FRI Data Science Project Competition 2023
Final report

Question answering pipeline for closed domain questions

Luka Škodnik and Robert Jutreša

Abstract

For this project, we are tasked with experimenting with possible implementations of closed-domain question-answering systems for the NLB Group. To this end we use large language models, focusing on two approaches: text extraction and text generation. In addition to this, a relevant context (one containing an answer) has to be passed into the models. To select one we employ a dense passage retrieval model together with a relevant document store. In order to find the optimal method, we fine-tune all of the components on the available data. The achieved results provide a good baseline, for further improvements and developments, using additional expert data and larger models.

Keywords

question answering, large language models, financial 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 productivity. To this end, we adapt existing models and approaches for open domain question answering (QA) to our specific domain - banking. Using various techniques, we implement and evaluate different question-answering pipelines, by fine-tuning open-source models with domain-specific data that has been provided to us.

In this paper, we first outline the state-of-the-art datasets and models for the domain of question answering. We then present the data, models, and evaluation procedures used, giving an outline of the structure and behavior of the final model. Next, we show both the quantitative and qualitative results and comment on them briefly. Finally, we discuss what these results mean, how we could improve them, and what the conclusions of our experiments are.

Related Work

Datasets

The most widely used dataset for open-domain question answering is Stanford Question and Answering Dataset (SQuAD) [1, 2]. Originally containing over 100k questions with corresponding passages (contexts) and answers, it was later extended to also include over 50k questions without an answer, with the intention of improving models by ensuring more accurate learning when a context does not contain an answer. TriviaQA [3] and WikiQA [4] are also general question-answering datasets, that use the same data structure as SQuAD. We use that same data structure for our own dataset. HotpotQA [5] is a dataset that aims to mitigate some of the drawbacks of extractive models and the SuperGLUE [6] benchmark contains datasets for different language understanding tasks including question answering. It formulates question-answering tasks 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 approach is to adapt the BERT [7] model on datasets such as SQuAD. Generative models such as GPT [8], T5 [9], LLaMA [10], and BLOOM [11], can also be trained to generate an answer to a question, either from a provided context or without a context whatsoever.

To obtain contexts from a larger text database from which to extract/generate an answer, many systems use Dense Passage Retrieval (DPR) [12]. It uses question and context encoders and returns a paragraph with the highest relevance to the answer.

Methodology

Data

The data available to us were the annual sustainability reports of the NLB banking group. These reports are publicly available and contain data about the development and working culture of the bank. To implement a QA model, question-context-answer pairs have to be extracted from selected sources, and formatted to suit the already pre-trained models. To achieve this, we used the pre-built QA generation pipeline from Haystack [13]. This data was filtered by NLB banking experts, and post-processed to ensure standard formatting. This process returned 185 questions from the 2020 yearly report and 355 from the 2022 report. Later on, we received 62 expert question-context-answer pairs of higher quality. All of these datasets are randomly split into train and test sets, using a 70-30 split.

In addition, we use a prebuilt script (provided by Haystack) to transform the question-context-answer pairs from the SQuAD to the DPR format, which is used to fine-tune the DPR model.

Question answering pipeline

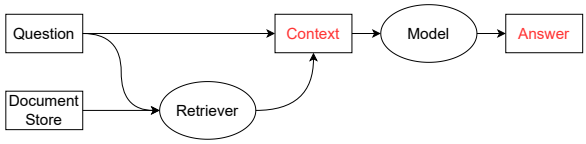

Figure 1. The diagram of our pipeline. Red denotes the output of the previous component.

The outline of our pipeline can be observed in Figure 1 and its components are briefly described below. All used models are part of the HuggingFace library [14].

We considered two approaches: extractive and generative. In the extractive approach, models try to find the answer in a given context. This approach is reliable for more straightforward 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. We explored two variants of BERT, called DistilBERT [15] and RoBERTa [16], which have already been pre-trained on the SQuAD dataset.

Generative models learn to generate a sequence of words from the input query (and context if provided). The benefit of these models is that they are able to answer more complex questions since they are not extracting the answer from the context but generating one based on language understanding and the provided context. For this approach, we use the small and base version of the T5 [9] model which has been fine-tuned for question-answering on the SQuAD dataset.

Since both the extractive and generative models require contexts our pipeline has to provide one containing the answer. A sliding window passing through all possible contexts would be computationally inefficient. Therefore, we employ a context retriever model which identifies a small number of relevant contexts to the given question. We employ the DPR [12] model which was shown to outperform traditional methods such as TF-IDF [17] or BM25 [18].

