University of Texas Rio Grande Valley
TREC LiveQA 2016:
Using Topic Modeling to Answer Complex Questions

Josue Balandrano Coronel
University of Texas Rio Grande Valley
jcoronel@tacc.utexas.edu

Abstract

This paper describes the system submitted to the TREC 2016 LiveQA track. This year, the TREC 2016 LiveQA track consists of implementing a web service to answer user-submitted questions. The newest unanswered question from Yahoo! Answers will be posted to the web service, a question every minute for 24 hours. The implementation described in this paper uses natural language processing (NLP) to extract keywords from the question given as input. A web search together with a Yahoo! Answer search is used to select candidate answers. A latent dirichlet allocation (LDA) model is trained in order to compute a topic distribution of the different candidate answers. Finally, the Jensen-Shannon distance is used as similarity measure between the candidate answers and the question given as input. This implementation performed better than the average scores.

1 Introduction

Question Answering Systems (QA-Systems) is a research field which has been improving since the 1950's. Answering open domain questions (questions which can belong to one or more topics in particular) is particularly difficult because of the vast amount of information that one has to analyze in order to retrieve a specific answer. This year the LiveQA track asks its participants to answer the latest unanswered question submitted to Yahoo! Answers. These questions are posted by real humans in real time. This means, that the question is not only an open domain question but also a complex question. Meaning, that the question will be verbose, will probably pertain multiple topics and, more often than not, it will contain repetitive information. Because of the details described above about these questions, implementing an automated way to answer them is an ongoing research field.

I chose a more statistical approach using, in the heart of the implementation, an unsupervised algorithm called latent dirichlet allocation (LDA). LDA helps in extracting the

topics and distribution of these topics in a set of documents. An LDA model still needs to be trained but it does not need any manual labeling of the data that is fed into the algorithm. This algorithm was chosen, mainly, because of the unknown nature of the questions that were going to be submitted throughout the LiveQA track. Using LDA is not enough to implement a QA-System, there is still the question of what information is given to the algorithm in order to train it. For this a Web Search is made using keywords extracted from the question given as input. The results are not only used to train the LDA model but also as an alternative corpus (set of documents) if the search results from Yahoo! Answers is not satisfactory. Using the distribution from the LDA model output the Jensen-Shannon distance (JSD) is calculated between every candidate answer and the given question. Finally, the candidate answer with the shortest JSD is chosen as the correct answer.

