Answer Keyword Generation for Community Question Answering by Multi-aspect Gamma-Poisson Matrix Completion

Qing Liu
Advanced Analytics Institute
University of Technology Sydney
Sydney, Australia
Qing.Liu-7@student.uts.edu.au

Trong Dinh Thac Do
Advanced Analytics Institute
University of Technology Sydney
Sydney, Australia
ThrongDinhThac.Do@uts.edu.au

Longbing Cao
Advanced Analytics Institute
University of Technology Sydney
Sydney, Australia
Longbing.Cao@uts.edu.au

***Abstract*—Community question answering (CQA) recommends appropriate answers to existing and new questions. Such answer recommendation is challenging since CQA data is often sparse and decentralized and lacks sufficient information to generate suitable answers to existing questions. Matching answers to new questions is more challenging in modeling Q/A sparsity, generating answers to cold-start/novel questions, and integrating metadata about Q/A into models, etc. This paper addresses these issues by a novel statistical model to automatically generate answer keywords in CQA with multi-aspect Gamma-Poisson matrix completion (MAGIC). MAGIC is the first trial in CQA to model multiple aspects of Q/A sentence information in CQA by involving Q/A metadata, Q/A sparsity, and both lexical and semantic Q/A information in a hierarchical Gamma-Poisson model. MAGIC can efficiently generate answer keywords for both existing and new questions against nonnegative matrix factorization (MF), probability MF, and relevant Poisson factorization models w.r.t. recommending appropriate and informative answer keywords.**

***Index Terms*—Poisson factorization, variational inference, answer keyword generation, matrix completion, community question answering**

I. Introduction

Community question answering (CQA) is a popular online service for people to share, search and produce knowledge. Such platforms as Stack Overflow, Quora and Yahoo!Answers not only provide a direct way to ask questions (Q) and acquire answers (A) but also allow users to interact with each other and judge the answer quality. For example, users can vote for answers and each answer receives a score equaling the upvote number minus the downvote number, which reflects the answer quality. The higher the score is, the more the answer matches its question. High quality Q/A pairs can be used as the knowledge base for many Q/A applications such as automatic Q/A dialogue and enabling automatic chatbot.

Compared to human-computer Q/A systems, CQA can generate more accurate and fluent human-generated answers. While users can benefit from high quality answers in CQA, they might have to wait for a long time before high quality answers are available, and it may also be uncertain whether and when a good answer may appear. This downgrades the applicability and value of CQA and motivates recent research on matching answers to questions to improve user Q/A experience and generate knowledge in CQA. The related work can be categorized into two: question routing, and answer selection.

Given a new question, *question routing* [1], [2] connects it to users who are likely the experts in the area. This method directs questions to qualified answerers but does not guarantee an answer to be produced, which is constrained by respondent's willingness and availability, etc. Instead, *answer selection* [3]–[5] first finds the mostly related questions with a given question and then recommends the correspondingly potential answers in the Q/A corpus. In general, this retrieval-based method can return more fluent and informative responses [6] in a short time. However, compared with answer generation-based methods, retrieval-based methods always lack flexibility and cannot deal with new questions or questions unmatchable to those in the Q/A database.

To address the weaknesses of traditional CQA research, this paper aims for automatically creating answer keywords, which is a new research direction in CQA. By the coupling learning of the relations, matching and similarity between questions and answers in a CQA corpus, we aim to capture multiple aspects of Q/A and their coupled interactions [7], the machine-generated answer keywords can then be produced in more meaningful way for both existing and new questions. However, this is a non-trivial task due to several reasons. First, it is quite challenging to analyze Q/A sentences by merely lexical information. Compared with documents, Q/A sentences are much shorter. The sparsity of word (keyword) co-occurrences in Q/A pairs cannot measure their similarity accurately. The second challenge is how to integrate the multiple aspects of information and their couplings effectively. CQA corpus provides abundant information including answer scores, user profile, and tags, etc. It is challenging to select and integrate the right information from the heterogeneous resources.

