

# Open Domain Real-Time Question Answering Based on Semantic and Syntactic Question Similarity

Vivek Datla, Sadid A. Hasan, Joey Liu, Yassine Benajiba\*

Kathy Lee,<sup>†</sup> Ashequl Qadir, Aaditya Prakash,<sup>‡</sup> Oladimeji Farri

Artificial Intelligence Laboratory, Philips Research North America, Cambridge, MA, USA

{vivek.datla, sadid.hasan, joey.liu}@philips.com, yassine@lukilabs.com  
{kathy.lee-1, ashequl.qadir, aaditya.prakash, dimeji.farri}@philips.com

## Abstract

In this paper, we describe our system and results of our participation in the Live-QA track of the Text Retrieval Conference(TREC) 2016. The Live-QA task involves real user questions, extracted from the stream of most recent questions submitted to the Yahoo Answers (YA) site, which have not yet been answered by humans. These questions are pushed to the participants via a socket connection, and the systems are needed to provide an answer which is less than 1000 characters length in less than 60 seconds. The answers given by the system are evaluated by human experts in terms of accuracy, readability, and preciseness. Our strategy for answering the questions include question decomposition, question relatedness identification, and answer generation. Evaluation results demonstrate that our system performed close to the average scores in question answering task. In the question focus generation task our system ranked fourth.

## 1 Introduction

Question Answering(QA) is a well-studied research area in natural language processing (NLP). Since the early days of artificial intelligence in the 60's, researchers have been fascinated with answering natural language questions (Kwok et al., 2001). Initial efforts for QA systems primarily focused on

domain-specific expert systems. The domain specific factoid questions have been answered well and the systems have achieved similar performance as human experts, where as answering open-domain questions in natural language is still an open challenge. The open-domain real life questions amplify the challenge many folds as natural language is ambiguous, and constructing the answer requires an elaborate understanding of the question being asked, expert domain knowledge, as well as language generation models.

The open domain real-time question answering task increases the complexity even further as one has to address the issues as mentioned previously and in addition to producing human-like response in less than 60 seconds. The properties of human-like response include structured grammatically correct sentences, which answer the question to the satisfaction of a human evaluator. Additionally, the answers need to be concise as they are restricted to a 1000 character limit. This is our first participation in the live-QA track and in the following sections we describe our model, results, and experiences.

## 2 Task Description

The LiveQA track was first started in TREC 2015. The competition runs for 24 hrs during which questions being posted on Yahoo Answers site<sup>1</sup>(after some preliminary cleaning) by the real users are posted on to the participating team's servers registered for the competition. The questions are selected from 7 distinct topics shown in Table 1

As Table 1 indicates, the topics are fairly different and have several sub-categories. The category of the

\*This author was affiliated with Philips during this work.

<sup>†</sup>The author is also affiliated with Northwestern University(kathy.lee@eecs.northwestern.edu)

<sup>‡</sup>The author is also affiliated with Brandeis University (aprakash@brandeis.edu).

<sup>1</sup>Yahoo Answers - <https://answers.yahoo.com/>

Table 1: Topic categories and no. of sub categories

Topic	#sub topics
Arts & Humanities	10
Beauty & Style	5
Health	10
Home & Garden	6
Pets	8
Sports	30
Travel	27

question being asked is selected from a predefined list by the person asking the question. The user selects only one category for the question being asked. The category of the question may overlap with other categories. For example “*Why does Labor back this kind of behavior?*” is identified by the user as in the topic category “*Travel*” and to the sub-topic category “*Australia*”. However, from the question, we can understand that it belongs to the category “*Politics and Government*”. Sometimes the questions belong to multiple categories and the user based on his interest/convenience picks only one topic.

