Using Re-ranking to Boost Deep Learning based Community Question Retrieval

Krishnendu Ghosh
Centre for Educational Technology
Indian Institute of Technology
Kharagpur
Kharagpur, West Bengal 721302
kghosh.cs@iitkgp.ac.in

Plaban Kumar Bhowmick
Center for Educational Technology
Indian Institute of Technology
Kharagpur
Kharagpur, West Bengal 721302
plaban@cet.iitkgp.ernet.in

Pawan Goyal
Dept. of Computer Science and Engg.
Indian Institute of Technology
Kharagpur
Kharagpur, West Bengal 721302
pawang@cse.iitkgp.ernet.in

ABSTRACT

The current study presents a two-stage question retrieval approach which, in the first phase, retrieves similar questions for a given query using a deep learning based approach and in the second phase, re-ranks initially retrieved questions on the basis of inter-question similarities. The suggested deep learning based approach is trained using several surface features of texts and the associated weights are pre-trained using a deep generative model for better initialization. The proposed retrieval model outperforms standard baseline question retrieval approaches. The proposed re-ranking approach performs inference over a similarity graph constructed with the initially retrieved questions and re-ranks the questions based on their similarity with other relevant questions. Suggested re-ranking approach significantly improves the precision for the retrieval task.

CCS CONCEPTS

• **Information systems → Information retrieval**; *Retrieval models and ranking*; *Probabilistic retrieval models*;

KEYWORDS

Community question answering, Question retrieval, Re-ranking

ACM Reference format:
Krishnendu Ghosh, Plaban Kumar Bhowmick, and Pawan Goyal. 2017. Using Re-ranking to Boost Deep Learning based Community Question Retrieval. In *Proceedings of WI '17, Leipzig, Germany, August 23-26, 2017,* 8 pages.
DOI: 10.1145/3106426.3106442

1 INTRODUCTION

Community question answering (CQA) services provide a platform where people from different background can fulfill their information need by browsing historical archive of question-answer pairs and asking/answering queries. With increasing popularity over time, CQA services stacked huge amount of question-answer pairs in their archive. This historical data can be used to retrieve similar questions and hereby avoiding the time-lag between posting a query and attaining a personal response [32]. Question retrieval returns direct and comprehensive answers to the user query unlike traditional web-search services [5].

One primary challenge in developing a question retrieval system is handling lexical gap between query and questions. Lexical gap is a word-mismatch problem which often misleads retrieval systems to regard relevant questions otherwise. For example in Table 1, lexical mismatch between the terms 'spice' and 'pepper' poses challenges in finding similar questions. This issue becomes even more crucial due to short length and informal nature of the web-queries. It also has been a major hindrance for directly using traditional information retrieval (IR) techniques (such as, vector space model, Okapi BM25 model, language model) in developing question retrieval systems.

Query	How to **get ghost peppers**?
Relevant Question	Best advice to grow spice?
Irrelevant Question	How to **get** over **ghosts**?

Table 1: Example of lexical gap in question retrieval

To handle lexical gap issue, various approaches have been reported among which translation, syntactic tree matching and topic model based approaches are significant. Assuming question-answer pairs as "parallel texts", translation-based approaches determine word-to-word (or, phrase-to-phrase) translation probabilities and hereby reduce the problem of lexical gaps [7, 26]. Syntactic analysis based approaches leverage syntactic and semantic similarities between query and questions to overcome lexical mismatch issue [14, 25]. Topic-based models determine similarity between question-pairs based on similarity between their respective topical distributions. On the other hand, recent studies tend to adopt distributed representations of texts to better comprehend the question-semantics in a low-dimensional vector space.

The current work proposes a two-stage retrieval approach: (i) question retrieval employing a deep learning based model trained using several surface features and (ii) re-ranking initially retrieved questions by exploiting inter-question similarity. The present study intuitively suggests various surface features, many of which are vastly discussed in the literature in relating a query-question pair. These features are fed as input in a neural network model which classifies a test query-question pair as relevant or irrelevant. Typical neural network models suffer from the issue of over fitting due to poor weight initialization. Therefore, present study suggested a deep generative model composed of numbers of restricted Boltzmann machines (RBM). In this model, each RBM infers associated

WI '17, Leipzig, Germany

DOI: 10.1145/3106426.3106442

non-linear features in the hidden layers for a given feature set in the visible layer and hereby, the generative model provide better initialization for the weights. The learned weights are used to initialize the hidden layers of the neural network. Finally, leveraging discriminative fine-tuning of proposed network, query-question pairs are classified as relevant and ranked based on scores. The initially retrieved questions is then re-ranked by performing inference over a network of questions constructed based on question-question similarity. The suggested re-ranking strategy assumes that, a question will be similar to an input query if it is similar to other questions that are relevant to the same query. Hence, based on inter-question similarity, initial retrieval score is re-weighed and a final ranked list is generated. A series of experiments are finally conducted to evaluate the performance of the proposed system. The contribution of the current work are as follows.

- The present study, leveraging on word alignments as well as the proposed combination of surface features, successfully reduces the issue of "lexical gap".
- The proposed deep neural network based approach unifies diverse surface features, including both discrete and continuous features effectively without making detailed assumptions about the data distribution. Hence, it reduces the burden to analyze complex syntactic structures or employing external knowledge-base (such as, WordNet).
- Pre-training the neural network using deep generative model such as, deep belief nets guarantees better performance as, weight parameters are properly initialized and chance of over fitting are minimized.
- The proposed re-ranking strategy tries to re-appropriate the significance of the query terms by scoring the questions based on inter-question similarity. Such method can reduce bias toward specific set of terms present in queries.

