PAPER Special Section on Data Engineering and Information Management

Modeling Joint Representation with Tri-Modal Deep Belief Networks for Query and Question Matching

Nan JIANG[†,††], ***Nonmember***, **Wenge RONG**[†,††a)], ***Member***, **Baolin PENG**[†††], **Yifan NIE**[††††], ***and*** **Zhang XIONG**[†,††], ***Nonmembers***

SUMMARY One of the main research tasks in community question answering (cQA) is finding the most relevant questions for a given new query, thereby providing useful knowledge for users. The straightforward approach is to capitalize on textual features, or a bag-of-words (BoW) representation, to conduct the matching process between queries and questions. However, these approaches have a lexical gap issue which means that, if lexicon matching fails, they cannot model the semantic meaning. In addition, latent semantic models, like latent semantic analysis (LSA), attempt to map queries to its corresponding semantically similar questions through a lower dimension representation. But alas, LSA is a shallow and linear model that cannot model highly non-linear correlations in cQA. Moreover, both BoW and semantic oriented solutions utilize a single dictionary to represent the query, question, and answer in the same feature space. However, the correlations between them, as we observe from data, imply that they lie in entirely different feature spaces. In light of these observations, this paper proposes a tri-modal deep belief network (tri-DBN) to extract a unified representation for the query, question, and answer, with the hypothesis that they locate in three different feature spaces. Besides, we compare the unified representation extracted by our model with other representations using the Yahoo! Answers queries on the dataset. Finally, Experimental results reveal that the proposed model captures semantic meaning both within and between queries, questions, and answers. In addition, the results also suggest that the joint representation extracted via the proposed method can improve the performance of cQA archives searching.

key words: *cQA, deep belief networks, joint representation, tri-modal deep belief network*

1. Introduction

Recently, community question answering (cQA) sites, such as Yahoo! Answers, Quora, and Baidu Knows have become popular platforms to maintain and distribute user-generated web content in the form of textual questions and answers. cQA sites have proven a success in knowledge sharing for their enabling of cQA users to get answers to personal, specific, and open-ended questions, and accordingly have accumulated extremely large information archives at ever-increasingly faster speed, which calls for an effective and efficient mechanism for leveraging information in cQA systems [1].

On the whole, cQA sites support general search engines or their own site search engines to retrieve information wherein short queries are widely used. But cQA users often ask questions via natural language, which is easier for users to express their knowledge request this way [2]. In these scenarios, one of the most common tasks is the matching of queries and questions to the underlying information needed by the user [3]. However, the process of matching queries and questions is error-prone and quite often irrelevant questions are returned [4]. Therefore it has become an important challenge to find similar and related questions from past archives, with regard to new queries, and obtain useful knowledge for the users from a linguistic as well as semantic point of view.

Traditionally, TF-IDF algorithm is widely used in search engines based on the bag-of-words (BoW) representation, and has proven its success due to its simplicity and robust implementation [5]. However, it cannot model semantic similarity when keyword-based matching fails, which is referred to as a lexical gap. In some cases, two sentences can hold the opposite meaning while they have the same BoW representation [6], whereas in other cases, different BoW representations can express similar meanings.

To handle this gap, LSA model [7], such as probabilistic latent semantic index (pLSI) [8] and Latent Dirichlet Allocation (LDA) [9] have been proposed to extract low-dimensional semantic features. Although integrating extra semantic information is promising, this kind of approach only build linear and shallow models that can only capture pairwise semantic relationships between words [10].

Although BoW and semantic oriented solutions have proven their capability and feasibility to some extent, both regard queries, questions, and answers in the same feature space. However, in reality queries are composed of short and incomplete sentences, the questions length is usually longer and will typically be a single sentence, while answers are the most informative part and generally comprise many sentences. As such, it can be assumed that there are three different modalities, lying in distinct feature spaces. Inspired by this hypothesis, we propose a tri-modal deep belief network (triDBN) to obtain a unified representation that captures highly non-linear correlation between different modal inputs, from three separate feature spaces for the queries, questions, and answers. The core procedure of the

Manuscript received June 22, 2015.
Manuscript revised November 8, 2015.
Manuscript publicized January 14, 2016.

†The authors are with State Key Laboratory of Software Development Environment, Beihang University, China.
††The authors are with School of Computer Science and Engineering, Beihang University, China.
†††The author is with Department of System Engineering and Engineering Management, The Chinese University of Hong Kong, China.
††††The author is with Department of Computer Science and Operations Research, Université de Montréal, Canada.
a) E-mail: w.rong@buaa.edu.cn

DOI: 10.1587/transinf.2015DAP0009

proposed method is as follows.

First, three separate deep belief networks (DBN) are employed to model the highly non-linear correlation between queries, questions, and answers in their own feature spaces. Then, another DBN is used to obtain a unified representation from them in the joint space. We perform an experimental study on a Yahoo! Answers dataset on "query and question semantic level classification". Our results depicted that using the unified representation as features to train the classifier significantly outperform baselines using BoW, LSA or unimodal DBN representation.

The main contributions of this paper are: 1) We thoroughly investigate the relationship between queries, questions, and answers. From real dataset, it is found that there exist quite a lot different patterns among queries, questions and answers. Inspired by that, in this paper, we assume that they are in three separate feature spaces instead of being represented in one feature space; 2) We propose a tri-modal deep belief network (triDBN) that is characterized by fusing queries, questions, and answers into a joint feature space. Our experimental study on Yahoo! Answers queries to question dataset show triDBN's promising potential.

