Meanings are more than just words: A Cross-Domain Question Answering Tool based on Unsupervised Semantic Feature Learning

[1]Nripa Chetry, [2]Debanjan Choudhary, [3]Arindam Chatterjee, [4]Dr. Gopichand Agnihotram

Wipro Limited Doddakannelli, Sarjapur Road Bangalore - 560 035

Abstract

The state-of-the-art Question Answering (QA) systems, either focus on the similarity in occur-rences of words, or similarity of passages. In this paper, we present a novel QA system, which strikes a harmony between word based similarity and context similarity based on short texts, rather than entire passages. Our system can be vaguely categorized as a cross domain FAQ based QA system. It tracks word based similarity through Latent Semantic Indexing (LSI) and then re-ranks the LSI results using an eXtreme Gradient Boosting(xgboost) classifier model. Sev-eral features trained from word embedding vectors, learned from the domain corpus are fed into the xgboost classifier. These features capture the semantic understanding of the questions or headings/sub-headings in our knowledge base. We have observed from our experiments that us-ing latent semantic indexing and re-ranking these results using the classifier gives better MRR at top 3 than information retrieval techniques. Its performance is comparable (if not better) to most state-of-the-art QA systems across domains.

1 Introduction

Question Answering (QA) systems understand natural language queries and respond with actual an-swers in natural languages. Knowledge based QA systems are generally based on Semantic Parsing, Information retrieval or Open Information Extraction. Current state-of-the-art QA systems make use of Information retrieval and Deep Learning techniques, but these (Kenter and de Rijke, 2015) are open domain systems, and dont generalize across all domains, as such natural language applications are do-main dependent. The domains that we are concerned with are mainly for commercial applications and the questions primarily focus on troubleshooting issues and informational queries. The task of our QA system is to provide the solution from the knowledge base, relevant to the issue faced by the user. Let us look at a few sample queries to our QA system:

> "It is taking too much time after a key press. Why?"
> "How can I get my system ID from the terminal?"

The answer to the queries contains a section from the manual or knowledge base and the relevant steps to be performed to solve the issue. For example, for the first question, the relevant section from the man-ual is: "Key presses are slow to respond". As our system is domain specific and contains numerous technical terms that germane to our domain, from the above example it might seem that using simple semantic parsing, ontology and information retrieval techniques might serve our purpose. Con-versely though, the pertinent section to the second question is: "Finding the serial number of system from command prompt". From this example, it is evident that key word based simi-larity measures, in our case will not suffice. We need to augment this information with a deep semantic understanding of the user query, in order to present the user with the correct manual section/sub-section.

In this paper, we have proposed a novel architecture for a question answering system using LSI and consequently re-ranking the retrieved answers from the LSI index, based on semantic similarity of the query and the manuals headings. We trained a classifier with features based on several metrics of se-mantic relatedness of sentences and apply the same between the query and the manuals headings/sub-headings to get the most relevant section of the manuals in the knowledge base for a particular user query. We would like to assert that the type of our knowledge base is quite different from ones used in (Wang and Ittycheriah, 2015) and (Kenter and de Rijke, 2015). Hence we have have come up with a novel architecture that works well across domains.

The remaining paper is organized as follows: In section 2 we explain our system in details. Sections 3 and 4 pertain to the experimental setup and subsequent results and observations. We conclude in section 5 with conclusion and future work.

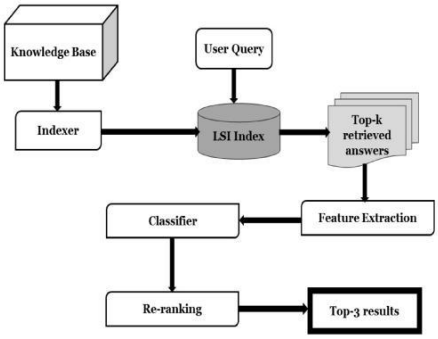
2 The System

In this section we describe our QA system in detail. We first exhibit the system architecture, followed by the two phases in which our system operates, namely LSI and xgboost based re-ranking.

