Open-Domain Non-factoid Question Answering

Maria Khvalchik(✉) and Anagha Kulkarni

Computer Science Department, San Francisco State University,
1600 Holloway Ave, San Francisco, CA 94132, USA
{mkhvalch,ak}@sfsu.edu

Abstract. We present an end-to-end system for open-domain non-factoid question answering. We leverage the information on the ever-growing World Wide Web, and the capabilities of modern search engines to find the relevant information. Our QA system is composed of three components: (i) query formulation module (QFM) (ii) candidate answer generation module (CAGM) and (iii) answer selection module (ASM). A thorough empirical evaluation using two datasets demonstrates that the proposed approach is highly competitive.

Keywords: Question answering · Learning to rank · Neural network · BLSTM

1 Introduction

The popularity of QA websites such as Quora, and Yahoo! Answers highlights users' preference to express information needs as natural language questions, rather than keyword based queries. Currently, the person who posts the question has to wait for the answer until someone responds with the correct answer. Often the answer to the posted question is already out there on the World Wide Web (WWW), as an answer to a similar question, or embedded in the content of a web page. This observation was inspired evaluation forums such as the TREC LiveQA Track[1], and QALD Challenge[2] that are facilitating the research on the automated QA problem.

Developing an automated QA system that is capable of answering factoid or non-factoid questions from any domain (e.g. health, sports, cooking, etc.) is a challenging problem. In this paper we present our take on this problem. The end-to-end system that we have developed consists of three modules: (i) QFM converts the free-text question into a boolean query that can be processed by a commercial search engine; (ii) CAGM extracts all the promising candidate answers from the top ranked web pages returned by the search engine and (iii) ASM employs different ranking and classification approaches to select the best answer from the set of candidates. In the subsequent sections we describe our

[1] https://sites.google.com/site/trecliveqa2015/.
[2] https://project-hobbit.eu/challenges/qald2017/.

K. Ekštein and V. Matoušek (Eds.): TSD 2017, LNAI 10415, pp. 290–298, 2017.
DOI: 10.1007/978-3-319-64206-2_33

novel approach which combines several machine learning models together and is trained on a large sanitized dataset.

The subproblem of answering factoid questions using a static collection of documents has been researched since late 60's [6,12,20]. One of the recent examples is work by Bian et al. [4] where they built a framework which allows to extract facts from the data source and rank them. Their work defines a set of textual features some of which we use in our work. Suryanto et al. [18] proposed a similar method using the reputation of the question asker and the answerer to determine the relevance of the answer. Both papers are focused on factoid QA using Yahoo! Answers data. Our method is focused on non-factoid questions with using the entire Web as the data source.

Soricut and Brill [16] published one of the first papers on non-factoid question answering, and many others have followed [13,14,17]. As a training set they used a corpus of 1M question-answer pairs from FAQ collected on the Web. To search for the answer candidates they used MSNSearch and Google. Our work uses different algorithm for QFM, is trained using Yahoo! Answers dataset and uses learning to rank techniques which started to advance in mid-00s. In recent years the advancements in NLP/ML techniques and availability of large QA datasets have propelled research and contests on answering open-domain non-factoid questions [3]. Wang et al. [21,22] works were the winner of two subsequent TREC LiveQA competitions. In the first paper they trained an answer prediction model using BLSTM Neural Network. In the second - Neural Machine Translation techniques to train the model which generates the answer itself given only a question. We use their method as a baseline comparing our work against to.

2 Open Domain Factoid/Non-factoid QA System

We use a typical architecture for our QA system. The natural language question is transformed to a keyword based boolean query by the QFM. A commercial search engine, Bing, is then used to obtain the relevant web pages to the boolean query. CAGM mines the downloaded web pages for candidate answers to the original question. Finally, ASM identifies the best answer from all the candidates, and presents it to the user.

Query Formulation Module (QFM): The QFM transforms the natural language question to a well-formed boolean conjunctive query that can be evaluated by a search engine. This is a challenging problem as questions are often verbose. They contain information that is useful for a human but is superfluous, or even misleading, if included in the query. We address this verbosity problem at multiple levels. First, not every sentence in the question contributes to the query. Only sentences that start with WH-words (e.g. Who, When, Where, Why) and end with a question mark do [19]. Second, within a sentence only certain parts of the question are included in the query. For example, transforming the following question *Why's juice from orange peel supposed to be good for eyes?* into a boolean query: *(orange) AND (peel) AND (juice) AND (good) AND (eyes)* is not effective because most of the retrieved web pages are about *orange juice* and