The questions being asked in Yahoo Answers are mostly subjective and describe a human experience which are often personal and relevant to the topic. The ability for a machine to replicate human understanding of the topic(s) and biases in a subjective question is a challenge. Also, the fact that these questions can represent multiple events and potential causal relationships further complicates the LiveQA task. For example, the question “*My fiance hates my dog. He ignores him and always complains. He started calling him names. Should I be worried?*” posted in the Pets topic shows three parties (me, my fiance, and my dog) with a mix of interpersonal relationships represented as emotions and actions (“hates my dog”, “ignores him”, “always complains”, “started calling him names”, and “should I be worried”). The answers given by people for this question include suggestions on personal relationships, pet behaviors, and further questions like “*Do you want to stay with the person who is cruel to animals?*”.

As the answers given to the question indicate that there is no one correct answer and answers provided by users indicate the different focus picked while

answering the questions. To answer such questions one needs not only to know about the sentiment and the focus of the question, but also needs to know the interactions between various facets of the problem. Given these open-domain questions, the big challenge that we need to address is how to create huge domain knowledge to answer such questions.

In addition to the main question answering task, a new pilot subtask was introduced this year for identifying the focus of the question. The goal of the task is to test whether the system understood the question. The task focuses on identifying focus words of the question that indicate the most important phrases in the question. The focus words can be generated from the question title and the body of the question. We used a localized keyword extraction methodology to identify the important words in the question.

### 3 TREC 2016 Competition

During the TREC competition, a question is pushed every minute by the track organizers onto the server registered with the competition. The question posted on our server is a JSON object with fields shown in Table 2. The category, sub-category and the question-title fields always have entries, while the other fields can sometimes be empty.

The questions from the topics shown in Table 1 sometimes relate to current events, and hence it becomes particularly challenging to address the questions on personal experiences related to current events, pandemics, or on going family issues like marriage, divorce, etc.

If, after 60 seconds there is no response provided from the server to the competition, then the response is assessed as negative, and would be penalized. The systems are ranked on two metrics 1) *success*: ratio of the aggregated scores of the answers to the total questions asked in the competition and 2) *precision*: ratio of the aggregated scores of the answers to the total number of the questions answered by the system.

### 4 System

Our LiveQA system comprises of the following modules:

- Question cleaning

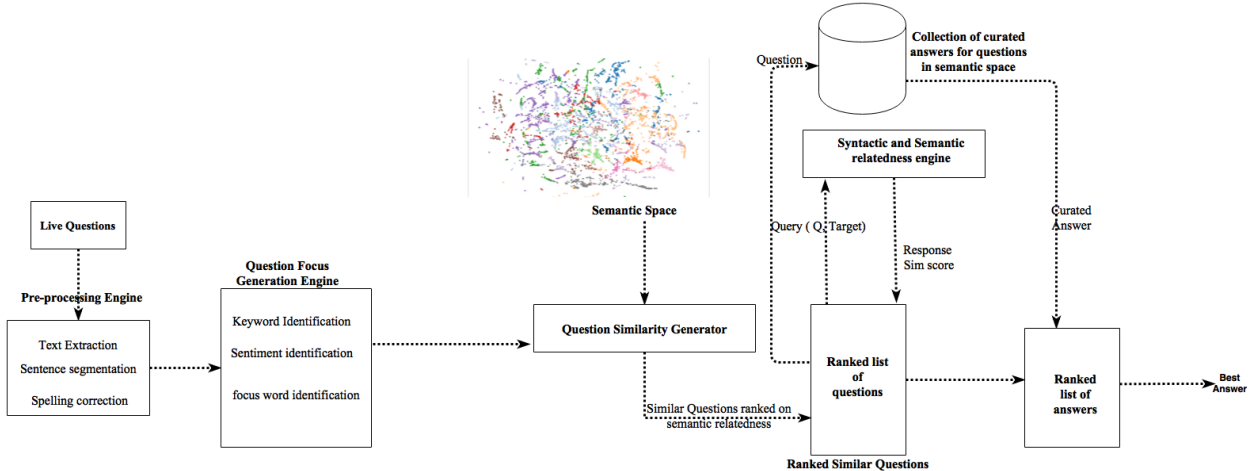


Figure 1: System Architecture

Table 2: Question Fields

Category	Topic of the question
Sub-category	sub-topic of the question
Question body	description of the question
Question title	actual question posted

