A Look at Question Answering Problem - as an Embedding based Retrieval

Jagan Lakshmipathy
University of California, Berekeley
jagannathan@berkeley.edu

Sweta Bhattacharya
University of California, Berekeley
swetabee@berkeley.edu

Abstract

We attempt to solve a Question Answering (QA) problem by treating it as a information retrieval problem. We derived inspiration for this work from Overveiw QA pipeline (OQAP) described in Bithiah Yuan [4] thesis work. In this thesis, to select relevant answers for a given question from a larger answer set the author employs a two step process that used non-factoid answer selection using inverted index retrieval scheme to gather a set of candidates. And subsequently, from this answer set the author used pretrained transformers to pick most relevant answers. Though this approach is impressive and produced some interesting results, the non-factoid selection require some word matches for an answer to qualify for a second level selection. As a result this approach will select only answers that have some common words in both the question and the answer. In this paper we will contrast the following approaches: (1) a pure informational retrieval (IR) based approach that uses TF-IDF (our baseline model) [6], (2) Yuan's OQAP using BERT transformer [4], (3) A simple non-transformer based Two-tower model (TOM I), and (4) An advanced transformer based Two Tower Model (TOM II). As the two tower models are based out of text embeddings and its similarity, we hope it will pick answers that have similar semantics with the question even if they don't share common words among them. Finally, we will compare our results with rank aware metrics [8].

1 Introduction

In recent years, QA systems have become very prevalent in assisting customer facing or frontline workforce of large institutions to answer or advise their clients. In order to answer questions, provide prompt and valuable guidance, and make suggestions to their clienteles', customer support personnel have a need to search through the knowledge base for relevant answers. The knowledge base could include some textual passages, letters, notes, multi-page documents, FAQs, etc. However, retrieving relevant materials from this knowledge base or information troves in real-time or near real-time does pose a serious challenge. As the experience levels of the customer support personnel could not only result in inconsistent interpretations of the information but also be difficult to keep this information at their fingertips. Moreover, the large magnitude of detailed and technically written incoming reports are infeasible to read through even for the best and most experienced client facing employees. As a result, the application of QA in any domain is critical due to the highly competitive and profitable nature of the industry. Since QA research targeting a dataset or a domain that is fairly under explored can result in large profits and competitive advantages with even a slight improvement in the process. With recent advancements in NLP and Deep Learning approaches and a need for a novel QA system has reinforced our motivation and has culminated in this research.

2 Related Work

This research is not a first of its kind. We surveyed the literature and we found some very interesting related papers. Ansari et.al. [10] used conventional neural networks to understand the contents of the documents. He divided the sentences into knowledge units and assign deep case to each word to improve the quality of knowledge extraction. Chali et.al. [9] proposed a graph-based random walk method to compute the relative importance of textual units. Abdi et.al. [11] developed an ontology-based domain specific QA system using natural language processing. Suman et. al. [5] provided a simplified approach and used SQuAD dataset to demonstrate their approach. Zhuang Liu et. al. [2] presented a domain specific language model pre-

trained on large-scale financial corpora that enabled the capture of language knowledge and semantic information. Boyer et.al. [3] proposed financial domain QA systems to retrieve the top descriptive text passage answers from a corpus of financial reports given a question. Boyer first developed a Naive Bayes binary question classifier to determine if a question is a financial outlook or informational question based on financial keyword counts. However, financial entities such as currency, assets, and industry occur more frequently, thus, causing domain term bias and misclassification. To address this issue, a rule-based system was developed where the domain terms were selected and replaced [3]. The outlook questions were then treated as factoid questions and the information questions as non-factoid questions. For the non-factoid questions, [5] used logistic regression operating over 80 proprietary linguistic scorers. Another feature-engineeing-based method for a general QA task proposed by [13] uses the lexical database, WordNet, to pair semantically related words and finding similarities between the questions and answers [12]. Since [3] and [13]'s QA systems are based on feature engineering and linguistic match, they could not represent domain-specific financial language [3]. Here we will examine how deep learning methods can be used to address this problem next. Bithiah Yuan focused on financial non-factoid answer selection and retrieved a set of passage-level texts and selected the most relevant as the answer.