The rest of the paper is organized as follows: Section 2 discusses existing works related to question retrieval and re-ranking approaches. Section 3 and 4 presents respectively the proposed approaches for question retrieval and re-ranking of initially retrieved questions. Sections 5 and 6 discuss the experimental methods, performances and corresponding analysis. Section 7 concludes the paper with proposed directions toward future research.

2 RELATED WORK

2.1 Related Work on Question Retrieval

With increasing popularity of community question answering (CQA) sites, enormous research efforts have been made to improve question retrieval performance. These studies can be approximately classified in five major groups. The first group considers question answer pairs as "parallel-texts" and learns translation models to overcome lexical gap. Context information have been incorporated later via phrase-based models generating more accurate translations [30]. Such models are further improved by modeling domain-specific semantics of the word/phrases by discriminating named entities and noisy (unimportant) words present in the questions [21]. Recently translation models have been developed by employing efficient paraphrasing technique [29]. Translation-based approaches consistently prove their mettle yielding better performance than the traditional IR based approaches (such as, VSM, BM25 and LM) even if the problem of lexical gap persists.

The second group of studies are focused to achieve better retrieval performance by leveraging categorization information. The key idea is filtering irrelevant questions from archive using category information which denotes topical or categorical aspects of a question. [5, 6] used question-category information for smoothing in language-based models whereas, [31] refines retrieved questions using a disjoint category hierarchy.

The third group of studies target modeling topic distributions of the questions. Topic distributions, modeled by probabilistic topic models, encode topical attributes of a question in a lower dimension offering better comparison. However, topic model based approaches often suffers due to data sparsity problems. Such inadequacies are handled by fusing these models with other retrieval approaches (such as, translation-based language model in [4]) or related tasks (such as, modeling user preferences).

The fourth group employs syntactic information for retrieving similar questions. [10] employed a tree-cut method to extract syntactic structure, topic and focus of the questions. The extracted information is incorporated in a language-model based retrieval approach. [24] proposed a syntactic tree matching model where syntactic trees extracted from questions are compared for (i) syntactic structure at tree level and (ii) semantic relatedness at node-level. [23] proposed a pattern-recognition based approach to compare syntactic similarity. However, determining accurate syntactic structure is difficult for casual web-queries.

The previous four groups of studies advocated the use of different surface representations of texts to retrieve similar questions. Despite finding new and significant aspects to better comprehend the underlying semantics of the questions and user's information need, previous approaches failed to combine these features in a common framework. Hence, recent studies lean toward employing distributed representations of texts. In these approaches questions are generally modeled in a lower dimensional semantic space [22] or concept space [32] and their similarity is determined. Questions and answers are often encoded separately using de-noising auto-encoders and combined later using a convolutional neural tensor network [19]. Recently, learning-to-rank (LTR) based framework are employed to retrieve the ranked list based on the distance-based probability scores. Using standard LETOR and semantic features, ranking frameworks (such as, SVMrank) are also trained to approximate distances between query-question pairs [11, 33]. The current study suggests a deep neural network model trained on a variety of surface features extracted from query-question pairs to retrieve similar questions.

2.2 Related Work on Re-ranking

Although re-ranking has been one of the top research topics in field of information retrieval, it has not been employed in question retrieval systems so far. In information retrieval systems, re-ranking is primarily achieved in two ways: (i) direct re-ranking on initially retrieved documents and (ii) indirect re-ranking via query expansion. Query expansion based approaches assume top ranked documents to be relevant and hence, augment the original query using the terms present in all of these documents. Using expanded queries,

retrieved lists are less affected by lexical gap. On the other hand, direct re-ranking re-orders the initially retrieved documents by adjusting their weights.

Direct re-ranking approaches engaged by retrieval systems, can be classified into three categories based on the resources utilized. The first category exploits inter-document similarities. In such approaches, each of the initially retrieved documents are scored based on their initial score and inter-document distances [1]. The inter-document similarities are typically determined using term-overlap [13, 16, 27], relative diversity and information richness [28], discussed topics [9]. The second category maneuvers various external resources to reweigh the initial retrieval scores such as, hand crafted thesaurus [20], rule-base in form grammar [2] or controlled vocabularies [12]. The third category re-weighs initial scores leveraging document metadata such as, title [17] or stemmed and un-stemmed words respectively from initially retrieved documents and queries [8]. There are some studies where cluster-based and structural re-ranking based approaches are integrated to achieve better retrieval performance in terms of both precision and recall [18]. The current study introduces re-ranking approach for question retrieval where the initially retrieved list of questions are re-weighed based on inter-question similarity determined using term-overlap.

3 PROPOSED QUESTION RETRIEVAL

Web-queries are short in length, hardly follows grammar and standard word-usages. It is difficult to extract grammatical surface features and learn a model using these features to determining similarity between question pairs. Considering the significance of surface features and inability of traditional retrieval systems in aggregating these features, the present study proposes a deep learning based approach leveraging several surface features.

3.1 Pre-training

The current study proposes a deep learning based approach where a neural network, consisting of multiple hidden layers of units between the input and output layers, is employed to classify query-question pairs as relevant or irrelevant. Various surface features representing similarity between a query-question pair as fed as input and final output classifies a pair. To overcome the common issue of over fitting, the current work suggests using a generative model such as, deep belief net (DBN) to generatively pre-train the neural network to learn the initial weights properly. A DBN is a stochastic generative model composed of multiple hidden layers and are trained in an unsupervised, layer-by-layer manner where the layers are typically made of restricted Boltzmann machines (RBM). Once an RBM is trained, another RBM is stacked atop it, taking its input from the final already-trained layer. The new visible layer is initialized to a training vector, and values for the units in the already-trained layers are assigned using the current weights and biases. The new RBM is then trained with the procedure above. This whole process is repeated until some desired stopping criterion is met. Finally, the pre-training step is pursued using the DBN model which initializes the weights using greedy layer-wise Contrastive Divergence (CD) technique.

