FAQ Search using Transformers

Anonymous ACL submission

Abstract

Many websites have bots as a guiding agent, for answering FAQ questions or directing users to human support. Many of them already have a curated FAQ page that can be used to bootstrap these bots. In this paper, we want to tackle a real-world problem of question answering for Bots. Given a user query, the system needs to pick the most relevant answer from a data source such as FAQ or Manuals. So, the ranking system needs to consider not just the passage but also the provided support questions or titles. This technique also provides the flexibility to add and delete support questions to continuously improve bot's quality, suggestions can be provided by system and the bot developer has control over their data instead of a black box system. We explore novel techniques to improve the results on a few public sets and on our own judged real user data. For the paper, We limit our experiments to transformers since it has proven to be significantly better in all question answering tasks. We show that significant gains can be observed using an extra segment embedding as well as pre-training new separators in transformers.

1 Introduction

We define a knowledge base as set of QnAs as shown in Table 1, where each QnA corresponds to a unique answer and set of support questions. Earlier work like (Agrawal et al., 2020) have used WordNet, DSSM, and TF-IDF based features to solve the problem, we observed significant gain using transformers. Recent years have seen a trend towards using Transformers for almost all NLP tasks. The biggest challenge with fine-tuning pre-trained transformers for this task is its inability to understand more than 2 segments since they are at most trained on Next Sentence Prediction tasks which have 2 segments. The simplest way to tackle this problem is to divide the task into sub-tasks and then merge the results based on an aggregate model or heuristics. We show that self-attention across all the contexts gives much better results compared to breaking it into different sub-tasks of Query-Question matching and Query-Answer matching. Then we explore adding new tokens and extra segment encoding to distinguish between the segments. We show significant gains in the model's ability to be able to understand FAQs compared to when everything is concatenated using delimiters or existing sentence separator.

2 Related Work

There has been extensive research on various types of question answering: closed book QA (Roberts et al., 2020) , QA over text, QA over SQL(Wang et al., 2021), knowledge graph-based QA (Lukovnikov et al., 2019), but FAQ based question answering hasn't been much explored apart from (Agrawal et al., 2020) and (Damani et al., 2020). In this paper, we repro both the papers and compare results with them, however, we use the same pretrained model (Chi et al., 2020) instead of MT-DNN as used in the paper. There are datasets for QuestionAnswering like SQUAD (Rajpurkar et al., 2016), GoogleNQ(Kwiatkowski et al., 2019) but there is a lack of data with the knowledge base of Questions and Answers combined. In the recent work, (Bruyn et al., 2021), where authors have called it to be a FAQ retrieval problem, but they specify that since their dataset does not contain live user queries, they only restrict the task to rank the FAQ's Answers based on a FAQ question. This becomes very similar to the standard Question-Answer retrieval task and makes it non-relevant for our research.

3 DataSets

Our existing system already powers a huge number of bots across the world with such data. We have gathered 30 such knowledge bases for English Language and 10 sets for other high-traffic languages.

QnAID	Questions	Answer
1	Medical reimbursement policy What is Medical reimbursement deadline Max limit for claiming medical expenses	Medical Reimbursement can be filed for expenses of employee and dependent in a financial year. Download claim form here
15	My PC is locked Lost my corporate card Files got corrupted	For any IT or technical queries , contact IT help desk at <PII> or mail to <PII>.
21	How can I Apply for paternity leave Paid leaves and sick leaves When can I take sabbatical Can leaves be carry forwarded to next year?	Link to Vacations Portal (http://vacations).
22	Comapany tax benefits Tax implications on Car lease	For tax filling, ESOPS gain and related queries , click here.

Table 1: This is an example of a knowledge base. Please note that only few selected rows are mentioned, this knowledge base contains 40 QnA pairs in total.

Initialization	High LR	Internal Set F_1
Baseline	NA	88.8
Random	Yes	90.1
Random	No	87.7
</s> token	Yes	2.2
";" and "</s>"	No	90.2
";" and "."	No	90

Table 2: New Separator Token Initialization techniques.

Initialization	High LR	Internal Set F_1
Random	Yes	90.1
Random	No	84.6
0th Segment	Yes	90.3
0th Segment	No	89.1

Table 3: Segment Encoding Initialization techniques.

This data was gathered when product was in a preview mode after consent of users. Our base model is (Chi et al., 2020) which is multilingual and we have shipped the technique for 50+ languages but experiments and evaluations specific to multilingual quality are out of scope for this paper. For getting user queries, we applied a few filters on the logs:

- Removing Chit-Chat user queries like "hi", "Hello" using a chat domain classifier (Akasaki and Kaji, 2018).
- Removing junk queries (queries containing majority of the words out of the language) using a pre-created language based dictionary.
- Grouping very similar queries to make the data more diverse and so that the judgement queries can cover more intents. We did using cosine similarity on Sentence Transformers (Reimers and Gurevych, 2019) and applied very high similarity threshold (0.9).