To leverage the additional information contained in the provided data, we fine-tune all the components on the available data. We test multiple combinations of the components, choosing between the extractive or generative model and different data used to fine-tune those models.

Evaluation metrics

For evaluation, we consider several metrics. Since SQuAD has its own evaluation benchmark, this is our first choice. The second metric, BLEU [19], is a benchmark for evaluating translation models. Here we use it to match the returned answer with the ground truth (GT). Lastly, we use Bertscore [20], which returns the precision, recall, and F1 score between the returned answer and the GT.

Results

Model performance

Table 1. Extractive model performance. *Fine* refers to the fine-tuned model.

	Model	Bertscore Precision	Bertscore Recall	Bertscore F1	Bleu	SQuAD Exact	SQuAD F1
20/22	distil	85.94	92.17	87.49	9.22	37.32	48.68
	distil-fine	88.13	92.54	89.52	11.69	38.73	50.50
	roberta	70.46	91.12	71.78	26.46	38.03	46.47
	roberta-fine	**89.07**	**93.64**	**90.76**	**32.09**	**47.18**	**59.32**
expert	distil	**87.71**	86.50	87.04	15.36	**15.79**	37.71
	distil-fine	87.68	**88.28**	**87.90**	37.81	10.53	45.52
	roberta	56.82	82.55	57.13	0.53	5.26	22.14
	roberta-fine	75.83	87.73	79.33	**47.66**	**15.79**	**47.09**

Table 2. Generative model performance. *Fine* refers to the fine-tuned model.

	Model	Bertscore Precision	Bertscore Recall	Bertscore F1	Bleu
20/22	t5-small	95.07	95.54	95.28	55.46
	t5-small-fine	97.03	98.16	97.57	54.31
	t5-base	88.29	90.95	89.55	12.71
	t5-base-fine	**98.07**	**98.44**	**98.25**	**68.27**
expert	t5-small	89.48	86.39	87.87	6.74
	t5-small-fine	92.26	88.93	90.52	22.11
	t5-base	83.70	84.70	84.13	12.81
	t5-base-fine	**93.21**	**92.25**	**92.68**	**52.30**

In Table 1 and Table 2 we can observe the results obtained by the extractive and generative models before and after additional fine-tuning. We fine-tuned the models on the train splits of automatically generated data from 2020, 2022, and combined 20/22, as well as on the expert data.

We observe that fine-tuning a model results in better metrics in most cases. On the automatically generated data, we get good results without additional fine-tuning and observe a similar increase after fine-tuning, regardless of the dataset. When evaluating on only a specific year, rather than the combined

data, we do see better performance for the respectfully fine-tuned models. However, we deem the combined data more important and thus exclude others from the tables for the sake of brevity.

The biggest difference in performance is observed with the expert data, where the increase is large, especially for the BLEU score. Many models don't gain a significant increase in performance after fine-tuning. This was expected since the automatically generated data resembles the data in SQuAD much more than the expert data. For this reason, the information gained was smaller in contrast to fine-tuning on the expert data, which is more distinct.

Using the automatically generated data we tried fine-tuning the models on only half of the train set to see if training on less data impacts the performance significantly. Albeit worse, the metrics, for models fine-tuned this way, are close enough to question whether the additional work required (filtering all of the automatically generated data) is worth the slight increase in performance.

Table 3. Mean and standard deviation scores for the T5-small model over 10 different train/test splits.

		Bertscore			Bleu
		Precision	Recall	F1	
20/22	mean	96.98	97.78	97.35	57.73
	std	0.35	0.34	0.33	3.25
expert	mean	91.79	88.48	90.06	19.65
	std	1.17	1.16	1.22	7.04

We decided to check the uncertainty of the dataset by training the same model 10 times with a different 70-30 train/test split on both the combined and expert datasets. The results of this can be seen in Table 3. We expected the standard deviation to be higher for the expert data set as there are not a lot of questions and they differ quite substantially, and this is indeed the case. Meanwhile, the standard deviation for the generated data is lower, due to the fact that we have a lot more (similar) data in that set.

Pipeline performance

Table 4. Pipeline performance using fine-tuned models.

