FRI Data Science Project Competition 2020
Interim report

Question answering pipeline for closed domain questions

Luka Škodnik and Robert Jutreša

Abstract

In this project we focus on the use of large language models for question answering, focusing on two approaches: extractive and generative. In the extractive approach, the position of the answer within the context is extracted, while in the generative approach, the answer is generated from the given question and relevant contexts. The choice of context is crucial for the performance of the system, and a retriever is typically used to select the most informative contexts. Extractive models perform well when the answer is a substring of the context, while generative models can handle more complex questions and reasoning on the context. We plan to implement and compare the performance of both approaches.

Keywords

question answering, large language models, banking domain

Advisors: prof. dr. Marko Robnik Šikonja, Grega Jerkič, Branislava Šandrih Todorović

Introduction

In the past couple of years the quality of Natural Language Processing (NLP) tools that use Large Language Models (LLM) drastically increased. This prompted a large number of companies to start introducing them into their work environment in order to cut down on time and increase performance. To this end we take already existing models and approaches for open domain question answering (QA) and adapt them to our specific domain - banking. Using various techniques we will implement and evaluate different question answering pipelines, by fine-tuning open source models with domain specific data that has been provided to us. The goal is to provide a production-ready model, that outperforms the baseline LLMs for the required tasks.

Related Work

Datasets

When it comes to open domain question answering, the most widely used dataset is Standford Question and Answering Dataset (SQuAD) [1, 2]. The basic version contains over 100k question with corresponding passages (contexts) and answers. The extended version of the dataset also includes over 50k questions without and answer written by crowdworkers, with the intention of improving models by ensuring more accurate learning of when a context does not contain an answer. TriviaQA [3] and WikiQA [4] are also general question answering datasets, that have the same data structure as SQuAD. A data structure that was also used when preparing our own dataset. To mitigate the drawback of extractive models HotpotQA [5] also includes the positions of multiple supporting facts which should help the models perform more complex reasoning and provide explanations for the answers. The SuperGLUE [6] benchmark contains datasets for different language understanding tasks including question answering. Here the question answering tasks are formulated in multiple ways: only with yes/no questions, questions for which we need to choose the correct alternative and queries that need to be filled in with an answer at a certain position.

Models

Question answering tasks that are most relevant to us include extractive QA and generative QA. For extractive QA model, which extract the answer from a given context, the most widely spread model is BERT [7] and it's many variations. These models are most often fine-tuned on datasets such as SQuAD in order to better satisfy the open domain question answering task. Models such as GPT [8], T5 [9], LLaMA [10] and BLOOM [11], have recently seen an increase in popularity, with GPT being at the forefront of the movement. Being generative models they can be trained for the task of generating an answer to the question, either by having a provided context from which to generate an answer, or generating one without a context whatsoever.

The above mentioned models (depending on implementations) require contexts from a larger text database in order to ex-

tract/generate an answer. For this propose Dense Passage Retrieval (DPR) [12] was developed for open domain QA. Using question and context encoders it returns a paragraph with the highest relevancy to the answer.

Methodology

Data

The data we were provided with are the yearly sustainability reports of the NLB banking group. These reports are available as **.pdf* files or websites, and contain data about the development and working culture of the bank. To implement a QA model, questions, answers and contexts had to be extracted from the sources, and formatted such that they would suite already pre-trained models. To extract the text and generate question-answer pairs from a selected **.pdf* file, a pre-built QA generation pipeline from Haystack [13] was used. The generated data was then filtered by the NLB banking experts, and further post-processed to ensure standard formatting.
Furthermore, we are expecting another set of question-answer pairs to be provided to us, written by the NLB banking experts. These will be processed and used at a later date.