After applying the above filters, human annotators were asked to select Answers (with supporting questions) which is relevant to the query from the respective Knowledge base. 2 annotators labeled each query, all the answers are chosen by both were taken directly whereas the answers chosen by 1 were outright ignored but made sure that they aren't added to the negative set. A negative set (for both training and testing) was prepared by selecting 5 random answers from the knowledge base.

4 Approach

Even if the task looks like a ranking task, the score given by the system is also important. Bot developers using our service use a score threshold to decide whether to show the answer. Thus, ranking metrics such as DCG or NDCG don't qualify. For both modeling and metrics purposes. we treat the problem as a triplet classification task. Let Q be the set of all user queries and KB = (sq_{11} ,sq_{12},sq_{13}....,a_1),... (sq_{n1},sq_{n2},...a_n) be the set of all QnAs in the Knowledge base. Given a user query $q_i \in Q$ and a QnA pair $kb_j \in KB$, output a relevance score h(q_i,kb_j) for kb_j with respect to q_i. We use Average Precision (AP) as the primary metric.

We define 3 experiments to achieve the task:

Technique	Internal Set	StackFAQ	FAQIR	Latency (ms)
Concatenating with whitespace (Baseline) (Damani et al., 2020)	89.6	82.6	91.3	48.5
Using SubTasks	91.0	82.8	93.5	206.16
Using Seperator Tokens	90.2	82.4	92.7	48.8
Using Segment Encodings	90.3	82.9	92.8	48.5
Using Seperator tokens and Segment Encoding	91.3	82.4	93.2	48.9
Non-Transformer based (Agrawal et al., 2020)	83.3	68.5	70.3	31.1 (CPU)

Table 4: Retrieval And Ranking Measurements

4.1 Using SubTasks

In this method, we break the task into 2 simple tasks, let UQ be the User Query:

$$QuesScore = \max_{sq_{ji} \in KB_j} Model1(sq_{ji}, UQ) \quad (1)$$

$$AnsScore = Model2(a_j, UQ) \quad (2)$$

Then we train a linear dense model to combine scores from the two models. While this architecture results in very high runtime computation depending on the number of Support Questions contained in a Knowledge Base, the explainability of the model is high as compared to other models. One more advantage that this approach provides is that support questions are processed independently of each other and final aggregated using Max, thus adding a support question can never lower a score. This is the top negative feedback we have gotten from customers after deploying the below technique.

4.2 Using New Seperator Tokens

Transformers use pre-trained tokens like [SEP] to differentiate between 2 segments of text. Here we use 2 extra separator tokens named "[SQSep]" and "[ASep]" denoting Support Question Separator and Answer Separator. As shown in 1, the input is now feeded like this: $UQ + [SEP] + sq_{j1} + [SQSeperator] + sq_{j2} + [SQSeperator] + sq_{j3} ... sq_{jm} + [ASeperator] + a_j$

One drawback of this method is it adds extra tokens thereby decreasing the length of text, the model can handle. 87.1 perc of Query-QnA pair from our internal set have less than 128 tokens since FAQ-based questions and answers are generally smaller in length. If we use a token length limit of 256, 93 perc of the dataset gets covered. So, the impact of these newly added separator tokens is only on a minority of the set. We show a comparison of using these tokens individually and together. We see that the AnswerSeperator is more important compared to SupportQuestionSeperator. Though, after using Segment Encoding, Answer Separator becomes redundant. One challenge with training new tokens in transformers is the initialization. We tried various techniques for that:

- Random Initialization: If doing Random Initialization, you can't learn the weights with standard finetuning Learning Rate which ranges from 0 to 2e-06. So, we use a separate learning rate for the embedding matrix of these tokens which is 1e05 -> 2e-03.
- Initializing with "[SEP]" Embeddings: We copy [SEP] or </s> token's embedding for initializing the embeddings. We observe this to be better than the technique used above. All Random initializations of weight were done using the constraint of 0 mean and 0.02 std. deviation as adopted in most transformer pre-training. We show that pre-training a new token with Random initialization doesn't work well for task-specific training.
- Using Existing tokens: In this method, we try using existing tokens as separators without treating them as new tokens.

We tried a few combinations of the 3 types of initialization for the Question and Answer Separator. As seen in Table 2, initializing Question Separator token with ";" and Answer Separator token with [SEP] gave the best results. Our baseline is using the [SEP] token for all Separators, no extra new tokens. Some other punctuations like comma and full-stop were also tried.

4.3 Using Segment Encodings

Segment Encodings can be used to differentiate between the 3 segments of UserQuery, KBQuestions, and KBAnswer. As shown in 1, KBQuestions and