Focusing on these difficulties, we propose a Multi-aspect Gamma-Poisson MatrIx Completion (MAGIC) model, which generates answer keywords to existing and new questions by coupling various Q/A metadata and tackling the Q/A sparsity. Given a question, MAGIC first models the essential character-

istics of Q/A sentences and abundant metadata information of Q/A (e.g., question tags, and answer scores). Then, to model the relations and similarity between questions and answers from the CQA corpus, we construct a Q/A interaction matrix where each row stands for a question and each column stands for an answer. Values in this matrix are the fusion of both explicit Q/A relations (i.e., Q/A lexical similarity) and implicit Q/A matching (i.e., answer scores as semantic similarity). The Q/A interaction matrix is then modeled w.r.t the Poisson distribution. With a hierarchical Poisson factorization (HPF) model, MAGIC factorizes the interaction matrix into latent spaces. We further extend HPF by involving a metadata layer to capture Q/A metadata and characterize the multiple aspects of each question and each answer. To solve the model, mean-field variational inference is applied as an approximate inference since no closed-form solution exists.

Consequently, this work makes the following contributions:

- A statistical approach MAGIC formulates Q/A in domain-specific CQA as an answer keyword generation problem. MAGIC recommends answer keywords to existing and new questions in a corpus. It can also serve for both answer selection by searching existing answers per the recommended answer keywords and automatic answer generation based on the generated answer keywords.
- A metadata-driven hierarchical Gamma-Poisson matrix completion mechanism (MHPF) is further introduced for MAGIC. MHPF couples the explicit and implicit relations and similarity between Q/A and involves Q/A metadata. Thus, MAGIC is the first CQA method that jointly learns abundant Q/A side information into Poisson-Gamma statistical models and the first attempt to apply hierarchical Poisson factorization to sparse Q/A and cold-start/novel questions in CQA.
- The experimental results show that the MAGIC model can generate valuable and diverse answer keywords with sparse Q/A matching on domain-specific CQA data. We also find that the metadata integrated into MAGIC plays an important role in guiding matrix factorization and topic intensity assignment of Q/A in latent spaces shown as the case study and key component visualization.

II. Related Work and Background

A. CQA and related work

Q/A analysis in CQA is an important yet challenging task. The relevant research can be roughly divided into two directions: answer selection and question routing. The methods for answer selection can be further divided into two categories: question semantic matching [8]–[11], and question answer matching [3], [12]–[14]. With an increasing number of questions available in open Q/A forums, a newly asked question may be similar with some existing ones. The aim of *question semantic matching* is to identify these questions similar to the new one. This similar question detection can not only reduce data redundancy but also respond to asker's question timely and improve user experience. *Question answer matching* is an issue widely explored in information retrieval. A question in a CQA platform is usually followed by multiple answers from different respondents. Since a CQA usually has good and bad answers intermingled, Q/A matching selects high quality answers and can be used for knowledge acquisition and generating a common sense Q/A corpus. Another problem in CQA is the high unanswered-question rate [15]. Due to the large number of Q/As in a CQA platform, newly posted questions may not be immediately identified and responded by answerers who likely know the answers (such answerers are subject experts in the areas). This problem has also attracted considerable attention with research known as *question routing* [16]–[19].

B. Probabilistic topic modeling

In this paper, we focus on a new task that is to automatically generate answer keywords for a given question by statistical generative models. Q/As are generally very short when compared with documents. Due to lack of enough word co-occurrences in Q/As, classic probabilistic generative topic models, i.e., Latent Dirichlet Allocation (LDA), suffer a lot from the performance degradation. To overcome the lack of contextual information, a variety of novel methods are available, matrix factorization (MF) models, especially non-negative matrix factorization (NMF) [20], are popularly used for short text understanding and topic modeling. In this paper, we mainly focus on statistical generative models and briefly review the related work below.

