Neurocomputing 334 (2019) 44–58

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Multi-modal multi-view Bayesian semantic embedding for community question answering

Lei Sang[a,b], Min Xu[b,*], ShengSheng Qian[c], Xindong Wu[d]

[a] Hefei University of Technology, China
[b] University of Technology Sydney, Australia
[c] National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, China
[d] University of Louisiana at Lafayette, USA

ARTICLE INFO

Article history:
Received 12 February 2018
Revised 10 August 2018
Accepted 26 December 2018
Available online 31 December 2018

Communicated by Dr. Tie-Yan Liu

Keywords:
Community question answering
Semantic embedding
Multi-modal
Multi-view
Topic model
Word embedding

ABSTRACT

Semantic embedding has demonstrated its value in latent representation learning of data, and can be effectively adopted for many applications. However, it is difficult to propose a joint learning framework for semantic embedding in Community Question Answer (CQA), because CQA data have multi-view and sparse properties. In this paper, we propose a generic Multi-modal Multi-view Semantic Embedding (MMSE) framework via a Bayesian model for question answering. Compared with existing semantic learning methods, the proposed model mainly has two advantages: (1) To deal with the multi-view property, we utilize the Gaussian topic model to learn semantic embedding from both local view and global view. (2) To deal with the sparse property of question answer pairs in CQA, social structure information is incorporated to enhance the quality of general text content semantic embedding from other answers by using the shared topic distribution to model the relationship between these two modalities (user relationship and text content). We evaluate our model for question answering and expert finding task, and the experimental results on two real-world datasets show the effectiveness of our MMSE model for semantic embedding learning.

1. Introduction

Community Question Answer (CQA) sites such as Quora, Yahoo! Answers, Baidu Knows and Zhihu are gaining popularity, for people can exchange knowledge with each other efficiently. These sites usually contain a large number of Question/Answer (Q/A) pairs with other metadata like question's categories and users' relationship. Semantic embedding is one of the most basic tasks to fully reuse these CQA archives, for its wide range of applications such as question retrieval [1,2], expert finding [3] and question answering [4,5]. The goal of semantic embedding is to represent CQA data in a latent semantic space. Let us take question answering as an example, which aims to identify the most relevant answers to the queried question within a collection of answers. Then a user asks a new question in CQA site and the best matched historical answer can be located, the lag time incurred by having to wait for a person to respond can be avoided, thus improving user satisfaction. This is what we call question answering in this article.

One of the greatest challenges in question answering is the lexical gap between the query questions and the candidate answers, since data in CQA often contain incomplete and ambiguous information [1], such as question "What is the best laptop?" with an answer "For a programmer, I recommend Macbook". Although this Q/A pair shares no words in common, they are strongly associated with synonyms, hyponyms, or other weaker semantic associations of words. Semantic embedding has been proven its ability to capture this latent similarity between different words and has attracted lots of attention in the fields of information retrieval and natural language processing research.

Semantic embedding for CQA has multi-view property. This is because semantic information with different granularities should be modeled simultaneously in CQA. In Fig. 1, we show an example about different answers for the question "What should I know before traveling to Sydney?" Since the two main semantic embedding methods, topic model and word embedding, calculate similarity from two different views, we get different answers. Word embedding, such as Skip-gram [19], learn a vector-space representation for each word, based on statistics about how often each word occurs within a local context window of another word. This local view based word semantic embedding method may choose the

* Corresponding author.
E-mail address: Min.Xu@uts.edu.au (M. Xu).

https://doi.org/10.1016/j.neucom.2018.12.067
0925-2312/

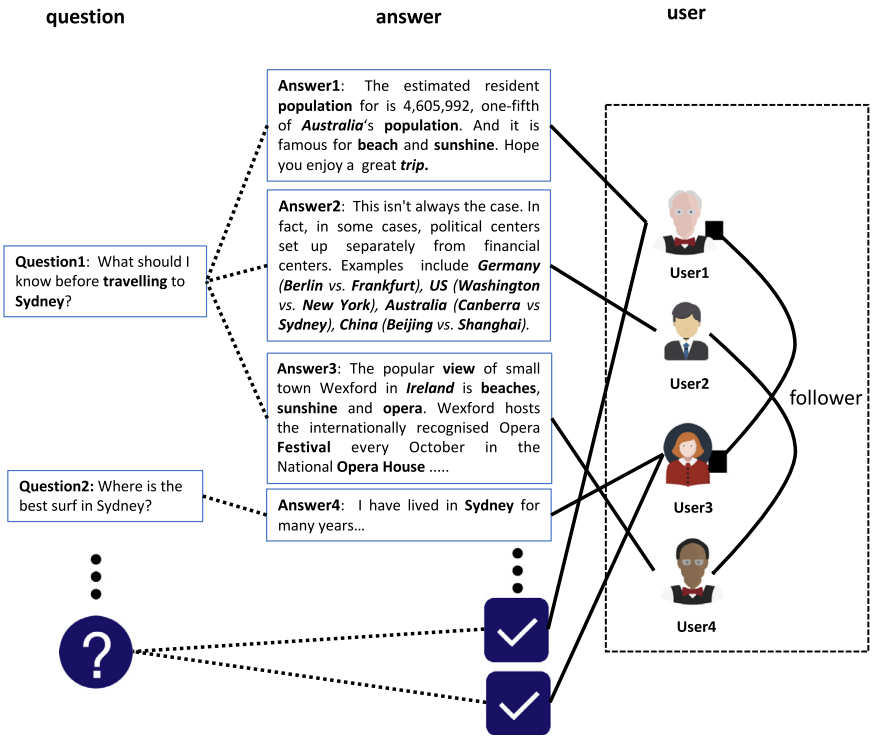


Fig. 1. An example of multi-modal multi-view semantic embedding for CQA data. (i) The two modalities, Q/A pairs and corresponding users are explored jointly. (ii) Semantic embedding was modeled from both local and global view. Single-view based embedding method may lead to a biased result, such as local embedding for answer 2 and global topical embedding for answer 3.

answer 2, because the similarity between city names is very high in local word embedding circumstance. In contrast, topic models, such as Latent Dirichlet Allocation (LDA) [3], take a more global view, which assumes that two words are similar if they are often found in the same document. With this global view, we may choose the answer 3, for "view", "opera house", "sunshine" and "festival" are more likely to appear in the same document to depict the the query term "Sydney" and "travel", while neglecting the important fact that local similarity between distinctive place name "Ireland" and "Sydney" is very low. So neither of this two views can generate unbiased semantic embedding solely. Since local view can help disambiguate word meanings, the global view can also provide useful topical information, it is natural for us to expect to model this two kinds of information simultaneously.

The Sparsity problem of CQA data is another challenging issue [5]. In CQA site, each question is associated with a few answers and thus the question answer pairs are very sparse. Most of the existing method which only utilizes the text content of the question answer pair can hardly learn the general semantic form other questions. As is shown in Fig. 1 each question only connect with few answers. Since user 1 and user 3 are friends with each other, what they answered such as answer 1 and answer 4 may have potential relation with each other, which can provide extra information for the semantic embedding of a single answer and can also alleviate the sparse problem of Q/A pairs. Specifically, the social relationship in CQA provides a natural avenue for enhancing the quality of general semantic embedding from other answers, because the Q/A pairs are all posted by users and CQA site is based on user interaction. Users and Q/A content are an indivisible whole and can promote each other. Therefore, it is necessary and important to take the multi-modal CQA data(text content and user relationship) into consideration simultaneously.

In this paper, we adopt a Bayesian embedding method to exploit the comprehensive text semantic information from both local view and global view, and exploit the user social relationship to solve the sparsity problem in CQA tasks. Our proposed framework is named as **MMSE** (**M**ulti-modal **M**ulti-view **S**emantic **E**mbedding). The goal is to learn latent semantic embedding for multi-modal data from multi-view. As shown in Fig. 2, the input consists of Q/A pairs and answers' social network. Given the input, the proposed MMSE is adopted to learn the semantic embedding, which can preserve the latent relation between Q/A pairs as well as the user interaction structure. With the derived multi-modal multi-view semantic embedding, when a certain question is queried, MMSE can retrieve the best answer for it. Moreover, the expertize of users learned by our MMSE can be beneficial for other CQA tasks, such as expert finding. Compared with existing methods, the contributions of this work are three-fold.

- We propose a novel multi-modal multi-view semantic embedding (MMSE) framework for CQA data. The proposed MMSE can effectively combine the advantage of both local and global context information via a joint Bayesian embedding, and is able to discover both topic structures and word embedding from the corpus.
- The proposed MMSE can collaboratively learn the shared feature embedding from both Q/A pairs and user social network to deal with the data sparse problem with considering the multi-modal property.
- The semantic embedding results can be applied to many applications such as question answering and expert finding, which can achieve much better performance than existing methods.

The rest of the paper is organized as follows. In Section 2, the related work is reviewed. Section 3 introduces the formulation of the MMSE. The applications for question answering is presented in Section 4. In Section 5, we report and analyse extensive experimental results. Finally, we conclude the paper with future work in Section 6.

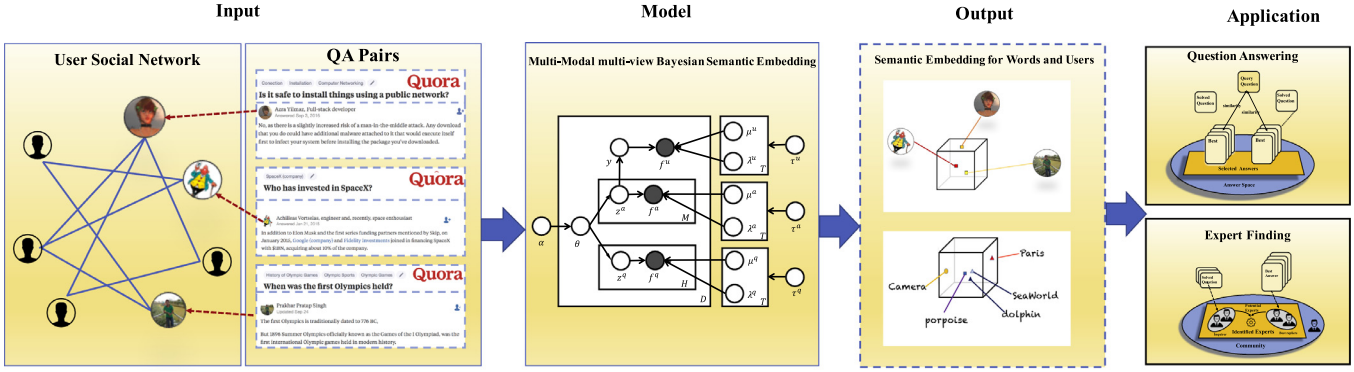


Fig. 2. The flowchart of the proposed multi-modal multi-view semantic embedding model.

2. Related work

In this Section, we briefly review previous methods which are most related to our work including semantic method and question answering.

Semantic embedding: There are two mainstream semantic embedding methods. On the one hand, topic models are a powerful unsupervised tool to reveal the latent semantic structure based on its global document-word context information. Several variants of topic model are proposed for multi-modal modeling. The Author-Topic Model [6] learns topics conditioned on the mixture of authors that composed a document. Mimno et al. [7] propose a Dirichlet-multinomial regression (DMR) topic model that includes a log-linear prior on document-topic distributions that is a function of observed features of the document, such as author, publication venue, references, and dates. However, these model use discrete representation for observed variables, and none of them are appropriate for capturing the directed interactions and relationships between user. On the other hand, continuous embedding learning has proven to be able to capture semantic regularities in both words and networks by learning the local co-occurrence context information [8–10]. Word embedding, such as Skip-gram [11] learning low-dimension vector representation for each word. Network learning such as DeepWalk and LINE [12,13] learn latent representations of a user in a network, which can also preserve the network structure information. Chang et al. [14] propose the embedding method for heterogeneous networks. However, the semantic embedding learned from this method either use global context, or the local context, which may not be suitable for our problem.

To make use of both the global context and the local context, many hybrid models have been proposed to combine topic model and embedding model [8,15,16]. Topical Word Embedding (TWE) [16] use latent topic models for assigning topics to each word in a corpus and learn topic specific word representations. The major difference between this work and ours is that they did not aim to integrate topic modeling and word embedding and yet only use pre-trained topic structures as the input of word embedding models. By replacing the original discrete word types in LDA with continuous word embedding, Gaussian-LDA [15] has shown that the additional semantics of word embedding can be incorporated into topic models and further enhance the performance. However, the aforementioned models fail to directly model the user relation information. More recently, Yang et al. [17] try to model the user information and knowledge concept embedding in a shared topic space. Different from [17], our work focuses on learning semantic embedding for CQA data to deal with the sparse problem.

