Extractive Long Form Question-Answering for Annual Reports using BERT

Anusha Kabber[1], V M Dhruthi[2], Raghav Pandit[3], and S Natarajan[4]

[1] Department of Computer Science and Engineering, PES University, Bangalore, India
anushakabber2001@gmail.com,
[2] Department of Computer Science and Engineering, PES University, Bangalore, India
dhruthi25vm@gmail.com,
[3] Department of Computer Science and Engineering, PES University, Bangalore, India
raghavvpandit81@gmail.com,
[4] Department of Computer Science and Engineering, PES University, Bangalore, India
natarajan@pes.edu

Abstract. This paper suggests a strategy to obtain extractive long form answers from Annual Reports. Perceiving most previous usage, BERT has been used to get short answers with either a phrase or a sentence. With annual reports that is simply not sufficient and most queries require a more well-rounded answer. In this paper, TF-IDF, FinBERT and BERT are used to build extractive long answers from annual reports - which is a very large context. The BERT model that is used in the pipeline is pre-trained on unsupervised textual data from annual reports and fine-tuned on FiQA.

Keywords: Annual Reports · Extractive Long Form Question Answering · Financial Domain Question Answering · TF-IDF · Unsupervised Pre-training on BERT · FinBERT · BERT · FiQA.

1 Introduction

Annual Reports are documents that companies provide shareholders as an insight into their financial and corporate health. They paint a picture of the company's activities and progress in the past year and provide forecasts to aid interested parties to make decisions on investments. They also hold SDG reports of the company that offer a wealth of information on how progressive a company is with respect to human resources and environmental awareness. Annual reports usually contain the company's goal or vision and how they plan to achieve it. The details of cash flow are also mentioned. All these details are wrapped in large amounts of text that is difficult to sift through. They are extremely lengthy documents of up to 200 to 300 pages. This signals the need for a framework that can extract the right explanatory paragraph or data given a prompt or query. This is the need we address in this paper with the following strategy.

The annual report in pdf form is fed into pdfminer which converts the documents into text. This text is then split into sentences and the sentences with maximum similarity to the question are obtained. Finbert Embeddings are used to rerank them and the next top few answers are sent into pre-trained, fine-tuned BERT to process and rerank one last time. The top candidates are then merged into an answer.

2 RELATED WORK

2.1 BERT

For Question Answering, context and a deeper understanding of text is vitally important. Prior to BERT[1], models like ELMo[2] and the original Open AI GPT[3] used unidirectional language processing. This limited the performance for downstream sentence-level tasks dependent on tokenization. BERT overcomes this by being primarily a deep bidirectional Transformer. The authors use two main objectives to train the transformer, ”Masked Language Modelling” or ”MLM”[4] and ”Next Sentence Prediction” or ”NSP”[5]. This model counters the need for task-specific models for downstream tasks like text classification, question answering and sentiment analysis.

BERT was pretrained using the BooksCorpus and Wikipedia. It was necessary to choose data sets of continuous, semantically meaningful text due to the nature of BERT's bidirectional context understanding. BERT can be fine-tuned relatively easily because of its self attention mechanism. BERT's GLUE[6] test results show that it outperforms its peers significantly. On average, taking all GLUE tasks into consideration, BERT(large) has a 8.1% advantage over Pre-OpenAI SOTA[7], 7% increase on OpenAI GPT and a whopping 11.1% increase over BiLSTM+ELMo+Attn[8]. BERT exhibits the importance of bidirectional context understanding and deep, unsupervised pre-training to better improve the performance of transformers on a number of downstream tasks.

2.2 FinBERT

The issue with financial domain and NLP tasks is that there is very limited availability of annotated data sets or labelled data. Neural Nets and Language models usually require a vast amount of data to perform adequately. The authors aim to build a financial sentiment analysis model that doesn't require huge initial input. They explore the effects on pre-training BERT on a relatively large financial corpus versus training BERT on only the sentences to be classified. They also record the performance of ULMFit[9], LSTM[10], LSTM with ELMO[11], the machine learning models LPS[12], HSC[13] and FinBERT[14] on the Financial PhraseBank Dataset. Their results show that ULMFit outperforms the machine learning models due to powerful pre-training strategies.