NMF methods in short text analysis. NMF models are widely used to learn topics in short text. Both LDA and NMF uncover the latent topics in unstructured short text. The work in [21] concludes that NMF tends to be superior to LDA for short text topic mining. Because of its good interpretability and performance, NMF has been applied in short text topic mining to factorize various non-negative latent spaces, e.g., mining query subtopics with a TF-IDF matrix and a question similarity matrix [22] and clustering web items with weighted data feature matrix [23]. Further, external corpus can also be combined with NMF. The work in [21] integrates the word-word semantic graph regularization learned from Wikipedia into the basic NMF model directly, while a question-question similarity matrix is trained on Wikipedia which is then used as a complementary part of NMF in [22].

Poisson matrix factorization. Our proposed MAGIC formulates the analysis of Q/A relations, similarity and matching as an automatic answer keyword generation task by a Poisson factorization (PF) model. PF is part of NMF and has been widely studied in collaborative filtering-based recommender systems [24]–[26]. For sparse and noisy data, PF has the desirable property that does not penalize missing values (i.e., 0) as strongly as Gaussian distribution [27]. Many PF variants have been proposed to combine more information into the basic PF model, e.g., content information [25] and metadata [28]–[30]. As shown in [28]–[30], PF models fit countable data like ratings, votings and word counts better than Gaussian distribution since the statistics of such data are

usually sparse and discrete, and this is exactly the case of Q/A in CQA tasks. In addition, compared with classic NMF and Gaussian probabilistic matrix factorization [31], PF delivers non-negative estimates of values and avoids to overfit missing (0) values. As a result, PF models can dramatically improve the prediction performance and reduces the computational cost on sparse and large data.

Mean-field variational inference. Since it is hard to obtain a closed-form expression of the posterior distribution of probabilistic graphical models, variational inference solves this problem by casting the inference problem as an optimization problem [32]. That means a family of distributions over the hidden variables is introduced as the variational distribution. The optimization is achieved by finding the member of the family that is closest to the actual posterior distribution. The Kullback-Leibler divergence (KL-divergence) is applied to measure the closeness between the variational distribution and the true distribution. To make this optimization easier, the mean-field method regards each hidden variable as independent of each other in the variational distribution. In brief, the basic idea behind the mean-field variational inference is to choose a factorized family of variational distributions to approximate the actual posterior distribution, so that the KL-divergence between the variational distribution and the true posterior is minimized [27]. In this work, our model takes the mean-field variational inference method.

III. MAGIC FOR ANSWER KEYWORD GENERATION

Here, we introduce the design of the proposed MAGIC for automatically generating answer keywords to both existing and new questions for CQA by involving Q/A metadata, explicit and implicit Q/A relations, matching, and similarity.

A. The MAGIC model

In CQA, Q/As are organized with respect to categories or topics. Under a specific topic, the related Q/As are posted by users. Each question is usually associated with several tags that summarize the topics of the question. CQA usually has a voting mechanism to judge the quality of answers by all users, who may upvote good answers and downvote bad ones. The final voting score of an answer to a given question is the result of the upvote number minus the downvote number. Accordingly, higher score indicates higher quality of answers, and vice versa.

With the above settings in CQA, we aim to automatically generate answer keywords for a question, which is valuable for both selecting proper answers and automatically generating answers to given questions (the latter is out of the scope of this paper). As shown in Fig. 1, MAGIC works as follows. For the input question q, it is first processed by the *pre-processor* and then converted to a list of question keywords. Related questions containing these question Keywords will be picked out by the *question finder*. With the completed matrix, answer candidates for q are generated by the *answer candidate selector*. Lastly, answer Keywords are finalized by the *answer keyword extractor* based on the Q/A keyword matrix. In