Community question answering: Question Answering is about selecting the best answer for a query question, which is an important task to fully use CQA corpus [18]. This is a well-researched problem and closely related to the retrieval task. State-of-the-art methods are based on language model (LM). For example, Ji et al. [19] propose a category-smoothed language model (CLM) for question retrieval, which views category-specific term saliency as the Dirichlet hyper-parameter that weights the parameters of LM. Xue et al. [20] propose a translation-based language model for question answer pairs. While useful, the effectiveness of LM approach is dependent on the availability of high-quality parallel question-answer pairs which are always troubled by noise issue. There are also some studies aim to better adapt semantic embedding methods to the needs of question answering. Zuccon et al. [21] propose to employ local word embedding within the translation language model for question answering by capturing latent semantic relations between the words in question and answer. Cai et al. [22] propose a topic model incorporated with the category information into the process of discovering the latent topical embedding in the content of questions.

Another crucial problem in CQA is expert finding, which is to choose the right experts for answering the questions posted by the users. The existing work for the problem of expert finding can be categorized into two groups: the authority-oriented approach [23,24], and the topic oriented approaches [25,26]. The authority-oriented expert finding methods are based on link analysis for the ask-answer relation between users in the rating matrix. For example, Bouguessa et al. [23] choose the experts to answer the questions based on the number of best answers provided by users, which is an In-degree-based method. Zhu et al. [24] measure the category relevance of questions and rank user authority in extended category link graph. The topic-oriented expert finding methods are based on latent topic modeling techniques for the content of the questions. Deng et al. [25] develop latent user model for the problem of expert finding in DBLP bibliography. Weng et al. [26] choose the topic-sensitive inuential users by leveraging topic models for answering the questions.

However, these semantic embedding based model ignore the multi-modal multi-view property and only utilize partial information in CQA. Different from the above methods, we propose a novel MMSE model to learn the semantic embedding which makes full use of the rich multimodal contents from both local and global view, and then incorporate this semantic embedding to a retrieval model for question answering.

3. The proposed MMSE model

We propose MMSE, a Bayesian embedding model for learning social graphs, which jointly models multiple modalities and multiple semantic views of CQA data. Our goal is to learn a shared latent topic space to generate network-based user embeddings and text-based word embeddings in two different embedding spaces.

Table 1
Key notations of our proposed MMSE model.

Notations	Description
D	The number of Q/A pair documents
T	The number of topics
M	The number of words in answer
H	The number of words in question
W	The number of unique words
E^u, E^w	The dimension of user and word embedding
α	The hyperparameter of the Dirichlet prior on θ
θ	The multinomial distribution of topic specific to the Q/A text
y	The topic of each user
z^q, z^a	The topic of single word in question or answer
f^u	The network based user embedding
f^a, f^q	The word embedding of Q/A text
μ^u, μ^q, μ^a	The mean of Gaussian distribution for user embedding and word embedding
λ^u, λ^q, λ^a	The precision of Gaussian distribution for user embedding and word embedding
τ^u, τ^q, τ^a	The hyperparameter of the normal Gamma distribution

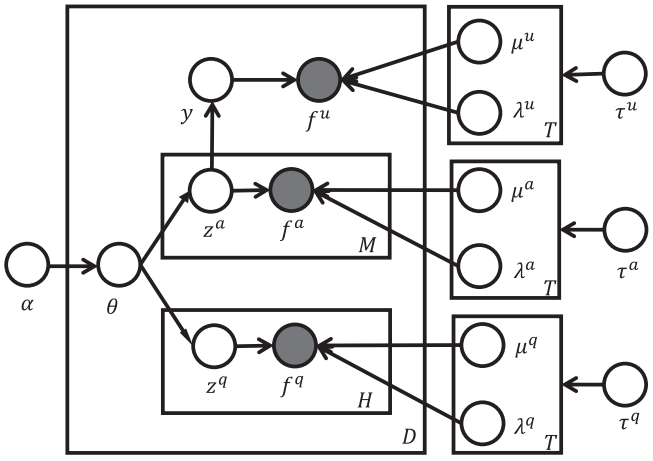


Fig. 3. The graphical representation of the proposed multimodal multi-view Semantic Embedding model (MMSE). Each document d contains a single social network user, one answer posted by the user and question related to the answer. Embeddings are observed variables.

The input of our problem is multi-modal CQA data, which includes pre-trained user embedding f^u and word embedding f^q, f^a of Q/A text. In other words, embeddings are given as observed variables in our model. We use the Skip-gram model [11] to learn word embeddings, and use DeepWalk [12] to learn network-based user embeddings. In brief, MMSE tries to get a more comprehensive word embedding and user embedding from pre-trained embedding representation by considering the multiple modalities and multiple semantic views.

3.1. Generative process

The graphical representation of the proposed MMSE is shown in Fig. 3. In our model, we assume there are two kinds of global topics in multi-modal CQA data: (1) z^q and z^a are latent topic assignments for question words and answer words, respectively. Since the asker and answerer may express similar meanings with different words, Q/A pairs should share the same topic space. (2) To jointly model multiple modalities, we assume that the latent topic of users is generated from their related answer topic space, thus each user is associated with a multinomial distribution over answer topics. Both user topic and Q/A content topic are modeled in a global semantic view. Table 1 lists the key notations.

To model the multi-view property of semantic embedding, MMSE introduce local semantically coherent to the topic model by replacing the original discrete word types in topic model with continuous word embedding and user embedding. This is implemented by considering embedding vector as drawing from several Gaussian distributions in the topic model framework, in view of word embedding and user embedding can capture a notion of centrality in space [15,17]. Thus, in the joint model, the semantic embedding for multi-modal user and Q/A pairs can be learned from both global topical view and local embedding view. The MMSE takes pre-trained word embedding f^q, f^a and user embedding f^u as inputs, where the f^u is obtained by DeepWalk [12] to preserve the user social network structure information and f^q, $andf^a$ are learned from Skip-gram model [11]. The dimension of word and user are E^w and E^u, respectively.

Our multi-modal semantic embedding is continuous vectors, we characterize each dimension of the embedding as a univariate Gaussian distribution with parameter (μ, λ) with Normal-Gamma distribution priors $Normal - Gamma(\tau = \{\mu_0, k_0, \alpha_0, \beta_0\})$ [27]. There are two major reasons to choose univariate Gaussian rather than multivariate Gaussian to generate the embedding. First, model with univariate Gaussian distribution can obtain better performances than that with multivariate Gaussian, which might be caused by the relatively independence between each dimension of embedding representation. The word analogical task introduced by Mikolov et al. [28] means that the influence between words can be simply computed by addition and subtraction, such as [king] - [man] + [woman] $\approx$ [queen]. Because of these, we can assume that the dimensions between a word embedding are relatively independent which means there are no cross influence between them [29]. So it is much better to use the univariate Gaussian to describe the distribution of each dimension. Second, model univariate Gaussian distribution is more computationally efficient. A univariate normal distribution is described using just the two parameters namely mean μ and precision λ, $X \sim N(\mu, \lambda)$. For a multivariate distribution, we need a third parameters Σ, i.e., the correlation between each pair of random variables, $X \sim N(\mu, \Sigma)$. This is what distinguishes a multivariate distribution from a univariate distribution. If there are n dimensions in the embedding, we will have $n * (n-1)/2$ pairs of correlations.

The choice of Gaussian distribution is also justified by that Euclidean distances between Gaussian distributions is straightforward to calculate, naturally asymmetric, and has a geometric interpretation as an inclusion between families of ellipses [30]. The θ_d is the multinomial topic distribution of document d, which shared by both question and answer text. α is the hyperparameter of the

Dirichlet distribution. $\tau = \alpha_0, \beta_0, k_0, \mu_0$ is hyperparameter of the normal Gamma distribution. Accordingly, the generative process of a document in the proposed MMSE model can be described as follows:

1. For each topic t, and for each dimension
   (a) Draw μ_t^u, λ_t^u from NormalGamma(τ^u)
   (b) Draw μ_t^q, λ_t^q from NormalGamma(τ^q)
   (c) Draw μ_t^a, λ_t^a from NormalGamma(τ^a)
2. For each document D_i
   (a) Draw a multinomial distribution θ from Dir(α)
   (b) Form each answer word w^a in d
      i. Draw a topic z^a from Multi(θ)
      ii. For each dimension of the embedding of w^a, draw f^a from $N(\mu_z^a, \lambda_z^a)$
   (c) Draw a topic y uniformly from all z^a
   (d) For each dimension of the embedding of user u, draw f^u from $N(\mu_z^u, \lambda_z^u)$
   (e) Form each question word w^q in d
      i. Draw a topic z^q from Multi(θ)
      ii. For each dimension of the embedding of w^q, draw f^q from $N(\mu_z^q, \lambda_z^q)$

where notations with superscript {u q a} denote parameters defined for user, question and answer respectively. Particularly, during the model learning process, we assume the prior (τ^u, τ^q, τ^a) distributions follow symmetric NormalGamma, which is the conjugate prior for Gaussian distribution.

3.2. Model inference

Exact inference is often intractable in many topic models and appropriate methods must be used, such as variational inference [31] and Gibbs sampling [32]. Let hyper-parameters {α, τ^u, τ^a, τ^q} be denoted as Ψ, and hidden variables {θ, μ^u, λ^u, μ^q, λ^q, μ^a, λ^a} as Φ and we need to estimate the latent variables conditioned on the observed variables, namely $p(z^q, z^a, y|f^u, f^q, f^a, \Psi, \Phi)$. We employ collapsed Gibbs sampling method to obtain samples of latent variables and estimate unknown parameters{θ, μ^u, λ^u, μ^q, λ^q, μ^a, λ^a} in the proposed MMSE. In a Gibbs sampler, one iteratively samples new assignments of latent variables by drawing from the distributions conditions on the previous state of the model. We list the update rules for the latent variables {y, z^a, z^q} as follows:

$$p(y_d|y_{-d}, z, f^u, f^q, f^a) \propto (n_d^t + l) \prod_{e=1}^{E^u} G'(f^u, y, t, e, \tau^u, d). \tag{1}$$

$$p(z_{dm}^a = t|z_{-dm}^a, z^q, y, f^u, f^q, f^a) \propto (n_{dm}^{y_d} + l)(n_{dm}^t + \alpha_t) \times \prod_{e=1}^{E^w} G'(f^a, z^a, t, e, \tau^a, dm). \tag{2}$$

$$p(z_{dh}^q = t|z_{-dh}^q, z^a, y, f^u, f^q, f^a) \propto (n_{dh}^t + \alpha_t) \times \prod_{e=1}^{E^w} G'(f^q, z^q, t, e, \tau^q, dh). \tag{3}$$

where the superscript $-d$ denotes a counting variable that excludes the user in document d, $-dm$ and $-dh$ means ruling out m-th answer word and h-th question word in a document d, respectively. z_{dm}^a is the topic of the m-th answer word in document d, and z_{dh}^q is the topic of the h-th question word in document d. Similarly, n_{dm}^t is the number of answer words assigned to topic t in document d, and n_{dh}^t is the number of question words assigned to topic t in document d. $\tau = \{\alpha_0, \beta_0, k_0, \mu_0\}$ are the hyperparameter of the NormalGamma distribution. To eliminate zero counts, we use Laplace smoothing parameter l, which can simply add smoothing one to each count.

The function $G'(\cdot)$ is defined as follows:

$$G'(f, y, t, e, \tau, d) = \frac{\Gamma(\alpha_n)}{\Gamma(\alpha_{n'})} \frac{\beta_{n'}^{\alpha_{n'}}}{\beta_n^{\alpha_n}} \left(\frac{k_{n'}}{k_n}\right)^{\frac{1}{2}} \frac{(2\pi)^{-n/2}}{(2\pi)^{-n'/2}}. \tag{4}$$

with

$$a_n = a_0 + n/2,\ k_n = k_0 + n,\ \mu_n = \frac{k_0\mu_0 + n\bar{x}}{k_0 + n},$$
$$\beta_n = \beta_0 + \frac{1}{2}\sum_{i=1}^{n_t}(x_i - \bar{x})^2 + \frac{k_{0n}(\bar{x} - \mu_0)^2}{2(k_0 + n)}. \tag{5}$$

where n denotes the times of ith dimension in vector embedding assigned to topic t; x denotes the concatenated vector of the e-th dimension of f_i with $y_i = t$; $n' = n - n_d$, n_d denotes the number of f which satisfies $y = t$.