The rest of the paper is organised as follows. The background about cQA and DBN is presented in Sect. 2. In Sect. 3, the proposed method is elaborated and in Sect. 4, we illustrate and discuss the experimental study on the Yahoo! Answers dataset. Finally, in Sect. 5, we conclude the paper and depict the future research directions.

2. Background

2.1 Community Question Answering

The value of user-generated question and answer has not been emphasized until recent years. Unlike traditional search engines, community question answering (cQA) sites can help users find and share knowledge with others. Furthermore, different from search engines, which employ keyword based searches, in cQA based forums users can use natural language to communicate with each other. Therefore, one of the most important challenges is to be able to match queries with questions, which is useful in supporting query suggestion, question recommendation [11], [12], and answer ranking in cQA sites [13].

In the process of matching queries and questions, Zhao et al. proposed an approach trying to automatically generate questions from query logs in cQA [3], *viz.* finding the most semantically similar questions to a query. Their model is built on templates introduced from search engine query logs. The core idea is to first extract pairs of queries and user-selected questions from query logs that are used for template extraction. Afterwards, when a new query is submitted, the model selects the most appropriate templates to generate a list of selected and sorted questions that are returned to the users.

Another approach is to to generate questions from queries in term of keywords by considering the query history and user feedback [14]. This method also introduces question templates while employing an adaptive language model in the process of question forming, which depends on templates and cannot grasp semantic information. Similarly, Figueroa and Neumann proposed an approach to extract query paraphrases to questions using learning to rank technology [15]. They first extract a corpus of query paraphrases from Yahoo! Search and Yahoo! Answers using the *query-question* click history. Second, they use query paraphrases to form positive and negative examples, which are then fed into the SVM-Ranking model [16]. This method can capture semantic meaning to some degree.

When we further investigate the widely employed methods in matching of queries and questions, we find that none of these methods distinguish between the feature space of queries, questions, and answers. However, in the daily use of cQA sites such as Yahoo! Answers or Baizhu Knows, queries are usually short and incomprehensible, while answers tend to be very informative and are usually provided in form of sentences or paragraphs and much longer than questions. In this paper we assume that queries, questions, and answers exist in different feature spaces. Accordingly, we employ a deep learning oriented method to capture the non-linear correlation between them.

2.2 Deep Belief Network

In recent years, deep learning has shown record performance across a variety of applications on language models and information retrieval [17]–[19]. Variant deep learning algorithms are designed to reveal hidden structures and features that are highly non-linear at different levels of abstraction from a huge amount of data, which is then used as features for classification or retrieval [10], [20].

Among deep learning algorithms, the Deep Belief Network (DBN) has been proven an efficient and effective generative model [21], [22]. At its base, it uses a restricted Boltzmann machine (RBM) as a building block. Many researchers have emphasized RBM and this has given rise to many RBM variants being successfully applied across different domains, such as Gaussian RBM [23] and replicated softmax Model [24]. In this section we give a brief introduction to these variants.

2.2.1 Restricted Boltzmann Machine

Restricted Boltzmann machines (RBMs) have been widely used as building blocks for deep learning algorithms due to their power in modelling distributions over binary-valued data [25], [26]. The traditional RBM is a two-layer undirected graphical model that has stochastic visible units $\mathbf{v}$ and stochastic hidden units $\mathbf{h}$, with visible units and hidden units being fully connected, as shown in Fig. 1, where circles in black denote observed states and white circles denote hidden states. Given $(\mathbf{v}, \mathbf{h})$ pair, where $\mathbf{v}$ is a stochastic visible state and $\mathbf{h}$ is a stochastic hidden state, its energy $E(\mathbf{v}, \mathbf{h}; \theta)$ is computed as follows [27]:

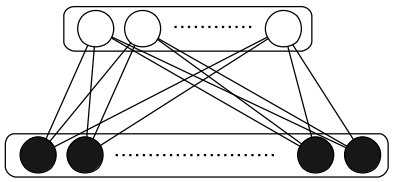


Fig. 1 Architecture of a Restricted Boltzmann Machine.

$$E(\mathbf{v},\mathbf{h};\theta) = -\sum_{i=1}^{m}\sum_{j=1}^{n} W_{ij}v_i h_j - \sum_{i=1}^{n} v_i b_i - \sum_{i=j}^{m} h_j a_j \quad (1)$$

where v_i is the binary state of visible unit i, h_j is the binary state of hidden unit j, a_i and b_j are bias terms for the states of visible unit v_i and hidden unit h_j, and W_{ij} is the connected weight between v_i and h_j.

Given this energy function, the model assigns a probability vector to every possible pair of visible and hidden states $(\mathbf{v},\mathbf{h})$:

$$P(\mathbf{v},\mathbf{h}) = \frac{1}{\mathcal{Z}}\exp(-E(\mathbf{v},\mathbf{h})) \quad (2)$$

with $\mathcal{Z}$ being a normalization term that guarantees that the sum of $P(\mathbf{v},\mathbf{h})$ is 1.

$$\mathcal{Z} = \sum_{\mathbf{v},\mathbf{h}} \exp(-E(\mathbf{v},\mathbf{h})) \quad (3)$$

The network assigns a probability to a visible vector v by summing all the possible hidden vectors $\mathbf{h}$, as shown below:

$$p(\mathbf{v}) = \frac{1}{\mathcal{Z}}\sum_{\mathbf{h}} \exp(-E(\mathbf{v},\mathbf{h})) \quad (4)$$