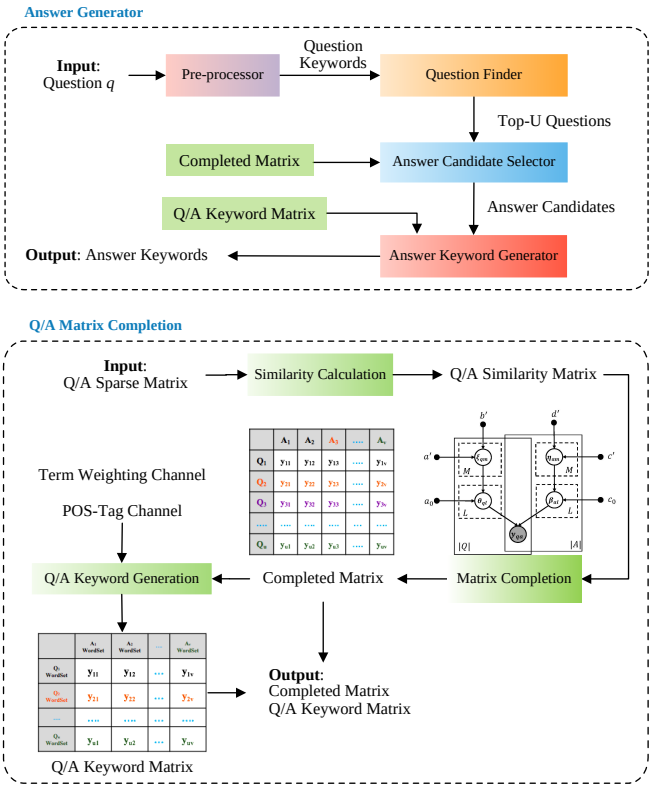

Fig. 1: The workflow of MAGIC to generate answer keywords for community question answering

achieving this, the completed matrix and the Q/A keyword matrix are generated as follows. Given a Q/A sparse matrix, the matching degree y_{qa} between each Q/A pair is computed per Eqn. (6). MAGIC first factorizes the Q/A sparse matrix into a low-dimensional space and then completes the missing values in this matrix to obtain the Q/A similarity matrix. The Q/A keyword generator takes the completed matrix as input, removes the noisy words through the data cleaning, and combines the term weighting channel and the POS-tag channel to generate the Q/A keyword matrix.

In the above, MAGIC consists of four major modules: pre-processor, question finder, answer candidate selector, and answer keyword generator, to achieve the goal of automatic answer keyword generation. We explain them below.

a) Pre-processor: The original Q/A sentences generated by users may contain typos and meaningless symbols given the nature of how CQA is generated. The keywords in a sentence capture the main information and are often used to represent the sentence in NLP. As shown in Fig. 1, the *pre-processor* first conducts a series of cleaning processes including removing stopwords, external links and special symbols, acronym restoration, tokenization and lemmatization for each question q. It then extracts keywords from all user-generated Q/As in a corpus. This pre-processing is also applied to newly posted questions. After this, each q is transformed into a set of n clean, standard and meaningful keywords, $w_1, w_2, ..., w_n$.

b) Question finder: With the keywords extracted from a given question q, the *question finder* aims to identify the most similar questions in the Q/A corpus. Based on the keyword-based Jaccard similarity between Qs and As, the $top-U$ most similar questions (represented as $q_1, q_2, \ldots, q_u$) are selected from the corpus. Here, U can be adjusted dynamically according to the coverage degree and a search range (upper limit of U value), and a minimum number of existing questions in the corpus to cover all the keywords of q that can be found.

c) Answer candidate selector: For each selected similar question q_i, the *answer candidate selector* finds the $Top-K$ best matched answers (represented by $a_{i1}, a_{i2}, ..., a_{ik}$). In CQA, although some answers already have scores associated, most of answers in the corpus do not have scores available, leading to the sparsity of answer quality scores, as shown in Table I). To estimate the missing scores, we conduct the Q/A matching matrix completion. MAGIC first constructs a Q/A relation matrix, where rows and columns stand for existing questions and answers in the corpus, respectively. Consequently, a cell value y_{qa} in the matrix measures the matching degree or relation strength between a question and an answer. MAGIC estimates the missing values in the matrix and converts a sparse Q/A matching matrix to a complete one. After the matrix completion, each value in the matrix is treated as a Q/A sentence matching degree (weight), which also supports the further answer candidate selection.

d) Answer keyword generator: Further, the *answer keyword generator* generates a list of answer keywords from answer candidate a_{ij} for question q. a_{ij} is the j^{th} answer candidate to the similar question q_i for q. Besides the Q/A sentence matching weights captured in the Q/A matching matrix, we further introduce two weighting channels on the word level: the term weighting channel and the POS channel, similar with [33]. The TF-IDF weight [34] is used as a term weighting channel in the implementation of our model. By integrating the Q/A matching weight, term weight, and POS weight, we uniformly rank all the keywords from answer candidates.