The conditional probabilities (1) can also be written as two terms p_{global} and p_{local}:

$$p(y_d|y_{-d}, z, f^u, f^q, f^a) \propto (n_{dm}^t + l) \prod_{e=1}^{E^u} G'(f^u, y, t, e, \tau^u, d) \triangleq p_{global} * p_{local}. \tag{6}$$

Das et al. [15] embed word types into the space of Gaussian distributions, and learn the embedding directly in that space, which demonstrated the effectiveness of Gaussian distribution on the modeling of word embedding. As we can see, p_{local} is related to the second part of (1) and depict the semantic embedding in local view. The first part of (3.1) $n_u^t + l$ is the statistic count of topic number, which can model the global topical context of the data. In fact, the conditional probabilities may largely depend on p_{local}, due to the high dimension of E^u. As a result, we cannot make full use of the global topic distribution information. To address this problem, we introduce a balance weight parameter c to control the weights of various parts. Then formula (1) can be written as :

$$p(y_d|y_{-d}, z, f^u, f^q, f^a) \propto (n_{dm}^t + l)^c \prod_{e=1}^{E^u} G'(f^u, y, t, e, \tau^u, d). \tag{7}$$

We can balance p_{local} and p_{global} by adjusting the weight c. When $c = 1$, it is the same with the original form in Eq. (1). Similarly, the conditional probabilities of the latent topic equation (A.28) (3) can also be modified by introducing this balance weight.

3.3. Parameter update

After finishing the Gibbs sampling training, we can estimate the parameters θ, μ^u, λ^u, μ^q, λ^q, μ^a, λ^a just like [17,32]. Therefore, we have:

$$\theta_d^t = \frac{n_a^t + n_q^t + \alpha_t}{\sum_{t=1}^{t}(n_a^t + n_q^t + \alpha_t)},$$
$$\mu_t = \frac{k_0\mu_0 + n\bar{x}}{k_0 + n},$$
$$\lambda_t = \frac{\alpha_0 + n/2}{\beta_0 + \frac{1}{2}\sum_i (x_i - \bar{x})^2 + \frac{k_0 n(\bar{x} - \mu_0)^2}{2(k_0 + n)}}. \tag{8}$$

Inspired by Yang et al. [17], we update the embedding during inference to fine-tuning pre-trained word embedding and user embedding for a task-specific representation with both global and local semantic. Let the S be the number of users, and the W be the vocabulary size of Q/A pairs content.The log-likelihood of the data given the model parameters is

$$L = \sum_{s=1}^{S}\sum_{t=1}^{T}\sum_{e=1}^{E^u}\left(-\frac{\lambda_{te}^u}{2}\right)(f_{se}^u - \mu_{te}^u)^2 + \sum_{w=1}^{W}\sum_{t=1}^{T} n_w^t \sum_{e=1}^{E^w}\left(-\frac{\lambda_{te}^q}{2}\right)(f_{de}^q - \mu_{te}^q)^2 + \sum_{w=1}^{W}\sum_{t=1}^{T} n_w^t \sum_{e=1}^{E^w}\left(-\frac{\lambda_{te}^a}{2}\right)(f_{de}^a - \mu_{te}^a)^2. \tag{9}$$

Note that we cannot directly solve for the optimal by setting the gradient to zero. Instead, we need to use the gradient ascent method to find the optimal result. This method changes the parameter values to increase the log-likelihood based on one example at a time. The gradients for f^q, f^a, f^u are then calculated as

$$\frac{\partial L}{\partial f_{se}^u} = \sum_{t=1}^{T} -\lambda_{te}^u (f_{se}^u - \mu_{te}^u). \tag{10}$$

$$\frac{\partial L}{\partial f_{we}^q} = \sum_{t=1}^{T} n_w^t (-\lambda_{te}^q)(f_{te}^q - \mu_{te}^q). \tag{11}$$

$$\frac{\partial L}{\partial f_{we}^a} = \sum_{t=1}^{T} n_w^t (-\lambda_{te}^a)(f_{te}^a - \mu_{te}^a). \tag{12}$$

The word embedding f^a and f^q share the same vocabulary, and they update in turn. To eliminate scale difference between user embedding and word embedding, the resulting embedding of f^q, f^a, f^u are then normalized by:

$$f \leftarrow \frac{f}{\| f \|_2}. \tag{13}$$

The model training procedure is summarized in Algorithm 1. With the conditional distributions and parameter update above, we can construct a Markov chain to learn our model with Gibbs sampling. The Gibbs sampling algorithm runs over the three periods: initialization, burn-in and sampling. (1) We first initialize the algorithm by randomly assigning a topic to y, z^q, z^a with uniform distribution. (2) And then we finish the burn-in stage in t_b iterations based on the Markov chain, as outlined in Algorithm 1. In the burn-in stage, to eliminate the impact of the initial value of the topic implicit variable on the model, we do not update the model parameters or embedding representations. (3) To obtain the resulting model parameters from a Gibbs sampler, several approaches exist. One is to just use only one read out, another is to average a number of samples which we used, and often it is desirable to leave an interval of t_p iteration between subsequent read-outs to obtain decorrelated states of the Markov chain. This interval is often called thinning interval or sampling lag.

In Algorithm 2, we give the detailed procedure of embedding update for each embedding representation. When training the topic model, it is often useful to reduce learning rate as the training progresses. Every time before the gradient decent, we first calculate the log-likelihood of the model, and then calculate the log-likelihood after iteration. If the log-likelihood increase, it means that the learning rate is appropriate, and we adopt the embedding expression after the gradient descent. If the log-likelihood seems to drop, indicating that the current learning rate is too high, we multiply the learning rate by a decay, and abandon the update of the current iteration's embedding representation. We set the learning rate or embedding update $lr^u = 1e{-}3$, $lr^w = 1e{-}5$, and the decay is set to 0.5.

Algorithm 1: Model Training.

Input : Hyperparameter of model τ^u, τ^q, τ^q;Initial embedding representation f^u, f^q, f^a; Iteration times of burn-in t_b; Maximum iteration times t_m; Iteration times of hidden topic t_l; Iteration time of parameter updating t_p
Output: Hidden topic y, z^q, z^a; Model parameter $\lambda^u, \lambda^q, \lambda^a, \mu^u, \mu^q, \mu^a, \theta$; Updated embedding representation f^u, f^q, f^a

```
// Initialization
Random sample topic for y, z^a, z^q
// Burn-in
for t ← 1 to t_b do
    foreach Q/A text word hidden topic z^q, z^a do
        Sample topic z^q, z^a with Eqs. (2)(3), respectively
    foreach User embedding topic y do
        Sample topic y with Eq. (1)
// Sampling
for t ← 1 to t_m do
    for t' ← 1 to t_l do
        foreach Q/A text word hidden topic z^q, z^a do
            Sample topic z^q, z^a with Eqs. (2)(3), respectively
        foreach User embedding topic y do
            Sample topic y with Eq. (1)
        if t_p iterations have occurred since the last time parameter was read in then
            Calculate the parameter with Eq. (8)
            Average the current parameter and last one
    Embedding updating(Cf.Algorithm 2)
```

Algorithm 2: Embedding updating.

Input : Model parameter $\lambda^u, \lambda^q, \lambda^a, \mu^u, \mu^q, \mu^a, \theta$; Old embedding representation f^u, f^q, f^a; Iteration time of updating t_e; Initial learning rate of user embedding lr^u;Initial learning rate of word embedding lr^w; learning rate decay *decay*
Output: New embedding representation f^u, f^q, f^a

```
for t ← 1 to t_e do
    llh_old ← current log-likelihood of MMSE
    foreach User embedding topic f^y do
        calculate gradient g with Eq. (10)
        f^u ← f^u + lr^u · g
    foreach User embedding topic f^q do
        calculate gradient g with Eq. (11)
        f^q ← f^q + lr^w · g
    foreach User embedding topic f^a do
        calculate gradient g with Eq. (12)
        f^a ← f^a + lr^w · g
    llh_new ← updated log-likelihood
    if llh_new ≥ llh_old then
        accept the embedding update
    else
        refuse the embedding update
        lr^u ← lr^u · decay
        lr^w ← lr^w · decay
return f^u, f^q, f^a
```

4. Question answering

In this Section, we introduce how to leverage our semantic embedding model for question answering task. We consider creating a retrieval model with the learned semantic embedding. There have been some successful attempts in retrieval task to use a more general representation method, namely Bag-of-Word-Embedding (BoWE), where both question and answer can be represented by variable length sets of word embeddings. Queried questions and the existing answers represented by BoWE, which provides a better foundation for semantic level matching than previous Bag-of-Words (BoW) method. A popular strategy for using BoWE in IR referred to as the Word Mover's Distance (WMD) [33], which estimates similarity between pairs of documents by estimating the minimum cumulative distance in the embedding space that words from a document need to travel to match words from a second document. In practice, however, this strategy in infeasible, since we

have to calculate the distance with each word in the vocabulary. The time complexity is $O(n^3 \log n)$, where p denotes the number of unique words in the documents. It is too expensive for online retrieval and ranking.

We borrow an idea form Non-linear Word Transportation (NWT) [34], and consider reducing the computational complexity by pruning the document nodes and corresponding edge if the document words are too distant from the query words. Specifically, suppose we learned a word embedding matrix $W \in \mathbb{R}^{K\times|V|}$ for a vocabulary with finite $|V|$ word by previous proposed MMSE, where the i-th column, $w_i \in \mathbb{R}^K$, represents the embedding of the i-th word in the K-dimensional space. Both question and answer are represented as BoWE. We denote the answer text as $A = \{(w_1^a, tf_1), \ldots, (w_n^a, tf_n)\}$ where tf_i denotes the term frequency, and similarly the queried question as $Q = \{(w_1^q, qtf_1), \ldots, (w_n^q, qtf_n)\}$ where qtf_i denotes the term frequency of j-th word in the question. In the view of transportation view, NWT assume each exiting answer has fixed information capacity, while query question has unlimited capacity to accumulate as much relevance information from an answer text. The information gain of transporting denote as "profit", and the total profit on each query question should obey the law of diminishing marginal returns. Finally, the target of NWT is to find the answer that can bring the maximum net returns for a given query.

Based on the above idea, we estimate relevance between a query question and an answer by finding a set of optimal flows $F = \{f_{ij}\}$ that satisfy

$$\begin{aligned} \max \quad & \sum_{j\in Q} \log \sum_{i\in A} f_{ij} r_{ij}, \\ \text{subject to}: \quad & f_{ij} \geq 0 \qquad \forall i \in A, \forall j \in Q, \\ & \sum_{j\in Q} f_{ij} = c_i \qquad \forall i \in A. \end{aligned} \tag{14}$$

where f_{ij} denotes how much capacity of the i-th answer text word flow to j-th query question word, r_{ij} denotes corresponding transportation profit, and c_i denotes the information capacity of the i-th answer text word. To alleviate the bias problem of varying lengths in IR, here we define the document word capacity c_i using Bayesian smoothing with Dirichlet priors:

$$c_i = \frac{tf_i + \mu \frac{cf_i}{|C|}}{|D| + \mu}. \tag{15}$$

A straightforward idea to define transportation profit is the semantic closeness between two words, such as cosine similarity between word embedding.

$$r_{ij} = \widehat{cos}(w_i^a, w_j^q) = \max(cos(w_i^a, w_j^q), 0), \forall i \in A, \forall j \in Q. \tag{16}$$

The truncated cosine similarity $\widehat{cos}(w_i^a, w_j^q)$ is used to avoid negative profit. However, simply using cosine similarity as the transportation profit would over-emphasize the importance of semantic matching. Following the literature [34], we introduce a matching risk parameter α to control the profit gap between exact matching and semantic matching.

$$r_{ij} = \widehat{cos}(w_i^a, w_j^q)^\alpha. \tag{17}$$

Since the risk of matching semantically related words highly depend on the discriminative power of word, here we simply define the risk parameter as a function of *idf* and obtain the following profit definition.

$$r_{ij} = \widehat{cos}(w_i^a, w_j^q)^{g(idf_i)} \quad \forall i \in A, \forall j \in Q. \tag{18}$$

where

$$g(idf_i) = idf_i + b, \quad idf_i = \frac{N - df_i + 0.5}{df_i + 0.5}. \tag{19}$$

Table 2
Statistics of the two CQA datasets.