3.1.1 Restricted Boltzmann Machine. An RBM, as shown in Figure 1, is composed of primarily a visible layer v and a hidden layer h. The visible and hidden units of an RBM are connected using a m x n matrix of weights W and associated offsets a_i and b_j are denoted as biases for the visible and the hidden units respectively.

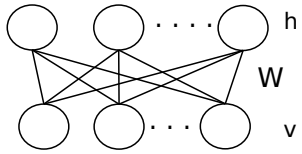

Figure 1: An RBM network, composed of a visible layer v and a hidden layer h, with a matrix of weights $W = \{w_{ij}\}$ where, w_{ij} connects hidden unit h_j and visible unit v_i. In the current work, series of RBMs are used to develop a deep generative model to initialize the weights for the proposed neural network model DNN

For an RBM, the energy of a joint configuration is given by:

$$E(\mathbf{v}, \mathbf{h}; \theta) = -a^T\mathbf{v} - b^T\mathbf{h} - \mathbf{v}^T W\mathbf{h} \tag{1}$$

where $\theta = (W, a, b)$. The probability the model assigns to the visible layer v is:

$$p(\mathbf{v}; \theta) = \frac{\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}}{\sum_{\mathbf{u}}\sum_{\mathbf{h}} e^{-E(\mathbf{u},\mathbf{h})}} \tag{2}$$

Since there are no hidden-hidden or visible-visible connections, the conditional distributions $p(v|h)$ and $p(h|v)$ are given by:

$$p(v_i = 1|\mathbf{h}) = \sigma(a_i + \sum_j w_{ji}h_j) \tag{3}$$

$$p(h_j = 1|\mathbf{v}; \theta) = \sigma(b_j + \sum_i w_{ij}v_i) \tag{4}$$

where σ is the logistic sigmoid function.

3.1.2 Training of an RBM. Following the gradient of the joint likelihood function for data and labels, the update rule for the weights is given by:

$$\Delta w_{ij} = \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{recon} \tag{5}$$

where $\langle v_i h_j \rangle_{data}$ is the frequency of visible unit v_i and hidden unit h_j occurring together in the training set and $\langle . \rangle_{recon}$ represents the same expectation with respect to the sample distribution obtained by running the Gibbs sampler initialized at the data for one full step.

3.2 Training

The present work uses a neural network with one input layer, two hidden layers and an output layer as shown in Figure 2. Weights, initialized in the pre-training phase, are used in the proposed network which is discriminatively fine-tuned by standard back propagation technique. Length and number of hidden layers are set to achieve optimal performance.

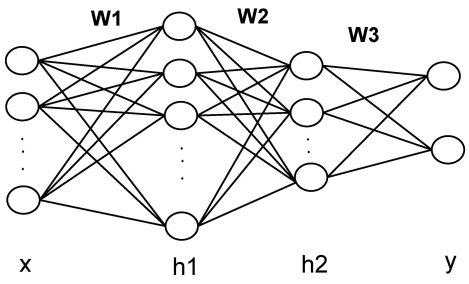

Figure 2: Proposed Neural Network model with input layer (x), two hidden layers (h1 and h2) and an output layer (y). W1, W2 and W3 are the associated weight matrices connecting different layers. In the present context, optimal set of features extracted from query-question pair is fed the input layer and the output layer represents whether the input pair is relevant to each other or not.

3.3 Feature Description

The current work comprises numerous surface features as discussed below:

Lexical Features: Lexical features pertain to the term itself and hence describes the components of a text aptly. The present study advocates for exploring following lexical features:

- *BM25 score:* It is traditional Okapi BM25 similarity between a query-question pair as obtained by Apache Lucene [1] with default parameter settings.
- *Word n-gram Overlap:* The current study extracts word n-gram counts for n = 1, 2 and 3 between query q and question Q as follows:

$$\text{Word n-gram Overlap} = \frac{\text{Count of n-grams common in } q \text{ and } Q}{\text{Total count of n-grams present in } Q}$$

- *Cosine Similarity:* In the present study, cosine similarity measures lexical similarity based on overlap between word n-grams(n=1,2,3) in the query and the question.

Syntactic Features: Syntax specific features (such as, POS tags, phrasal or syntactic structures) are useful to disambiguate between different lexical and semantic aspects. The current work proposes following syntactic features:

- *Noun Overlap:* It is the count of common nouns between a query-question pair. The count is normalized by the number of nouns present in the question. Different POS tags (nouns or verb) are determined in the current study using Stanford POS tagger[2].
- *Verb Overlap:* Similarly, the count of common verbs normalized by the number of verbs in the questions is considered as verb overlap.
- *Dependence-pair overlap:* It is the normalized count of common dependence-pairs between a query-question pair. Dependencies are determined using Stanford parser [3] in the question. Dependence-pair is the pair of words which are connected via a dependency relationship. For example, the sentence 'Barack Obama was born in Hawaii has dependencies: compound(Obama, Barack), nsubjpass(born, Obama), auxpass(born, was), root(ROOT, born), case(Hawaii, in) and nmod(born, Hawaii). Hence, the associated dependence-pairs are: (Obama, Barack), (born, Obama), (born, was), (ROOT, born), (Hawaii, in) and (born, Hawaii).
- *Named-entity overlap:* It is the normalized count of common named-entities between a query-question pair. Named-entities have been recognized using Stanford NER (Named Entity Recognizer) [4].