On training an RBM and acquiring a high probability for the observed training data, weights and biases between visible and hidden units need to be adjusted to lower the energy of the observed training vector and raise the energy of other training vectors, in particular those that have very low energy [28]. It has been proven that lowering the training vector energy is equal to maximizing the log probability of the training data [21]. As such, parameters of RBM can be learned using the following partial derivation:

$$\begin{aligned}
\frac{\partial \log p(\mathbf{v})}{\partial W_{ij}} &= \left\langle v_i h_j \right\rangle_{data} - \left\langle v_i h_j \right\rangle_{model} \\
\frac{\partial \log p(\mathbf{v})}{\partial a_i} &= \langle v_i \rangle_{data} - \langle v_i \rangle_{model} \\
\frac{\partial \log p(\mathbf{v})}{\partial b_j} &= \left\langle h_j \right\rangle_{data} - \left\langle h_j \right\rangle_{model}
\end{aligned} \quad (5)$$

where angle brackets $\langle\cdot\rangle_P$ denote the expectation governed by the subscript P distribution.

Since there is no connections between hidden units, it is easy to obtain an unbiased value of $\left\langle v_i h_j \right\rangle_{data}$. Given a randomly selected training vector $\mathbf{v}$, the hidden units states h_j have the following probability set to 1.

$$p(h_j = 1|\mathbf{v}) = \sigma(\sum_{i}^{m} v_i w_{ij} + b_j) \quad (6)$$

where $\sigma(x) = 1/(1+e^{-x})$ is the sigmoid function.

In the same way, it is also easy to obtain the probability of visible units to be set to 1 given a hidden vector, as shown below:

$$p(v_i = 1|\mathbf{h}) = \sigma(\sum_{j}^{m} w_{ij} h_j + a_i) \quad (7)$$

The training procedure will be elaborated on in detail in the next section.

2.2.2 Replicated Softmax Model

In this research, a variant RBM named Replicated Softmax Model (RSM) is employed to build the block since it has proven useful for modelling sparse data [24]. Similar to the traditional RBMs, RSM consists of two layer states, i.e., stochastic visible state $\mathbf{v}$ and stochastic hidden state $\mathbf{h}$. The energy function of state $\mathbf{v}$ and $\mathbf{h}$ is computed as follows:

$$E(\mathbf{v},\mathbf{h};\theta) = -\sum_{i=1}^{m}\sum_{j=1}^{n} W_{ij}v_i h_j - \sum_{i=1}^{m} v_i b_i - M\sum_{j=1}^{n} h_j a_j \quad (8)$$

where $\theta = (W, a, b)$ are model parameters to be learned and M is the total number of words occuring in a query or question.

Given the hidden states, the following conditional distribution is realized:

$$p(v_i = 1|\mathbf{h}) = \frac{\exp(b_i + \sum_{j=1}^{n} h_j W_{ij})}{\sum_i \exp(b_i + \sum_{j=1}^{n} h_j W_{ij})} \quad (9)$$

$$p(h_j = 1|\mathbf{v}) = \sigma\left(a_j + \sum_{i=1}^{m} v_i W_{ij}\right) \quad (10)$$

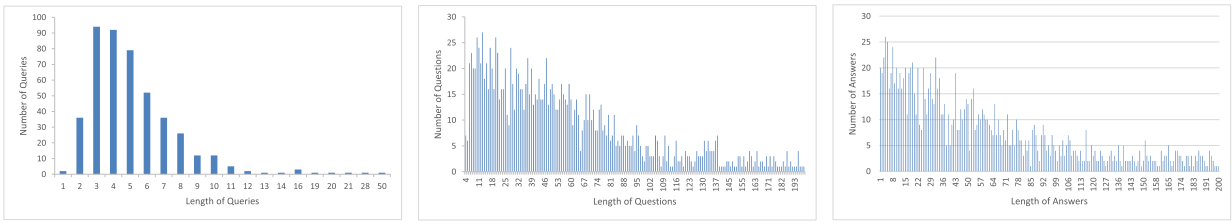
3. Tri-Modal DBN Based Approach

3.1 Data Observation

When users use traditional search engines for searching cQA sites, they usually issue several keywords to represent their information needs in a query. Whereas in a cQA site, users normally express their questions with sentences as it is more convenient to use natural language to ask personal and specific questions.

In this section, we first analyse the Yahoo! Answers query to questions dataset[†] for the discriminative task of judging the semantic similarity level. As illustrated in Fig. 2 (a), in this dataset queries tend to be short and their length aggregates in the range of 3 to 8. Every query is composed of several concise keywords, each of which is dispensable in defining the topics. Speaking of questions, it is found that they are totally different from queries and they are

[†]http://webscope.sandbox.yahoo.com



(a) (b) (c)

Fig. 2 Comparison of content length in queries, questions, and answers. 2 (a) illustrates the relationship between the number of queries and the length of queries, 2 (b) and 2 (c) are questions and answers, respectively

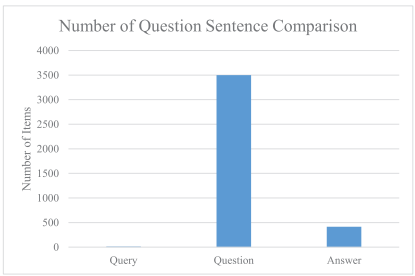


Fig. 3 Number of question sentences in queries, questions, and answers

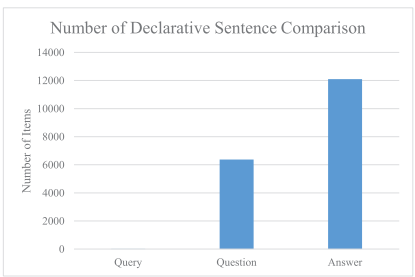


Fig. 4 Number of declarative sentences in queries, questions, and answers.