The rest of this paper is organized as follows. Section 3 outlines 3 approaches that we experimented here in this paper. Section 4 outlines the details of the dataset. And section 5 lists rank aware metrics that we will use to compare our approaches. Section 6 outlines our experimental setup. Sections 7, 8 and 9 discusses results, future work and limitations.

3 Our work

We will try the following three approaches (1) The classical retrieval approach (our baseline) approach, (2) QA Pipeline approach (OQAP) [4], (3) A simple non-transformer based Two-tower model (TOM I), and (4) An advanced transformer based Two Tower Model (TOM II). In the following subsections we will provide details for each of the above approaches.

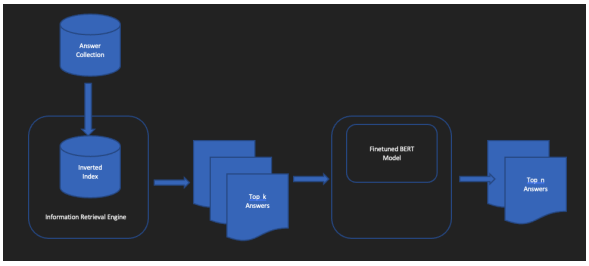

Figure 1: QA Pipeline approach (OQAP).

3.1 Baseline approach

Our baseline system is an IR based approach that uses a simple weighted inverted index system based on TF-IDF. TF-IDF computes the importance factor by computing the product of a term's frequency (TF) and the inverse document frequency (IDF). Our Answer Retriever uses Anserini's [7] a BM25 implementation to retrieve 50 candidate answers for each question. Anserini is an open-source IR toolkit built on top of Lucene (SimpleSearcher). We use the list of answers from the FiQA dataset to first build an inverted index, then we use Pyserini, a Python interface of Anserini, to generate the candidate answers.

3.2 QA Pipeline approach (OQAP)

As shown in figure 1, the inverted index retriever first returns the top k candidate answers for a given question. These selected k candidate answers are later passed through a finetuned pretrained BERT model. For each test sample in the dataset, the output of the BERT model is later passed through a feed forward softmax layer that will assign probability for each of the k answers. The top n high probability answers are output for each test question. The pretrained BERT is finetuned for each question answer pair in the training dataset in FiQA. The tokens for question and answer pair from the BERT tokenizer is fed into the BERT model. The output embeddings from the BERT is passed through a Dense and Dropout layers in that order before it is fed into a sigmoid classifier which classifies it as either 1 or 0.

3.3 Simple Two Tower Approach (TOM I)

Simple two tower architecture is shown in Fig. 2. It has separate question and answer towers. The embeddings from those towers is passed to a dot product evaluator to get the similarity measure or the relevance score. We will use the training data to maximize this similarity measure.

We leveraged the tensorflow recommender system package namely the tensorflow_recommenders for this implementation. We created two towers that output the embedding (tf.keras.embedding) and we assembled a tfrs.Model by overiding its compute_loss function by tasking it to compute tfrs.metrics.FactorizedTopK by retrieving the embedding from the output layers of the towers. We trained the above network by going through our training samples. At the end of the training, we will have embeddings for every answer and question in the dataset. Later, we took all the answer embeddings and load into a embedding space so that we could lookp the answer for any given question embedding. So, for any given question we lookup the question embedding and its most k relevant answer embeddings and return the answers for those embeddings. We leveraged the ScaNN (tfrs.layers.factorized_top_k.ScaNN) module to create the embedding space partitioning. The following are some important steps in this approach: (i) Training Step, (ii) Model Evaluation step, (iii) Create ScaNN Index for the answer embeddings, (iv) ScaNN index lookup to obtain candidates and candidates scores for a given question.

3.4 Advanced Two Tower Approach (TOM II)

Advanced two tower architecture is shown in Fig. 3. While the overall workflow of TOM II looks similar to that of TOM I, TOM II employs pretrained BERT transformer to generate embeddings as opposed to TOM I which learns all the embeddings from the scratch. The towers in TOM II stacks a BERT preprocessor, BERT, Dense and Dropout layers in that order. We merged the towers using the keras.Model by overiding its compute_loss and train_step functions. In the compute_loss function we extract the embeddings from the question and the answer layers and then compute the logits from the dot product of the embeddings and then we calculate the cross entropy losses for both question and answer and return their mean as the total loss. We trained the above network by going through our training samples as before. At the end of the training we will have embeddings for every answer and question in the dataset. Unlike in the case of TOM I, we did not load the embeddings into an embedding space instead we extracted the question embedding for an incoming question and we filtered top scoring answers by leveraging the tf.math.top_k to get the relevant answers. The following are some important steps in this approach: (i) Training Step, (ii) Model Evaluation step, and (iii) Lookup relevant answers.