not about *orange peel juice*. In order to achieve this, QFM performs grammatical analysis of the question. Specifically, we use the Stanford Dependency Parser [8] to obtain the grammatical structure of the sentence, and then apply a recursive logic to identify various phrases (noun, verb, preposition, and adjective phrases). This allows us to identify important phrases, rather than just individual words. For the above question this approach selects a noun phrase *orange peel*, an adjective phrase *good for eyes*, and a single word *juice*. The final boolean conjunctive query is constructed as follows: *(juice) AND (orange peel) AND (good for eyes)*. This query is successful at retrieving web pages about *orange peel juice* rather than about *orange juice* even though the latter has the more dominant presence on the web.

The English *closed class* terms (pronouns, determiners, prepositions) in the question are often ignored since they do not capture the topic of the question. However, in certain situations the prepositions should be included in the query. In case of the following question *How much should I pay for a round trip direct flight from NYC to Chicago in early November?*, if the preposition words, *from* and *to*, are ignored then the information about the travel direction is lost. To address this issue, the grammatical tree structure of the sentence is leveraged (using Stanford parser) to identify the preposition phrases, such as, *from NYC* and *to Chicago*, and these are included as-is in the boolean query.

The verb phrases are also important because the verb alone is too broad to be a standalone keyword in the query. For question *What should I have in my disaster emergency kit stored outside my house?*, without the verb phrase detection, the system generates *(disaster emergency kit) AND (outside house) AND (store)*. Some of the web pages retrieved by this query are about stores that sell disaster emergency kit. Whereas with verb phrase detection logic, a better query is generated: *(disaster emergency kit) AND (store outside house)*.

Candidate Answer Generation Module (CAGM): The boolean query created by QFM is executed against the commercial search engine, Bing. The top 20 web pages returned for the query are downloaded, and each page is passed through the following text processing pipeline. The first step extracts ASCII text from the web page using an html2text library[3]. We refer to the extracted text as a document. This document is next split into *passages*, where each passage consists of four consecutive sentences, the most popular answer length in Yahoo! Answers dataset. A sliding span of four consecutive sentences is used to generate the passages. Thus a document containing 5 sentences would generate two passages. This approach generates many passages, specifically, $1 + (n - 4)$, where n is the total number of sentences in the document. Passages that do not contain any of the query terms, or that contain more than 2 line breaks, or more than 10 punctuation marks, or non-printable symbols are eliminated. Also, passages that are not in English are filtered out. The langdetect library[4] is employed for language identification. All the passages that survive the filtering step are considered as *candidate answers*.

[3] https://pypi.python.org/pypi/html2text.
[4] https://pypi.python.org/pypi/langdetect.

Answer Selection Module (ASM): In this final step of the QA pipeline, the best answer from all the candidate answers is chosen. We experiment with three algorithms for this task: 1. Learning To Rank based LambdaMart algorithm [7], 2. Neural Network based BLSTM algorithm [11], and 3. A combination approach that employs both, LambdaMart and BLSTM.

There is a rich history of LeToR approaches being applied to automated QA [1,5,17]. Following on this tradition, for the baseline approach, we employ the LambdaMart algorithm to learn a ranking model for scoring the candidate answers, and the highest scored answer is selected as the final answer. We refer to this answer selection approach as LLTR. A subset of the *Webscope Yahoo! Answers L6* dataset[5] is used for training the LLTR model. For many questions in this dataset one of the answers for the question is identified as the best answer. For training LLTR the best answer is assigned the highest rank label, and the remaining answers are assigned a rank label proportional to their BM25 score with the best answer. The following feature set is computed for each <question, answer>pair: Okapi BM25 score, cosine similarity, number of overlapping terms, number of punctuation marks in the passage, number of words in the answer, number of characters in the answer, query likelihood probability, largest distance between two query terms in the answer, average distance between two terms, number of terms in longest continuous span, maximum number of terms matched in a single sentence, maximum number of terms in order. Before computing each of these features, all terms from query and candidate answer were stemmed using Porter.

