Information Retrieval for Question Answering Systems using a Statistical Mixture Model

Uffaz Nathaniel

Abstract
The paper investigates using a Mixture Model to build an IR4QA system that recommends an answer to a natural language question from precompiled list of Yahoo Answers question-answer pairs. We describe the system, results, and some of the key takeaway lessons.

1. Introduction
An Information Retrieval for Question Answering (IR4QA) or as they are often called Question Answering in Information Retrieval (QAIR) are systems that try to find answers to given questions from an existing FAQ database. Unlike AI question-answering systems that focus on generation of new answers, IR4QA retrieves existing answers from a FAQ file (Burke 1997).

The Text REtrieval Conference (TREC) is a series of workshops organized by the National Institute of Standards and Technology (NIST) and designed to advance the state-of-the-art in information retrieval (IR). The workshops have focused primarily on the traditional IR problems of retrieving a ranked list of documents in response to a given query. However, in many cases a user has a specific question and would much prefer that the system return the answer itself rather than a list of documents that contain the answer. To address this need, the Question Answering (QA) track has focused on retrieving answers rather than document lists (Voorhees and Tice 2000).

For many years in TREC QA-Track's, the focus has been on answering factual questions that are extracted from corpus of news articles. In 2015, TREC introduced a LiveQA track that focused on answering questions submitted on Yahoo Answers (YA). The questions were extracted and submitted to the registered participants during a time period of 24 hours starting August 31, 2015 (Agichtein 2015). Thus the main focus for participants was to map new questions to already known questions and chose the best answer.

The participants had 60 seconds to answer the question after which their responses were judged and scored by TREC organizers using the 4-level scale shown below:

4: Excellent – a significant amount of useful information, fully answers the question.
3: Good – partially answers the question.
2: Fair – marginally useful information.
1: Bad – contains no useful information for the question.
-2: Unreadable

For our research, we focused on the 2016 LiveQA track. Since the conference already taken place, we had access to a list of questions[1] and ranked list of answers[2] by various participants. We used the list of 1015 questions as our "live" unknown questions and suggested an answer. To score our answer, we used the 4-level scale proposed by TREC organizer and gave our answer a rating. We then computed the average to understand how well our system did.

2. System Overview

There are 3 main components to our system that dealt with different aspects of answering a YA question:

1. Analyzing questions
2. Training
3. Searching and scoring

2.1 Analyzing questions

A YA question usually consists of a *title* and a *body*. Ideally, the title contains the main question, however, often this is not the case. This means that the relevant question is distributed across the title and body. For example:

> *Pregnant cat? What to do?!!*
>
> *Yesterday we rescued a pregnant cat. I've her running around with her huge belly for quite a while now, so we decided to help her. Her breathing is quite heavy and some of her nipples are white but with not a lot of milk in them. I really want to be able to tell roughly when the babies are coming because her belly is enormous!!*

Some researchers (Varanasi and Neumann 2015) have suggested giving more weight to the title or looking at the title, pronouns, or question key-words (e.g. 'who', 'what', 'where') to extract a summary and using that. However, we chose not to go down that route or apply any type of heuristic in fear of missing vital information.

Thus given a question Q, we construct a new question Q' that consists of the title+body. Q' is also stripped of any HTML tags/entities and is normalized for further processing later.

2.2 Training

The core idea of training phase is to build a model that can match new questions to existing ones.

Our dataset for training *Yahoo! Answers Comprehensive Questions and Answers version 1.0*[3] is a collection of more than 4 million question-answer pairs provided by Yahoo Labs on the WebScope site. However, we chose a subset of 834,000 questions that belonged to the categories specified in the TREC-2016 guidelines[4]. For each of the Q', we normalized them by expanding

[1] https://trec.nist.gov/data/qa/2016_LiveQA/interpretations.xml
[2] https://trec.nist.gov/data/qa/2016_LiveQA/anon-qrels.txt.gz
[3] https://webscope.sandbox.yahoo.com/catalog.php?datatype=l&did=11
[4] https://trec.nist.gov/data/qa/2016_LiveQA/categories.txt

contractions like *don't* to *do not* or fixing any spelling mistakes. Q' is also stemmed to remove morphological variations.

At this point, we had a collection of Q' that needed to be vectorized. We chose a simple count-based vectorizer with unigram sequencing, a feature size of 3000 words, and stop-word removal.

We then trained a finite-dimensional mixture model. We chose a mixture model mostly because it is easy to understand and implement in code. It offers properties similar to those of TF-IDF along with similarity measure similar to cosine similarity. Instead of building an indexing engine, a mixture model offers us the best of both worlds.

Lastly, in this model, we chose λ=0.5. This was based on research done by McNamee and Mayfield 2004 that showed the value of lambda doesn't really matter. They also proposed that λ=0.5 works well for most situations.

2.3 Searching and scoring

Given a new question, we find the relevant top 50 documents from our collection of over 834-thousand question-answer pairs.

Researchers (Varansai et. al. 2015 and Burke et. al. 1997) have suggested a two stage approach for ranking and retrieval:

Stage 1: The first step is to narrow the search to a small set of FAQ files likely to contain an answer to the user's question.
Stage 2: Each QA pair is matched against the user's question to find the ones that best match it.