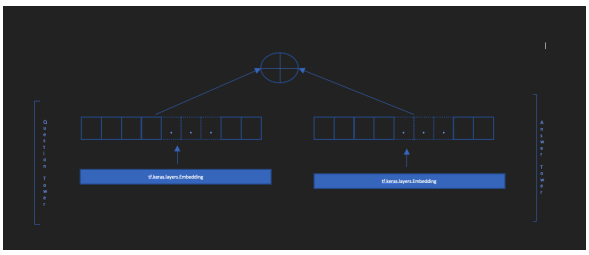

Figure 2: Simple Two Tower Model (TOM I).

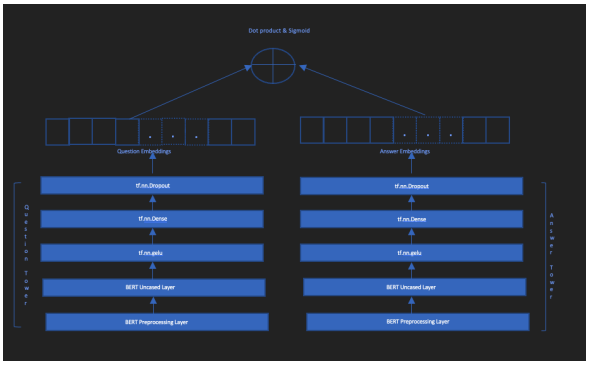

Figure 3: Advanced Two Tower Model (TOM II).

4 Dataset

We have used the FiQA 2018 open challenge dataset for our experiments. The FiQA 2018 [1] open challenge based on the use of unstructured text documents from different financial open data sources in English. FiQA 2018, is an open challenge of International World Wide Web Conference with two tasks (a) Task 1: sentiment analysis train, and (b) Task 2: Opinion-based QA. We are interested only in the QA dataset in task 2 collection. In the dataset, for each question we have one or more answers in the order of relevance. For our baseline approach above, We take the list of answers from the FiQA dataset to build an inverted index. We then use the Python interface of Anserini to map 50 candidate answers for each question. Thus, for each question in the dataset we have the answers as provided in the dataset and the 50 candidate answers as mapped by the Anserini. As shown below we have 6315 training questions and 333 testing questions. For each data sample we will have a question, and ground truth answers for that question

	Questions
Training	6315
Testing	333

Table 1: Dataset

5 Metrics

In order to compare our approaches, we shuffled our dataset and split the dataset by 95:5 ratio to create train and test datasets for training and testing respectively. As each our above approches output one or more answer predictions for a given question it is necessary to evaluate our answers against the ground truth with a rank aware scheme. So, we chose the following rank aware metrics: (1) Mean Reciprocal Rank (MRR), (2) Mean Average Precision (MAP), and (3) Normalized Discounted Cumulative Gain (NDCG). Taifi [8] provides a comprehensive survey on this topic.

6 Experiments

FiQA Task 2 dataset comes in the form of 3 .tsv files (a) document file of (docId, document) pairs, (b) question answer file of (docId, answerId) pairs, and (3) question file of (qId, question) pairs. We mapped the above data such that each data sample contains a question id (referring a question text in the dataset) and a sequence of answer labels (each label referring to a answer text in the dataset). Unlike the OQAP or the TOM approaches above, the baseline approach is not a model based. The **baseline approach** uses a simple relevance weight based on TF-IDF for each document for an incoming question. In order to compute this weight quickly, we use Anserini, an open-source IR toolkit built on top of Lucene (SimpleSearcher) to build the inverted index for the keywords in the documents. For each incoming question, we look up the cached inverted indices to match, score and sort documents quickly and output the list of k document ids with highest relevance scores. So basically, we could have run the metrics on both test or train data. However, for fairness we chose the test dataset to calculate the above metrics. For each data sample we will evaluate the candidates against the answers to measure the Average Precision (AP), Reverse Rank (RR) and Cumulative Gain (CG).