Recurrent Neural Network (RNN) based approaches have received a lot of attention from the QA community recently [10,15,21,22]. Since carefully feature engineering is completely unnecessary for NNs these networks lend themselves very well to the QA problem where it is difficult to defining features that generalize well. In fact, the best performing system (Encoder-Decoder) at the TREC 2016 LiveQA track employed a recurrent neural network based approach. In our work we have employed the Bidirectional Long Short Term Memory (BLSTM) neural network because it adapts well to data with varying dependency spans length. The bidirectional property of this network allows for tracking of both, forward and backward relations in the text. We use a modification of network architecture implemented in [21]. The network consists of several layers: the word embedding layer followed by BLSTM layer, dropout layer to reduce overfitting, mean pooling, and dense layer for the output. The output for the network is a number from 0 to 1 identifying how likely the answer matches the question. It was trained with *ADAM* optimizer, with binary cross-entropy as a target loss function. To train the network a subset containing 384K <question, answer>pairs from the *Webscope Yahoo! Answers L6* dataset was used.

The third answer selection approach that we investigate simply combines the above two approaches. The score assigned by BLSTM to each <question, answer>pair is used as an additional feature in the feature set used by the LLTR ranking algorithm.

[5] http://webscope.sandbox.yahoo.com.

3 Experimental Setup and Evaluation Data

In order to compile the subsets of L6 dataset that are used to train the LLTR and BLSTM models, we used two steps to filter out low-quality question-answers. We were discarding: questions with less than 3 answers, questions (or answers) that were too small (less than two sentences) or too long (greater than 1000 characters). A subset of 48,000 question-answers from the L6 dataset was used to train the LLTR ranking model[6].

For training BLSTM, the answers voted as the best one for the questions were assigned the positive label, and answers for another random question were assigned the negative label. For the embedding layer we used the pretrained Google News word2vec model[7]. It was found that the most efficient to use the word2vec vector size of 200 with 15000 most popular words (i.e. all words except these are discarded). We couldn't use more words or word2vec dimensionality because of the overfitting. The input size for BLSTM was 128 words and the dropout was set to 0.5.

To evaluate the ASM we employed the LiveQA track data from TREC 2015 and TREC 2016, which both contain the answers from all participant systems for approximately 1000 questions. Each answer is rated by human judges on the scale from 1 (poor) to 4 (excellent). The effectiveness of the three answer selection approaches was evaluated with the above two datasets. Standard evaluation metrics were used for this task: NDCG (Normalized Discounted Cumulative Gain), MAP (Mean average precision) at rank X, and MRR (Mean Reciprocal Rank) at rank X. As a point of reference we also define a baseline QA approach: the original question is used as-is for the query (stopwords excluded), the top web page retrieved by Bing is downloaded, and the passage with highest BM25 score with respect to the question, is selected as the final answer.

4 Results and Analysis

Table 1 provides the results for the evaluation of the ASM. For the TREC 2015 LiveQA evaluation set, the results for the best performing system for this task are available [22], and are included in the table (Encoder-Decoder). LLTR is less effective than the state-of-the-art approach, Encoder-Decoder, across all the metrics. However, the neural network based approach, BLSTM performs substantially better than Encoder-Decoder and LLTR for both datasets. The results for LLTR+BLSTM illustrate that two approaches have complementary strengths that can be combined to obtain the best results for the task. The difference between LLTR and LLTR+BLSTM is statistically significant.

We believe that quality of the model can be improved by sanitizing the training dataset. Currently, two main problems are: (i) presence of words with misspellings which make computations of statistical features imprecise; (ii) quality of the best answers manually selected by voters. There exists a few approaches

[6] The datasets will be shared after publication.

[7] https://code.google.com/archive/p/word2vec/.

Table 1. Results of answer ranking

	NDCG	MAP@			MRR@		
		2	3	4	2	3	4
TREC 2015							
Encoder-Decoder	0.6346	0.5124	0.3390	0.1657	0.5645	0.3672	0.1779
LLTR	0.6222	0.4843	0.3162	0.1551	0.5490	0.3522	0.1562
BLSTM	0.6562	0.5462	0.3470	0.1744	0.5874	0.3790	0.2046
LLTR+BLSTM	0.6602	0.5498	0.3487	0.1763	0.5901	0.3810	0.2059
TREC 2016							
LLTR	0.6484	0.5124	0.3463	0.2165	0.6211	0.3806	0.2410
BLSTM	0.6712	0.5591	0.3788	0.2541	0.6478	0.4033	0.2879
LLTR+BLSTM	0.6754	0.5674	0.3835	0.2567	0.6504	0.3990	0.2928

to diminish impact of both issues such as [9] for misspellings and [2] for keeping only high-quality answers.

Evaluating the end-to-end QA system is tricky because the generated answer might change if the search engine results change, and thus manual assessment of answer relevance cannot be a one-time activity. As a compromise, we attempt to provide quantitative evaluation by computing similarity between the answer generated by our system and the best answer assessed by TREC annotators. The intuition being that higher the similarity score the more effective the system is.