Dataset	Quora	Zhihu
Question number	444,138	73,619
Answer number	887,771	421,029
User number	95,951	36,428

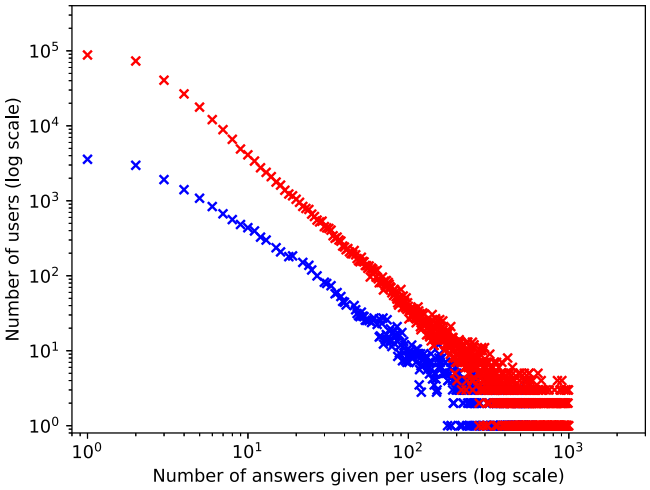


Fig. 4. Distribution of number of answers given per user. A small fraction of users answer a lot of questions while many users answer a few number of questions.

In [34] df_j denotes the document frequency of the j-th query word, N denotes the total number of documents in the corpus, and b is a free parameter denoting the default offset of the risk.

5. Experiments

In this section, we present the experiments on two popular used question answering datasets to show the effectiveness of the proposed method.

5.1. Datasets and evaluation

The two datasets are Quora in English and zhihu in Chinese. Table 2 gives the details of the two datasets.

- *Quora:* For English, we use a publicly available Question-Answering data [35] sets which is obtained from a popular question answering site, Quora. The dataset contains 444,138 questions, 95,915 users and 887,771 answers from Quora, together with user's following relationship in Twitter.
- *Zhihu:* For Chinese, we use a crawler to collect the question-answer pairs and user relationship information from a popular Chines question answering site Zhihu. Chinese sentences are segmented into word in advance. According to the Weibo IDs extracted from the Zhihu user account, we have crawled their follower relationship from Weibo's social network (Limited by weibo API, only 200 followers at most can be obtained for single queried ID).

Due to the nature of the community, the contribution of each user is different based on their interests and availability. As each user directly connects to their answers, we plots the distribution of number of answers given per user in Fig. 4. We observe user answer distribution is a power-law distribution, which means the answers for most users are relatively small.

For the ground truth, we generate (q, a^+, a^-), where the (q, a^+) is original question/answer pair, and y^- is randomly selected answers. In our experiment, a set of candidate answers is created

with size 10 (one positive + nine negative) for each question in the testing. All models parameters are learned from the training question answer pairs data. The hyperparameters are tuned on a validation set (as part of the training set). For all English data, we remove stop words and conduct stemming, and for all Chinese data, we conduct Chinese word segmentation.

Since question answering is similar to the task of question answer retrieval, we use nNDCG and P@N [36] on random answers pool with size 10 (one positive + nine negative) to measure the performance of different retrieval models. The premise of nDCG is that highly relevant documents appearing lower in a ranking list should be penalized as the graded relevance value is reduced logarithmically proportional to the position of the result. nDCG accumulated at a particular rank position p is defined as:

$$nDCG = \frac{DCG}{IDCG}, \quad DCG = rel_1 + \sum_{i=2}^{\pi_i} \frac{rel_i}{log_2(i+1)}.$$

where the one positive answer $rel = 1$, for other answers $rel = 0$. Note that in a perfect ranking algorithm, the NCG will be the same as the IDCG producing an nDCG of 1.0. All nDCG calculations are then relative values on the interval 0.0–1.0 and so are cross-query comparable.lu2013deep, shen2015question.

P@N This criterion is the fraction of the top N retrieved questions that are relevant to the queried questions, given by

$$P@N = \frac{1}{|Q|} \sum_{q \in Q} \frac{N_{q,N}}{N}.$$

where $N_{q,N}$ denotes the number of relevant questions among the top N returned rank list for question q. We use P@1 for evaluation in our experiment.

5.2. Baselines

We take the BOW based traditional information retrieval models such as BM25 and LM as the baseline models. In addition to that, we also compared our model with several state-of-art BoWE based method. The detailed descriptions of these methods are listed as follows.

- *BoW* is a simple representation method used in natural language processing and information retrieval. In our experiment, questions and answers are represented by BoW feature and then the similarity between them is calculated to rank the candidate answers for each query question.
- *BM25* [37] is a classical probabilistic IR model that consider the number of occurrences of each query term in the document (term-frequency) and the corresponding inverse document frequency of the same terms in the full collection.
- *LM* (Language Model) [38] approach build a probabilistic language model from each answer, and ranks answers based on the probability of the model generating the question.
- *LDA* [31] is a generative model that seeks to discover the global latent semantic embedding of question and answer.
- *DeepWalk* [12] learn the semantic embedding of CQA data merely utilizing the structure information from the user social network.
- *LDA+LM* Language model with LDA smooth feature.
- *Doc2Vec* [39] provide document-level low-dimension embedding for question and answer text.
- *GLM* [40] (Generalized Language Model) use the word similarity in the Skip-gram embedding space as a way to estimate term transformation probabilities in a language modeling setting for retrieval.
- *CNTN* [2] (Convolutional Neural Tensor Network) use CNN to build the embedding representation for each question and answer and then calculate their semantic matching score into a single model.
- *LDA + NWT* We use the low-dimension topic distribution as the representation of question and answer word.
- *Skip-gram + NWT* learn a relevance model by extracting features from Skip-gram embedding method in addition to corpus statistics such as inverse document frequency.
- *TWE + NWT* Topic Word Embedding [16] trains a topic model and word embedding on the same corpus, and then represents question and answer by concatenating average topic embedding and average word embedding.
- *MMSE-sim + NWT* In the previous, we demonstrate that the utilization of social relationship can mitigate the sparsity problem in CQA tasks. In order to verify this assumption, we evaluate a simplified version of MMSE without the user information.

Table 3
Evaluation results on Quora and Zhihu data.

Model	Objectives	Quora		Zhihu	
		nDCG	P@1	nDCG	P@1
BoW	–	0.714	0.487	0.683	0.473
BM25	–	0.738	0.541	0.721	0.485
LM	–	0.783	0.547	0.757	0.504
LDA	global	0.749	0.493	0.719	0.476
DeepWalk	user	0.666	0.458	0.639	0.446
LDA+LM	global	0.804	0.549	0.752	0.517
Doc2vec	local	0.761	0.521	0.723	0.494
GLM	local	0.821	0.579	0.773	0.532
CNTN	local	0.849	0.614	0.847	0.624
LDA+NWT	global	0.803	0.587	0.78	0.527
Skip-gram+NWT	local	0.839	0.601	0.817	0.568
TWE+NWT	local+global	0.852	0.613	0.825	0.588
MMSE-sim+NWT	local+global	0.871	0.635	0.837	0.623
MMSE+NWT	**local+global+user**	**0.902**	**0.679**	**0.862**	**0.673**

5.3. Parameter setting

We tune parameters for different models based on their performance on nDCG. In our experiment, the following three representative methods are selected to train the semantic embedding: Skip-gram (baseline), TWE and MMSE. We set the count of negative samples in 4; the context window size in 5; the learning rate is initialized as 0.03 and is set to decrease linearly so that it approached zero at the end of training [5]. LDA uses the specified 26 topics on Quora dataset, as this is the optimal topic number according to [41]. For Zhihu dataset, we set topic number in 30. For NWT, we tune the smoothing parameter μ in (100, 200,..., 1000) and the offset b in (0, 1, 2, 3) suggested by the original papers [34]. We empirically set the hyperparameter of MMSE $\mu_0 = 0, k_0 = 1e{-}5, \beta_0 = 1, \alpha_0 = 1e{-}3, T = 300, \alpha_0 = 0.3$.

5.4. Performance evaluations and comparisons

As mentioned previously, we argue that the utilization of both local and global semantic context for multi-modal data in CQA can effectively mitigate the lexical gap problem. To investigate the capability of comprehensive semantic modeling, we compare its performance with existing solely topic model based methods and word embedding based models. Besides, the social network information is crucial to alleviate the problem of data sparsity. Thus, the content information is utilized by all the method except DeepWalk, and the social information is introduced by DeepWalk and our proposed MMSE.

The question answering retrieval results of each method on Quora and Zhihu are reported in Table 3. The highest scores were highlighted with boldface. Based on these results, we have the following observations:

Table 4
Retrieval Results for "What should I know before **traveling** to **Sydney**?".

	Answer1: The estimated resident population for is 4,605,992, one-fifth of Australia's population. And it is famous for beach and sunshine. Hope you enjoy a great trip.	Answer2: In fact, in some cases, political centers set up separately from financial centers. Examples include Germany (Berlin vs. Frankfurt), US (Washington vs. New York), Australia (Canberra vs Sydney), China (Beijing vs. Shanghai).	Answer3: The popular view of small-town Wexford in Ireland is beaches, sunshine and opera. Wexford hosts the internationally recognized Opera Festival every October in The National Opera House.
LDA+NWT	Rank 3	Rank 7	Rank 1
Skip-gram+NWT	Rank 4	Rank 1	Rank 6
MMSE+NWT	Rank 1	Rank 5	Rank 7

1. For language model, semantic embedding based language model (local embedding view for GLM, global topical view for LDA+LM) outperform the original LM, BoW and BM25 methods and yield large improvements in most cases, which demonstrates that both local and global view based method can effectively address the word lexical gap problem.
2. Local view based method Doc2vec, GLM, Skip-gram+NWT obtain better performances than global view based method LDA, LDA + LM, LDA + NWT, respectively. These results indicate that the local semantic embedding is more effective at capturing semantic relations of words than that of global topic approach.
3. We also note that CNTN achieve comparable performance with the local semantic embedding, which indicates the effectiveness of the deep learning method for question answering with the help of large scale of dataset.
4. TWE+NWT and MMSE-sim+NWT outperform the LDA+NWT and Skip-gram+NWT, respectively, which tells us that semantic embedding with multi-view can obtain more comprehensive semantic relation between words.
5. DeepWalk performs worse than other models in most cases, which indicates that the content analysis plays a more important role than the simple utilization of social information in CQA tasks.
6. Since our method MMSE+NWT achieve much better performance than MMSE-sim+NWT and TWE+NWT, which shows that it is useful to model and exploit the multi-modal information to mitigate the sparsity problem in CQA tasks. We conclude that user social information can reinforce the effect of semantic embedding, and user similarity calculated in the vector space is very useful for identifying the semantic similarities between questions.

5.5. Impact of parameter values and case study

To get a better understanding how the multi-modal multi-view property benefit the question answering task, Table 4 gives part of the results of an example question "What should I know before traveling to Sydney?" The semantic embedding modeled by different methods for query words "traveling" and "Sydney" indicates different semantic similarity with words in answers. The two baseline can be easily misled by some distinctive words in single-view. In contrast, our proposed MMSE can give consideration to both local and global similarity-based words.

In our approach, there are two essential parameters, which are the dimension of semantic embedding and the balance weight c. Here we evaluate performance results on the dataset using semantic embedding trained by MMSE model. As shown in Table 5, by setting dimensions on 50, 100, 300, and 500 dimensions, the performance first increases and then slightly drops as the embedding dimension size increasing. As we know, different dimensionality of embedding provides different levels of granularity of semantic similarity, which may also require different amounts of training data.

Table 5
Performance comparison of MMSE over different dimensionality of word embedding.