Semantic Features: Semantic features capture meaning of a term in the context they have been used. Such features can handle the lexical gap issue and hence, considered as significant aspects while determining similarity between query-question pairs.

- *Word alignments:* It is the translation probability score between two texts. To determine word-to-word translation probabilities, a word-alignment model using Giza++[5] is trained on similar question-pairs from held-out data. Finally, employing statistical machine translation using IBM model-1, the translation probability between the question pairs is calculated.
- *Common frames:* Using Frame net, semantic frames associated with the lexemes present in both questions are determined. Finally, the normalized count of the common frames have been considered as one of the features.

3.4 Optimal Feature Selection

Present study conducts a statistical significance test using chi-square and mutual information (MI) scores between the features for each query-question pairs and associated class labels. Based on significance as shown in Table 2, top 10 features from the entire set of 13 features have been selected.

Features	Chi-square	MI
BM25 score	6231.33	0.4
Word 1-gram Overlap	5817.15	0.2
Word 2-gram Overlap	5523.67	0.15
Word 3-gram Overlap	4972.72	0.08
Cosine Similarity (Word 1-gram)	5224.44	0.34
Cosine Similarity (Word 2-gram)	5007.32	0.31
Cosine Similarity (Word 3-gram)	3923.8	0.12
Noun Overlap	3825.7	0.2
Verb Overlap	3593.18	0.13
Dependence-pair overlap	1276.2	0.01
Named-entity overlap	1312.59	0.01
Word alignments	2648.93	0.14
Common frames	812.33	0.004

Table 2: Performance of statistical significance test of features using chi-square and mutual information (MI) scores. Features mentioned in bold are the top 10 features selected in optimal set

[1] http://lucene.apache.org/
[2] http://nlp.stanford.edu/software/tagger.html
[3] http://nlp.stanford.edu/software/lex-parser.html
[4] http://nlp.stanford.edu/software/CRF-NER.shtml
[5] http://www.statmt.org/moses/giza/GIZA++.html

3.5 Data Representation and Ranking

A question is represented with the following meta data: unique id, title and body where only id and title are the mandatory fields. Standard preprocessing steps (such as, stop word removal, lemmatization and stemming) have been carried out wherever required before feature generation. During testing, the proposed neural network model assigns a probability score for each test query-question pairs and based on these scores, an initial retrieval list of questions is generated.

4 PROPOSED RE-RANKING

Being inspired by [13], present study suggests a re-ranking technique to improve question retrieval performance. Basic hypothesis behind the proposed structural re-ranking approach is: *A question (Q_i) is ranked higher with respect to a query (q) if it is similar to most of the other questions retrieved for the same query.* The re-ranking strategy adopts a scoring function that is computed by considering the amount of support it receives from other questions.

For a target query q, an initial set of questions $\mathcal{Q}_{init}$ is retrieved from question repository C. We define a scoring function $IS_Y(X)$ to be a score that is assigned to question Y for an input query X. Each of the retrieved questions $Q_i \in \mathcal{Q}_{init}$ obtains an score $IS_{Q_i}(q)$ during initial retrieval. The re-ranking of initial retrieved questions $\mathcal{Q}_{init}$ is achieved in three steps: (i) support graph generation and (ii) determining cumulative support and (iii) final scoring. The detailed steps for re-ranking are illustrated below in Figure 3.

4.1 Support Graph Generation

A *support graph* for a set of retrieved questions $\mathcal{Q}_{init}$ encodes support relationship between different pairs of questions in $\mathcal{Q}_{init}$. A question Q_j is assumed to receive support from another question $Q_i \in \mathcal{Q}_{init}$ if Q_i is retrieved against a query in form of Q_j. In this case, we establish an edge $Q_i \rightarrow Q_j$ in the support graph.

Algorithm 1: Support Graph Construction with $\mathcal{Q}_{init}$

Input : Initial Retrieve Set ($\mathcal{Q}_{init}$)
Output: Support Graph $G = (\mathcal{V}, \mathcal{E})$

$\mathcal{V} \leftarrow \emptyset$;
$\mathcal{E} \leftarrow \emptyset$;
for $i \leftarrow 1$ **to** $|\mathcal{Q}_{init}|$ **do**
  $v_i \leftarrow$ `CreateNode`(Q_i);
  $\mathcal{V} \leftarrow \mathcal{V} \cup \{v_i\}$;
end
for $j \leftarrow 1$ **to** $|\mathcal{Q}_{init}|$ **do**
  $Top_\alpha(Q_j) \leftarrow$ `RetrieveTop`$(Q_j, \mathcal{Q}_{init} - \{Q_j\}, \alpha)$;
  for $i \leftarrow 1$ **to** $|Top_\alpha(Q_j)|$ **do**
    $e \leftarrow$ `CreateEdge`$(Q_i \rightarrow Q_j)$;
    $wt(Q_i \rightarrow Q_j) \leftarrow$ Compute weight with Eqn 6;
  end
end

Definition 4.1 (Support Graph). A support graph is a directed and weighted graph $G = (\mathcal{V}, \mathcal{E})$. Each question from the initial retrieved set $\mathcal{Q}_{init}$ is represented by a node $v \in \mathcal{V}$. A directed edge $Q_i \rightarrow Q_j$ in $\mathcal{E}$ represents the amount of support Q_j receives from Q_i. The amount of support is quantified by a weight function $wt(Q_i \rightarrow Q_j)$ as defined in 4.2.