Table 2. Overall system quality results

Metric name	Jaccard coefficient			Cosine similarity			KLD similarity		
Question length (in tokens)	[0;10]	[11;32]	[33;)	[0;10]	[11;32]	[33;)	[0;10]	[11;32]	[33;)
Number of questions	134	152	148	134	152	148	134	152	148
BM25	0.05 ± 0.01	0.05 ± 0.01	0.06 ± 0.01	0.07 ± 0.01	0.08 ± 0.02	0.07 ± 0.01	0.43 ± 0.05	0.48 ± 0.05	0.44 ± 0.06
LLTR	0.12 ± 0.02	0.11 ± 0.01	0.11 ± 0.02	0.23 ± 0.03	0.25 ± 0.02	0.25 ± 0.03	0.64 ± 0.04	0.65 ± 0.05	0.67 ± 0.06
BLSTM	0.13 + 0.02	0.12 ± 0.02	0.12 ± 0.02	0.23 ± 0.02	0.26 ± 0.03	0.26 ± 0.04	0.65 ± 0.05	0.65 ± 0.06	0.68 ± 0.06
LLTR+ BLSTM	0.13 ± 0.02	0.13 ± 0.02	0.13 ± 0.02	0.24 ± 0.03	0.26 ± 0.03	0.27 ± 0.04	0.67 ± 0.04	0.68 ± 0.05	0.70 ± 0.05

Table 3. Answers returned by the system compared to best human answers

Question: How do you fix a computer programme that run for itself instead of serving you? **System's Answer:** Run AdwCleaner. Click the "Scan" button in AdwCleaner to scan your computer for infections. This may take 20 min or so to complete. Once the "Clean" to remove any infections that AdwCleaner found. **Best Human Answer:** More information would be useful.
Question: What's the best tablet to get? **System's Answer:** 7-inch Apple iPad Pro is the best tablet yet. We love the sleek design and stunning True Tone display, which makes the most of the 1 million tablet-specific apps available in the iOS App Store. Not only can this slate's 12-megapixel camera shoot in 4K, the A9X processor inside can also edit multiple 4 K streams without a hiccup. Plus, this more portable Pro tablet lasts a remarkable 10 h and 53 min on a charge, and it supports the $99 Apple Pencil. **Best Human Answer:** Transformer prime if your looking for customization and uniqueness. iPad if you like it simple.
Question: Should the WWE hire AJ Styles? What do you think? **System's Answer:** WWE would hire Sting before they hire AJ Styles.not only that.they'd make him cover his tattoo of "AJ" and then change his name and make out of some NXT rookie and give him some generic name like, "Kyle Evans". AJ isnt a rookie and shouldnt be treated as such. **Best Human Answer:** WWE has tried several times to sign AJ Styles. AJ Styles has turned WWE down every time. AJ Styles can make more money working for New Japan Pro Wrestling, Ring of Honor and Independent Professional Wrestling Promotions all over the world. AJ Styles works where he wants, when he wants and as often as he wants.
Question: Can midline shift be corrected by elastics? I got upper braces and I'm getting the bottom ones soon. **System's Answer:** You should let your dentist know so he can correct his own work (that's what you paid him for, right?) or at least ask for more rubber bands. I wanted to prevent my midline from shifting due to my braces but my dentist waited to correct mine, so I certainly understand your concern. I began asking for extra rubber bands of the kind I needed at every dentist visit so I could start correcting it on my own. **Best Human Answer:** What you need to realize is that what "a lot of people" have had has nothing to do with what you need. Every case is different. Patient "A" can have a midline shift and will be corrected with elastics. Patient "B" can have a midline shift that may need surgery. There are many other factors, including the formation of the jaw, that come into play. Mention using elastics to your orthodontist and s/he will be able to tell you, and show you in x-rays, why surgery is being recommended for your particular situation.
Question: Which is worse? Gambling or cheating with PEDs on the game of baseball? Why? **System's Answer:** That gambling creates the moral hazard of being potentially so in hock to bookies that they require you to throw a game – something that apparently didn't happen to Rose but quite easily could have – must be noted as well. No one takes PEDs to lose or only takes them before important games. Results are more important than statistics. This would seem self-evident, but it's a claim Rose makes explicit. **Best Human Answer:** PEDs. using a illegal substance and not working hard to improve your craft is worse. Gambling is just stupid, as there's still a chance you could gamble wrong

The absolute similarity scores are not the focus, but the relative difference in system scores is worth analyzing.