Dataset	Dimension	nDCG	P@1
Quora	50	0.869	0.648
	100	0.883	0.662
	300	0.902	0.679
	500	0.889	0.654
Zhihu	50	0.831	0.635
	100	0.835	0.639
	300	0.862	0.673
	500	0.842	0.645

Generally, larger dimension size will give better quality and may need more data to prevent overfitting problem. Our results suggest that 300 dimensions is sufficient for learning semantic embedding on both datasets.

Besides, balance weight c used in MMSE also plays an important role in producing high-quality semantic embedding. Overemphasizing the weight of the original objective of p_{local} may result in the weakened influence of global topical context, while putting too large weight on p_{globle} may hurt the generality of learned local word embedding. Based on our experience, it is a better way to decode the objective balance weight based on the scale of their respective derivatives during optimization. Therefore, we carry out an experiment on the validation set to determine the best value among {1, 50, 100, 150, 200}. Fig. 5 shows the evolution of for the balance weight c on Quora data and Zhihu data. For all these plots, the two group methods are displayed. We can see that the best value achieved when setting the balance weight around 100.

5.6. Expert finding

In CQA, one of central issues is to find users with expertize and willingness to answer the given questions. Expert finding provides a platform to connect questions with experts who can contribute quality answers [42]. For example, CQA can help to find a mathematician for a chef with a math problem. At the same time, cooking tips from the chef will be returned to the mathematician if necessary. The goal of expert finding is to return a ranked list of experts with expertize on a given question. An essential part of an expert finding task is the ability to model the expertize of a user based on her answering history.

The textual contents of questions are fed into the corresponding CNN to calculate the latent representation with the leaned word embedding f^a, f^q [43]. Each question q_i is denoted by a d dimensional word vector. We then denote the collection of questions by $Q = \{q_1, \ldots, q_n\} \in R^{d\times n}$ where n is the total number of the questions. We denote the set of user embeddings by $U = \{u_1, u_2, \ldots, u_m\} \in R^{d\times m}$where u_i is the embedding vector for the latent expertize of the i-th user with the learned user embedding f^u.

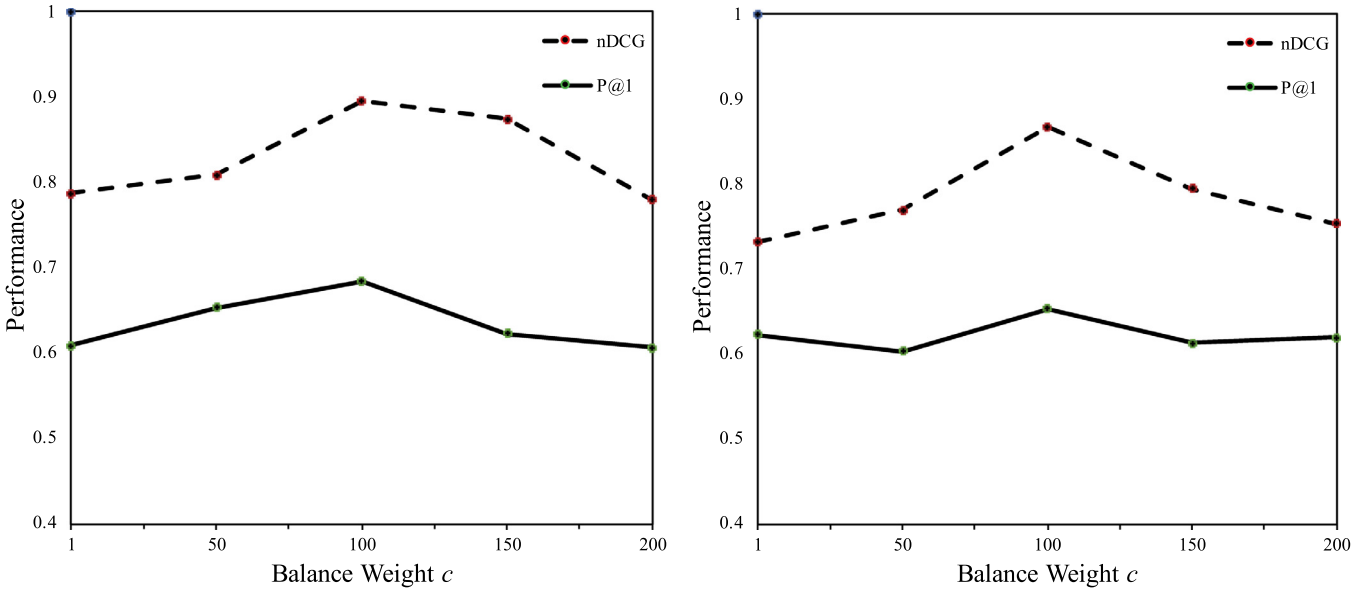


Fig. 5. Effect of varying the balance weight c.

For ground truth, we consider all the answerers for each question as the target user set, and their received thumbs-up/down as the ground truth rating scores. We use the relative quality rank to model the performance of users for answering the questions, which is in the form of triplet constraints. Unlike the previous studies, our method MMSE learns the user embedding and word embedding from the proposed CQA network. We denote a triplet constraint by the ordered tuple (j, i, k), meaning that "the i-th user obtains more votes than the k-th user for answering the j-th question". Let $Tri = \{(j, i, k)\}$ denote the set of triplet constraints obtained from the community votes for a set of m users answering n different questions. More formally, we aim to learn the ranking metric function that for any $(j, i, k) \in Tri$, the inequality holds: $f_{u_j}(q_j) > f_{u_k}(q_j) \iff q_i^T u_j > q_i^T u_k$. We compare our proposed method with other popular expert nding algorithms in CQA systems as follows:

- *ExpertsRank* algorithm [44]: uses question ask-answer relation to construct the graph of users, and then finds the experts with link structure analysis based on PageRank algorithm.
- *AuthorityRank* algorithm [23]: computes user authority based on the number of provided best answers, which is an in-degree method.
- *DRM* method [45] is a topic-sensitive probabilistic model, which learns question representation via PLSA-based model.
- *DeepWalk* algorithm [12]: learns the embedding of both questions and users based on the network structure.
- *TSPM* algorithm [46]: The TSPM algorithm is a topic sensitive probabilistic method for expert nding in CQA systems, which ts a LDA-based probabilistic model to the question-answering activities.

Experimental results. We summarize our results for expert finding in Tables 6 and 7. The evaluation were conducted with 60%, 70% and 80% of data for training. The DeepWalk, AuthorityRank and ExpertsRank methods are based on link analysis of users, which only consider the network structure. While DRM and TSPM methods are based on probabilistic model with question content. These experiments reveal several key points:

- The topic-oriented methods, both TSPM and DRM, outperform the AuthorityRank and ExpertsRank methods, which suggests the effectiveness of the latent user model for the problem of expert finding.
- The supervised methods, AuthorityRank, TSPM, DRM and ExpertsRank outperform the unsupervised DeepWalk method, which suggests that the supervised information such as users' relative quality rank and question contents are critical for the problem.
- In all the cases, our MMSE method achieves the best performance. This fact shows that the ranking metric network learning framework that exploits both deep representation of question contents and users relative quality rank can further improve the performance of expert finding.

Table 6
Experimental results on nDCG with different proportions of Quora data for training.

Methods	nDCG		
	60%	70%	80%
ExpertsRank	0.6702	0.6926	0.6948
AuthorityRank	0.6723	0.7014	0.7108
DRM	0.6585	0.6646	0.6707
DeepWalk	0.6238	0.6296	0.6381
TSPM	0.6316	0.645	0.6627
MMSE	**0.6937**	**0.7428**	**0.7582**

Table 7
Experimental results on Precision@1 with different proportions of Quora data for training.

Methods	P@1		
	60%	70%	80%
ExpertsRank	0.4223	0.4637	0.4859
AuthorityRank	0.4441	0.4785	0.4904
DRM	0.4183	0.4242	0.4311
DeepWalk	0.3659	0.3717	0.3815
TSPM	0.3781	0.3944	0.4177
MMSE	**0.4827**	**0.5348**	**0.5527**

6. Conclusions

In this paper, we propose a novel multi-modal multi-view semantic embedding learning method for community question answer analysis. The proposed MMSE can simultaneously model multi-modal CQA data as well as their corresponding semantic information from both local and global view in a unified and principled way. Experiments conducted on question answering task for both English and Chinese CQA datasets demonstrate the effectiveness of our approaches. In the future, we will evaluate our model on other CQA task and investigate more applications, such as the answers quality evaluation.

Acknowledgments

This work has been supported by the National Key Research and Development Program of China under Grant No.2016YFB1000901, the Innovative Research Team in University (PCSIRT) of the Ministry of Education under grant IRT17R32.

Appendix A. Full Derivation of Model Inference

In Bayesian probability theory, if the posterior distributions $p(\theta|x)$ are in the same probability distribution family as the prior probability distribution $p(\theta)$, the prior and posterior are then called conjugate distributions. There are two types of conjugate distributions in the proposed methods MMSE. (1) We characterize each dimension of the embedding $\{f^u, f^a, f^q\}$ as a univariate Gaussian distribution $N(\mu, \lambda)$ with Normal-Gamma distribution priors $NormalGamma(\tau = \{\mu_0, k_0, \alpha_0, \beta_0\})$ [27]. (2) Each topic of question and answer $\{z_q, z_a\}$ are draw from Multinomial distribution $Multi(\theta)$ with Dirichlet priors $Dir(\alpha)$ [27].

$$\begin{aligned}&\alpha \xrightarrow[\text{Dirichlet}]{} \beta \xrightarrow[\text{Multinomial}]{} z,\\ &\tau \xrightarrow[\text{Normal Gamma}]{} \{\mu, \lambda\} \xrightarrow[\text{Gaussian}]{} f.\end{aligned} \tag{A.1}$$

A1. The joint distribution

Looking at the topology of the Bayesian network, we can specify the complete-data likelihood of a document(Q/A pair with user), i.e., the joint distribution of all known and hidden variables $\{\theta, \mu^u, \lambda^u, \mu^q, \lambda^q, \mu^a, \lambda^a, z^a, z^q, y, f^u, f^q, f^a\}$ given the hyperparameters $\{\alpha, \tau^u, \tau^q, \tau^a\}$:

$$\begin{aligned}&p(\theta, \mu^u, \lambda^u, \mu^q, \lambda^q, \mu^a, \lambda^a, z^a, z^q, y, f^u, f^q, f^a; \alpha, \tau^u, \tau^q, \tau^a)\\ &= \underbrace{p(\theta|\alpha)}_{①}\underbrace{p(z_q|\theta)p(z_a|\theta)}_{②}\underbrace{p(y|z_q)}_{③}\\ &\quad\underbrace{p(\mu^u, \lambda^u|\tau^u)p(\mu^q, \lambda^q|\tau^q)p(\mu^a, \lambda^a|\tau^a)}_{④}\\ &\quad\underbrace{\cdot p(f^u|y, \mu^u, \lambda^u)p(f^q|z_q, \mu^q, \lambda^q)p(f^a|z_a, \mu^a, \lambda^a)}_{⑤}.\end{aligned} \tag{A.2}$$

For each part of the joint distribution:

1. θ_d draw from Dirichlet distribution with hyperparameter α.

$$p(\theta_d|\alpha) = \frac{1}{\Delta(\alpha)}\prod_{t=1}^{T}\theta_{dt}^{\alpha_t - 1}, \tag{A.3}$$

where subscript d denote each Q/A pair as a document, t denote topic.

2. The parameter of Gaussian distribution μ and λ are generated by normal Gamma distribution with hyperparameter $\tau = \{\mu_0, k_0, \alpha_0, \beta_0\}$. According to the definition of normal Gamma distribution, we derive distribution of μ and λ as :

$$\begin{aligned}p(\mu^u_{te}, \lambda^u_{te}; \tau^u_e &\overset{def}{=} \{\mu_0, k_0, \alpha_0, \beta_0\})\\ &= \frac{\beta_0^{\alpha_0}\sqrt{\lambda_0}}{\Gamma(\alpha_0)\sqrt{2\pi}}\lambda_{te}^{u\,\alpha_0 - 1/2}e^{\beta_0\lambda^u_{te}}e^{-\frac{\lambda_0\lambda^u_{te}(\mu^u_{te}-\mu_0)^2}{2}},\\ p(\mu^q_{te}, \lambda^q_{te}; \tau^q_e &= \{\mu_0, k_0, \alpha_0, \beta_0\})\\ &= \frac{\beta_0^{\alpha_0}\sqrt{\lambda_0}}{\Gamma(\alpha_0)\sqrt{2\pi}}\lambda_{te}^{q\,\alpha_0 - 1/2}e^{\beta_0\lambda^q_{te}}e^{-\frac{\lambda_0\lambda^q_{te}(\mu^q_{te}-\mu_0)^2}{2}},\\ p(\mu^a_{te}, \lambda^a_{te}; \tau^a_e &= \{\mu_0, k_0, \alpha_0, \beta_0\})\\ &= \frac{\beta_0^{\alpha_0}\sqrt{\lambda_0}}{\Gamma(\alpha_0)\sqrt{2\pi}}\lambda_{te}^{a\,\alpha_0 - 1/2}e^{\beta_0\lambda^u_{te}}e^{-\frac{\lambda_0\lambda^a_{te}(\mu^a_{te}-\mu_0)^2}{2}}.\end{aligned} \tag{A.4}$$

where the subscript t denote topic, e denote one dimension of embedding representation.