- Question focus generation
- Question Similarity based on latent semantic analysis (LSA) (Landauer, 2007)
- Semantic Similarity based on WordNet (Fellbaum, 1998)
- Answer selection module
- Pushing the answer response

Figure 1 shows the flow of the several modules used in our system. For question cleaning, we focused on the removal of noisy characters (emojis, repeated characters, html tags etc.), correction of spellings, and sentence segmentation. The language used in the questions is informal, as expected in social media (Twitter, Facebook, etc.). We used similar steps in cleaning the body and title of the question. After cleaning the question, we perform question decomposition by focusing on keyword extraction, sentiment analysis and focus word generation. Keyword extraction is done using a localized term frequency inverse document frequency(TF-IDF) based model (Rose et al., 2010).

The keywords indicate the important words and the sentiment is identified with respect to these keywords. These keywords are used as our focus words while constructing the answer for the question. For identifying the sentiment we used Vader sentiment analysis tool (Hutto and Gilbert, 2014).

## 4.1 LSA-based Question Similarity

### 4.1.1 Building LSA Space

Latent Semantic Analysis (LSA) is a statistical language algorithm that captures semantic relations by mapping initially meaningless words into a continuous high dimensional semantic space (Landauer, 2007). More specifically, a first-order process associates stimuli (words) and the contexts they occur in (documents). Stimuli are paired based on their contiguity or co-occurrence in the document. These local associations are next transformed by means of Singular Value Decomposition (SVD) into a small number of dimensions (typically 300) yielding more unified knowledge representations by removing noise (Hutchinson et al., 2012; Datla et al., 2012).

For instance, if there are  $m$  terms in  $n$  documents, a matrix of  $A = (F_{ij} \times G(j) \times L(i, j))_{m \times n}$ , was obtained. The value of  $f_{ij}$  is a function of the integer that represents the number of times term  $i$  appears in document  $j$  :  $L(i, j)$  is a local weighting of term  $i$  in document  $j$ ; and  $G(j)$  is the global weighting for term  $j$ . The matrix of  $A$  has, however, lots of redundant information. Singular Value Decomposition (SVD) reduces this noise by decomposing the

matrix  $A$  into three matrices  $A = U\Sigma V^t$ ; where  $U$  is a  $m \times m$  and  $V$  is a  $n \times n$  square matrix, with  $\Sigma$  being an  $m \times n$  diagonal matrix with singular values on the diagonal (Hutchinson et al., 2012).

By removing dimensions corresponding to smaller singular values, the representation of each word is reduced as a smaller vector with each word now becomes a weighted vector on 300 dimensions, with only the most important dimensions that correspond to larger singular values being kept (Landauer, 2007). The semantic relationship between characters can then be estimated by taking the cosine distance measure between the two feature vectors.

We built an LSA space using the Yahoo 4.4 million question answer corpus<sup>2</sup>. We selected all the question titles from the corpus and cleaned them using stop-word removal and stemming. Each question represents a document in our model. Using *gensim* (Řehůřek and Sojka, 2010) we built the semantic space with 300 dimensions. We retrieve the semantically similar questions to a given question with the trained LSA model. We then set a threshold of 0.70 for the cosine similarity score such that all answers above this score are candidates. We selected this threshold value after experimentation and expert opinions.

The output of the LSA module is a ranked list of candidate questions which are semantically similar. Despite being semantically similar, the list of candidate questions may not be related in terms of polarity and subject-object relationship as LSA is a bag-of-words model. In order to extract semantically and syntactically similar questions we filtered the candidate questions further based on similar keywords and word order.

## 4.2 Keyword Extraction

Localized keyword extraction based on pre-trained term frequency inverse document frequency (TF-IDF) scores helped identify the important words in the question. These words are used to get the word overlap score and re-rank the questions obtained from the LSA model. We used the Rake software for keyword extraction (Rose et al., 2010).