In **OQAP approach** figure 1, our first step is still the same as our baseline approach however we pass the k selected candidates to flow through our BERT pipeline (subsection 3.2) to pick the top n from that k documents. In the case of **TOM I** and **TOM II** approaches, the workflow is the same. The structure of inputs and outputs remain the same but the internal architecture is different (check out subsections 3.3 and 3.4). All these approaches, produces one or more relevant documents for a given question.

6.1 Experimental Setup

To cater to our above approaches, we have organized our workflow into following stages: (a) data preparation, (b) baseline, (c) TOMI, (d) TOMII, and (e) OQAP. The data preparation and baseline stages have to precede in the order of execution as they create interim data for the last three stages as they depend on it. The last 3 stages could be run in any order. The data preparation stage basically loads the FiQA dataset, creates inverted index and runs the Answerini for each question and create a data sample format that is very similar to the structure shown in table 3 but not including the last column (BERT score). At the end of data preparation stage, we have stored the train, test, dictionaries to map between questions, answers, and their IDs as interim data. At the second stage (baseline stage) we will read the interim data and process Baseline Metrics and generate Question/Answer pairs. TOM and OQAP stages basically will read all the interim data and create and store respective models for future use and finally generate metrics for respective approaches.

7 Results

In Table 2, we show MRR, MAP, and NDCG scores for each of the above approaches however with one caveat. Except for baseline approach, all other approaches are embedding based and so they are sensitive to the embedding size. As we see in Fig. 4, each question and answer pair are of varying sizes and they cluster mostly under 1000. While it is very memory intensive to run with the embedding size at 1000, we had run our experiments with different sequence lengths (128, 256, 512, respectively) and we have presented it in our plots (figures 4, 5, and 6). For a side-by-side comparision we have captured the scores for 256 embedding size and have displayed it for your review. Methods TOM I, TOM II and OQAP all out performed the baseline approach across all metrics. To our surprise TOM II is the best performer amongst all our ap-

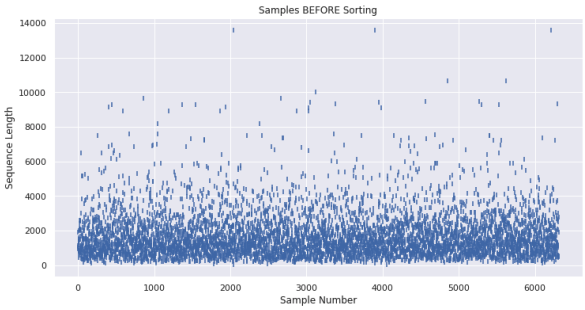

Figure 4: Plot showing the of sample (question, answer) pairs and their sequence lengths distribution.

proaches despite the dot product we use to calculate the similarity of the output from question and answer towers. We were expecting OQAP to perform better as it is taking in the tokens from question and answer simultaneously and makes a classification 1/0 based on the relevance. We attribute two reasons for this behaviour: (1) We employed TF/IDF based indexing to preselect the candidates before we run it through the BERT, and (2) The question and answer text together need a longer sequence length for the model to learn from the data. As we fixed the embedding size to range only between 128 and 512, the model's learning capability was limited. Finally, between TOM I and TOM II approaches, TOM II performed better than TOM I however only marginally. TOM I ran way faster than TOM II and OQAP methods. Considering the fact TOM I is 20X faster than TOM II and performing only marginally less accurately than TOM II, **TOM I is the star performing method** in the list. As shown in the appendix A below, embedding based approaches picked answers that were semantically closer even when words didn't overlap between the question and the answert that much.

Approach	MRR	MAP	NDCG
Baseline	0.3012	0.2578	0.2955
TOM I	0.5787	0.5787	0.5275
TOM II	0.5854	0.6405	0.4917
OQAP	0.415	0.3201	0.4455

Table 2: Approaches and scores (Two Tower with 128 Embedding Sz.)

8 Limitations

The TOM methods lend naturally to match answers with semantic relevance even when there is very little word overlap between answers and questions.