Definition 4.2 (Edge Weight Function $wt(.)$). Let us assume that $Top_\alpha(Q_j)$ is the set of top α questions $\{Q_i | Q_i \in \mathcal{Q}_{init} - \{Q_j\}$ for query Q_j with respect to a scoring function $IS_{Q_i}(Q_j)$. The amount of support that Q_j receives from Q_i is computed as follows:

$$wt(v[Q_i] \rightarrow v[Q_j]) = \begin{cases} IS_{Q_i}(Q_j) \text{ if } Q_i \in Top_\alpha(Q_j), \\ 0 \text{ otherwise} \end{cases} \tag{6}$$

The construction of supports graph starts with a set of node created out of the set $\mathcal{Q}_{init}$. For each question Q_i the construction algorithm (see 1) identifies top α similar questions in $\mathcal{Q}_{init} - \{Q_j\}$ and establishes edges for each respective pair of questions. This process is applied iteratively until all the question in $\mathcal{Q}_{init}$ are processed. A visual description of the construction process in presented in Figure 3.

In boundary conditions, supports from all the questions to a question in concern may be zero or insignificant. Consequently, contribution from structural re-ranking stage will become insignificant. This will heavily penalize a question that has got moderate score in the first phase. In order to handle this issue, we have employed Pagerank's smoothing technique in [3]. According to this technique, a graph G is smoothed to $G^{[\lambda]}$ to avoid zero edge-weights using smoothing parameter $\lambda \in [0, 1)$ where, $G^{[\lambda]} = (\mathcal{V}, \mathcal{E}^{[\lambda]})$ has smoothed edge weights ($wt^{[\lambda]}$) as defined below:

$$wt^{[\lambda]}(v[Q_i] \rightarrow v[Q_j]) = (1-\lambda)\frac{1}{|\mathcal{Q}_{init}|} + \lambda \frac{wt(v[Q_i] \rightarrow v[Q_j])}{\sum_{Q'_j \in \mathcal{Q}_{init}} wt(v[Q_i] \rightarrow v[Q'_j])} \tag{7}$$

for every $Q_i, Q_j \in \mathcal{Q}_{init}$.

4.2 Cumulative Support Determination

A straightforward way to determine the total amount of support other questions provide to a particular question Q_j with respect to a given graph $G^{[\lambda]} = (\mathcal{V}, \mathcal{E}^{[\lambda]})$ is to set it to Q_j's weighted in-degree. We call it a Non-Recursive (CS_{NR}) support model which is given by:

$$CS_{NR}(Q_j, G^{[\lambda]}) = \sum_{Q_i \in \mathcal{Q}_{init}} wt^{[\lambda]}(v[Q_i] \rightarrow v[Q_j]) \tag{8}$$

In another consideration, the amount of support question Q_i propagates to question Q_j will be dependent on how much support Q_i is getting from others. Hence, another recursive version of cumulative support (CS_R) is determined as:

$$CS_R(Q_j, G^{[\lambda]}) = \sum_{Q_i \in \mathcal{Q}_{init}} wt^{[\lambda]}(v[Q_i] \rightarrow v[Q_j]). CS_R(Q_i, G^{[\lambda]}) \tag{9}$$

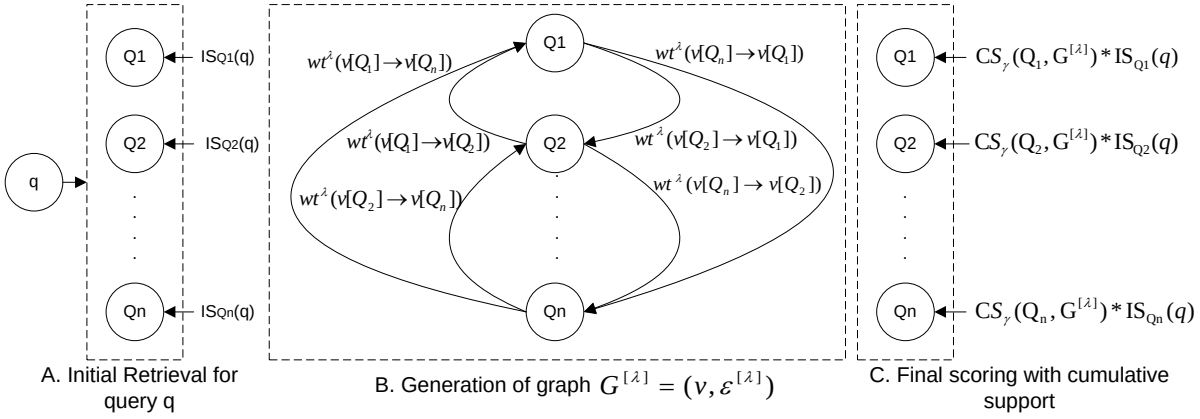

Figure 3: Re-ranking of initially retrieved questions: (i) generating support graph to represent the similarity between questions and (ii) determining cumulative support each question obtains from other questions in the graph and (iii) final scoring for each questions for further ranking

4.3 Final Scoring

For a query-question pair (q, Q_j), the final score of Q_j is a function of initial retrieval score ($IS_{Q_j}(q)$) obtained from stage 1 and cumulative support score (CS_R or CS_{NR}) computed in stage 2. The final score function is given by:

$$SR_\gamma(Q_j, q) = CS_\gamma(Q_j, G^{[\lambda]}) * IS_{Q_i}(q) \tag{10}$$

where $\gamma \in [NR, R]$. Hence, the present study generates two re-ranking models namely: recursive and non-recursive re-ranking models. Retrieval performances are compared for different values of the parameters α and λ within ranges [5,10,15,20,..,$|Q_{init}|$-1] and [0,0.05,0.1,0.15,..,0.95] respectively. Finally, the parameters α and λ have been set to values 15 and 0.05 respectively to achieve optimal retrieval performance with respect to performance metric Precision@5.