2 System Overview

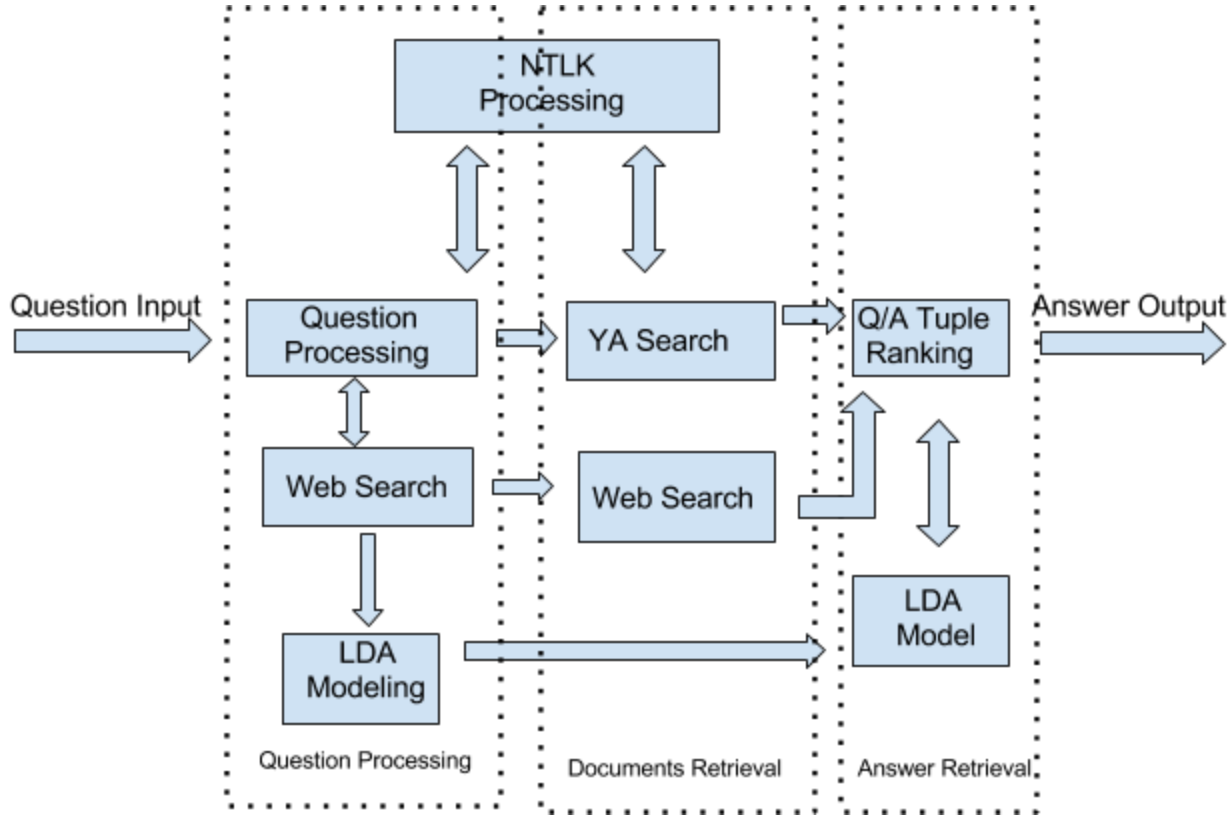

Figure 1

Figure 1 represents a block diagram of the QA-System architecture implemented in this paper. The “question processing” block uses the “NLTK processing” block to extract the keywords for further queries. The “NLTK processing” uses the python NLTK library to remove

stop words, punctuation and then uses a lemmatizer to normalize the keywords. This process takes place on every document in the corpus constructed. The “Web Search” block uses the Bing Web API and the previously processed question to extract document related to the given query. The “YahooAnswers Search” block is used to retrieve candidate related question/answer tuples from the YahooAnswers database. The “LDA Modeling” block uses the gensim python library to construct a model based on LDA (Blei et al. 2003). Gensim implements an online LDA model, meaning that the probabilistic variables are calculated with each iteration of the algorithm. The Answer ranking model feeds every candidate question/answer tuple into the “LDA Modeling” to get a topic probabilistic distribution of each of the question/answer tuples and then calculates the Jensen-Shannon distance (JSD) to get a similarity measure between the given question and the candidates question/answer tuples. Finally, the answer from the question/answer tuple with the shortest JSD to the given question is selected as the answer.

There are a few inferences made based on a quick analysis of the YahooAnswers database. First, it is observed that the title of each question includes the most important keywords referent to that question. This observation is used when searching for related documents using the Bing Web API. Second, it is noted that the first word usually describe the type of question that is being asked. For instance “Which teams are going to participate in March Madness?”, they word “Which” together with the rest of the keywords (“team”, “go”, “participate”, “march”, “madness”) yields better results when searching for related documents in the web. Third, when searching for candidate questions in YahooAnswers and using a large amount of keywords, the results are not very good and often times the only result is the given question itself. In order to get better results when searching YahooAnswers multiple searches are conducted by randomly removing keywords until the results yield at least 10 candidate related question/answer tuples. The steps of the QA-System implemented in this paper are the following.

2.1 Question Processing

The first block of the system is in charge of the question processing. When processing a given question the system retrieves the title and body of the question. Second, the question is process to extract the keywords by removing stop words, punctuation and lemmatizing the keywords using the NLTK package with the WordNet database. After extracting the keywords, these are fed into the Bing Web Search API and a set of 20 documents are processed the same way the question is processed. Then, an LDA model is constructed to infer the topics the given question contains.

2.2 Documents Retrieval

The keywords extracted from the question are then used as a query to the YahooAnswers service to retrieve a set of 50 candidate related questions. The retrieved questions and answers are used to construct the corpus, for this they are processed by removing stop words, punctuation and lemmatizing the keywords. If this query does not yields enough candidate question/answer tuples then results from the web search are incorporated into the corpus. The web search results are processed in the same way as the candidate question/answer tuples are in order to normalize the entire corpus.

2.3 Answer Retrieval

The LDA model is used to calculate the probability distribution of the topics for each document in the corpus. Then, the Jenssen-Shannon distance (JSD) divergence is calculated for each of the documents. The JSD is calculated as a similarity measure between pairs of question/answers from the corpus and the given question. The candidate related questions are then ranked from more related to less related. The system then returns the more related pair of candidate questions/answers pairs. With the top related candidate question/answer pair the more upvoted answer is then return as the result. The JSD was used a similarity measure based on the work of Celikyilmaz et al. (2010). The JSD measures the shannon entropy between two probability measures. The JSD is often used instead of the KL-Divergance because it is symmetric and, as such, a true metric.

3 Evaluation and Results

The LiveQA track is evaluated by TREC editors using a 4 level scale:

- 4: Excellent - a significatn 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: the answer is unreadable.