often in a interrogative mood and contain some meaningless conjunction words. Shown in Fig. 2 (b), a question is usually composed of one or more sentences and only the keywords in question can stand for its topic. The different characteristics of questions and queries make it inappropriate to model them in the same feature space. In addition, from Fig. 2 (c), answers are in paragraphs but seem to have a similar pattern with questions. However, in Figs. 3 and 4, answers are described in the form of declarative sentences, denoting that it is hard to pick up keywords in paragraphs as there too many of them. Zhou et al. conclude some historical methods in solving this problem [29]. As a result, answers' characteristics are also different from the former queries and questions. Therefore, it might be feasible to assume that queries, questions, and answers should be represented in three different feature spaces rather than a single BoW or LSA.

3.2 Tri-Modal Deep Boltzmann Machine

The inspiration to design such a Tri-modal DBN is, as mentioned, queries, questions, and answers have quite different statistical characteristics that make it difficult for a model to directly capture correlations between different input modalities. Moreover, since there are a lot of structures in different input channels, it is difficult to capture highly non-linear correlations across different input modals. Consequently, one more DBN layer is proposed to conduct joint learning of these three different spaces.

3.2.1 Architecture

Figure 5 (b) depicts the architecture of triDBN. First, three separate two-hidden layer DBNs are employed to model correlation within each input mode, as shown in Fig. 5 (b) (right panel). Each DBN with replicated softmax model as a building block is assigned a probability to each training vector:

$$p(\mathbf{v}_{query}) = \sum_{\mathbf{h}^1\mathbf{h}^2} p(\mathbf{h}^1, \mathbf{h}^2) p(\mathbf{v}_{query}|\mathbf{h}^1) \tag{11}$$

$$p(\mathbf{v}_{question}) = \sum_{\mathbf{h}^1\mathbf{h}^2} p(\mathbf{h}^1, \mathbf{h}^2) p(\mathbf{v}_{question}|\mathbf{h}^1) \tag{12}$$

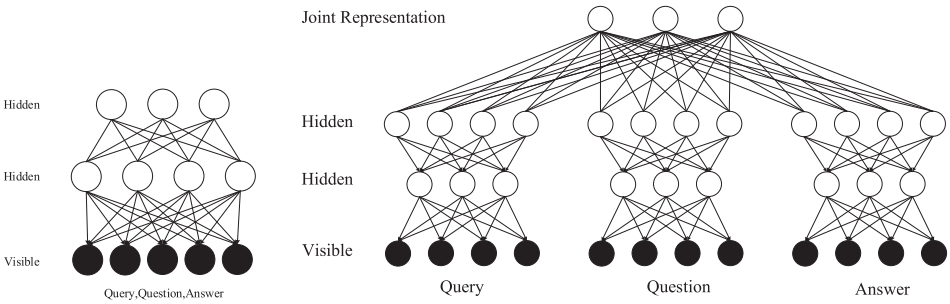
$$p(\mathbf{v}_{answer}) = \sum_{\mathbf{h}^1\mathbf{h}^2} p(\mathbf{h}^1, \mathbf{h}^2) p(\mathbf{v}_{answer}|\mathbf{h}^1) \tag{13}$$

Secondly, three DBNs are combined together by adding an additional layer of DBN to model non-linear correlation among these three different input modals. The joint distribution can be calculated according to the graphic model shown in Fig. 5 (b).

$$\begin{aligned} & p(\mathbf{v}_{query}, \mathbf{v}_{question}, \mathbf{v}_{answer}) \\ &= \sum_{\mathbf{h}^2_{query}, \mathbf{h}^2_{questoin}, \mathbf{h}^2_{answer}, \mathbf{h}^3} p(\mathbf{h}^2_{query}, \mathbf{h}^2_{questoin}, \mathbf{h}^2_{answer}, \mathbf{h}^3) \\ &\times \sum_{\mathbf{h}^1_{query}} p(\mathbf{v}_{query})|\mathbf{h}^1_{query}) p(\mathbf{h}^1_{query}|\mathbf{h}^2_{query}) \\ &\times \sum_{\mathbf{h}^1_{question}} p(\mathbf{v}_{questoin})|\mathbf{h}^1_{questoin}) p(\mathbf{h}^1_{questoin}|\mathbf{h}^2_{questoin}) \\ &\times \sum_{\mathbf{h}^1_{answer}} p(\mathbf{v}_{answer})|\mathbf{h}^1_{answer}) p(\mathbf{h}^1_{answer}|\mathbf{h}^2_{answer}) \end{aligned} \tag{14}$$

3.2.2 Learning Procedures

In the learning procedure, the objective is to minimize all



(a) Architecture of uniDBN (b) Architecture of triDBN

Fig. 5 **Left:** Two-hidden layer DBN using Replicated Softmax Model to model the distribution over sparse word features regarding query, question, and answer in a single feature space. **Right:** Tri-model DBN that models joint representation for queries, questions, and answers that have been viewed in the three feature spaces.

visible units reconstruction error or Kullback-Leibler divergence [30]. First, it is necessary to learn parameters for the three DBNs separately. Take a DBN model of queries for example, as Eq. (5) show, the form of parameter updating equation is quite brief. Simple summation of the data is sufficient to get the expectation of the data distribution. Where the difficulty lies is in obtaining expectation of the model distribution. Since this term is intractable, we have to resort to a sampling method, such as Gibbs-sampling. However, the above method is time-consuming when sampling data from distributions. Hinton et al had proposed k-step Contrastive Divergence algorithm in a greedy layer-wise training fashion [31].