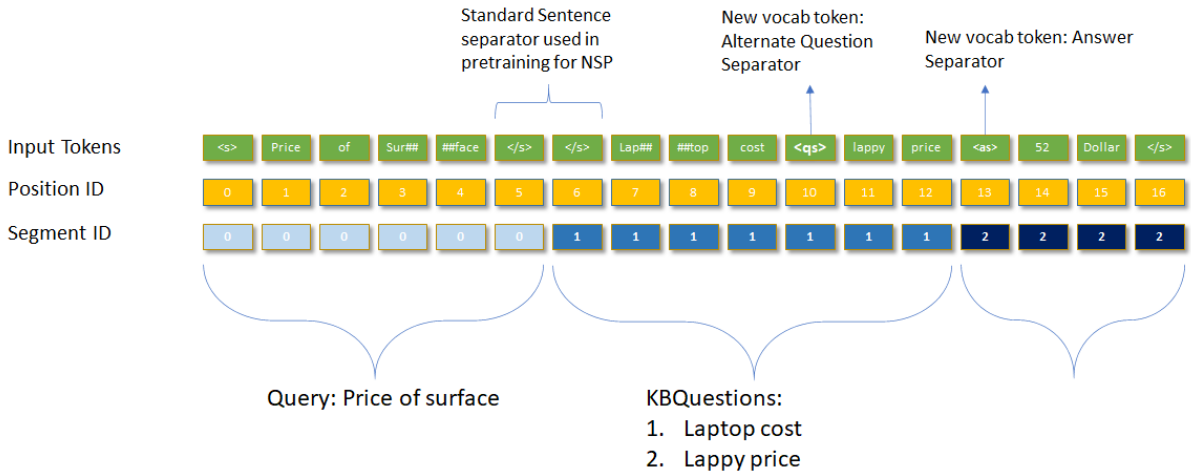

Figure 1: QnAMaker Runtime Pipeline

KBAnswer are assigned Segment Encodings of 0 and 1 respectively. These new Segment Encodings are trained using the training set. We experimented few initialization techniques for Segment Encodings as well:

- Random Initialization: Same as new tokens, we used a separate Learning Rate group for this 2*768 Embeddings while finetuning and constraints for initialization.
- Initializing with 0th Segment: We initialized the embeddings for Segment 1 and Segment 2 with embeddings of Segment 0. While finetuning, we tried both the setting of normal LR and high LR for these embedding weights.

As seen in Table 3, for both kinds of initializations, high LR gave better results. Overall, initializing with existing segment embedding worked little better over random initialization.

5 Result

All experiments are ran using Grid Search from Azure ML and only best of the experiment is reported. We first ran individual experiments on internal set for figuring the best strategy for pre-training separator tokens and new segment embeddings as shown in Table 2 and 3. Row no. 2 and 3 in Table 4 denotes the best strategy from those evaluations. We then combined those two techniques to see if they can give incremental gains. Qualitative analysis shows that Segment Encoding is good for making the model understand the difference between a support question and answer whereas the Separator token helps the model understand that the support questions are independent and if the user-query is matching any of them, it is good enough. We also provide the average time taken to run inference on our evaluation set with batch size of 5 on V100 GPU (replicating the production scenario in closest way possible). We see that Approach 4.1 is more than 4 times higher latency even if having better results. Adding Segment Encoding doesn't have any impact on the computation over baseline whereas using separator tokens have negligible impact. 4

6 Current limitations and Future Work

We are doing experiments with training the separator tokens better using MLM tasks. One of the user feedback we have been getting about our system is from customers with knowledgebases containing very domain-specific knowledge. Transformer-based models only take the specific QnA and User query as input and ignore any information about the entire knowledge base. For example, in a knowledge base for a car store that sells cars of various brands, a query about "audio not working in my car" should not result in "audio not working in BMW" because they can be talking about BMW or Mercedes or Audi whereas if there is a knowledge base created for BMW products having the same QnA, the query "audio not working in my car" can be mapped with it. We are trying to solve this problem by appending local (KB-specific) IDF information per word along with word embedding and segment embedding.

References

Parag Agrawal, Tulasi Menon, Aya Kamel, Michel Naim, Chaikesh Chouragade, Gurvinder Singh, Rohan A Kulkarni, Anshuman Suri, Sahithi Katakam, Vineet Pratik, Prakul Bansal, Simerpreet Kaur, Neha Rajput, Anand Duggal, A. Fattah Chalabi, Prashant Choudhari, Reddy Satti, and Niranjan Nayak. 2020. Qnamaker: Data to bot in 2 minutes. *Companion Proceedings of the Web Conference 2020*.

Satoshi Akasaki and Nobuhiro Kaji. 2018. Chat detection in an intelligent assistant: Combining task-oriented and non-task-oriented spoken dialogue systems.

Maxime De Bruyn, Ehsan Lotfi, Jeska Buhmann, and Walter Daelemans. 2021. Mfaq: a multilingual faq dataset.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2020. Infoxlm: An information-theoretic framework for cross-lingual language model pre-training. arXiv.

Sonam Damani, Kedhar Nath Narahari, Ankush Chatterjee, Manish Gupta, and Puneet Agrawal. 2020. Optimized transformer models for faq answering. In *Advances in Knowledge Discovery and Data Mining*, pages 235–248, Cham. Springer International Publishing.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*.

Denis Lukovnikov, Asja Fischer, and Jens Lehmann. 2019. Pretrained transformers for simple question answering over knowledge graphs. In *International Semantic Web Conference*, pages 470–486. Springer.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model?

Bailin Wang, Richard Shin, Xiaodong Liu, Oleksandr Polozov, and Matthew Richardson. 2021. Rat-sql: Relation-aware schema encoding and linking for text-to-sql parsers.