<sup>2</sup>L6 - Yahoo! Answers Comprehensive Questions and Answers version 1.0 (multi-part)

## 4.3 WordNet based Semantic and Syntactic Similarity

We used the method proposed by (Li et al., 2006) to augment the similarity measure between the questions. The method is heavily dependent on the word-order and uses WordNet to identify the strength of the relationship among the words. The words belonging to the same synset (synonymous words conveying the same sense) have a higher weight than the words belonging to different synsets. Also, if the words have a hypernymy or hyponymy relationships, then the weights are lower compared to the synonyms.

By computing the similarity of the words based on their meaning and maintaining the word order helps us augment the macro similarity obtained from the LSA module, with the micro similarity obtained in this module. The output of this module is a score indicating if the two questions are similar. This method is computationally expensive as it is sensitive to the word order as well as length of the questions. We used a caching mechanism to improve the computational speed of the algorithm. The scores of the word pairs are calculated only once.

## 4.4 Answer Generation

After getting the final ranked list of the questions from the previous step, we extract the answers from the Yahoo Answers corpus associated with each candidate question. We rank the answers based on the keyword overlap and alignment with the focus of the question. Since the answers cannot be longer than 1000 characters long, we select the sentences that are most representative of the focus and weighted keywords extracted from the question title and body. We pick the best answer for the highest ranked candidate question and greatest alignment with the question keyword and topic.

## 4.5 Knowledge Graph-based Question Answering

There are many questions (50%) that could be answered by our pipeline. The 4.4 million question answer corpus from Yahoo wouldn't account for all possible questions that can be asked. To answer the questions that our system found difficult to address, we used the Google knowledge

graph (Google, 2016). We used the keywords extracted from the question title and body as queries for the knowledge graph. We constructed the final answer based on the top three results retrieved from the knowledge graph application program interface (API).

If the knowledge graph could not give any results we responded by using a random response from a bag of 15 responses we prepared in advance. An example of such response “*This is a profound question. Sorry I cannot answer this difficult question at this time*”. This approach hurt us badly as the answers given were not accurate and sometimes completely irrelevant. From a deeper analysis, we see that the questions being asked are subjective and answers expected are opinionated answers. Having a system that works for factoid based questions adapted to opinionated and subjective questions was not much helpful.

## 5 Results and Discussion

Results from our system were evaluated based on the scoring system shown below:

- avgScore(0-3): The average score over all questions. This is the main score used to rank the participating system runs.
- succ@i+: the number of questions with score  $i$  or above ( $i \in \{2..4\}$ ) divided by the total number of questions. For example, succ@2+ measures the percent of questions with at least fair grade answered by the run.
- prec@i+: the number of questions with score  $i$  or above ( $i \in \{2..4\}$ ) divided by number of questions answered by the system. This measures the precision of the run, designed not to penalize non-answered questions.

Our system attempted to answer 899 out of 1088 questions, which is much higher than the average number of questions answered. Our servers timed out on 117 questions for which 60 secs was not enough to construct an appropriate answer. Results from our system were comparable to the mean metrics of all systems in the competition as shown in Tables 3-5.

We implemented two methods to answer the live questions. 1) Generate an answer from responses to similar questions asked in Yahoo question answer corpus; 2) The questions for which there was no answer in our LSA space and were answered using Google knowledge graph API to retrieve appropriate web links. Tables 6-7 show the breakup of the results after splitting the questions answered into the two methods explained above. The questions for which we answered confidently have prec@2+, prec@3+, and prec@4+ scores higher than the average scores as shown in Table 7.

Our results show that leveraging knowledge graph to answer subjective questions adversely affected the performance of the system. Analysis of the questions answered based on the semantic similarity with questions in Yahoo corpus performed much better than the questions answered by the knowledge graph. This can be attributed to the systematic approach we adopted in identifying the questions which are similar not only semantically but also in focus and sentiment with respect to the main focus of the question. We computed the similarity by respecting the word-order, which was computationally very expensive for longer questions (Li et al., 2006). We overcame this limitation by implementing a caching scheme where we greedily cached all the word-pairs for which we calculated the similarity score, and retrieved them efficiently when needed. This helped us to answer the questions in the prescribed time limit of 60 secs.