They also show that despite ULMFit performing better than most models, FinBERT greatly outperforms all the other models. The gains in F1 scores on other models range from 7% for FinSSLX[15] to 21% for LSTMs. FinBERT also requires a significantly lesser number of samples compared to its peers for the same F1 scores. Vanilla BERT is also compared with FinBERT and FinBERT achieves a slight improvement on the test dataset. They also show that all 12 layers achieve the best performance while classifying financial data, while the sixth layer contributes most to model performance. FinBERT shows that deep learning models that usually perform the best but require an exceptionally large amount of data might not be the only option available for domain-specific NLP tasks. The authors then conclude with the suggestion of extending FinBERT to more downstream NLP tasks like question answering and entity recognition.

2.3 ELI5 Long Form Question Answering

Questions on Annual Reports more often that not require long passage answers. The current scene with question answering is heavily focused on short answers. The field has improved greatly with significant strides in extractive, fact-based question answering, with datasets like SQuAD[16], TriviaQA[17], Adversarial QA[18], MS MARCO[19], CoQA[20], Quasar[21] all available for training models to obtain accurate short answers. There is a distinct lack of long form question answering datasets. One of the most popular ones is ELI5[22] or ”Explain to me like I’m 5”.

The authors have built a dataset by collecting responses from an online reddit forum. The dataset has numerous diverse questions requiring long answers. The authors also show the performance of multiple models for extractive as well as abstractive question answering. They’ve explored the use of BiDAF[23] for extractive question answering given a sub-sample of a document. For abstractive question answering they’ve looked at Seq2Seq multi-task, Seq2Seq Q + D to A[24] and others. Seq2Seq multi-task outperforms the others suggesting that multi-task objective aids the performance Seq2Seq models.

2.4 BERTSerini

The authors show an end-to-end question answering framework that coordinates BERT with the open-source Anserini IR tool kit. As opposed to most question answering and perusing cognizance models today, which work over limited quantities of information text, their framework incorporates best practices from IR with a BERT-based peruser to recognize replies from a huge corpus of Wikipedia articles in a start to finish design. They report huge enhancements over past outcomes on a standard benchmark test assortment, showing that adjusting pretrained BERT with SQuAD is adequate to accomplish high precision in recognizing answer spans.

To conclude they present BERTserini[25], their end-to-end open domain question addressing framework that incorporates BERT and the Anserini IR toolbox. With a basic two-stage pipeline engineering, they are capable of accomplishing enormous upgrades over past frameworks. Error analysis focuses on opportunities to get better in IR, answer extraction, and reply collection—all of which address progressing endeavors. Furthermore, they are additionally keen on extending the multilingual abilities of their framework.

3 DATASET

Annual Reports from various companies regardless of domain but limited to Indian Annual Reports from 2020 were selected. The annual reports were in the form of .pdf. Text was extracted and then pre-processed to get sentences and paragraphs. The result was more than 20 thousand lines of text.

The question answering dataset used was FiQA. It offers opinion-based question answers over financial data. The dataset provides question answer pairs without a context. There are approximately 6000 questions available for training in this dataset.

4 METHOD

The method proposed in this paper aims to provide long answers to open ended questions in the financial domain from a relatively large corpus of text that is an annual report with a moderate time improvement compared to some strategies. After trial and error, this method was chosen as the best performing in terms of accuracy, semantic relationship and time taken.

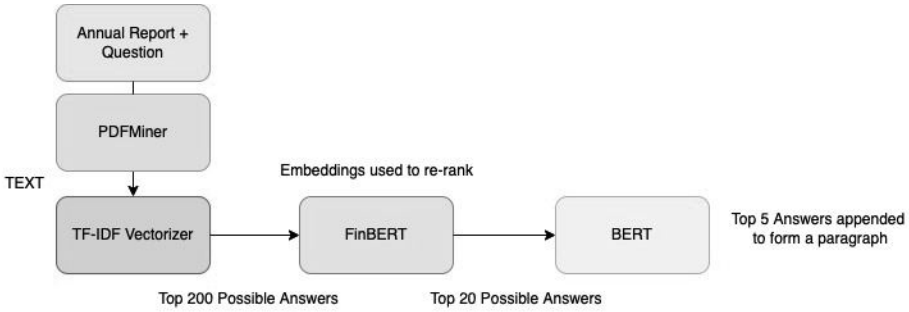

Fig. 1. A Diagramatic representation of the strategy used in this paper.

4.1 Extraction

Annual reports are very text heavy with a lot of tabular and graphical data interrupting text extraction. They also generally come in the form of .pdf and can have numerous embellishments to make the document more pleasing to the human eye. A tool that could handle all these issues was vital to the overall strategy.