5 EXPERIMENTAL SETUP

5.1 Data preparation

To evaluate the proposed retrieval approach, query and question archives from AskUbuntu forums has been used as dataset [15]. This dataset comes with a gold-standard where for each of the queries similar questions from the repositories are manually annotated. Table 3 presents an overview of the datasets.

	Train Set	Dev Set	Test Set
Query	4341	200	186
Question	167765	167765	167765

Table 3: Overview of AskUbuntu dataset

Apart from these questions, the current study uses two held-out data: (i) AskUbuntu held-out data as used in [15] to empirically optimize the neural network parameters and (ii) Stack Exchange held-out data of 100K similar question-pairs to train word alignment models.

5.2 Experiments

In the current work, four retrieval approaches are employed as baselines: (a) Okapi BM25 model (BM25), (b) language model using Jelinek-Mercer smoothing (LM), (c) translation-based language model (TRLM) and, (d) syntactic tree matching (STM) [24]. These baseline models are compared against the deep learning approach proposed in the current study, named as DNN, using six metrics: Mean Average Precision (MAP), R-precision, Mean Reciprocal Rank (MRR), Precision@1, Precision@5 and Precision@10. Implemented retrieval models are also compared against a RCNN (Recursive Convolutional Neural Network) based model in [15] deployed on same AskUbuntu dataset. To inquire the classifier's performance in retrieval, the proposed deep learning based approach has been substituted with a SVM classifier with default parameters as suggested by LIBSVM [6]. Experimental results are compared against the provided gold-standard data and the performance is shown in Table 4.

Model	MAP	RP	MRR	P@1	P@5	P@10
BM25	45.49	54.2	57.8	48.4	36.7	19.9
LM	45.2	52.8	58.4	46.2	36.7	17.6
TRLM	42.6	50.4	56.6	43.1	33.6	16.6
STM	38.9	44.8	46.6	39.8	28.3	11.6
SVM	59.2	66.7	68.6	60.1	48.3	30.6
RCNN	62.3	_	**75.6**	62.0	47.1	_
DNN	**64.4**	**69.8**	72.8	**67.9**	**54.1**	**31.7**

Table 4: Performance of question retrieval models on AskUbuntu test data. Only the available performance measures have been mentioned for model RCNN as collected from [15]

DNN is further re-ranked separately with both non-recursive and recursive cumulative support as discussed in equation 6 and 7 respectively to form models: $\mathcal{R}^{NR}$ and $\mathcal{R}^{R}$. While determining top generators $Top_\alpha(.)$ for questions in the above mentioned models,

[6] https://www.csie.ntu.edu.tw/ cjlin/libsvm/

present study employs two different retrieval model: (i) language model using Jelinek-Mercer smoothing (LM) and, (ii) proposed model DNN. For different retrieval model used to determine top generators $Top_\alpha(.)$, two different models are formed as: $\mathcal{R}(LM)$ and $\mathcal{R}(DNN)$. Therefore, employing (i) different versions of cumulative support and (ii) different retrieval model used to determine top generators, current study developed four retrieval models: $\mathcal{R}^{NR}(LM)$, $\mathcal{R}^{R}(LM)$, $\mathcal{R}^{NR}(DNN)$ and $\mathcal{R}^{R}(DNN)$.

The objective of re-ranking is to re-order a list of initially-retrieved questions to improve precision at the top ranks of the final results. Hence, these models are evaluated using the same performance metrics. The parameters α and λ have been optimized with respect to the measure P@5, not with respect to all the evaluation measures. The optimal value of α has been noted in the range 5-15 with respect to P@5, suggesting that propagating cumulative supports from relatively small number of questions can effectively re-rank the questions. In contrast, λ exhibited substantial variance in optimal value for P@5. In the current study, $|Q_{init}|$ is set as 50 as, for higher values similar performances were obtained. Experimental results are compared against the provided gold-standard data and the performance is shown in Table 5.

	MAP	RP	MRR	P@1	P@5	P@10
DNN	64.4	**69.8**	72.8	67.9	54.1	**31.7**
DNN + $\mathcal{R}^{NR}(LM)$	60.2	62.4	63.7	58.1	47.3	27.5
DNN + $\mathcal{R}^{R}(LM)$	64.0	64.7	66.6	63.2	51.7	30.8
DNN + $\mathcal{R}^{NR}(DNN)$	60.8	62.2	65.2	59.6	49.1	28.3
DNN + $\mathcal{R}^{R}(DNN)$	**65.8**	68.5	**73.3**	**68.6**	**56.8**	30.8

Table 5: Performance of re-ranking on AskUbuntu test data where, R (or, NR) denotes the associated cumulative support determined in recursive (or, non-recursive) fashion and LM (or, DNN) denotes the retrieval model employed to determine top generators for a question

6 PERFORMANCE ANALYSIS

The observations that can be made from the experimental results are: (i) traditional baseline models like BM25 and LM performed similarly, (ii) translation-based model (TRLM) and syntactic-matching model (STM) could not perform at par with traditional models, (iii) proposed approach DNN outperforms traditional baseline models with significant margin, (iv) DNN consistently performed better than SVM & RCNN and, (v) using re-ranking, further improvements are reported in retrieval performance.

Translation-based model failed to improve retrieval performance due to domain-specific nature of AskUbuntu dataset. Similarly, due to informal nature of texts, syntactic models also performed poorly. For the following example, top retrieved questions from TRLM and STM are irrelevant compared to the same retrieved by proposed DNN model.