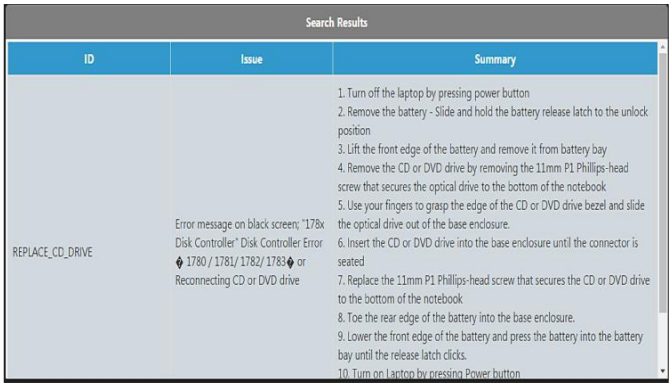
2.1 System Architecture

Our proposed system contains two stages:

1. An information retrieval system based on LSI
2. A classifier to re-rank the answers/documents retrieved through LSI



(a) System Architecture Diagram.



(b) A snapshot of the QA tool.

Figure 1: Block Diagram and Snapshot of the QA system.

Figure 1a shows the block diagram of the system. We index different sections/sub-sections of the product manuals in the knowledge base and fetch relevant ones for every user query using latent semantic indexing. We train the xgboost classifier based on the semantic features from the user query and the retrieved documents from LSI as shown in 1a, using word embeddings and consequently re-rank them to get the final list of relevant answers. The next two sections enumerate the two phases of our QA system in more detail. 1b shows a snapshot of our system.

2.2 Latent Semantic Indexing

Vector space information retrieval systems rely on measuring the similarity between the term-document matrix and the vector formed from the query. The similarity measurement is generally done by measuring the cosine distance between the vectors of the queries with those of the documents. Latent semantic Indexing, is a method which relies on representation of the documents in the collection into a new, low rank matrix space using Singular Value Decomposition (SVD) (Deerwester et al., 1990). To cope with synonymy and polysemy LSI uses SVD to form a low rank representation k of the term document matrix, where the documents are represented in a different space and the synonyms and polysemous words are captured based on the co-occurrence of the terms in similar contexts. The query from the user is represented in the k-dimensional LSI space and then the cosine distance is calculated with the transformed matrix, which handles both synonymy and polysemy.

2.3 Re-ranking: The xgboost classifier model

In this section we elaborate on how we train our eXtreme Gradient BOOSTing(XGBOOST) model and the set of features we use to capture semantic relatedness.

For a given query Q and top 10 relevant answers A0; A1; : : : ; A9, we calculate the semantic similar-ities of the query and the section headings of the answers based on word alignment. If we denote the query words as q_i and the answer words as a_j , the similarity between two words is given by equation (1)

$$sim(q_i; a_j) = max(0; cos(v_{qi}; v_{qj})) \quad (1)$$

Where v_{qi} and v_{qj} are the vectors of the terms in the query and answers. Then we obtain the similarity matrix for the question-answer pair by calculating the similarity of all such pairs, as described in (1) to measure the alignment and semantic similarities. The features used for this model are:

1. Similarity: which measures the pair similarity based on aligned words.
2. Dispersion: measures the contiguous query and answer pair words.
3. Penalty: which penalizes the unaligned words in the pair.
4. 5 important words: which consists of 5 features consisting of alignment scores of the words based on their inverse document frequencies.
5. Cosine distance between the averaged word vectors of the query and answer pairs.
6. The semantic similarity feature as mentioned in equation (3).

$$f_{sts}(s_l; s_s) = \sum_{w \; s_I} IDF(w): \frac{sem(w; s_s):(k_1 + 1)}{s_{sj}} \quad (2)$$

Where, s_l is the longest text of the two, s_s is the corresponding short text and avg_{sl} is the average length of the sentence in the training corpus. The semantic similarity of the term w with respect to short text is represented by sem(w; s) as shown in equation (3)

$$f_{sem(w;s)} = \max_{w \; s} f_{sem}(w; w^0) \quad (3)$$

The function f_{sem} measures the semantic similarity between two terms which we have used as cosine distance between the term vectors. Values of k_1 and b used are 1.2 and 0.75 respectively as suggested in (2). We train a xgboost classifier with the above mentioned seven features with classes relevant (1) or irrelevant (0). We use 'AUC' as the evaluation metric and objective function as 'binary: logistic. We re-rank the queries only if the probability of any of the answers from the candidate set is greater than 0.4 which is our observed confidence threshold value.