3. the topic are generated by multinomial distribution:

$$\begin{aligned}&p(z^q_{dm}|\theta_d) = \theta_{dz_{dm}},\\ &p(z^a_{dm}|\theta_d) = \theta_{dz_{dh}}.\end{aligned} \tag{A.5}$$

where d denote each Q/A pair, m denote an answer word embedding representation and h denote an question word embedding representation

4. User embedding's topic y is generated from the topic of answer z^a_{dm} with a uniform distribution. Because each topic may occure many times, the probability of user embedding's topic is proportional to the number of occurrences

$$p(y_d|z^a_d) = \frac{\sum_{m=1}^{M_d}\mathbb{I}(z_{dm} = y_d)}{M_d}. \tag{A.6}$$

where d denote each Q/A pair, and m denote an answer word embedding representation. In practice, however, the topic of a user embedding may not occurred in the topic of answer. Thus a correction is need apply to this probabilities calculations to ensure that none of the probabilities is 0. This correction, known as Laplace smoothing, is given by the following:

$$p(y_d|z^a_d) = \frac{\sum_{m=1}^{M_d}\mathbb{I}(z_{dm} = y_d) + l}{M_d + Tl}. \tag{A.7}$$

where the Laplace smoothing parameter $l \in (0, 1)$

5. Each dimension of user embedding f^u is draw from a univariate Gaussian distribution:

$$p(f^u_{dm}|y, \mu^u, \lambda^u) = \frac{1}{\sqrt{2\pi}}\sqrt{\lambda^u}e^{-\frac{\lambda^u}{2}(f^u_{dm}-\mu^u)^2}. \tag{A.8}$$

Each dimension of question embedding f^q is draw from a univariate Gaussian distribution:

$$p(f^q_{dh}|z_{dh}, \mu^q, \lambda^q) = \frac{1}{\sqrt{2\pi}}\sqrt{\lambda^q}e^{-\frac{\lambda^q}{2}(f^q_{dh}-\mu^q)^2}. \tag{A.9}$$

Each dimension of answer embedding f^a is draw from a univariate Gaussian distribution:

$$p(f^a_{dm}|z_{dm}, \mu^a, \lambda^a) = \frac{1}{\sqrt{2\pi}}\sqrt{\lambda^a}e^{-\frac{\lambda^a}{2}(f^a_{dh}-\mu^a)^2}. \tag{A.10}$$

A2. Compute the integral for parameters

After we get the the joint distribution of all known and hidden variables, we calculate the joint distribution of hyperparameters by integral for model parameters θ, μ^u, λ^u, μ^q, λ^q, μ^a, λ^a.

1. Integral for parameter θ

$$\int p(\theta; \alpha)p(z|\theta)d\theta = \prod_{d=1}^{D}\frac{\Delta(n_q + n_a + \alpha)}{\Delta(\alpha)}. \tag{A.11}$$

where n_a, n_q is a vector with length of T. Each dimension of the vector n_a^t, n_q^t denote the number of words allocated with topic t in answer and question.α is the hyperparameter of Dirichlet distribution, and is a vector with length of T

2. Integral for parameter μ^u, λ^u

$$\int p(\mu^u, \lambda^u; \tau^u) p(f^u|\mu^u, \lambda^u, y) d\mu^u d\lambda^u = \prod_{t=1}^{T} \prod_{e=1}^{E^u} G(f^u, y, t, e, \tau^u). \tag{A.12}$$

where $G(\cdot)$ is defined as:

$$G(f, y, t, e, \tau = \mu_0, k_0, \alpha_0, \beta_0) = \frac{\Gamma(\alpha_n)}{\Gamma(\alpha_0)} \frac{\beta_0^{\alpha_0}}{\beta_n^{\alpha_n}} . (\frac{k_0}{k_n})^{\frac{1}{2}} (2\pi)^{-n/2}. \tag{A.13}$$

where n is the number of embedding f with $y = t$. If x is the vector of e dimension of all the embeddings with $y = t$, thus:

$$\begin{aligned} a_n &= a_0 + n/2, \\ k_n &= k_0 + n, \\ \mu_n &= \frac{k_0\mu_0 + n\bar{x}}{k_0 + n}, \\ \beta_n &= \beta_0 + \frac{1}{2}\sum_{i=1}^{n_t}(x_i - \bar{x})^2 + \frac{k_0 n(\bar{x} - \mu_0)^2}{2(k_0 + n)}. \end{aligned} \tag{A.14}$$

where $\bar{x}$ is the mean of all the element in x.
Likewise, we can get the ntegral for parameter $\{\mu^q, \lambda^q\}$ and $\{\mu^a, \lambda^a\}$:

$$\begin{aligned} &\int p(\mu^q, \lambda^q; \tau^q) p(f^q|\mu^q, \lambda^q, z_q) d\mu^q d\lambda^q \\ &= \prod_{t=1}^{T} \prod_{e=1}^{E^w} G(f^q, z_q, t, e, \tau^q), \\ &\int p(\mu^a, \lambda^a; \tau^a) p(f^a|\mu^a, \lambda^a, z_a) d\mu^a d\lambda^a \\ &= \prod_{t=1}^{T} \prod_{e=1}^{E^w} G(f^a, z_a, t, e, \tau^u). \end{aligned} \tag{A.15}$$

3. In conclusion, we can get the joint distribution for hyperparameters:

$$\begin{aligned} &p(y, z_a, z_q, f^u, f^q, f^a; \alpha, \tau^u, \tau^q, \tau^a) \\ &= \prod_{d=1}^{D} \frac{\Delta(n_q + n_a + \alpha)}{\Delta(\alpha)} \prod_{t=1}^{T} \prod_{e=1}^{E^u} G(f^u, y, t, e, \tau^u) \\ &\times \prod_{t=1}^{T} \prod_{e=1}^{E^w} G(f^q, z_q, t, e, \tau^q) \prod_{t=1}^{T} \prod_{e=1}^{E^w} G(f^a, z_a, t, e, \tau^u). \end{aligned} \tag{A.16}$$

where E^u and E^w are the dimension of user and word embedding respectly.

A3. Inference via Gibbs sampling

Exact inference for topic model is generally intractable. The solution to this is to use approximate inference algorithms, such as mean-eld variational expectation maximization [BNJ02], expectation propagation [MiLa02], and Gibbs sampling.

Gibbs sampling is a special case of Markov-chain Monte Carlo (MCMC) simulation [MacK03,Liu01] and often yields relatively simple algorithms for approximate inference in high-dimensional models such as LDA. MCMC methods can emulate high-dimensional probability distributions $p(\vec{x})$ by the stationary behavior of a Markov chain. This means that one sample is generated for each transition in the chain after a stationary state of the chain has been reached, which happens after a so-called burn-in period that eliminates the inuence of initialization parameters. Gibbs sampling is a special case of MCMC where the dimensions x_i of the distribution are sampled alternately one at a time, conditioned on the values of all other dimensions, which we denote $\vec{x}_{\neg i}$. The algorithm works as follows: (1) choose dimension i (random or by permutation). (2) sample x_i from $p(x_i|\vec{x}_{\neg i})$.

1. For each Q/A pair document d, the conditional distribution of user embedding y_d is:

$$p(y_d|y_{-d}, z, f^u, f^q, f^a) \propto (n_d^t + l) \prod_{e=1}^{E^u} G'(f^u, y, t, e, \tau^u, d). \tag{A.17}$$

where n_d^t is the number of topic t appeared in document d. $G'(\cdot)$ is defined as:

$$G(f, y, t, e, \tau, d) = \frac{\Gamma(\alpha_n)}{\Gamma(\alpha_{n'})} \frac{\beta_{n'}^{\alpha_{n'}}}{\beta_n^{\alpha_n}} \left(\frac{k_{n'}}{k_n}\right)^{\frac{1}{2}} \frac{(2\pi)^{-n/2}}{(2\pi)^{-n'/2}}. \tag{A.18}$$

where n denotes the times of ith dimension in vector embedding assigned to topic t; x denotes the concatenated vector of the e-th dimension of f_i with $y_i = t$; $n' = n - n_d$, n_d denotes the number of f which satisfies $y = t$.

2. sample x_i from $p(x_i|\vec{x}_{\neg i})$

$$\begin{aligned} p(z_{dm}^a = t|z_{-dm}^a, z^q, y, f^u, f^q, f^a) &\propto (n_{dm}^{y_d} + l)(n_{dm}^t + \alpha_t) \prod_{e=1}^{E^w} G'(f^a, z^a, t, e, \tau^a, dm), \\ p(z_{dh}^q = t|z_{-dh}^q, z^a, y, f^u, f^q, f^a) &\propto (n_{dh}^t + \alpha_t) \prod_{e=1}^{E^w} G'(f^q, z^q, t, e, \tau^q, dh). \end{aligned} \tag{A.19}$$

With the conditional distributions above, we can construct a Markov chain to learn our model. In training, we finish the burn-in stage in T iterations based on the Markov chain.

A4. Parameter estimation

After training, we need to obtain the multinomial parameter sets θ that correspond to the state of the Markov chain z, and Gaussian parameter set μ, λ that correspond to the embdddding f.

$$\textit{Dirichlet prior} + \textit{Data from multinomial} \rightarrow \textit{posterior distribution is Dirichlet distribution} \tag{A.20}$$

1. According to the definitions of multinomial distributions with Dirichlet prior, applying Bayes' rule on the component $z = k$, we can get the posterior distribution for parameter θ.

$$\begin{aligned} p(\theta_d|\alpha) &= Dir(\theta_d|\alpha), \\ p(\theta_d|z, \alpha) &= \frac{1}{Z_\theta} \prod_{d=1}^{D} p(z|\theta_d) \cdot p(\theta|\alpha) = Dir(\theta_d|n_a + n_q + \alpha). \end{aligned} \tag{A.21}$$

where n_a, n_q is the vector of topic observation counts for Q/A document respectively. After we get the posterior distribution, the most common way to estimate the parameter is the maximum the posterior estimation(MAP). The MAP can be used to obtain a point estimate of an unobserved quantity on the basis of empirical data. But in this paper, we use the the expectation of the Dirichlet distribution as the estimate of the unknown parameter, $< Dir(X) >= \alpha_i / \sum_i \alpha_i$, on these results yields:

$$\theta_d^t = \frac{n_a^t + n_q^t + \alpha_t}{\sum_{t=1}^{t}(n_a^t + n_q^t + \alpha_t)}. \tag{A.22}$$

2. In a similar way, we derive posterior distribution for Gaussian parameter μ, λ:

$$\begin{aligned}
\text{Normal-Gamma prior} &: p(\mu,\lambda|\tau) = NG(\mu,\lambda|\tau=\mu_0,k_0,\alpha_0,\beta_0),\\
\text{Normal-Gamma posterior} &: p(\mu,\lambda|f,\tau) = NG(\mu,\lambda|\tau=\mu_n,k_n,\alpha_n,\beta_n),\\
\alpha_n &= \alpha_0 + n/2,\\
k_n &= k_0 + n,\\
\mu_n &= \frac{k_0\mu_0 + n\bar{x}}{k_0+n},\\
\beta_n &= \beta_0 + \frac{1}{2}\sum_{i=1}^{n_t}(x_i-\bar{x})^2\\
&\quad + \frac{k_0 n(\bar{x}-\mu_0)^2}{2(k_0+n)}. \qquad \text{(A.23)}
\end{aligned}$$

We see that the posterior sum of squares, β_n, combines the prior sum of squares, β_0, the sample sum of squares, $\sum_i(x_i-\bar{x})^2$, and a term due to the discrepancy between the prior mean and sample mean. We use the the expectation Normal-Gamma posterior to estimate the parameter μ, λ. When we get Normal-Gamma posterior $p(\mu, \lambda|f, \tau)$, the expectation of parameters are $E(\mu)=\mu_n, E(\lambda)=\alpha_n/\beta_n^{-1}$

$$\begin{aligned}
\mu_t &= \mu_n = \frac{k_0\mu_0 + n\bar{x}}{k_0+n},\\
\lambda_t &= \alpha_n/\beta_n^{-1} = \frac{\alpha_0 + n/2}{\beta_0 + \frac{1}{2}\sum_i(x_i-\bar{x})^2 + \frac{k_0 n(\bar{x}-\mu_0)^2}{2(k_0+n)}}. \qquad \text{(A.24)}
\end{aligned}$$

where the n is the number of embedding with $z=t$.