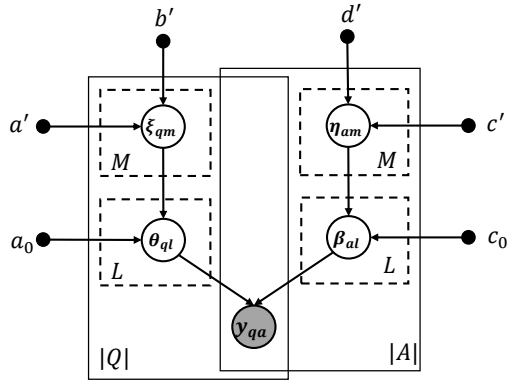

Fig. 2: The graphical model of MAGIC.

B. Metadata hierarchical Poisson factorization

The graphical model of the metadata-driven hierarchical PF in MAGIC is shown in Fig. 2. It plays an important role in the sparse Q/A interaction matrix completion, which further determines the Q/A matching degree. Below, we introduce the graphical model in the bottom-up order.

a) Poisson factorization for the Q/A interaction matrix: We construct the Q/A interaction matrix Y with the dimension of $|Q| \times |A|$ to capture the matching degree and relation strength between Qs and As under a certain topic. We factorize Y with PF by assuming y_{qa} follows a Poisson distribution, then y_{qa} can be factorized to a question vector θ_q and an answer vector β_a. Both vectors contain L latent semantic intensity components. We assume that each question semantic intensity component θ_{ql} and each answer semantic intensity component β_{al} follow Gamma distributions, which are the conjugate priors of the Poisson distribution.

b) Construction of the Q/A interaction matrix: The Q/A interaction matrix embodies both explicit (lexical) similarity and implicit (semantic) similarity between Qs and As. Specifically, the lexical similarity between a question and an answer is captured by the Jaccard similarity as shown in the first part of Eqn. (1), and the Q/A sentence semantic similarity is captured by the answer score. In CQA, one question may have one to multiple answers, leading to a one-to-one or one-to-many relation between a question and an answer. Answers may have different scores which reflect how well answers match questions. Hence, the answer score is an implicit indicator of Q/A semantic similarity judged by users through their votings and is complementary to the lexical similarity. Eqn. (1) obtains the final Q/A matching degree value y_{qa}:

$$y_{qa} = \begin{cases} T & score \geq T \\ [T \cdot \frac{Q \cap A}{Q \cup A}] + score_{qa} & score < T \end{cases} \quad (1)$$

where T, determined by domain experts, is the truncation level of voted scores on answers. When the answer score is larger than T, it indicates that the corresponding Q/A pair has been fully evaluated by users, and this score alone is sufficient to represent the Q/A matching degree. When the score is less than T, it means the Q/A pair is not discovered by enough users, and the voting scores have some bias. To leverage this, we use the Jaccard similarity to make up the shortage of the insufficient voting scores made by users on the lexical level. In addition, we obtain the approximation with $[\cdot]$ to ensure y_{qa} is an integer.

