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

Anonymous ACL submission

Abstract

Here 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], and (3) A Two-tower model (TOM) that uses the question and answer towers. As the TOM is based out of text embeddings and its similarity, 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 and contrast 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 personnels 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 personnels could not only result in inconsistent interpretations of the information but also be difficult to keep these information at their finger tips. 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 improvements 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]. Thus, 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 and contrast 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], and (3) Two tower model approach. In the following subsections we will provide details for each of the above approaches.

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.

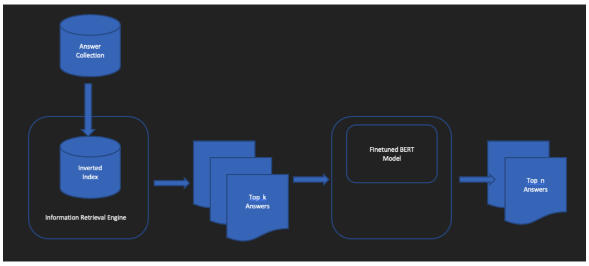

Figure 1: QA Pipeline approach (OQAP).

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 passed through a finetuned pretrained BERT model. For each test sample in the dataset, the output of the BERT mdoel 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 from the OQAP. The pretrained BERT is finetuned for each question answer pair in the training dataset in FiQA.

3.3 Two Tower Approach (TOM)

The overview of the two tower architecture is shown in Fig. 2. It has a separate question tower and an answer tower with a final dot product that computes the relavance between a given question and an answer. The answer tower takes the features from the given answer and generates an answer embedding for it. The features here are the encodings for the words found in the answer. Similarly, the question tower takes the encodings for the words in the question. Finally, we do a simple dot product based on the two embeddings as a measure of how likely the pairs are relevant. So, basically we trained the network for each positive question and answer combination such that we maximize the dot product measure. At the end of the training we

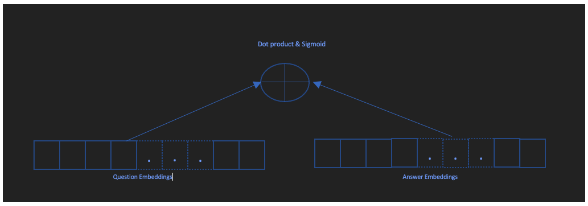

Figure 2: Two Tower Model (TOM).

will have embeddings for each answer and question. We took the answer embeddings and load into a embedding space so that we could lookp the answer for any given question embedding. We used the ScaNN module to do the embedding similarity. The following are some important steps in this approach (check the provided notebook for details): (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.

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. This data comes in two flavors (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 many relevant answers in no specific order. So, we have assumed all answers as equally relevant. 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, its corresponding answers and questions' Anserini candidates.

	Questions
Training	6315
Testing	333

Table 1: Dataset

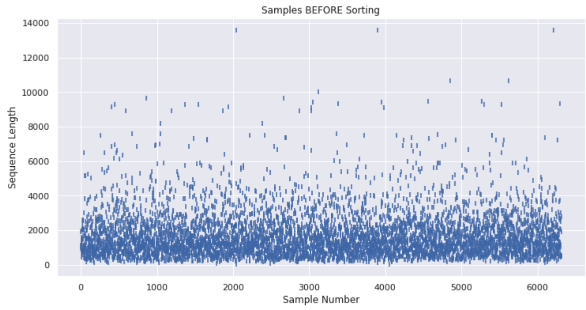

Figure 3: Plot showing the of sample (question, answer) pairs and their sequence lengths distribution.

5 Metrics

In order to compare and contrast our approches, we used a test set extracted from the FiQA dataset. We used the Mean Reciprocal Rank (MRR), Mean Average Precision (MAP), and Normalized Discounted Cumulative Gain (NDCG) for the top 10 answers to evaluate the systems' performances.

6 Experiments

Unlike the QA Pipeline and the Two Tower approches, the baseline approach is not a model based. The baseline approach uses a simple weighted inverted index system based on TF-IDF. 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 Commulative Gain (CG).

In the case of Two Tower model as outlined above, for each data sample we looked up the candidates and candidate scores from the ScaNN index. Similarly, as in the case of baseline approach, We will evaluate the candidates against the answer to generate AP, RR and CG scores for each data sample.

We had run our experiments with different sequence lengths for the overview pipeline and two tower approaches respectively. We have plotted the metrics in figures 3 and 4.

7 Results

As shown in table 2, MRR and MAP scores have shown considerably improved with the two tower model as opposed to the baseline and NDCG scores showed a 5 percentage point improvement at embedding size 128. From the two tower approach

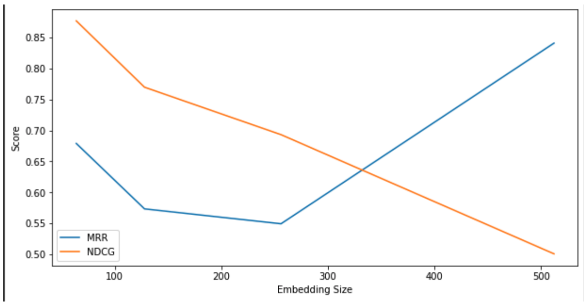

Figure 4: Score vs. Embedding Sz. plot for two-tower

plot shown in figure 4 we see a linear fall off in the MRR score with the increase in embedding size where as the NDCG score show a slight fall with the embedding size between 50 and 275 and show a considerable improvement there after.

Approach	MRR	MAP	NDCG
Baseline	0.2943	0.2810	0.7067
QA Pipeline	0.5976	0.5976	1.0368
Two Tower	0.5736	0.5736	0.7698

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

8 Limitations

The two tower and overview pipeline approaches not only lend naturally match answers with semantic relevance even when there is very little word overlap between answers and questions it also perform overall in accuracy performance. However, these methods grow prohibitively in time complexity with the increase in embedding size. So matching relevant documents becomes a challenge with the increase in document size. For our experimental dataset sequence length of 512 reasonably covers the text however the data with moderate or large sized documents this method could slow down drastically.

9 Future work

We have measured mean average precision (MAP), normalized discounted cummulative gain (NCDG) score

10 conclusion

Papers that have been or will be submitted to other meetings or publications must indicate this at submission time in the START submission form, and must be withdrawn from the other venues if accepted by ACL 2020. Authors of papers accepted for presentation at ACL 2020 must notify the program chairs by the camera-ready deadline as to whether the paper will be presented. We will not accept for publication or presentation the papers that overlap significantly in content or results with papers that will be (or have been) published elsewhere.

Acknowledgments

First and foremost, we would like to thank our instructors Prof. Mark Butler, and Prof. Natalie for their valuable time and feedback. We are very grateful for their encouragement through out this course.

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

[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.

A Appendices

Appendices are material that can be read, and include lemmas, formulas, proofs, and tables that are not critical to the reading and understanding of the paper. Appendices should be **uploaded as supplementary material** when submitting the paper for review. Upon acceptance, the appendices come after the references, as shown here.