The core idea is to train one RBM at a time, then freeze the parameters and take the activation value of its hidden unit as the training data to train the next RBM. Stacking these RBMs gets a DBN. The other two DBNs are constructed in a similar manner. Finally, the three models are combined by adding another DBN above them. The training procedure is similar to the above described except that the training data is generated by concatenating the activation values of these three DBNs.

3.2.3 Inference

After learning the parameters, it is a must to use the joint representations for classification or retrieval. Here we use two-layer DBN for clarity. Given an instance, $\mathbf{v}_{query}, \mathbf{v}_{question}, \mathbf{v}_{answer}$, the proposed method will first compute the value of hidden variable $\mathbf{h}^1_{query}, \mathbf{h}^1_{question}, \mathbf{h}^1_{answer}$ according to Eq. (10). Afterwards, it utilizes activation from the previous hidden layer to compute the next layer, $\mathbf{h}^2_{query}, \mathbf{h}^2_{question}, \mathbf{h}^2_{answer}$. Finally, joint representation can be obtained as follows:

$$\begin{aligned} &p(h^3|\mathbf{h}^2_{query}, \mathbf{h}^2_{question}, \mathbf{h}^2_{answer}) \\ &= \sigma(\mathbf{W}^3_{query}\mathbf{h}^2_{query} + \mathbf{W}^3_{question}\mathbf{h}^2_{question} \\ &+ \mathbf{W}^3_{answer}\mathbf{h}^2_{answer} + \mathbf{b}^3) \end{aligned} \tag{15}$$

After the above illustrated steps, a tri-model DBN network is built that can generate a joint representation for queries, questions and answers. This joint representation that captures correlations across different input modes can be further used for classification or retrieval tasks. The whole process is depicted in Algorithm 1.

Algorithm 1 1-step Contrastive Divergence Algorithm

Input: $(V_1, \ldots, V_m, H_1, \ldots, H_n)$ *training batch* S
Output: approximate gradient $\Delta\mathbf{W}, \Delta\mathbf{a}, \Delta\mathbf{b}$
  init $\Delta\mathbf{W} = \Delta\mathbf{a} = \Delta\mathbf{b} = 0$
  for all $v \in S$ **do**
    $v^{(0)} \leftarrow v$
    for $i = 1, \ldots, n$ **do**
      sample $h^1_i \sim p(h_i|v^0)$
    end for
    for $j = 1, \ldots, m$ **do**
      sample $v^1_j \sim p(v_j|h^1)$
    end for
    for $i = 1, \ldots, n, j = 1, \ldots, m$ **do**
      $\Delta w_{ij} \leftarrow \Delta w_{ij} + p(H_i = 1|v^{(0)})v^{(0)}_j - p(H_i = 1|v^{(1)})v^{(1)}_j$
      $\Delta b_i \leftarrow \Delta b_j + v^{(0)}_j - v^{(1)}_j$
      $\Delta a_j \leftarrow \Delta a_i + p(H_i = 1|v^{(0)}) - p(H_i = 1|v^{(1)})$
    end for
  end for

4. Experimental Study

4.1 Dataset and Feature Extraction

For the purpose of figuring out whether joint representation models extracted by triDBN can capture correlations across queries, questions, and answers, we apply our proposed method to a well known Yahoo! Answers query to questions dataset for a discriminative task of judging semantic similarity level. This dataset compromises 12850 $< query, question, answer >$ triples. Each item is manually labelled 1, 2, 3 indicating the semantic matching level.

Since we put queries, questions, and answers in different feature spaces, we extract three dictionaries to represent them using a bag-of-words in the visible layer. In order to make the text representation tractable, we only keep frequent words that occur more than five times. The final

size of the respective vocabularies are 1320, 3792, 6210 and these sizes will be used as the size of the visible units.

4.2 Experiment Setting and Evaluation Metric

In the proposed model, the query DBN is composed of an RSM with 1320 visible units and 500 hidden units, followed by another layer that comprises 128 hidden units. Question DBN consists a RSM with 3792 visible units and 500 hidden units, followed by another layer that is composed of 128 hidden units. The answer DBN consists of 6210 visible units and 500 hidden units, followed by another layer with 128 hidden units. On the joint representation layer, an RBM with 384 visible units and 128 hidden units is employed. Each layer is greedy pre-trained for 300 passes using a 1-step CD algorithm which is shown in Algorithm 1. Furthermore, weight-decay with coefficient 0.0001 is used during the training stage in case of over-fitting and the learning rate for each model is 0.02. We also adopt a momentum of 0.5 to get a more reasonable result. All these hyperparameters can be tuned freely. Our study indicates that the result is not sensitive to the number of hidden units but very sensitive to the learning rate and weight decay rate.

We evaluate performance by the effectiveness of classifying the semantic matching degree between query and question. As such the metrics employ accuracy as the main measurement.

$$accuracy = \frac{\sum_i^n 1\{p_i == g_i\}}{\#test\ instance} \tag{16}$$

where g_i is the desired value on semantic matching degree and y_i the predicted value. The dataset is randomly split into training set (70%) and test set (30%). Each experiment is conducted ten times and the average accuracy is presented as the final result.

