78.81**	**83.24**	**75.40**	**84.30**	**82.14**	**82.14**
Article	Sliding Window	Fine-tuned BERT	NA	52.28	55.93	64.19	71.56	40.59	40.59
Article	Sliding Window	Fine-tuned BERT	Enriched	65.96	69.35	62.30	69.15	69.56	69.56
Article	Sliding Window	Fine-tuned RoBERTa	NA	71.07	74.69	64.50	71.85	77.48	77.48
Article	Retriever BM25@5	RoBERTa SQuAD 2.0	NA	69.93	72.82	65.21	71.03	74.57	74.57

Table 2. Results

Automation

When pushing the model into production, it's crucial to be able to make automatic decisions with high confidence. Our work allows to answer questions automatically on long documents with high confidence, which results in fast quality answers for the users while lowering costs of human labeling for the company. We used the validation set to select the best threshold that can cover as many questions as possible while maintaining the 90% Exact Match score.

- **For HasAnswer questions:** We used the softmax score for the "start" token logit and 'end' token logit for thresholding
- **For NoAnswer questions:** We used the NoAns gap (difference in NoAns logit and HasAns logit) for thresholding

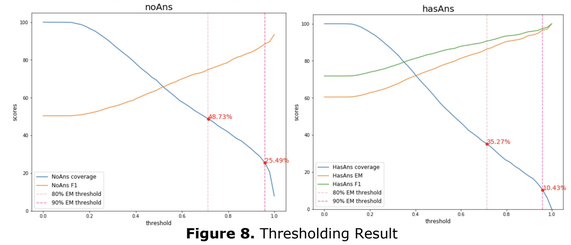

Figure 8. Thresholding Result

We listed 90% and 80% thresholds as for some use cases a 80% Exact Match might be enough, in which case we can increase the automation and lower costs of human review.

Conclusion and Future Work

- The sliding window method performance is not that far from the "performance ceiling". RoBERTa with sliding window is the best model from our experiments with a EM score gap of 7.74 and F1 score gap of 8.55 from the "performance ceiling".
- For a small decrease in performance we can reduce the computational costs largely. BM25@5 + RoBERTa shows great potential it could filter the whole article to 5 short contexts as the input to the reader.
- For model automation, we could cover 42 % questions which a human does not need to review the answer while maintaining the 80 Exact Match score by setting the threshold based on the validation set.
- We can explore on retriever since it is a bottleneck for our model performance, and we can finetune the retriever in future work.

Acknowledgements

We would like to thank our amazing mentor Jocelyn Beauchesne,Hyperscience for all his help and support throughout the project and also for all the effort he put in getting us quality datasets to work with. His ideas helped a lot in giving a proper direction to our project. We would also like to thank Najoung Kim, New York University for her constructive feedback and all his suggestions about question answering models.

References

1. *M. Ott and S. Edunov and A. Baevski and A. Fan and S. Gross and N. Ng and D. Grangier and M. Auli. 2019. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. Proceedings of NAACL-HLT 2019: Demonstrations*
2. *D.Jacob and C.Ming-Wei and L.Kenton and T.Kristina. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805*
3. *Karpukhin, Vladimir et al. 2020. Dense Passage Retrieval for Open-Domain Question Answering. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). https://www.aclweb.org/anthology/2020.emnlp-main.550*
4. *C. Danqi and F. Adam and W.Jason and B.Antoine. 2017. Reading Wikipedia to Answer Open-Domain Questions. Association for Computational Linguistics (ACL)*