Query	Wifi issue after ubuntu upgrade
TRLM	**Wifi** and ethernet stopped working in **ubuntu** 14.04
STM	**Upgrade** to 14.04 from 13.10 Not Working
DNN	**Wifi** not working after **upgrading Ubuntu** 16.04

The proposed model, DNN, effectively unifies the optimally selected surface features. As a result, DNN performs consistently better than RCNN and SVM. In the following table, RCNN and SVM [15] puts more weight on lexical matches, but DNN comprehend the information need better. Due to feature 'word alignment', lexical gap is handled between terms *flash* and *USB* and the central action of the query: mounting has been supported with feature: 'verb overlap'.

Query	How to mount flash drive
RCNN	**How** to install ubuntu on **flash drive**
SVM	**How** to **mount** USB **drive** from the terminal
DNN	USB devices not **mounted**

Detailed analysis concludes that re-ranking approach failed to improve retrieval performance where, (i) questions of different syntactic structures have been compared, (ii) presence of terms which are relevant but used rarely, (iii) domain-specific semantics of the questions could not be captured well and (iv) extracted features were erroneous. For example, for a query '*normal mod not found*' re-ranking improved ranking of questions due to absence of inconsistencies with respect to syntactic structures for the second question. On the other hand, presence of rare terms like *BTRFS* or *GRUB2* deteriorates the rank for the last question.

Question	Rank before re-ranking	Rank after re-ranking
error: file '/grub/i386-pc /normal.mod' not found	1	1
How to deal with "'normal.mod' not found" in grub rescue	4	2
error: BTRFS and GRUB2 issue	5	11

Using re-ranking retrieval performance is enhanced for all evaluation measures although, parameters related to re-ranking approaches have been optimized with respect to measure Precision@5. This indicates the effectiveness of the re-ranking approach in reordering initially retrieved list appropriately so that the most similar questions are placed in top ranks enhancing all the measures together. The propagated support, in terms of similarity between questions, to one question always enhanced its final score. As a result, recursive cumulative support based re-ranking achieved better performance than using non-recursive cumulative support based re-ranking.

Although the current study showed significant improvement in retrieval performance, there are still plenty scope to improve the proposed approach:

- Using advanced surface features (such as, overlap in dependency relation triples and overlap in semantic roles) and distributed features (such as, distributional similarity between texts, nouns or verbs etc).
- Using denoising auto-encoders (DAE) in pre-training phase, so that the features can better discriminate between questions.
- Optimizing parameter with respect to metrics MRR or P@1, better retrieval performance can be achieved in top ranks.

- Exploring hybrid re-ranking approaches on improved retrieval model, inter-question similarities can be captured efficiently and hence, better precision can be achieved.

7 CONCLUSION

The current paper presents a two-stage approach combining question retrieval system using a deep learning based approach DNN and re-ranking. In the first phase, DNN is trained over significant lexical, syntactic and semantic features to map the similarity between a query-question pair. DNN approach outperformed traditional information retrieval based systems with significant margin. In the second phase, retrieval performance is further improved with the help of the inter-question similarity. The performance gain achieved by the re-ranking module speaks for its mettle wherever applied. A comparison has been made with for the AskUbuntu dataset where, the proposed approach $\mathcal{R}^{NR}(DNN)$ surpass state-of-the-art model RCNN and classifiers like SVM with significant margin.

REFERENCES

[1] Jaroslaw Balinski and Czeslaw Danilowicz. 2005. Re-ranking Method Based on Inter-document Distances. *Inf. Process. Manage.* 41, 4 (July 2005), 759–775. DOI: http://dx.doi.org/10.1016/j.ipm.2004.01.006

[2] John Bear, David Israel, Jeff Petit, and David Martin. 1998. *Using information extraction to improve document retrieval.* Technical Report. DTIC Document.

[3] Sergey Brin and Lawrence Page. 1998. The Anatomy of a Large-scale Hypertextual Web Search Engine. *Comput. Netw. ISDN Syst.* 30, 1-7 (April 1998), 107–117. DOI: http://dx.doi.org/10.1016/S0169-7552(98)00110-X

[4] Li Cai, Guangyou Zhou, Kang Liu, and Jun Zhao. 2011. Learning the Latent Topics for Question Retrieval in Community QA. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, Vol. 1. 273Ű281.

[5] Xin Cao, Gao Cong, Bin Cui, and Christian S. Jensen. 2010. A Generalized Framework of Exploring Category Information for Question Retrieval in Community Question Answer Archives. In *Proceedings of the 19th International Conference on World Wide Web (WWW '10)*. ACM, New York, NY, USA, 201–210.

[6] Xin Cao, Gao Cong, Bin Cui, Christian Søndergaard Jensen, and Ce Zhang. 2009. The Use of Categorization Information in Language Models for Question Retrieval. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM '09)*. ACM, New York, NY, USA, 265–274.

[7] Doo Soo Chang and Yong Suk Choi. 2014. Weighted Combination of Q&A Retrieval Models Based on Part-of-Speech of Question Word. In *Information Science and Applications (ICISA), 2014 International Conference on*. IEEE, 1–4.

[8] Carolyn J Crouch, Donald B Crouch, Qingyan Chen, and Steven J Holtz. 2002. Improving the retrieval effectiveness of very short queries. *Information processing & management* 38, 1 (2002), 1–36.

[9] Fernando Diaz. 2005. Regularizing ad hoc retrieval scores. In *Proceedings of the 14th ACM international conference on Information and knowledge management*. ACM, 672–679.

[10] Huizhong Duan, Yunbo Cao, Chin-Yew Lin, and Yong Yu. 2008. Searching Questions by Identifying Question Topic and Question Focus. In *Proceedings of ACL-08: HLT*. Association for Computational Linguistics, Columbus, Ohio, 156–164.