4.3 Supervised Training and Classification

In the experimental study, the problem of query and question matching is addressed as a classification issue. After getting joint representation of the queries, questions, and answers, we feed them to **liblinear**, a library for large linear regression [32]. Other classifiers such as support vector machine (SVM), neural network can also be used.

4.4 Baseline Methods

In order to show the potential of joint representation extracted by the proposed model on query and question matching, we compared the best version of tri-DBN with three sets of baseline representation methods that puts them in the same feature space or in different feature space but utilizing a shallow model. The first is a simple and widely used BOW representation, the second is an LSA based representation approach, and the final one is a DBN based representation that treats queries, questions, and answers to be in the same feature space.

Randomly guessing (Row 1) is the overall lower bound. Since we have three labels altogether, the accuracy of a random guess is 33.33%.

Bag-of-words (Row 2) is a very traditional representation approach, which simply represents queries, questions, and answers as an unstructured bag of words using a vector whose size is equal to the vocabulary size and puts them in the same feature space. Formally, suppose $x = (q, qu, a)$ is a triple of queries q, questions qu, and answers a. Let $V = (w_{q_1}, w_{q_2}, \ldots, w_{q_m}, w_{qu_1}, w_{qu_2}, \ldots, w_{qu_m}, w_{a_1}, w_{a_2}, \ldots, w_{a_m})$ be the set of words. To make the experimental results comparable, the vocabulary size is kept the same. When judging the matching degree, it is possible to simply feed this representation to the classifier and conduct supervised training, we use the **liblinear** toolkit to implement this method.

uniLSA (Row 3) puts queries, questions, and answers into the same feature space. First, based on a BoW representation, this method conducts LSA, mapping them into a low-dimensional concept vector L. Second, it uses vector L to train the classifier.

triLSA without joint representation layer (Row 4) represents a method regarding queries, questions, and answers to be in different feature spaces. Furthermore, it executes LSA to extract low-dimensional representation $L_{query}, L_{question}, L_{answer}$, respectively. Finally this method concatenates them into a long vector L_{all} and feeds this to **liblinear** to train the classifier.

triLSA (Row 5) puts queries, questions, and answers into different feature spaces. But the difference to the above mentioned method is that after concatenating them into a long vector L_{all}, we further execute LSA to map this representation into a lower one, intending to infer the joint representation using LSA. Furthermore, we utilize this joint representation to train the classifier.

uniDBN (Row 6) uses a single DBN to model correlations between queries, questions, and answers by putting them into the same feature space. Its architectures are depicted in Fig. 5 (a). Visible units are extracted as in the BoW approach. After getting the low-dimensional representation, we also feed them to the **liblinear** toolkit to see the classification effect.

triDBN without joint representation layer (Row 7) regards queries, questions, and answers to lie in different feature spaces. However, we use three deep models rather than LSA to capture correlations. After getting representations, we concatenate them into a long vector to train the classifier.

triDBN (Row 8) is our proposed model. The difference from above mentioned method is that after concatenating them into a long vector L_{all}, we add a further DBN to model correlations between different input modes and get a joint representation. After getting the representation, we also feed them to **liblinear** toolkits to see the classification effect.

The architecture of all the baseline models are described in Table 1 where #dict is the size of vocabulary.

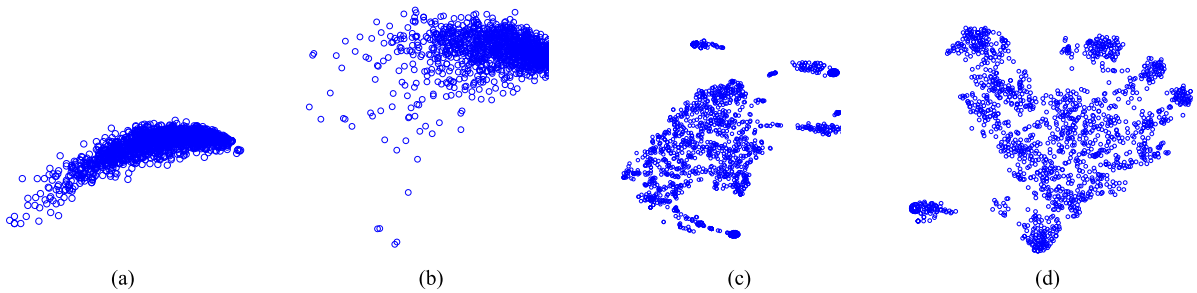


Fig. 6 6 (a) illustrates questions and answers 6 (b) in 2-dimensional representation produced by LSA; 6 (c) depict questions and answers 6 (d) in a 2-dimensional representation introduced by tri-DBN without joint representation layer.

Table 1 Architectures of different methods

#	Method	Architecture
1	Random	None
2	Bag-of-Words	None
3	uniLSA	6520 - 128
4	triLSA without JRL	#dict - 128
5	triLSA	#dict - 128, 384 - 128
6	uniDBN	6520 - 500 - 128
7	triDBN without JRL	#dict - 500 - 128
8	triDBN	#dict - 500 - 128, 384 - 128

Table 2 Accuracy Summary over different methods

#	Method	Accuracy (%)
1	Random	33.33
2	Bag-of-Words	51.22
3	uniLSA	50.44
4	triLSA without JRL	52.85
5	triLSA	43.90
6	uniDBN	45.53
7	triDBN without JRL	44.72
8	triDBN	**64.23**

4.5 Result Analysis and Discussion

To clearly illustrate what DBN has modelled, we further project question and answer representations into a two-dimensional space, as shown in Fig. 6, where each blue circle represents a question or answer in the two-dimensional space.