A5. Embedding representation updating

We update the embedding during inference to fine-tuning pre-trained word embedding and user embedding for a task-specific representation with both global and local semantic. We define the objective functions as the log-likelihood of embedding representation given certain hidden parameters.

For simplicity, we first consider about one dimension of user embedding in document f_{de} as variables x. Our sample is made up of the first n terms of an IID sequence $\{X_n\}$ of Gaussian random variables having mean μ and variance λ. The probability density function of a generic term of the sequence is:

$$G_X(x_i) = \frac{1}{\sqrt{2\pi}}\sqrt{\lambda}e^{-\frac{\lambda}{2}(x_i-\mu)^2}. \qquad \text{(A.25)}$$

The mean μ and the variance λ are the two parameters that need to be estimated.

1. Given the assumption that the observations from the sample are IID, the likelihood function can be written as:

$$\begin{aligned}
L(\mu,\lambda;x_1,x,\ldots,x_n) &= \prod_{i=1}^{n} G_X(x_i;\mu,\lambda)\\
&= \prod_{i=1}^{n}\frac{1}{\sqrt{2\pi}}\sqrt{\lambda}e^{-\frac{\lambda}{2}(x_i-\mu)^2}\\
&= \left(\frac{\lambda}{2\pi}\right)^{n/2} e^{-\frac{\lambda}{2}\sum_{i=1}^{n}(x_i-\mu)^2}. \qquad \text{(A.26)}
\end{aligned}$$

2. By taking the natural logarithm of the likelihood function, we get the log-likelihood function:

$$\begin{aligned}
l(\mu,\lambda,;x_1,x,\ldots,x_n) &= \ln(L(\mu,\lambda;x_1,x,\ldots,x_n))\\
&= \ln\left(\left(\frac{\lambda}{2\pi}\right)^{n/2} e^{-\frac{\lambda}{2}\sum_{i=1}^{n}(x_i-\mu)^2}\right)\\
&= \ln\left(\left(\frac{\lambda}{2\pi}\right)^{n/2}\right) + \ln\left(e^{-\frac{\lambda}{2}\sum_{i=1}^{n}(x_i-\mu)^2}\right)\\
&= -\frac{n}{2}\ln\left(\frac{2\pi}{\lambda}\right) + \sum_{i=1}^{n}\left(-\frac{\lambda}{2}\right)(x_i-\mu)^2\\
&= -\frac{n}{2}\ln(2\pi) + \frac{n}{2}\ln(\lambda)\\
&\quad + \sum_{i=1}^{n}\left(-\frac{\lambda}{2}\right)(x_i-\mu)^2. \qquad \text{(A.27)}
\end{aligned}$$

3. The first two terms of log-likelihood function for single embedding dimension are constant number. After omitting the constant term, log-likelihood function can be writtern as:

$$l(\mu,\lambda,;x_1,x,\ldots,x_n) = \sum_{i=1}^{n}\left(-\frac{\lambda}{2}\right)(x_i-\mu)^2. \qquad \text{(A.28)}$$

Let W be the number of word embedding. We write the log likelihood of the all the data given the model parameters as:

$$\begin{aligned}
L &= \sum_{s=1}^{S}\sum_{t=1}^{T}\sum_{e=1}^{E^u}\left(-\frac{\lambda_{te}^u}{2}\right)(f_{se}^u-\mu_{te}^u)^2\\
&\quad + \sum_{w=1}^{W}\sum_{t=1}^{T} n_w^t \sum_{e=1}^{E^w}\left(-\frac{\lambda_{te}^q}{2}\right)(f_{de}^q-\mu_{te}^q)^2\\
&\quad + \sum_{w=1}^{W}\sum_{t=1}^{T} n_w^t \sum_{e=1}^{E^w}\left(-\frac{\lambda_{te}^a}{2}\right)(f_{de}^a-\mu_{te}^a)^2. \qquad \text{(A.29)}
\end{aligned}$$

where n_w^t is the number of embedding with topic t.

4. We employ gradient ascent to maximize the log likelihood by updating the embeddings f^u, f^a and f^q. The gradients are computed as

$$\begin{aligned}
\frac{\partial L}{\partial f_{se}^u} &= \sum_{t=1}^{T} -\lambda_{te}^u(f_{se}^u-\mu_{te}^u),\\
\frac{\partial L}{\partial f_{we}^q} &= \sum_{t=1}^{T} n_w^t(-\lambda_{te}^q)(f_{te}^q-\mu_{te}^q),\\
\frac{\partial L}{\partial f_{we}^a} &= \sum_{t=1}^{T} n_w^t(-\lambda_{te}^a)(f_{te}^a-\mu_{te}^a). \qquad \text{(A.30)}
\end{aligned}$$

Appendix B. Normal-gamma prior

The Gaussian or normal distribution is one of the most widely used in statistics. Estimating its parameters using Bayesian inference and conjugate priors is also widely used. The use of conjugate priors allows all the results to be derived in closed form. In Bayesian probability theory, if the posterior distributions $p(\theta|x)$ are in the same probability distribution family as the prior probability distribution $p(\theta)$, the prior and posterior are then called conjugate distributions, and the prior is called a conjugate prior for the likelihood function. For example, the Gaussian family is conjugate to itself (or self-conjugate) with respect to a Gaussian likelihood function: if the likelihood function is Gaussian, choosing a Gaussian prior over the mean will ensure that the posterior distribution is also Gaussian. This means that the Gaussian distribution is a conjugate prior for the likelihood that is also Gaussian. There are total three types of conjugate prior for normal distribution for different parameters.

Likelihood	Model parameters	Conjugate prior distribution	Prior hyperparameters
Normal with known precision λ	μ (mean)	Normal	μ_0, λ_0
Normal with known mean μ	λ (precision)	Gamma	α, β
Normal	μ and λ	**Normal-gamma**	$\mu_0, k_0, \alpha_0, \beta_0$

Normal-gamma

We will now suppose that both the mean μ and the precision $\lambda = \sigma^{-2}$ are unknown.

1. **Likelihood**
Let $D = (x_1, \ldots, x_n)$ be the data, which means each dimension of embedding representation f^u, f^q, f^q in MMSE. The likelihood of Normal distribution can be written in this form:

$$p(D|\mu, \lambda) = \frac{1}{(2\pi)^{n/2}} \lambda^{n/2} \exp\Big(-\frac{\lambda}{2}(x_i - \mu)^2 \Big). \tag{B.1}$$

2. **Prior**
The conjugate prior is the **normal-gamma**:

$$\begin{aligned} NG(\mu, \lambda|\mu_0, k_0, \alpha_0, \beta_0) &\stackrel{def}{=} \mathcal{N}(\mu|\mu_0, (k_0\lambda)^{-1}) Ga(\lambda|\alpha_0, rate = \beta_0) \\ &= \frac{1}{Z_{NG}(\mu_0, k_0, \alpha_0, \beta_0)} \lambda^{1/2} \exp \\ &\quad \times \Big(-\frac{k_0\lambda}{2}(\mu - \mu_0)^2 \Big) \lambda^{\alpha_0 - 1} e^{-\lambda\beta_0} \\ &= \frac{1}{Z_{NG}} \lambda^{\alpha_0 - 1/2} \exp\Big(-\frac{\lambda}{2}[k_0(\mu - \mu_0)^2 + 2\beta_0] \Big). \\ Z_{NG}(\mu_0, k_0, \alpha_0, \beta_0) &= \frac{\Gamma(\alpha_0)}{\beta_0^{\alpha_0}} \Big(\frac{2\pi}{k_0} \Big)^{\frac{1}{2}}. \end{aligned} \tag{B.2}$$

3. **Posterior**
The posterior can be derived as follows.

$$\begin{aligned} &p(\mu, \lambda|D, \mu_0, k_0, \alpha_0, \beta_0) \\ &\quad \propto NG(\mu, \lambda|\mu_0, k_0, \alpha_0, \beta_0) p(D|\mu, \lambda) \\ &\quad \propto \lambda^{\frac{1}{2}} e^{-(k_0\lambda(\mu - \mu_0)^2)/2} \lambda^{\alpha_0 - 1} e^{-\beta_0\lambda} \times \lambda^{n/2} e^{-\frac{\lambda}{2}\sum_{i=1}^n (x_i - \mu)^2} \\ &\quad \propto \lambda^{\frac{1}{2}} \lambda^{\alpha_0 + n/2 - 1} e^{-\beta_0\lambda} e^{-(\lambda/2)[k_0(\mu - \mu_0)^2 + \sum_i (x_i - \mu)^2]}. \end{aligned} \tag{B.3}$$

We can rewrite the term $\sum_i (x_i - \mu)^2$ in the exponent as flollows:

$$\begin{aligned} \sum_i (x_i - \mu)^2 &= \sum_i [(x_i - \bar{x}) - (\mu - \bar{x})]^2 x \\ &= \sum_i (x_i - \bar{x})^2 + \sum_i (\bar{x} - \mu)^2 - \sum_i (x_i - \bar{x})(\mu - \bar{x}) \\ &= ns^2 + n(\bar{x} - \mu)^2. \end{aligned} \tag{B.4}$$

Since the empirical mean and variance are defined:

$$\begin{aligned} \bar{x} &= \frac{1}{n}\sum_{i=1}^n x_i, \\ s^2 &= \frac{1}{2}\sum_{i=1}^n (x_i - \bar{x})^2. \end{aligned} \tag{B.5}$$

And

$$\begin{aligned} \sum_i (x - \bar{x})(\mu - \bar{x}) &= (\mu - \bar{x}) \left(\left(\sum_i x_i \right) - n\bar{x} \right) \\ &= (\mu - \bar{x})(n\bar{x} - n\bar{x}) = 0. \end{aligned} \tag{B.6}$$

Also, it can be shown that

$$k_0(\mu - \mu_0)^2 + n(\mu - \bar{x})^2 = (k_0 + n)(\mu - \mu_n)^2 + \frac{k_0 n(\bar{x} - \mu_0)}{k_0 + n}. \tag{B.7}$$

where

$$\mu_n = \frac{k_0\mu_0 + n\bar{x}}{k_0 + n}. \tag{B.8}$$

Hence

$$\begin{aligned} k_0(\mu - \mu_0)^2 + \sum_i (x_i - \mu)^2 &= k_0(\mu - \mu_0)^2 + n(\mu - \bar{x})^2 + \sum_i (x_i - \bar{x})^2 \\ &= (k_0 + n)(\mu - \mu_n)^2 + \frac{k_0 n(\bar{x} - \mu)^2}{k_0 + n} \\ &\quad + \sum_i (x_i - \bar{x})^2. \end{aligned} \tag{B.9}$$

So

$$\begin{aligned} p(\mu, \lambda) &\propto \lambda^{\frac{1}{2}} e^{-(\lambda/2)(k_0 + n)(\mu - \mu_n)^2} \\ &\quad \times \lambda^{\alpha_0 + n/2 - 1} e^{-\lambda_0\lambda} e^{-(\lambda/2)\sum_i (x_i - \bar{X})^2} e^{-(\lambda/2)\frac{k_0 n(\bar{X} - \mu_0)^2}{k_0 + n}} \\ &\propto \mathcal{N}(\mu|\mu_n, ((k + n)\lambda)^{-1}) \times Ga(\lambda|\alpha_0 + n/2, \beta_n). \end{aligned} \tag{B.10}$$