However, due to the large quantity of Q/A items and diversified user interest, not every Q/A pair can be discovered by users and fortunately receive unbiased scores. Further, the answer scores only exist between paired Qs/As. This means even a question and an answer match well, while the answer does not address the question, there is still no score existing between the question and the answer. Hence, the answer scores reflect an incomplete aspect of Q/A matching, and many Q/A pairs receive no scores. To mitigate the possible bias and missing answer scores in CQA, MAGIC completes the missing

values in matrix Y built on the integrated weights of Q/A matching, answer scores, and Q/A lexical similarity.

c) Metadata layer upon Poisson factorization: A question or answer sentence contains multiple aspects of information. For example, in reading the question *What's the history of short and tall sizes?*[1] under the *coffee* topic, it is obvious that this question involves several aspects, e.g., coffee size-related terminology, historical evolution and probably Starbucks of tall and short coffee, even though there may not be words about some of them in this question. The Q/A semantic matching is expected to capture these multi-aspects embedded in Q/A sentences. Accordingly, we add a metadata layer on top of the PF architecture to form a hierarchical PF structure. In this work, the Q/A metadata captures the Q/A tagging information since Q/A tags reflect the different semantic and categorization aspects of a question or answer. Taking the aforementioned question as an example, the tags of this question are *Starbucks*, *history* and *terminology*. These tags complement the question and provide more abstract or commonsense information about the question sentence. Hence, our model uses tags to capture multiple aspects of Q/A.

To facilitate the PF calculation, we assume the Q/A metadata follows a Gamma distribution, which is the conjugate prior of Poisson when the shape parameter is known. The indicator function δ shows whether a question or answer is associated with a tag. If question q has the j^{th} tag, then $\delta(q, j) = 1$; otherwise, $\delta(q, j) = 0$. The same for answers.

The generative process of MAGIC is as follows, where variables shp and rte represent the shape and rate of the Gamma prior, variables a_0, c_0, a', c', b', d' are hyper parameters.

(1) For the m^{th} aspect of Q/A metadata:

(a) Sample its weight $\xi_{q,m}$ in question q:

$$\xi_{q,m} \sim Gamma(a', b') \tag{2}$$

(b) Sample its weight $\eta_{a,m}$ in answer a:

$$\eta_{a,m} \sim Gamma(c', d') \tag{3}$$

(2) For each component l in the Poisson factorization:

(a) Sample the question's semantic intensity θ_{ql}:

$$\theta_{ql} \sim Gamma(a_0, \prod_{m=1}^{M} \xi_{q,m}^{\delta(q,m)}) \tag{4}$$

(b) Sample the answer's semantic intensity β_{al}:

$$\beta_{al} \sim Gamma(c_0, \prod_{m=1}^{M} \eta_{a,m}^{\delta(a,m)}) \tag{5}$$

(3) For each Q/A pair, sample their similarity y_{qa}:

$$y_{qa} \sim Poisson(\boldsymbol{\theta}_q^T \boldsymbol{\beta}_a) \tag{6}$$

[1]https://coffee.stackexchange.com/

C. Mean-field variational inference for MAGIC

The mean-field variational inference (VI) is applied to solve the posterior approximation problem in MAGIC since VI tends to be faster and easier to scale to large data than the Markov chain Monte Carlo (MCMC) sampling. We introduce latent variables $z_{qal} \sim Poisson(\theta_{ql}\beta_{al})$ to make the model conditionally conjugate and apply the closed-form coordinate ascent. updatesAccording to the additive property of Poisson distribution, entry y_{qa} can be expressed as:

$$y_{qa} = \sum_l z_{qal} \tag{7}$$

The variational prior distributions are chosen from the same family as the true distributions. Accordingly, the mean-field variational distribution is:

$$\begin{aligned} q(\theta, \beta, \xi, \eta, z) = &\prod_{q,l} q(\theta_{ql}|\gamma_{ql}) \prod_{a,l} q(\beta_{al}|\lambda_{al}) \\ &\prod_{q,m} q(\xi_{qm}|\kappa_{qm}) \prod_{a,m} q(\eta_{am}|\tau_{am}) \\ &\prod_{q,a} q(z_{qa}|\phi_{qa}) \end{aligned} \tag{8}$$

Solving this optimization with the standard results [35], we can obtain the iterative updates[2]:

(1) For all questions and answers, initialize the $shape$ parameter for each m^{th} aspect of weightings:

$$\kappa_{qm}^{shp} = a' + Ka \tag{9}$$

$$\tau_{am}^{shp} = c' + Kc \tag{10}$$

(2) For each question,

(a) Update the $rate$ parameters for the m^{th} aspect of weightings:

$$\kappa_{qm}^{rte} = b' + \delta W_{qm} \sum_l \frac{\gamma_{ql}^{shp}}{\gamma_{ql}^{rte}} \tag{11}$$

where $W_{qm} = \prod_{M_{-m}} \xi_{qm}^{\delta(q,m)}$ (M_{-m} means the set of tags without the m^{th} tag).