Figures 6 (a) and 6 (b) show the respective questions and answers as generated by LSA based on representation in Row 4. Figures 6 (c) and 6 (d) show the respective questions and answers as generated by adding another RBM with size of hidden units being 2 based on representation in Row 7. As depicted in Fig. 6, we can see that DBN effectively captures semantic meaning for each instance; In Figs. 6 (d) and 6 (c), questions and answers are grouped into clusters wherein each question or answer has similar meaning. However, in Figs. 6 (a) and 6 (b) and, most of the questions and answers are gathered in a single large indistinguishable blob. This indicates that DBN can outperform LSA in capturing semantic meaning between questions and answers. However, when we plot queries onto a two-dimensional space, we find that neither LSA nor DBN work well: this issue deserves further effort in the future.

Table 2 summarizes the experimental results of our proposed method against the baseline algorithms. Row 1 is the method with randomly assigned similarity values. Row 2 is the term vectors based method. Rows 3 to 5 present results of LSA representation. uniLSA indicates mapping Row 2 representation into a lower dimension and then using it for classification. Row 4 indicates mapping queries, questions, and answers into three separate lower dimensions and concatenating them for classification. Row 5 expresses further mapping the above concatenated representation into a lower dimension to model correlation between input modes. Rows 6 to 8 show the results with DBNs approaches.

From Rows 3 and 4 we can see that putting queries, questions, and answers in different feature spaces can work well, indeed much better than simply putting them in the same feature space. One reason that uniDBN or triDBN without joint representation layer (JRL) lags behind LSAs may be that LSAs can maintain the original information to some degree since it is a shallow model and can only capture pairwise semantic similarity whereas DBN can model highly non-linear correlation. However, simply concatenating their lower dimension representation can hurt performance. But, as row 8 shows, when adding another layer to capture correlations between these separate representations, triDBN can significantly boost the overall performance.

5. Conclusions and Future Work

In this paper, we have investigated the relationship between queries, questions, and answers and found different presentation patterns among them. Inspired by this, we have hypothesized that they locate in different feature spaces. Consequently, we have proposed a tri-modal DBN to extract a unified representation between queries, questions, and answers into a joint feature space. Our experimental study has been conducted on the Yahoo! Answers dataset to compare the unified representations against baselines that utilize BoW or LSA representation as features, both regarding them in the same feature space and distinct feature spaces, in a discriminative task judging query to question seman-

tic matching level. The experimental results reveal that tri-modal DBN can capture semantic relationships between queries, questions, and answers. Taking this unified representation as a feature can significantly improve the overall performance.

In future work, we intend to investigate how this joint representation works in a retrieval task and whether joint representation can help in question recommendation systems that are designed to improve the answer rate in cQA.

Acknowledgements

This work was partially supported by the State Key Laboratory of Software Development Environment of China (No. SKLSDE-2015ZX-23), the National Natural Science Foundation of China (No. 61472021), the National High Technology Research and Development Program of China (No. 2013AA01A601), and the Fundamental Research Funds for the Central Universities.

References

[1] Y. Wu, Q. Zhang, and X. Huang, "Efficient near-duplicate detection for q&a forum," Proceedings of 5th International Joint Conference on Natural Language Processing, pp.1001–1009, 2011.

[2] M.W. Bilotti, J.L. Elsas, J.G. Carbonell, and E. Nyberg, "Rank learning for factoid question answering with linguistic and semantic constraints," Proceedings of 19th ACM Conference on Information and Knowledge Management, pp.459–468, 2010.

[3] S. Zhao, H. Wang, C. Li, T. Liu, and Y. Guan, "Automatically generating questions from queries for community-based question answering," Proceedings of 5th International Joint Conference on Natural Language Processing, pp.929–937, 2011.

[4] Y. Liu, S. Li, Y. Cao, C.-Y. Lin, D. Han, and Y. Yu, "Understanding and summarizing answers in community-based question answering services," Proceedings of 22nd International Conference on Computational Linguistics, pp.497–504, 2008.

[5] M.W. Bilotti, P. Ogilvie, J. Callan, and E. Nyberg, "Structured retrieval for question answering," Proceedings of 30th Annual International ACM Conference on Research and Development in Information Retrieval, pp.351–358, 2007.

[6] R. Socher, J. Pennington, E.H. Huang, A.Y. Ng, and C.D. Manning, "Semi-supervised recursive autoencoders for predicting sentiment distributions," Proceedings of 2011 Conference on Empirical Methods in Natural Language Processing, pp.151–161, 2011.

[7] T.K. Landauer and S.T. Dumais, "Latent semantic analysis," Scholarpedia, vol.3, no.11, p.4356, 2008.

[8] T. Hofmann, "Probabilistic latent semantic analysis," CoRR, vol.abs/1301.6705, 2013.

[9] D.M. Blei, A.Y. Ng, and M.I. Jordan, "Latent dirichlet allocation," Proceedings of Advances in Neural Information Processing Systems 14, pp.601–608, 2001.

[10] R. Salakhutdinov and G. Hinton, "Semantic hashing," International Journal of Approximate Reasoning, vol.50, no.7, pp.969–978, 2009.