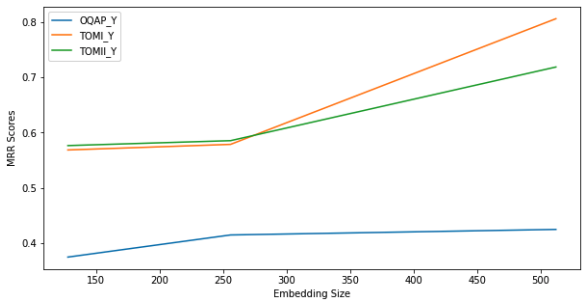

Figure 5: MRR Score vs. Embedding Size

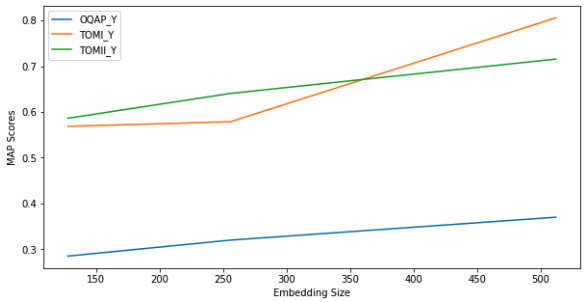

Figure 6: MAP Score vs. Embedding Size

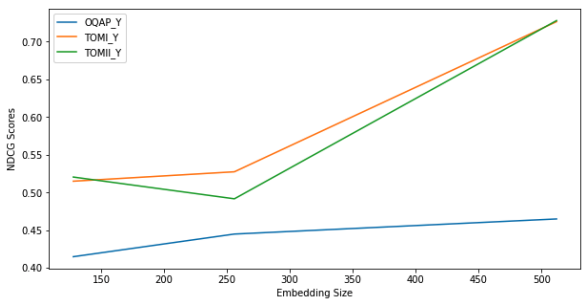

Figure 7: NDCG Score vs. Embedding Size

On the other hand, in the OQAP, we feed the documents with highest score from TF/IDF restricts the model from selecting only the documents with word overlap. This creates a rather uneven comparison with TOM methods.

Transformer based approaches are prohibitvely slow and are very sensitive with the embedding sizes. In general, these algorithms' performance depends highly on the compute, memory, time and special hardware resources. Though our experimental data is mostly centered around 1000 tokens we could manage to run it with 512 embedding size. The performance could be drastically improved if we could increase the embedding size to 700 or 1000. In real-world problems, document size could easily cross over this limit and hence transformer based IR algorithms could fail to scale up. We leveraged the google colab environment to run our experiments. Our transformer based methods took 15 to 20 hours to run with even 2 to 3 epochs.

With better compute resources, more experiments on number of training layers and even more training data (other Pretrained models or even a Bert model from scratch) can improve the model performance was to increase the number of training layers. A Bert model with LSTM can also improve the results.

9 Future work

We noticed several opportunities in this research. First and foremost, to our surprise TOM I came out as a star performer. TOM II and other Transformer based retrieval systems showed a lot of potential for increased performance. With increase in sequence length, number of epocs, and by tweaking hyperparameters these models could learn drastically and produce significant improvement in accuracy. We like to continue this research and hope to improve the model performance to produce SOTA results. We hope to extend these models to a more real world dataset and unleash the power of embedding based retrieval to a next level.

10 Conclusion

In this paper we explored question answer problem as a information retrieval problem. We tried 4 approaches namely the baseline approach, OQAP approach, TOM I approach and TOM II approach. We compared these approaches using rank aware metrics like MMR, MAP, and NDCG. While the sophisticated transformer based approaches like OQAP, and TOM II showed tremendous promise, simple TOM I approach came out as a surprise star performer.

Acknowledgments

First and foremost, we would like to thank our professors Mark Butler, and Natalie Ahn for their valuable time and feedback. We are very grateful for their encouragement throughout this course. Without their valuable support and knowledgeable advice this research would not have been possible.