Extraction was done using a python tool called PDFminer. PDFMiner is a tool for extracting data from PDF archives. Not at all like other PDF-related tools, it centers totally around getting and examining text information. PDFMiner permits one to get the specific area of text in a page,in addition to other data such as font and lines. It incorporates a PDF converter that can change PDF documents into other formats of text (like HTML). It has an extensible PDF parser that can be utilized for different purposes than text analysis. It is written entirely in Python. (for version 2.4 or newer)and is used for parsing, analyzing, and converting PDF documents.

Initially, a few methods were implemented like Google's Cloud Vision API, PyPDF2, pdftotext and microsoft's Azure AI for document processing. PDFminer was selected due to its high accuracy and ability to extract text with color, text against a colorful background and its overall speed and ease with which it can be incorporated with other components in this method.

4.2 TF-IDF

On average annual reports when divided can easily result in thousands of paragraphs of text and double the number of sentences. This is a huge amount of context for a BERT model to analyse and extract an answer from. To narrow the search space, an information retrieval tool is required. Anserini was first considered but it was found that a basic TF-IDF implementation gives almost the same results in a fraction of the time. Word2vec was also trained on around 2 lakh sentences from annual reports but did not perform well.

TF-IDF stands for "Term Frequency – Inverse Document Frequency" and is a technique to quantify the words in a set of documents. This is used to signify the importance of each word in a document by computing a score for each word. It is a popular approach used to weigh terms for NLP tasks as it assigns a value to each word according to its importance in a document scaled by its importance across all documents in the corpus, which eliminates mathematically naturally occurring words in English.

NLTK tokenizer was used to obtain sentences from paragraphs. Sentences from the converted pdf are passed to TF-IDF along with the question. A similarity matrix based on cosine is built and the top 200 sentences most similar to the question are passed to the next component in the pipeline.

4.3 FinBERT Embeddings

As mentioned in the section 'Related Works', a model for financial sentiment analysis was built. The sentence/ token-level embeddings from the model can be used to perform a variety of downstream NLP tasks. We use the embeddings to re-rank sentences by cosine distance from the question. This is done because BERT has a much greater depth of semantic understanding of language than TF-IDF can hope to achieve. FinBERT is used because it has been trained on financial data and thus is familiarised with the context of words specific to the financial domain. It narrows the search space further with only 20 possible answer sentences.

4.4 Pre-Trained and Fine-Tuned BERT

Literature suggests pre-training BERT on domain-specific data improves performance on downstream NLP tasks. BERT-BASE(uncased) which is a 12-layer, 768-hidden, 12-heads, 110M parameters model was used for this implementation.

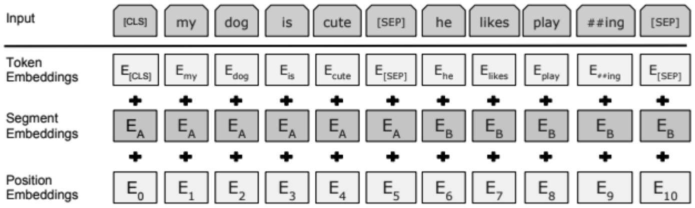

Fig. 2. The initial embeddings is the sum of the position, the segmentation and token embeddings

BERT was initially pre-trained on 10,000 lines of text for 3 epochs. We fixed the batch-size to be 32, max sequence length as 128 and learning rate to be 2e-5. To be noted, the default optimizer for BERT is Adam. BERT was trained from a fixed checkpoint, it was not pre-trained from scratch. The author suggests a 2e-5 learning rate works best when using a checkpoint. A GPU, Nvidia K80/T4 with 0.82Ghz/1.59Ghz was used. The results for masked language model accuracy was fairly low along with the Next Sentence Prediction accuracy. The time taken to train was approximately 15 minutes. The model size after pre-training was 1.22 GB. The input data was increased to 20,000 with the same parameters. It gave the following results given below.

Table 1. MLM and NSP Accuracy for Pre-trained BERT

Input Size	MLM Accuracy	NSP Accuracy
10000	53.96%	84.62%
20000	57.41%	90.12%

The model trained on 20000 sentences from annual reports was chosen to further fine-tune.

The pre-trained BERT model was fine-tuned on the FiQA data set. 2500 Question Answer pairs from FiQA were used. We tested out two fine-tuned versions of BERT - without pre-training on annual report data and with pre-training. We found that it performed slightly better on the FiQA dataset with pre-training. In the below table you can see the probability score for a question and answer pair after fine-tuning the two versions.