The performance measures are:

- avg-score(0-3) - average score over all queries (transferring 1-4 level scores to 0-3, hence comparing 1-level score with no-answer score, also considering -2-level score as 0)
- succ@i+ - number of questions with i+ score (i=2..4) divided by number of all questions

- prec@i+ - number of questions with i+ score (i=2..4) divided by number of answered only questions

Table 1 shows the results of the implementation described in this paper

Run	#Answers	avgScore (0-3)	succ@2+	succ@3+	succ@4+	prec@2+	prec@3+	prec@4+
UTRGV-JBC-TREC2016	882	0.7271	0.3704	0.2433	0.1133	0.4263	0.2800	0.1304
Average	771.0385	0.5766	0.3042	0.1898	0.0856	0.3919	0.2429	0.1080

Table 1

Table 2 shows the overall results.

Pl.	Run	#Answered questions	avgScore(0-3)	*succ*@2+	*succ*@3+	*succ*@4+
-	HumanQUAL	778	1.561	0.655	0.530	0.375
-	HumanSPEED	849	1.440	0.656	0.482	0.302
1	*Emory-EmoryCrowd*	976	1.260	0.620	0.421	0.220
2	CMU-OAQA	954	1.155	0.561	0.395	0.199
3	Emory-OutOfmEmory	995	1.054	0.519	0.355	0.180
4	YahooLabs-Q2A	798	0.996	0.465	0.343	0.188
5	QatarUniversity-QU3	1007	0.900	0.463	0.298	0.140
6	QatarUniversity-QU2	946	0.877	0.467	0.296	0.114
7	UniversityofMaryland-CLIP-YA	642	0.850	0.400	0.298	0.153
8	ECNU-ECNU	834	0.836	0.411	0.291	0.135
9	RMIT-RMIT-11	1008	0.786	0.428	0.252	0.106
10	QatarUniversity-QU	973	0.784	0.424	0.253	0.107
11	UTRGV-JBC-TREC2016	882	0.727	0.370	0.243	0.113
12	RMITUniversity-RMIT-1	1006	0.723	0.384	0.239	0.100
13	SFSU-IRSFSU	886	0.626	0.364	0.188	0.074
14	RMIT-RMIT-12	1012	0.447	0.273	0.137	0.037
15	PhilipsResearchNorthAmerica-prna	899	0.428	0.275	0.108	0.044
16	RMITUniversity-RMIT-2	1001	0.422	0.250	0.132	0.039
17	NUDT-NUDT681-3	627	0.375	0.187	0.126	0.062
18	NUDT-NUDT681-2	610	0.346	0.181	0.112	0.052
19	UWL-UWaterloo	387	0.292	0.191	0.081	0.020
20	EastChinaNormalUniversity-ECNUCS	749	0.274	0.187	0.067	0.020
21	NUDT-NUDTMDP2	314	0.236	0.116	0.083	0.037
22	NUDT-NUDTMDP1	262	0.232	0.117	0.080	0.034
23	DFKI-dfkiqa	260	0.112	0.072	0.033	0.008
24	UniversityofLeipzig-SMART	433	0.112	0.072	0.033	0.007
25	NUDT-NUDT681	824	0.071	0.043	0.022	0.006
26	NUDT-NUDT681-1	762	0.070	0.070	0.051	0.030
	Average	774	0.643	0.329	0.212	0.104

Table 2

4 Conclusions

As we can see my approach successfully scored above the average score. Looking at the average score does not gives us the complete story. If we pay attention to the overall results we can see that this implementation scored in #11 out of 26 runs. This tells us that here is still a lot of room from improvement, specially if we notice that the best performance score almost double as my implementation (1.260 vs 0.727).

Next year I will look into improving the different parts of this implementation. First, a better NLP processing for the input as well as the corpus. Second, a pre filter of candidate documents to train the LDA model as well as select the list of candidates question/answer tuples. Finally, adding more measurements to the answer selection other than JSD. Hopefully there will be time for all of these improvements.

References

Blei David, Ng Andrew, Jordan I. Michael. Latent Dirichlet Allocation. Journal of Machine Learning Research 3 (2003) 993-1022

Daniel Z, Ján S, Jozef J, Anton C. Text Categorization with Latent Dirichlet Allocation. Journal of Electrical and Electronics Engineering. 2014;7(1):161-4.

Li X, Ouyang J, Zhou X, Lu Y, Liu Y. Supervised labeled latent Dirichlet allocation for document categorization. Applied Intelligence. 2015;2014;42(3):581.

Asli Celikyilmaz , Dilek Hakkani-tur , Gokhan Tur. Lda based similarity modeling for question answering. In Proceedings of the NAACL HLT 2010 Workshop on Semantic Search.

T. Brants, F. Chen, I. Tsochantaridis, Topic-based document segmentation with probabilistic latent semantic analysis, in: Proceedings of the 11th International Conference on Information and Knowledge management (CIKM-02), 2002, pp. 211–218.

Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R. Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science (1986-1998). 1990;41(6):391.