3 Experimental Setup

We have trained the system on a corpus of 814 documents and 784 queries. For each query, there are 10 relevant documents retrieved, using LSI. The rank of the document term matrix used in the experiments is k = 300. The total training set consists of 7840 query-answer pair. As this is a real-time system, we have considered only one relevant/correct answer. From the training set, we tag relevant answers as 1 and irrelevant ones as 0. For cross domain, the training data is taken from laptop domain and re-ranking is done on printer domain, which is our test data.

4 Results and Observations

We index the section and sub-sections from product manuals as separate documents, the manuals can be both in pdf or html formats. The knowledge base is created by parsing the latter manuals. From this indexed documents, we firstly get the top 10 relevant answers/documents using Latent Semantic Indexing from the knowledge base, which decrease the candidate search space, and then re-rank the answers based on a xgboost classifier based on semantic similarities. As our QA system is developed to return the most relevant answer from a set of candidate answers, the metric we use is MRR at top 3. The results we obtained are for:

1. MRR at top 3 from Lucene (term document matrix)
2. MRR at top 3 on LSI and
3. MRR at top 3 from re-ranking the candidate answers using xgboost on 20-fold cross validation

Method	MRR @ Top 3	
	SAME Domain	ACROSS Domain
Bag of Words (Baseline)	0.316	0.324
Lucene(tf-idf document term matrix)	0.479	0.417
Latent Semantic Indexing	0.529	0.611
Latent Semantic Indexing + xgboost	0.566	0.614

Table 1: MRR@Top 3 results.

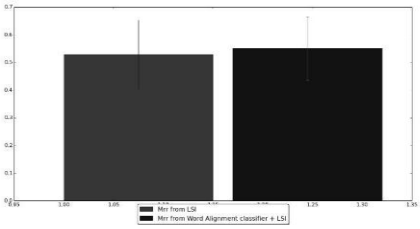


Figure 2: Error bars from LSI and LSI + xgboost on 20-fold CV

The error bars from experimental methods 2 and 3 are shown in figure 2. The left bar is for LSI and right bar is for LSI + classifier(xgboost) using semantic features. Table 1 shows that LSI+Classifier method achieves significant accuracy above baseline, which is Bag of Words match, and is comparable to state-of-the-art QA systems. It can be observed that there is a significant increase in MRR using LSI than lucene based indexing, because the latter does not capture the semantic similarity of the terms in the documents. Table 1 also reports the performance of the same using the same metric on 20 fold cross validation set. From Table 1 it is also evident that our system performs better than state-of-the-art cross domain QA systems. There is not much improvement in the MRR value, but that can be attributed to the fact that a MRR of more than 0.5 already confirms that most relevant documents are already present in the top 2 results.

5 Conclusion and Future Work

In this paper, we have suggested a Question Answering system based on semantic understanding of queries across domains in specific knowledge bases. We have also reported that the classification model trained in one domain can be reused in other domains too only using the semantic features. That is because they will be learned in an unsupervised way from domain data, provided we have enough data for training the word embeddings.

We want to further modify our system by trying to grasp an understanding of the entities con-tained in the user queries and add features in the classifier, based on entities present in the top-10 rel-evant/candidate answers. This should further increase the performance of our system both for same domain and across other domains.

References

1) Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. 1990.

2) Indexing by latent semantic analysis. JASIS, 41(6):391–407.

3) Tom Kenter and Maarten de Rijke. 2015. Short text similarity with word embeddings. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, CIKM '15, pages 1411–1420, New York, NY, USA. ACM.

4) Zhiguo Wang and Abraham Ittycheriah. 2015. Faq-based question answering via word alignment. CoRR, abs/1507.02628.