[11] H. Ma, M.R. Lyu, and I. King, "Diversifying query suggestion results," Proceedings of the 24th AAAI Conference on Artificial Intelligence, pp.1399–1404, 2010.

[12] P.-A. Chirita, C.S. Firan, and W. Nejdl, "Personalized query expansion for the web," Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, pp.7–14, ACM, 2007.

[13] M. Surdeanu, M. Ciaramita, and H. Zaragoza, "Learning to rank answers to non-factoid questions from web collections," Computational Linguistics, vol.37, no.2, pp.351–383, 2011.

[14] Z. Zheng, X. Si, E.Y. Chang, and X. Zhu, "K2q: Generating natural language questions from keywords with user refinements," Proceedings of 5th International Joint Conference on Natural Language Processing, pp.947–955, 2011.

[15] A. Figueroa and G. Neumann, "Learning to rank effective paraphrases from query logs for community question answering," Proceedings of 27th AAAI Conference on Artificial Intelligence, pp.1099–1105, 2013.

[16] H. Yu, J. Kim, Y. Kim, S. Hwang, and Y.H. Lee, "An efficient method for learning nonlinear ranking svm functions," Information Sciences, vol.209, pp.37–48, 2012.

[17] Y. Bengio, "Learning deep architectures for ai," Foundations and Trends in Machine Learning, vol.2, no.1, pp.1–127, 2009.

[18] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P.P. Kuksa, "Natural language processing (almost) from scratch," Journal of Machine Learning Research, vol.12, pp.2493–2537, 2011.

[19] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck, "Learning deep structured semantic models for web search using clickthrough data," Proceedings of 22nd ACM International Conference on Information and Knowledge Management, pp.2333–2338, 2013.

[20] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," Journal of Machine Learning Research, vol.11, pp.3371–3408, 2010.

[21] G.E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," Neural Computation, vol.18, no.7, pp.1527–1554, 2006.

[22] I.J. Goodfellow, A.C. Courville, and Y. Bengio, "Joint training of deep boltzmann machines," CoRR, vol.abs/1212.2686, 2012.

[23] K.H. Cho, T. Raiko, and A. Ilin, "Gaussian-bernoulli deep boltzmann machine," Proceedings of 2013 International Joint Conference on Neural Networks, pp.1–7, 2013.

[24] R. Salakhutdinov and G.E. Hinton, "Replicated softmax: an undirected topic model," Proceedings of 23rd Annual Conference on Neural Information Processing Systems, pp.1607–1614, 2009.

[25] A. Fischer and C. Igel, "Training restricted boltzmann machines: An introduction," Pattern Recognition, vol.47, no.1, pp.25–39, 2014.

[26] A. Fischer and C. Igel, "An introduction to restricted boltzmann machines," Proceedings of 17th Iberoamerican Congress on Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, vol.7441, pp.14–36, 2012.

[27] J.J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," Proceedings of the national academy of sciences, vol.79, no.8, pp.2554–2558, 1982.

[28] G.E. Hinton, "A practical guide to training restricted boltzmann machines," Neural Networks: Tricks of the Trade (2nd ed.), pp.599–619, 2012.

[29] G. Zhou, L. Cai, J. Zhao, and K. Liu, "Phrase-based translation model for question retrieval in community question answer archives," Proceedings of 49th Annual Meeting of the Association for Computational Linguistics, pp.653–662, 2011.

[30] G.E. Box and M.E. Muller, "A note on the generation of random normal deviates," The Annals of Mathematical Statistics, vol.29, no.2, pp.610–611, 1958.

[31] G.E. Hinton, "Training products of experts by minimizing contrastive divergence," Neural Computation, vol.14, no.8, pp.1771–1800, 2002.

[32] R.E. Fan, K.W. Chang, C.J. Hsieh, X.R. Wang, and C.J. Lin, "Liblinear: A library for large linear classification," Journal of Machine Learning Research, vol.9, pp.1871–1874, 2008.

Nan Jiang received his BSc from Zhejiang University of Technology, Zhejiang, China, in 2015. He is currently a MSc student in School of Computer Science and Engineering at Beihang University, China. His research interests include machine learning, information retrieval, and natural language processing.

Wenge Rong is associate professor at Beihang University, China. He received his PhD from University of Reading, UK, in 2010; MSc from Queen Mary College, University of London, UK, in 2003; and BSc from Nanjing University of Science and Technology, China, in 1996. He has many years of working experience as a senior software engineer in numerous research projects and commercial software products. His area of research covers machine learning, data mining, service computing, and information management.

Baolin Peng is currently a PhD student at the Chinese University of Hong Kong. His research focuses on deep learning for natural language processing, especially end-to-end and data-driven dialog modeling and document summarization. Before coming to Hong Kong, he obtained his MSc from Beihang University and BSc from Yantai University, respectively.

Yifan Nie received his MSc in Sino-French Engineer School at Beihang University, China in 2015 and his BSc in applied mathematics from Beihang University, China in 2012. He is currently pursuing his PhD in the Department of Computer Science and Operations Research at Université de Montréal. His research interests include machine learning, artificial intelligence, information retrieval, bioinformatics, etc.

Zhang Xiong is a professor in School of Computer Science of Engineering of Beihang University and director of the Advanced Computer Application Research Engineering Center of National Educational Ministry of China. He has published over 100 referred papers in international journals and conference proceedings and won a National Science and Technology Progress Award. His research interests and publications span from smart cities, knowledge management, information systems, intelligent transportation systems and etc.