			Bertscore			Bleu	SQuAD	
		Model	Precision	Recall	F1		Exact match	F1
20/22	DPR	distil-fine	91.75	91.41	91.57	12.22	23.94	27.64
		roberta-fine	**92.85**	**92.84**	**92.83**	**29.09**	**35.21**	**37.89**
		t5-small-fine	92.18	92.50	92.33	15.24	/	/
		t5-base-fine	91.64	92.02	91.81	16.79	/	/
	DPR-fine	distil-fine	91.88	91.61	91.73	15.18	27.46	31.70
		roberta-fine	**93.18**	**93.03**	**93.09**	**33.23**	**36.62**	**39.93**
		t5-small-fine	91.94	92.68	92.04	14.53	/	/
		t5-base-fine	92.06	92.49	92.25	18.28	/	/
expert	DPR	distil-fine	86.98	86.94	86.93	0.0	**4.76**	11.58
		roberta-fine	88.15	87.33	87.63	4.61	**4.76**	**21.53**
		t5-small-fine	**88.78**	87.44	**88.04**	13.48	/	/
		t5-base-fine	86.11	**88.20**	87.06	**24.60**	/	/
	DPR-fine	distil-fine	87.41	87.19	87.25	0.0	4.76	16.87
		roberta-fine	**88.78**	**87.75**	**88.21**	5.10	**14.29**	**27.57**
		t5-small-fine	86.92	85.23	86.03	0.88	/	/
		t5-base-fine	85.45	87.42	86.38	**18.75**	/	/

In Table 4 we can observe the results obtained using fine-tuned models in combination with a DPR that has or hasn't been fine-tuned (on automatically generated data). The DPR retrieves the contexts from both the reports for the combined 20/22 data or only from the 2022 report for the expert data. We can see that fine-tuning the DPR has a positive effect as a higher number of exact matches are achieved (alongside an increase in other metrics). This means that the DPR managed to retrieve more relevant contexts and the models can provide better answers.

Qualitative analysis

Table 5. Qualitative analysis of the QA pipeline on combined and expert data

		Correctness				Sensibility				"None"
	Model	1	2	3	avg [%]	1	2	3	avg [%]	
20/22	distilbert	13	1	7	**57**	7	5	9	**70**	0
	roberta	13	1	7	**57**	7	5	9	**70**	0
	t5 small	13	1	7	**57**	7	6	8	68.3	0
	t5 base	13	1	7	**57**	7	5	9	**70**	0
expert	distilbert	17	3	1	**41.3**	8	6	7	**65**	0
	roberta	18	3	0	38	11	4	6	58.7	0
	t5 small	16	5	0	**41.3**	12	3	6	57	7
	t5 base	17	3	1	**41.3**	14	1	6	54	0

To better gauge how well the pipeline works we look through the entire test set (21 examples) of expert questions and a subset of the automatically generated questions (21 examples that were selected arbitrarily). We score the answers based on sensibility - how closely the retrieved context matches the topic of the question, and correctness - how close is the obtained answer to the GT. We assign a number from 1 (worst) to 3 (best) for each example and mark the number of questions for which we get no answer - the model thinks it cannot answer the question from the provided context.

We can observe the results in Table 5. The results are quite poor since a lot of the time the "wrong" context is retrieved. Sensibility always scores better than correctness since the context usually contains some information that is related to the question. The answer, however, still might not be considered correct, since the returned answer doesn't match the GT. We observe that the more straightforward questions from the automatically generated data yield better results than the expert ones. This is not surprising as some of the expert questions are fairly long and complex. From the model and pipeline evaluation, we expected better performance from the generative models at least in the case of the more complex (expert) data. The scores do not indicate this without looking at the number of times that we get the answer "None". The small variant of T5 is the only model to correctly identify that the retrieved context does not contain the answer. It does this for 7 of the 21 selected examples.

In Table 6 we can see 2 examples of retrieved contexts and the obtained answers. When the context contains the GT answer we obtain the correct answer (1st example). We also noticed two situations. Either the retrieved context contains a slightly different answer than the GT, which still could be considered correct (2nd example), or the retrieved context is irrelevant and all of the QA models fail. We also include the answers obtained from a significantly larger model (ChatGPT) for com-

Table 6. Examples of retrieved contexts and obtained answers from the pipeline. An answer generated using ChatGPT is provided for comparison.