Varansai et. al. 2015 suggested a calculation which is the weight of our original rank multiplied by the cosine similarity between the user's question and the selected question. This didn't work for us primarily because our probability values are too small and not between [0, 1]. We tried normalizing but not much luck. Lastly, we tried to take our top 50 relevant document and compute cosine similarity between user's question and Q' and returned that ranked list. This again didn't work too well.

Since we didn't see much improvement in stage 2 (see Table 2), we decided to ignore it. It added extra computation without improving the quality of responses. Our final implementation then is just 1-stage where we return the ranked list produced by our mixture model.

We should also note that semantic knowledge bases like WordNet can be extremely useful to expand a question (Burke et. al. 1997). However, we didn't have enough time to implement any type of semantic analysis.

3 Experiments

We tried to run the original 1015 questions from TREC 2016 through our system. We think we got pretty good results. For example, one interesting question that was asked during the conference was:

TREC Question:
My mom died when I was 8 and I have an awful dad when it comes to health and body things. I never knew u needed to clean your belly button? I recently took a look at it and it is very dirty I manage to clean it but deeper in I can't seem to clean it cause it is really deep, how can I clean my belly button out completely.

Top 3 matched questions:
1. *Weird! Puss coming out of my bellybutton....is this normal? So just recently, I started noticing that I had like dried puss or something around my belly button. I took a Q-tip and cleaned it out and by the end of the night it was there again. It's really disgusting and it smells really bad. I have my belly button pierced but I don't think that is the cause. It seems to be coming from inside the belly button. What could be the cause of this and is it serious???*
2. *my finace's belly button is pretty deep how deep should they be? are there any health issues that come with a deep navel*
3. *How do you clean ur belly button?So it wont look brown and dirty?*

Top 3 matched answers:
1. *Obviously you have an infection. I would be VERY surprised if it didn't have something to do with your piercing. But you should go to the doctor and have them check it. Whatever is going on, you need at least an antibiotic...possibly pills AND ointment. Depends on what is infected and how badly.*
2. *deep navel??? is he fat or too thin ? may b itz just natural or i think u should consult a doctor and ask..coz if it is then u know the sooner you know bout the problem the better...best of luck!*
3. *A good way is to wet a cotton swab and use that to clear out dirt. Don't press to hard with it or it will hurt though.....just enough to get the gunk out.*

Looking at the top 10 results, we note that this typically contained the answer to the question. In our system, we would always return the top 1 ranked question-answer pair as the best choice but generally speaking, our ranked list of about top 10 contained the ideal answer.

4 Results

In our experiment, we did not have the ability to have our responses by judged by TREC organizers. We instead used the 4-level scale that was used by TREC judges to rate our own response.

For the first 10 questions we scored ourselves using the already judged responses as reference. Our results are shown in Table 1:

Table 1 – Using the 1-stage approach (only the mixture model).

Question	Score
Q1	2
Q2	3
Q3	4
Q4	3
Q5	1
Q6	2
Q7	1
Q8	1
Q9	1
Q10	2
average	**2**

We got an average score of 2 with about 40% of the results scoring a 3 or better.

Table 2 – Using a 2-stage approach where in stage-2, we compute the cosine similarity between original question and the top 50 matching questions. Then new ranked list is then returned.

Question	Score
Q1	3
Q2	2
Q3	1
Q4	1
Q5	1
Q6	1
Q7	1
Q8	2
Q9	1
Q10	1
average	**1.4**

In Table 2, we applied the 2-stage pipeline for ranking. We note that the average score was a 1.4 and only 10% of the documents scored 3 or better.

5 Conclusion

As we discussed in the *Experiments* section, generally answers that would qualify as Excellent or Good were in the top-10 results. With relying only on the answers from our model (1-stage pipeline), we got about 40% of the results that would score 3 or better. We note that one of the problems with mixture model (as opposed to something like cosine similarity) is that the probability values are not bounded (e.g. [0,1]). This made it difficult for us to factor in the rank from stage 1 into stage 2. For the next iteration, it would be interesting to explore a IR4QA system using a TF-IDF scheme along with cosine similarity.

References

[1] E. Agichtein, D. Carmel, D. Harman, D. Pelleg, and Y. Pinter. Overview of the TREC 2015 LiveQA Track. In *Proceedings of the Twenty-Fourth Text REtrieval Conference*. (2015)

[2] E. M. Voorhees and D. M. Tice. Building a Question Answering Test Collection. In *SIGIR conference on Research and development in information retrieval*. pp. 200-207. (2000)

[3] P. McNamee and J. Mayfield. Character N-Gram Tokenization for European Language Text Retrieval. In *Information Retrieval Journal*. Vol 7, 2. pp 73-97. (2004)

[4] R. D. Burke, K. J. Hammond, V. Kulyukin, S. L. Lytinen, N. Tomuro, and S. Schoenberg. Question Answering from Frequently Asked Question Files: Experiences with the FAQ FINDER System. In *AI Magazine*. Vol 18, 2. (1997)

[5] S. Varansai and G. Neumann. Question/Answer Matching for Yahoo! Answers Using a Corpus-Based Extracted Ngram-based Mapping. In *The Twenty-Fourth Text REtrieval Conference (TREC 2015) Proceedings*. (2016)