In summary,

$$\begin{aligned} p(\mu, \lambda|D, \mu_0, k_0, \alpha_0, \beta_0) &= NG(\mu, \lambda|\mu_n, k_n, \alpha_n, \beta_n), \\ \mu_n &= \frac{k_0\mu_0 + n\bar{x}}{k_0 + n}, \\ \alpha_n &= \alpha_0 + n/2, \\ k_n &= k_0 + n, \\ \beta_n &= \beta_0 + \frac{1}{2}\sum_{i=1}^{n_t} (x_i - \bar{x})^2 \\ &\quad + \frac{k_0 n(\bar{x} - \mu_0)^2}{2(k_0 + n)}. \end{aligned} \tag{B.11}$$

We see that the posterior sum of squares, β_n, combines the prior sum of squares, β_0, the sample sum of squares, $\sum_i (x_i - \bar{x})^2$, and a term due to the discrepancy between the prior mean and sample mean

4. Marginal likelihood
To derive the marginal likelihood, we just dererive the posterior, but this time we keep track of all the constant factors. Let $NG'(\mu, \lambda|\mu_0, k_0, \alpha_0, \beta_0)$ denote an unnormalized Normal-Gamma distribution, and let $Z_0 = Z_{NG}(\mu_0, k_0, \alpha_0, \beta_0)$ be the normalization constant of the prior; similarly let Z_n be the normalization constant of the posterior. Let $N'(x_i|\mu, \lambda)$denote an unnormalized Gaussian with normalization constant $\frac{1}{\sqrt{2\pi}}$. Then

$$p(\mu, \lambda|D) = \frac{1}{p(D)} \frac{1}{Z_0} NG'(\mu, \lambda|\mu_0, k_0, \alpha_0, \beta_0) \Big(\frac{1}{2\pi} \Big)^{n/2} \prod_i N'(x_i|\mu, \lambda). \tag{B.12}$$

The $NG'$and N' terms combine to make the posterior NG':

$$p(\mu, \lambda|D) = \frac{1}{Z_n} NG'(\mu, \lambda|\mu_n, k_n, \alpha_n, \beta_n). \tag{B.13}$$

Hence

$$\begin{aligned} P(D) &= \frac{Z_n}{Z_0} (2\pi)^{-n/2} \\ &= \frac{\Gamma(\alpha_n)}{\Gamma(\alpha_0)} \frac{\beta_0^{\alpha_0}}{\beta_n^{\alpha_n}} \Big(\frac{k_0}{k_n} \Big)^{\frac{1}{2}} (2\pi)^{-n/2}. \end{aligned} \tag{B.14}$$

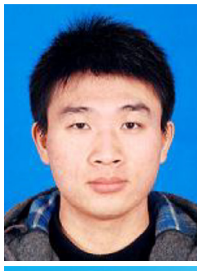
References

[1] K. Zhang, W. Wu, F. Wang, M. Zhou, Z. Li, Learning distributed representations of data in community question answering for question retrieval, in: Proceedings of Conference on Web Search and Data Mining, WSDM, 2016.
[2] X. Qiu, X. Huang, Convolutional neural tensor network architecture for community-based question answering., in: Proceedings of International Joint Conferences on Artifical Intelligence, IJCAI, 2015.
[3] Z. Zhao, Q. Yang, D. Cai, X. He, Y. Zhuang, Expert finding for community-based question answering via ranking metric network learning., in: Proceedings of International Joint Conferences on Artifical Intelligence, IJCAI, 2016.
[4] T. Sagara, M. Hagiwara, Natural language neural network and its application to question-answering system, Neurocomputing 142 (2014) 201–208.
[5] H. Fang, F. Wu, Z. Zhao, X. Duan, Y. Zhuang, M. Ester, Community-based question answering via heterogeneous social network learning, in: Proceedings of AAAI, 2016.
[6] M. Rosen-Zvi, T. Griffiths, M. Steyvers, P. Smyth, The author-topic model for authors and documents, in: Proceedings of Conference on Uncertainty in Artificial Intelligence, UAI, 2004.
[7] D. Mimno, A. McCallum, Topic models conditioned on arbitrary features with dirichlet-multinomial regression, in: Proceedings of Conference on Uncertainty in Artificial Intelligence, UAI, 2008.
[8] G. Xun, Y. Li, J. Gao, A. Zhang, Collaboratively improving topic discovery and word embeddings by coordinating global and local contexts, in: Proceedings of Conference on Knowledge Discovery and Data Mining, KDD, 2017.

[9] P. Qin, W. Xu, J. Guo, An empirical convolutional neural network approach for semantic relation classification, Neurocomputing 190 (2016) 1–9.
[10] P. Wang, B. Xu, J. Xu, G. Tian, C.-L. Liu, H. Hao, Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification, Neurocomputing 174 (2016) 806–814.
[11] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Proceedings of Neural Information Processing Systems, NIPS, 2013.
[12] B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: online learning of social representations, in: Proceedings of Conference on Knowledge Discovery and Data Mining, KDD, 2014.
[13] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, Q. Mei, Line: large-scale information network embedding, in: Proceedings of the 24th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2015, pp. 1067–1077.
[14] S. Chang, W. Han, J. Tang, G.-J. Qi, C.C. Aggarwal, T.S. Huang, Heterogeneous network embedding via deep architectures, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 119–128.
[15] R. Das, M. Zaheer, C. Dyer, Gaussian lda for topic models with word embeddings., in: Proceedings of Association for Computational Linguistics, ACL, 2015.
[16] Y. Liu, Z. Liu, T.-S. Chua, M. Sun, Topical word embeddings., in: Proceedings of AAAI, 2015.
[17] Z. Yang, J. Tang, W.W. Cohen, Multi-modal Bayesian embeddings for learning social knowledge graphs., in: Proceedings of International Joint Conference on Artificial International, IJCAI, 2016.
[18] X.-L. Mao, Y.-J. Hao, D. Wang, H. Huang, Query completion in community-based question answering search, Neurocomputing 274 (2018) 3–7.
[19] Z. Ji, F. Xu, B. Wang, A category-integrated language model for question retrieval in community question answering., in: Proceedings of Asia Information Retrieval Societies Conference, AIRS, 2012.
[20] X. Xue, J. Jeon, W.B. Croft, Retrieval models for question and answer archives, in: Proceedings of SIGIR, ACM, 2008.
[21] G. Zuccon, B. Koopman, P. Bruza, L. Azzopardi, Integrating and evaluating neural word embeddings in information retrieval, in: Proceedings of Australasian Document Computing Symposium, ADCS, 2015.
[22] L. Cai, G. Zhou, K. Liu, J. Zhao, Learning the latent topics for question retrieval in community qa, in: Proceedings of International Joint Conference on Natural Language Processing, IJCNLP, 2011.
[23] M. Bouguessa, B. Dumoulin, S. Wang, Identifying authoritative actors in question-answering forums: the case of yahoo! answers, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2008, pp. 866–874.
[24] H. Zhu, E. Chen, H. Xiong, H. Cao, J. Tian, Ranking user authority with relevant knowledge categories for expert finding, World Wide Web 17 (5) (2014) 1081–1107.
[25] H. Deng, I. King, M.R. Lyu, Formal models for expert finding on dblp bibliography data, in: Proceedings of the Eighth IEEE International Conference on Data Mining, ICDM'08, IEEE, 2008, pp. 163–172.
[26] J. Weng, E.-P. Lim, J. Jiang, Q. He, Twitterrank: finding topic-sensitive influential twitterers, in: Proceedings of the Third ACM International Conference on Web Search and Data Mining, ACM, 2010, pp. 261–270.
[27] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, Berlin, Heidelberg, 2006.
[28] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv:1301.3781 (2013).
[29] W. Hu, J. Zhang, N. Zheng, Different contexts lead to different word embeddings, in: Proceedings of the 26th International Conference on Computational Linguistics, COLING 2016, 2016, pp. 762–771. Technical Papers
[30] L. Vilnis, A. McCallum, Word representations via gaussian embedding, in: Proceedings of International Conference on Learning Representations, ICLR, 2015.
[31] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. 3 (Jan) (2003) 993–1022.
[32] T.L. Griffiths, M. Steyvers, Finding scientific topics, Proc. Natl. Acad. Sci. 101 (suppl 1) (2004) 5228–5235.
[33] M. Kusner, Y. Sun, N. Kolkin, K. Weinberger, From word embeddings to document distances, in: Proceedings of International Conference on Machine Learning, ICML, 2015.
[34] J. Guo, Y. Fan, Q. Ai, W.B. Croft, Semantic matching by non-linear word transportation for information retrieval, in: Proceedings of Conference on Information and Knowledge Management, CIKM, 2016.
[35] Z. Zhao, L. Zhang, X. He, W. Ng, Expert finding for question answering via graph regularized matrix completion, IEEE Trans. Knowl. Data Eng. 27 (4) (2015) 993–1004.
[36] Y. Shen, W. Rong, Z. Sun, Y. Ouyang, Z. Xiong, Question/answer matching for cqa system via combining lexical and sequential information., in: Proceedings of AAAI, 2015, pp. 275–281.
[37] S. Robertson, H. Zaragoza, et al., The probabilistic relevance framework: Bm25 and beyond, Found. Trends® Inf. Retr. 3 (4) (2009) 333–389.
[38] C. Zhai, J. Lafferty, A study of smoothing methods for language models applied to ad hoc information retrieval, in: Proceedings of SIGIR, 2017.
[39] Q. Le, T. Mikolov, Distributed representations of sentences and documents, in: Proceedings of International Conference on Machine Learning, ICML, 2014.
[40] D. Ganguly, D. Roy, M. Mitra, G.J. Jones, Word embedding based generalized language model for information retrieval, in: Proceedings of SIGIR, 2015.
[41] H. Wu, W. Wu, M. Zhou, E. Chen, L. Duan, H.-Y. Shum, Improving search relevance for short queries in community question answering, in: Proceedings of Conference on Web Search and Data Mining, WSDM, 2014.
[42] S. Yuan, Y. Zhang, J. Tang, J.B. Cabotà, Expert Finding in Community Question Answering: A Review, arXiv preprint arXiv:1804.079581 (2018).
[43] Y. Kim, Convolutional neural networks for sentence classification, in: Proceedings of Empirical Methods in Natural Language Processing, EMNLP, 2014.
[44] J. Zhang, M.S. Ackerman, L. Adamic, Expertise networks in online communities: structure and algorithms, in: Proceedings of the 16th International Conference on World Wide Web, ACM, 2007, pp. 221–230.
[45] F. Wu, X. Lu, J. Song, S. Yan, Z.M. Zhang, Y. Rui, Y. Zhuang, Learning of multimodal representations with random walks on the click graph, IEEE Trans. Image Process. 25 (2) (2016) 630–642.
[46] J. Guo, S. Xu, S. Bao, Y. Yu, Tapping on the potential of Q & A community by recommending answer providers, in: Proceedings of the 17th ACM Conference on Information and Knowledge Management, ACM, 2008, pp. 921–930.

Lei Sang is currently pursuing the Ph.D. degree with the School of Computer Science and Information Engineering, Hefei University of Technology, China, and also with the Faculty of Engineering and Information Technology, University of Technology, Sydney, Ultimo, NSW, Australia. His current research interests include natural language processing and recommender system.

Min Xu received the B.E. degree from University of Science and Technology of China, in 2000, M.S degree from National University of Singapore in 2004 and Ph.D. degree from University of Newcastle, Australia in 2010. She is currently a Senior Lecturer at University of Technology, Sydney. Her research interests include multimedia data analytics, pattern recognition and computer vision. She has published over 100 research papers in high quality international journals and conferences.



Shengsheng Qian received the B.E. degree from the Jilin University, Changchun, China, in 2012, and the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2017. He is currently an Assistant Professor with the Institute of Automation, Chinese Academy of Sciences. His current research interests include social media data mining and social event content analysis.

Xindong Wu received the Ph.D. degree in artificial intelligence from The University of Edinburgh, Edinburgh, U.K. He is a professor of computer science at the University of Louisiana at Lafayette, USA. His current research interests include data mining, knowledge-based systems, and Web information exploration. He is the Steering Committee chair of IEEE International Conference on Data Mining (ICDM). He is the editor-in-chief of Knowledge and Information Systems (KAIS) and ACM Transactions on Knowledge Discovery from Data (TKDD). He is a fellow of IEEE and the AAAS.