For the pilot task of identifying the focus of the question we submitted the output of the keyword extraction module discussed in the section 4.2. We used both question title and body to generate the keywords and since we used these words as the anchor words for identifying the similar questions, the same keywords are our focus words. Table 8 shows that our team ranked second among all the competing teams. Overall, our run ranked 4<sup>th</sup> in the task.

Table 3: Questions attempted by PRNA

Run	#Answers
prna	899
avg_score	771.0385

Table 4: Avg and succ scores

Run	avg	succ@2+	succ@3+	succ@4+
prna	0.4276	0.2749	0.1084	0.0443
avg	0.5766	0.3042	0.1898	0.0856

Table 5: Precision scores

Run	prec@2+	prec@3+	prec@4+
prna	0.3103	0.1224	0.0501
avg_score	0.3919	0.2429	0.108

Table 6: Break up of Answered and avg. score according to answer strategy

Run	#Answers	avgScore(0-3)
prna(default <sup>3</sup> )	459	0.216
prna(lsa_based)	439	0.763

Table 7: Break up of prec@{2-4} according to answer strategy

Run	prec@2+	prec@3+	prec@4+
prna(default <sup>3</sup> )	0.310	0.122	0.05
prna(lsa_based)	0.42	0.246	0.188

## 6 Conclusion

The performance of our open domain real-time question answering system is close to the Avg. runs of the competition. We answered 899 questions out of 1088 questions posted by moderators. By analyzing the questions that we answered using our pipeline we performed close to the avg. scores of all the runs. By analyzing the answers given based on the method used, our semantic similarity method performed much better than the average across all the participants. The knowledge graph approach affected the scores adversely. In the pilot task of identifying the focus of the question being asked our system is the 4<sup>th</sup> ranked system among all the participant systems.

In future, we would change our strategy to reduce the dependency on a knowledge graph and use more curated knowledge sources.

<sup>3</sup>default response is based on Google Knowledge Graph API and random excuse response

Table 8: Focus generation task results

Run	Meteor Score
baseline: title+body	0.260
baseline: title	0.212
NUDT NUDT681	0.177
NUDT NUDT681 1	0.167
NUDT NUDT681 3	0.136
<b>prna</b>	0.116
ECNU ECNU	0.089
DFKI dfkiga	0.065
NUDT NUDTMDP1	0.050
NUDT NUDTMDP2	0.048

## References

- Vivek Datla, King-Ip Lin, and M Louwerse. 2012. Capturing disease-symptom relations using higher-order co-occurrence algorithms. In *Bioinformatics and Biomedicine Workshops (BIBMW), 2012 IEEE International Conference on*, pages 816–821. IEEE.
- Christiane Fellbaum. 1998. A semantic network of english verbs. *WordNet: An electronic lexical database*, 3:153–178.
- Google. 2016. Google knowledge graph api.
- S. Hutchinson, V. Datla, and M.M. Louwerse. 2012. Social networks are encoded in language. In *Proceedings of the 34th Annual Conference of the Cognitive Science Society*. Austin, TX: Cognitive Science Society.
- Clayton J Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth International AAIL Conference on Weblogs and Social Media*.
- Cody Kwok, Oren Etzioni, and Daniel S Weld. 2001. Scaling question answering to the web. *ACM Transactions on Information Systems (TOIS)*, 19(3):242–262.
- T.K. Landauer. 2007. *Handbook of Latent Semantic Analysis*. University of Colorado Institute of Cognitive Science Series. Lawrence Erlbaum Associates.
- Yuhua Li, David McLean, Zuhair A Bandar, James D O’shea, and Keeley Crockett. 2006. Sentence similarity based on semantic nets and corpus statistics. *IEEE transactions on knowledge and data engineering*, 18(8):1138–1150.
- Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May. ELRA. <http://is.muni.cz/publication/884893/en>.
- S. Rose, D. Engel, N. Cramer, and W. Cowley. 2010. Automatic Keyword Extraction from Individual Documents. *Text Mining*, pages 1–20.