[11] Marc Franco-Salvador, Sudipta Kar, Thamar Solorio, and Paolo Rosso. 2016. UH-PRHLT at SemEval-2016 Task 3: Combining Lexical and Semantic-based Features for Community Question Answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*. 814–821. http://aclweb.org/anthology/S/S16/S16-1126.pdf

[12] Jaap Kamps. 2004. Improving retrieval effectiveness by reranking documents based on controlled vocabulary. In *European Conference on Information Retrieval*. Springer, 283–295.

[13] Oren Kurland and Lillian Lee. 2005. PageRank Without Hyperlinks: Structural Re-ranking Using Links Induced by Language Models. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '05)*. ACM, New York, NY, USA, 306–313. DOI: http://dx.doi.org/10.1145/1076034.1076087

[14] Akhil Langer, Rohit Banga, Ankush Mittal, and L. Venkata Subramaniam. 2010. Variant Search and Syntactic Tree Similarity Based Approach to Retrieve Matching Questions for SMS Queries. In *Proceedings of the Fourth Workshop on Analytics for Noisy Unstructured Text Data (AND '10)*. ACM, New York, NY, USA, 67–72.

[15] Tao Lei, Hrishikesh Joshi, Regina Barzilay, Tommi S. Jaakkola, Kateryna Tymoshenko, Alessandro Moschitti, and Lluís Màrquez. 2016. Semi-supervised Question Retrieval with Gated Convolutions. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*. 1279–1289. http://aclweb.org/anthology/N/N16/N16-1153.pdf

[16] Yang Lingpeng, Ji Donghong, Tang Li, and Niu Zhengyu. 2005. Chinese information retrieval based on terms and relevant terms. *ACM Transactions on Asian Language Information Processing (TALIP)* 4, 3 (2005), 357–374.

[17] Robert WP Luk and Kam-Fai Wong. 2004. Pseudo-Relevance Feedback and Title Re-Ranking for Chinese Information Retrieval.. In *NTCIR*.

[18] Seung-Hoon Na, In-Su Kang, and Jong-Hyeok Lee. 2008. Structural re-ranking with cluster-based retrieval. In *European Conference on Information Retrieval*. Springer, 658–662.

[19] Xipeng Qiu and Xuanjing Huang. 2015. Convolutional Neural Tensor Network Architecture for Community-based Question Answering. In *Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI'15)*. AAAI Press, 1305–1311. http://dl.acm.org/citation.cfm?id=2832415.2832431

[20] Youli Qu, Guowei Xu, and Jun Wang. 2001. Rerank Method Based on Individual Thesaurus. In *NTCIR*.

[21] Amit Singh. 2012. Entity Based Q&A Retrieval. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL '12)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 1266–1277.

[22] Baoxun Wang, Bingquan Liu, Xiaolong Wang, Chengjie Sun, and Deyuan Zhang. 2011. Deep learning approaches to semantic relevance modeling for chinese question-answer pairs. *ACM Transactions on Asian Language Information Processing (TALIP)* 10, 4 (2011), 21.

[23] Kai Wang and Tat-Seng Chua. 2010. Exploiting Salient Patterns for Question Detection and Question Retrieval in Community-based Question Answering. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING '10)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 1155–1163.

[24] Kai Wang, Zhaoyan Ming, and Tat-Seng Chua. 2009. A Syntactic Tree Matching Approach to Finding Similar Questions in Community-based Qa Services. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '09)*. ACM, New York, NY, USA, 187–194.

[25] Haocheng Wu, Wei Wu, Ming Zhou, Enhong Chen, Lei Duan, and Heung-Yeung Shum. 2014. Improving Search Relevance for Short Queries in Community Question Answering. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining (WSDM '14)*. ACM, New York, NY, USA, 43–52.

[26] Xiaobing Xue, Jiwoon Jeon, and W. Bruce Croft. 2008. Retrieval Models for Question and Answer Archives. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '08)*. ACM, New York, NY, USA, 475–482.

[27] Lingpeng Yang, Donghong Ji, and Munkew Leong. 2005. Chinese document re-ranking based on term distribution and maximal marginal relevance. In *Asia Information Retrieval Symposium*. Springer, 299–311.

[28] Benyu Zhang, Hua Li, Yi Liu, Lei Ji, Wensi Xi, Weiguo Fan, Zheng Chen, and Wei-Ying Ma. 2005. Improving web search results using affinity graph. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 504–511.

[29] Weinan Zhang, Zhaoyan Ming, Yu Zhang, Ting Liu, and Tat-Seng Chua. 2015. Exploring Key Concept Paraphrasing Based on Pivot Language Translation for Question Retrieval.. In *AAAI*, Blai Bonet and Sven Koenig (Eds.). AAAI Press, 410–416.

[30] Guangyou Zhou, Li Cai, Jun Zhao, and Kang Liu. 2011. Phrase-based Translation Model for Question Retrieval in Community Question Answer Archives. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1 (HLT '11)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 653–662. http://dl.acm.org/citation.cfm?id=2002472.2002555

[31] Guangyou Zhou, Yubo Chen, Daojian Zeng, and Jun Zhao. 2013. Towards faster and better retrieval models for question search. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management (CIKM '13)*. ACM, New York, NY, USA, 2139–2148.

[32] Guangyou Zhou, Yang Liu, Fang Liu, Daojian Zeng, and Jun Zhao. 2013. Improving Question Retrieval in Community Question Answering Using World Knowledge. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI '13)*. AAAI Press, 2239–2245.

[33] Guangyou Zhou, Yin Zhou, Tingting He, and Wensheng Wu. 2016. Learning Semantic Representation with Neural Networks for Community Question Answering Retrieval. *Know.-Based Syst.* 93, C (Feb. 2016), 75–83. DOI: http://dx.doi.org/10.1016/j.knosys.2015.11.002