Question: How many branches does NLB Group have in all markets where it operates? **GT Answer**: 440 **Retreived context**: NLB Group in numbers (as at 31 December 2022) • An extensive network of 440 branches in all markets where NLB Group operates. • More than 2.7 million active customers. • With successful operations, NLB Group generated a record high profit of €446.9 million in 2022, with the acquisition of N Banka in March significantly influencing the full-year business results of the Group. • The balance sheet total of the NLB Group amounts to €24.2 billion.
Answers | distilbert: "440" | roberta: "440" | t5-small: "440" | t5-base: "440"
ChatGPT Answer: Based on the provided context, the number of branches NLB Group has in all markets where it operates is 440.
Question: How much did the bank reduce the use of paper in 2022 compared to the previous year? **GT Answer**: 19% **Retreived context**: In 2022, another step to achieving this goal has been made as paper usage (prints) was reduced by 17% compared to 2021. Thus, we saved 866 trees. Paper usage in NLB was lower by 26%, which is the highest reduction group – wide. All banking subsidiaries experienced reduction as well, ranging from 8% to 19%, respectively.
Answers | distilbert: "17%" | roberta: "17%" | t5-small: "17%" | t5-base: "17%"
ChatGPT Answer: Based on the given context, the bank reduced the use of paper by 17% compared to the previous year (2021). However, the exact amount of paper usage in 2021 or any specific quantity of paper used is not mentioned.

parison. As expected the use of a larger model yields much better results. However, in the case of the second example, this one also "fails".

Discussion

We consider two important factors when drawing conclusions from obtained results. These factors are the data and the models as well as the pipeline. We believe that each of these two factors plays an important role and making alterations to each of them could improve the performance.

First let's consider the data. We were somewhat limited by the data available to us. We haven't mentioned thus far that over 4000 question-context-answer pairs were generated. These were filtered by the NLB banking professionals, to only include those that would be useful to them. The remaining 540 pairs, together with the expert examples, provide a good starting point to fine-tune the models. However, better results could be achieved with more data. This especially seems to be the case with the DPR model. It was noticed during qualitative analysis that the pipeline is bottle-necked by its first component, and additional fine-tuning could alleviate this. Better results could be achieved by using more expert data. As stated previously, the improvements seen when fine-tuning using expert data are the largest. Having more data of this type could increase the performance beyond the results reported here. This wouldn't only allow us to answer more complex questions, but also achieve better results with smaller models. Better results could be possible, if we focused on a specific type of questions, for example, yes/no or multiple choice questions.

As already mentioned, to improve the pipeline we need to improve the DPR. Another possible method to achieve this would be to modify the pipeline to use more than one context from the retriever to obtain one that actually has the answer. On the other hand, the pipeline could also be improved by trying different models, e.g. ones with a longer context. Implementing some Long Form Question-Answering (LFQA) [21] models could possibly yield better results, as these can both accept longer questions and process much longer contexts. Another improvement would be to use larger models. As we noticed, ChatGPT outperforms our fine-tuned models, thus we can expect that using it, LLaMA, LaMDA [22] or BART [23] in our pipeline would yield better results. This would, however, require more computational power.

During the qualitative analysis we observe discrepancies with the quantitative results. The first reason is that the quantitative metrics can fail in some cases. For example, it has been studied that Bertscore fails when evaluating wrong answers with high lexical overlap [24]. The second reason is that qualitative analysis is in itself a biased and subjective process. We assign much higher scores even if only one part of the pipeline works well which can't be easily included when obtaining the quantitative results.

Despite this we believe that the models do have their usefulness. The results show that they are good at answering more straightforward answers, most commonly those with a numeric answer. However, we don't deem these models production ready. We assume the models would require further fine-tuning with additional data or we could adapt larger models. With the currently available results, we conclude that the best cost-to-performance model is the small variant of T5. While the base variant and RoBERTa outperformed it in some metrics, it is the only model that correctly identifies when the answer isn't contained in the context.

Acknowledgments

We would like to thank our faculty advisor prof. dr. Marko Robnik Šikonja for providing irreplaceable guidance and advice. We would also like to thank dr. Branislava Sandrih Todorović and the NLB group for their cooperation, and our industry advisor Grega Jerkič for helping set the correct path and foundations of our work.

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] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[15] 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.

[16] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[17] Amir Jalilifard, Vinicius Fernandes Caridá, Alex Fernandes Mansano, Rogers S Cristo, and Felipe Penhorate Carvalho da Fonseca. Semantic sensitive tf-idf to determine word relevance in documents. In *Advances in Computing and Network Communications: Proceedings of CoCoNet 2020, Volume 2*, pages 327–337. Springer, 2021.

[18] Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.

[19] 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.

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

[21] Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. Eli5: Long form question answering. *arXiv preprint arXiv:1907.09190*, 2019.

[22] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.

[23] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

[24] Michael Hanna and Ondřej Bojar. A fine-grained analysis of bertscore. In *Proceedings of the Sixth Conference on Machine Translation*, pages 507–517, 2021.