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 use large language models for question answering, focusing on two approaches: extractive and generative. In the extractive approach, the span of the answer within the context is extracted, while in the generative approach, the answer is generated based on the given question and relevant context. The choice of a relevant context from a document or a set of documents is crucial for the performance of the system, and a separate 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 both approaches and compare their performance.

Keywords

question answering, large language models, banking domain

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

Introduction

In the past 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 employees time and increase performance. To this end we adapt existing models and approaches for open domain question answering (QA) 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 build a production-ready model, that outperforms baseline approaches 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 an answer written by crowd-workers, 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. We used the same data format for our own dataset. To mitigate the drawback of extractive models, HotpotQA [5] 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: yes/no questions, multiple choice questions and queries where an answer is located at a certain position.

Models

Question answering tasks that are the most relevant to us include extractive QA and generative QA. For extractive QA models, which extract the answer from a given context, the most widely used model is BERT [7] and it's variations. These models are most often fine-tuned on datasets such as SQuAD in order to adapt to the question answering task. Generative models such as GPT [8], T5 [9], LLaMA [10] and BLOOM [11], can be trained to generate an answer to the question, either from a provided context or without a context whatsoever.

Some of the above mentioned models require contexts from a larger text database to extract/generate an answer. For this propose Dense Passage Retrieval (DPR) [12] was developed for open domain QA. Using a 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 annual sustainability reports of the NLB banking group. These reports are available as *pdf* files or a website, 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 to 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 post-processed to ensure standard formatting. The data was then randomly split into train and test sets, based on a 80-20 split. Half of the test set was then used for validation during fine-tuning.
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. When this data is acquired, the data will then be joined an formally split into the train, validation and test sets.

Question answering pipeline

When constructing the pipeline of our model, two approaches were considered. The first 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 been shown to be reliable for simpler questions. There are multiple limitations of this approach since models can accept only a limited length context and the answer might not always be contained in the context. Our preliminary model was a smaller variant of BERT called DistilBERT [14] which has already been pre-trained on the SQuAD dataset. *Additional fine-tuning will have to be done after expanding the dataset. The full process will be explained here.*

Our second approach is generative. Such models learn 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 the context but generating the answer based on understanding the language and the provided context. *This approach could be better suited for 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 may be paraphrased or expressed implicitly. The baseline models used for this approach will likely include T5 and LLaMA.*

Since both the extractive and generative approach require contexts our pipeline has to provide one with the answer. A sliding window passing through all possible contexts is possible, but would be computationally inefficient. Therefore, we will employ a context retriever model which will identify a small number of relevant contexts to the given question. *Here the DPR model will be used as the first part of the pipeline, to find and extract the required context, and its output will be fed to the QA model to return the answer.*

Evaluation

For evaluation, we consider several metrics. Since SQuAD has it's own evaluation benchmark, this is out first choice. The two main components of this benchmark are the percentage of exact matches between the predicted and the ground truth answers, 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 calculated. The second metric, BLEU [15], is a benchmark for evaluating translation models. It's idea is to match the *n*-grams of the generated text, to those of the ground truth. These matches are then averaged geometrically. Next, Bertscore [16], 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. *Based on our dataset, we will also consider the SuperGLUE ReCoRD and MultiRC tasks in our evaluation.*

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. We can see marginal improvements across the board. We also ran similar testing using a model that was not pre-trained for the QA task, and we saw drastically better improvements. However, we choose to omit those tests, as we would like to leverage transfer learning in our model, and thus we should evaluate it as such.

Discussion

Thus far, we have identified the main approaches for solving our problem and defined the pipeline. 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 experts we plan on using that data to fine tune our models as it should be of better quality. Knowing that this approach works we plan on adapting a context retriever to extract the relevant contexts from the annual reports. After this, we plan to fine tune additional model(s) for the extractive approach and test the generative approach. Finally, we plan to identifying more realistic evaluation protocols including human evaluation where we would manually check a sample of the answers generated by the models.

Acknowledgments

References

[1] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, 2016.

[2] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, 2018.

[3] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, 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 Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, 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 *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. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[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. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[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 Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, 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.