Question answering pipeline

When constructing the pipeline of our model, two approaches were considered. The first of which is the extractive approach. In this approach, models try to find the answer in a given context, which is a task well suited for the data generated with the previously mentioned pipeline. This approach has also been proven to be reliable for simpler questions, which should be sufficient for our use case. There are multiple limitations for this since models can accept only a limited length context and the answer might not always be contained in the context. To achieve preliminary benchmark results, the model of choice was a smaller variant of BERT called DistilBERT [14] which has already been pre-trained on the SQuAD dataset.

The second viable approach is of generative nature. Here the models learns to generate a sequence of words from an the input query (and context if provided). The benefit of these models is that they should be able to answer more complex questions since they are not only extracting the answer from but generating the answer based on some understanding of the natural language and the provided context. *This approach could be better suited for the question-answer pairs that are written and not automatically generated, as we have no guarantee that the answer will be provided directly in the context, but perhaps paraphrased or something similar. The models used for these approach will most likely include T5 and LLaMA.*

Since both the extractive approach and the selected generative approach require contexts to be provided together with the question, a pipeline has to be devised that provides the model with the required context for that answer. A sliding window implementation that looks through all of the possible contexts is possible, however it would be computationally inefficient and it might also yield worse results in the end. Therefore it is beneficial if we employ the use of a context retriever model which will identify a small number of relevant contexts to the questions which needs to be answered. *Here an DPR model will be used, as the first part of the full pipeline, that will find and extract the required context, and then feed it to QA model to return the answer.*

Evaluation

For evaluation several metrics were considered. Since SQuAD has it's own evaluation benchmark, this was the first relevant one. The two main components of this benchmark is the percentage of exact matches between the predicted and the ground truth answers for the test set, and the average overlap (F1 score). In this case the predictions and ground truths are transformed into bags of tokens, for which the overlap is then calculated. The second relevant metric BLEU [15], is a benchmark for evaluating translation models. Since it's idea is to match the n-grams of the translated text, to those of the human generated translation or ground truth. These matches are then averaged geometrically. The metric can be extended to other tasks with a similar set up, such as an QA task. Next we have Bertscore [16], which is a benchmark which leverages contextual embeddings to represent tokens, and then computes precision, recall and F1 score using cosine similarity between the embeddings of the predicted answer and those of the ground truth. *The SuperGLUE ReCoRD or MultiRC task benchmarks could also be used as evaluation metrics.*

Results

Table 1. Table of baseline results using DistilBERT pre-trained on SQuAD before and after fine-tuning on our data.

	Bertscore				SQuAD	
Model	F1	Precision	Recall	Bleu	Exact Match	F1
Basline	87.5	85.9	89.4	8.9	35.7	48.0
Fine-Tuned	89.0	87.5	91.0	9.2	37.5	49.1

Table 1 shows the effects of fine-tuning the DistilBERT model, that has already been pre-trained for the QA task.

Discussion

Thus far we have identified the main approaches for solving our problem and defined the pipeline we need to implement. We started by fine tuning a model for the extractive approach and shown that fine tuning on our domain specific data improves the performance of the model. When we receive the data written by the banking expert we plan on using that data to fine tune our models as it should be higher in quality. Knowing that this approach should work we now plan on firstly adapting a context retriever to extract the relevant contexts from the yearly reports. After this we plan to fine tune additional model(s) for the extractive approach and also

try out the generative approach. Finally we plan on identifying more evaluation protocols one of which would possibly be human evaluation where we would manually check the answers output by the models.

Acknowledgments

References

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

[2] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*, 2018.

[3] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

[4] Yi Yang, Wen-tau Yih, and Christopher Meek. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2013–2018, 2015.

[5] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*, 2018.

[6] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

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

[8] OpenAI. Gpt-4 technical report, 2023.

[9] 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. *CoRR*, abs/1910.10683, 2019.

[10] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[11] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022.

[12] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*, 2020.

[13] Deepset AI. Haystack: End-to-End Open-Source Framework for Transformers-based NLP. https://haystack.deepset.ai, 2021. Accessed: April 12, 2023.

[14] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[15] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[16] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.