Table 2. Probability score for a QA pair with and without Pre-training BERT

Q+A	With Pre-training	Without Pre-training
1	0.999	0.922

Returning to our pipeline, the 20 possible answer pairs are then given to pre-trained, fine-tuned BERT along with the question. We re-rank the answers based on the question and answer scores again since none of the previous tools are trained on financial question answering or even general question answering specifically. The top 5 sentences are appended to one another to form a paragraph that is a long form answer that descriptively answers the question asked.

We also attempted to use deepset's roberta-base-squad2 from HuggingFace, replacing the pre-trained, fine-tuned BERT from our pipeline. Owing to its large training data set it performed really well on a lot of questions, one of which is given below. The context for the question is the entire document of TCS's 2020-21 Annual Report.

Table 3. QA pair obtained by using the pipeline described in the paper and replacing our Pre-trained, Fine-tuned BERT with roberta-base-squad2

Question	Answer
What is the brand statement of TCS?	TCS adopted a new brand statement Building on Belief to convey how its partnership with customers goes beyond technology deployment. What does your new brand statement 'Building on Belief' mean to you and to customers? To better articulate its mission and its aspirations your company adopted a new brand statement this year 'Building on Belief'. TCS Research marked its 40th year by adopting a new brand identity with the tagline 'Inventing for Impact'.TCS is committed to using zero- ozone depleting potential (ODP) refrigerants in its operations.

BERT fine-tuned on a general or a financial QA data set would take very long to process each paragraph of an annual report and locate potential answer statements. The pipeline as it currently stands takes around 2-3 minutes to complete execution. Thus, this pipeline is required.

5 Results and Evaluation

For evaluation, F1 scores where chosen since the scoring was most appropriate to the situation. F1 scores primarily measure how many shared words exist between the predicted answer and the true answer for question answering. F1 is a function of recall and precision.

$$F_1 = \frac{2}{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}} = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} = \frac{\mathrm{tp}}{\mathrm{tp} + \frac{1}{2}(\mathrm{fp} + \mathrm{fn})}.$$

Fig. 3. F1 Score

Here, recall score would be the ratio of the number of shared words to total word in true answer. Precision would be measured as the ratio of the shared words to the number of words in the predicted answer. The F1 score we've achieved on the FiQA dataset is 0.426. Another measure we've used is cosine distance using FinBERT embeddings. Comparison of the true answer to the question versus the predicted answer cosine similarity to the question is given below.

Table 4. Cosine Similarity of True answer and Predicted answer with question

True Answer	Predicted Answer
0.7426	0.7335

6 Conclusion

This is the strategy we propose for extractive long form question answering using BERT for financial domain. It executes in relatively shorter lengths of time and returns with sentences ranked according to the order in which they answer the question as well as how similar they are to the question on a financial basis. We begin our pipeline with extracting text from pdfs using PDFMiner, and then use TF-IDF to pick out 200 sentences similar to the question. FinBERT embeddings are then used to measure the cosine similarity between the sentences and the question. The top 20 are selected. BERT then re-ranks them to most suitable answers to the question. We use a BERT pre-trained on unlabeled text from annual reports and fine-tuned in question-answering using FiQA.Investors and shareholders require tools to streamline the process of analysis of annual reports.

A question answering framework that allows professionals to handle this is very much in demand.

This implementation can be improved and extended in multiple ways. It currently used two BERT models which is very hard on systems' memory as they are around 1GB each. The framework hasn't been deployed for use, the implementation only runs locally. Evaluation is not highly empirical and the system can be tested on more evaluation metrics. The model hasn't yet been trained on larger amounts of data. Only BERT has been experimented with, newer models like T5[26] and other forms of BERT haven't been tested out. Collection of questions from open-source can be done to expand the current availability of questions on annual reports.