References

[1] Macedo Maia, and Markus Endres *A comparative study using different question context information on pairwise learning-to-rank CQA transformer models in the home improvement domain*, Journal of Data Intelligence, Vol. 3, No. 1 (2021) 131–148

[2] Zhuang Liu , Degen Huang, Kaiyu Huang, Zhuang Li and Jun Zhao *FinBERT: A Pre-trained Financial Language Representation Model for Financial Text Mining*, Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20) Special Track on AI in FinTech

[3] John M. Boyer. *Natural language question answering in the financial domain* In: Proceedings of the 28th Annual International Conference on Computer Science and Software Engineering, CASCON 2018, Markham, Ontario, Canada, October 29-31, 2018. Ed. by Iosif-Viorel Onut et al. ACM, 2018, pp. 189–200.

[4] Bithiah Yuan, *FinBERT-QA: Financial Question Answering with pre-trained BERT Language Models*, Master's Thesis, Albert-Ludwigs-University Freiburg, Faculty of Engineering, Department of Computer Science, 2020.

[5] Suman Karanjit, *Question and Answering Using BERT*, Computer Science Major Minnesota State University Moorhead, 2021.

[6] Christopher Manning and Pandu Nayak. Information Retrieval and Web Search. 2019. url: https://web.stanford.edu/class/cs276/.

[7] Peilin Yang, Hui Fang, and Jimmy Lin. *Anserini: Enabling the Use of Lucene for Information Retrieval Research*. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017. Ed. by Noriko Kando et al. ACM, 2017, pp. 1253–1256. doi: 10.1145/3077136.3080721. url: https://doi.org/10.1145/3077136.3080721.

[8] Moussa Taifi, *MRR vs MAP vs NDCG: Rank-Aware Evaluation Metrics And When To Use Them*, https://medium.com/swlh/rank-aware-recsys-evaluation-metrics-5191bba16832

Qid	Answer Label	Candidates	BERT Score
12	[573518, 175824, 483282, 479203]	175824	0.154781
13	[573518, 175824, 483282, 479203]	479203	0.861560
14	[573518, 175824, 483282, 479203]	573518	0.926755

Table 3: Data sample examples

[9] Y. Chali, S.A. Hasan, S.R. Joty, *Improving graph-based random walks for complex question answering using syntactic, shallow semantic and extended string subsequence kernels* Inf. Process. Manage., 47 (6) (2011), pp. 843-855.

[10] A. Ansari, M. Maknojia, A. Shaikh *Intelligent question answering system based on artificial neural network* 2016 IEEE International Conference on Engineering and Technology (ICETECH), IEEE (2016), pp. 758-763.

[11] A. Abdi, N. Idris, Z. Ahmad *Qapd: an ontology-based question answering system in the physics domain* Soft. Comput. (2016), pp. 1-18, 10.1007/s00500-016-2328-2.

[12] Ming Tan, Bing Xiang, and Bowen Zhou. *LSTM-based Deep Learning Models for non-factoid answer selection*. In: CoRR abs/1511.04108 (2015).

[13] Wen-tau Yih et al. *Question Answering Using Enhanced Lexical Semantic Models* In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers. The Association for Computer Linguistics, 2013, pp. 1744–1753.

[14] *https://www.tensorflow.org/recommenders/examples/basic_retrieval*

A Appendices

In the following example we tried searching answer for Qid 444 to compare the output generated with OQAP vs. TOM I vs. TOM II approaches.

Question (444): Why do most banks in Canada charge monthly fee?

OQAP approach picked the following answers:
Answer 1 (573518): Arguably, "because they can". Canada's banking industry is dominated by five chartered banks .....
Answer 2 (479203): You have to check your contract to be sure what is it you're paying for. Typically, you get .....
Answer 3 (175824): Lending isn't profitable when interest rates are this low. Consider what's involved to .....

TOM I approach picked the following answers:
Answer 1 (483282): The other answers in this thread do a fine job of explaining the .....
Answer 2 (175824): Lending isn't profitable when interest rates are this low. Consider what's involved to .....
Answer 3 (573518):Arguably, "because they can". Canada's banking industry is dominated by five chartered banks .....

TOM II approach picked the following answers:
Answer 1 (573518): Arguably, "because they can". Canada's banking industry is dominated by five chartered banks .....
Answer 2 (504709): Draw up a budget and see where most of you expenses go to. See if you can cut .....
Answer 3 (483282): The other answers in this thread do a fine job of explaining the .....