Table 2 reports the similarity score using Jaccard, Cosine, and Negative & Symmetric KL Divergence metrics for the TREC 2015 LiveQA evaluation dataset. The presented results are also binned based on question-length, so as to analyze the effects of question-length on the system's performance. Compared to the baseline (BM25-based QA system), all the versions of our system perform better, with the combination approach (LLTR+BLSTM) being the best one. This trend is seen across the three metrics, and the different query-lengths. In future we plan to employ more powerful metrics, such as, METEOR[8] and doc2vec[9] that can capture synonyms and paraphrasing. In Table 3 we present examples of the answers given by our system for some of the dataset questions. In some cases, such as a question about computer programs the QA system produces more comprehensive answer than humans. Also, it is worth noting that the quality of the questions varies dramatically which is one of the big challenges for this problem.

5 Conclusions and Future Work

In this paper we presented our attempt at tackling the challenging problem of answering open-domain non-factoid questions. The empirical evaluation illustrates that the simple approach of combining LLTR and BLSTM outperforms the state-of-the-art system. The qualitative evaluation shows that the system is capable of producing high-quality answers. Possible directions for future research include use of recurrent neural networks for summarizing the question and answer to generate better queries and more concise answers[10]. The quality of the training dataset could also be improved to increase the performance of answer selection models.

References

1. Agarwal A., et al.: Learning to rank for Robust question answering. In: Proceedings of CIKM (2012)
2. Agichtein E., et al.: Finding high-quality content in social media. In: Proceedings of WSDM (2008)
3. Agichtein E., et al.: Overview of the TREC 2015 LiveQA track. In: Proceedings of TREC (2015)
4. Bian J., et al.: Finding the right facts in the crowd: factoid question answering over social media. In: Proceedings of WWW (2008)
5. Bilotti M.W., et al.: Rank learning for factoid question answering with linguistic and semantic constraints. In: Proceedings of CIKM (2010)

[8] http://www.cs.cmu.edu/~alavie/METEOR/.
[9] https://radimrehurek.com/gensim/models/doc2vec.html.
[10] https://research.googleblog.com/2016/08/text-summarization-with-tensorflow.html.

6. Bobrow D.G.: A question-answering system for high school algebra word problems. In: Proceedings of FJCC (1964)
7. Burges C.: From ranknet to lambdarank to lambdamart: an overview. Learning **11**, 81 (2010)
8. Chen D., Manning, C.: A fast and accurate dependency parser using neural networks. In: Proceedings of EMNLP (2014)
9. Chen, Q., Li, M., Zhou, M.: Improving query spelling correction using web search results. In: Proceedings of EMNLP-CoNLL (2007)
10. Cohen, D., Croft, B.: End to end long short term memory networks for non-factoid question answering. In: Proceedings of ICTIR (2016)
11. Graves, A., Schmidhuber, J.: Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Netw. **18**(5), 602–610 (2005)
12. Green, C.: Theorem proving by resolution as a basis for question-answering systems. Mach. Intell. **4**, 183–205 (1969)
13. Higashinaka, R., Isozaki, H.: Corpus-based question answering for why-questions. In: Proceedings of IJCNLP (2008)
14. Oh, J.H., et al.: Why question answering using sentiment analysis and word classes. In: Proceedings of EMNLP-CoNLL (2012)
15. Severyn A., Moschitti A.: Learning to rank short text pairs with convolutional deep neural networks. In: Proceedings of SIGIR (2015)
16. Soricut, R., Brill, E.: Automatic question answering using the web: beyond the factoid. Inf. Retrieval. **9**, 191–206 (2006)
17. Surdeanu M., et al.: Learning to rank answers to non-factoid questions from web collections. Comput. Linguist. **37**, 351–383 (2011)
18. Suryanto, M.A., et al.: Quality-aware collaborative question answering: methods and evaluation. In: Proceedings of WSDM (2009)
19. Varanasi, S., Neumann, G.: Question/answer matching for Yahoo! Answers using a corpus-based extracted ngram-based mapping. In: Proceedings of TREC (2015)
20. Waltz, D.L.: An English language question answering system for a large relational database. Commun. ACM. **21**, 526–539 (1978)
21. Wang, D., Nyberg, E.: CMU OAQA at TREC 2015 LiveQA: discovering the right answer with clues. In: Proceedings of TREC (2015)
22. Wang, D., Nyberg, E.: CMU OAQA at TREC 2016 LiveQA: an attentional neural encoder-decoder approach for answer ranking. In: Proceedings of TREC (2016)