References

1. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova Google AI Language, 2019
2. Deep contextualized word representations, Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Zettlemoyer, arXiv, cs.CL, 2018
3. Improving Language Understanding by Generative Pre-Training,Radford, Alec and Narasimhan, Karthik and Salimans, Tim and Sutskever, Ilya, 2018
4. Adaptation of Deep Bidirectional Multilingual Transformers for Russian Language Yuri Kuratov, Mikhail Arkhipov arXiv cs.CL, 2019
5. NSP-BERT: A Prompt-based Zero-Shot Learner Through an Original Pre-training Task–Next Sentence Prediction Yi Sun, Yu Zheng, Chao Hao, Hangping Qiu arXiv cs.CL, 2021
6. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding, Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, Samuel R. Bowman, arXiv, cs.CL, 2018
7. BERT and PALs: Projected Attention Layers for Efficient Adaptation in Multi-Task Learning Asa Cooper Stickland, Iain Murray Proceedings of the 36th International Conference on Machine Learning, PMLR 97:5986-5995, 2019
8. BiLSTM with Multi-Polarity Orthogonal Attention for Implicit Sentiment Analysis Jiyao Wei, Jian Liao Zhenfei Yang Suge Wang QiangZhaoa Neurocomputing Volume 383, 2020, Pages 165-173
9. Low Resource Text Classification with ULMFit and Backtranslation, Sam Shleifer, Stanford University, arXiv, cs.CL, Mar 2019
10. Bidirectional LSTM-CRF Models for Sequence Tagging Zhiheng Huang, Wei Xu, Kai Yu arXiv cs.CL, 2015
11. Neural Architectures for Nested NER through Linearization Jana Straková, Milan Straka, Jan Hajič arXiv cs.CL, 2019
12. Machine learning-based longitudinal phase space prediction of particle accelerators C. Emma, A. Edelen, M.J. Hogan, B. O'Shea, G. White, and V. Yakimenko Phys. Rev. Accel. Beams 21, 2018
13. Photometric classification of HSC transients using machine learning Ichiro Takahashi, Nao Suzuki, Naoki Yasuda, Akisato Kimura, Naonori Ueda, Masaomi Tanaka, Nozomu Tominaga, Naoki Yoshida PASJ, 2020

14. FinBERT: Financial Sentiment Analysis with Pre-trained Language Models Dogu Tan Araci dogu.araci@student.uva.nl University of Amsterdam Amsterdam, The Netherlands, arXiv, cs.CL, 2019
15. FinSSLx: M. Maia, A. Freitas and S. Handschuh, "FinSSLx: A Sentiment Analysis Model for the Financial Domain Using Text Simplification," 2018 IEEE 12th International Conference on Semantic Computing (ICSC), 2018, pp. 318-319
16. SQuAD: 100,000+ Questions for Machine Comprehension of Text, Pranav Rajpurkar and Jian Zhang and Konstantin Lopyrev and Percy Liang Computer Science Department Stanford University, 2017
17. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension Mandar Joshi† Eunsol Choi† Daniel S. Weld† Luke Zettlemoyer†‡ † Paul G. Allen School of Computer Science Engineering, Univ. of Washington, Seattle, WA Allen Institute for Artificial Intelligence, Seattle, WA, arXiv, cs.CL, 2017
18. Model Agnostic Answer Reranking System for Adversarial Question Answering, Sagnik Majumder, Chinmoy Samant Greg Durrett Department of Computer Science The University of Texas at Austin, arXiv, cs.CL, 2021
19. MS MARCO: A Human Generated MAchine Reading Comprehension Dataset Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, Li Deng, CoCo@ NIPS, 2016
20. Question Classification of CoQA (QCoC) Dataset Abbas Saliimi Lokman; Mohamed Ariff Ameedeen; Ngahzaifa Ab. Ghani Publisher: IEEE Aug. 2021
21. QUASAR: DATASETS FOR QUESTION ANSWERING BY SEARCH AND READING Bhuwan Dhingra Kathryn Mazaitis William W. Cohen School of Computer Science Carnegie Mellon University, 2017
22. ELI5: Long Form Question Answering Angela Fan1, Yacine Jernite Ethan Perez David Grangier Jason Weston1 Michael Auli Facebook AI Research LORIA NYU Google AI, 2019
23. W. Zhang and F. Ren, "ELMo+Gated Self-attention Network Based on BiDAF for Machine Reading Comprehension," 2020 IEEE 11th International Conference on Software Engineering and Service Science (ICSESS), 2020, pp. 1-6, doi: 10.1109/ICSESS49938.2020.9237663.
24. Q. Qi, X. Wang, H. Sun, J. Wang, X. Liang and J. Liao, "A Novel Multi-Task Learning Framework for Semi-Supervised Semantic Parsing," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 28, pp. 2552-2560, 2020, doi: 10.1109/TASLP.2020.3018233.
25. End-to-End Open-Domain Question Answering with BERTserini Wei Yang,Yuqing Xie,Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin1, David R. Cheriton School of Computer Science, University of Waterloo, arXiv, cs.CL, 2019
26. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer, Colin Raffel, Noam Shazeer, Adam Roberts,Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J. Liu, 2020
27. Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing,ACM Transactions on Computing for Healthcare, 2021
28. A Robustly Optimized BERT Pre-training Approach with Post-training, Zhuang Liu, Wayne LinYa, ShiJun Zhao, CCL, 2021