(b) Update the $shape$ and $rate$ parameters for the semantic intensity:

$$\gamma_{ql}^{shp} = a_0 + \sum_a y_{qa}\phi_{qal} \tag{12}$$

$$\gamma_{qa}^{rte} = \prod_m (\frac{\kappa_{qm}^{shp}}{\kappa_{qm}^{rte}})^{\delta(q,m)} + \sum_a \frac{\lambda_{al}^{shp}}{\lambda_{al}^{rte}} \tag{13}$$

(3) For each answer,

(a) Update the $rate$ parameters for the m^{th} aspect of weightings:

$$\tau_{am}^{rte} = d' + \delta W_{am} \sum_l \frac{\lambda_{al}^{shp}}{\lambda_{al}^{rte}} \tag{14}$$

[2]Due to space limitation, the derivation is ignored here. Readers can refer to [35] for more details.

where $W_{am} = \prod_{M_{-m}} \eta_{am}^{\delta(a,m)}$.

(b) Update the $shape$ and $rate$ parameters for the semantic intensity:

$$\lambda_{al}^{shp} = c_0 + \sum_q y_{qa}\phi_{qal} \tag{15}$$

$$\lambda_{al}^{rte} = \prod_m (\frac{\tau_{am}^{shp}}{\tau_{am}^{rte}})^{\delta(a,m)} + \sum_q \frac{\gamma_{ql}^{shp}}{\gamma_{ql}^{rte}} \tag{16}$$

(4) For each Q/A pair with $y_{qa} > 0$, update the multinomial:

$$\phi_{qal} \propto exp\Psi(\gamma_{ql}^{shp}) - log\gamma_{ql}^{rte} + \Psi(\lambda_{al}^{shp}) - log\lambda_{al}^{rte} \tag{17}$$

The mean-field variational inference for MAGIC is introduced in Algorithm 1.

Algorithm 1 The Mean-field Variational Inference for MAGIC

```
Initialize the variational parameters {γ, λ, κ, τ, φ}.
Set hyper-parameters {a_0, a', b', d', c', c_0}.
Set the number of latent components L.
Set the m^th aspect's shape parameters by Eqns. (9) and (10).
repeat
    for matching the degree of question q to answer a per y_qa ≠ 0 do
        Update the multinominal as in Eqn. (17).
    end for
    for each question do
        for each aspect do
            Update the aspect's rate parameter per Eqn. (11).
        end for
        for each latent component do
            Update the semantic shape per Eqn. (12).
            Update the semantic rate per Eqn. (13).
        end for
    end for
    for each answer do
        for each aspect do
            Update the m^th aspect's rate per Eqn. (14).
        end for
        for each latent component do
            Update the semantic shape w.r.t. Eqn. (15).
            Update the semantic rate w.r.t. Eqn. (16).
        end for
    end for
until convergence
```

IV. Experiments and Evaluation

In this section, we evaluate MAGIC in terms of its prediction performance against typical MF models. We particularly test MAGIC against its variant HPF without involving metadata, and show case studies of the MAGIC results for answer keyword generation.

TABLE I: Statistics of the StackExchange datasets.

dataset	# question	# answer	# score	sparsity	# tag
coffee	1017	2036	2016	0.097%	5
beer	914	2048	1991	0.106%	6
hardware	2370	4542	4423	0.041%	10
bioinfo	1594	2277	2263	0.062%	5

A. Experimental settings

a) Datasets: We randomly select four domain-specific CQA datasets from the latest data dump (published in December 2018) of StackExchange[3]. Table I summarizes the statistics of the randomly selected datasets, which include question number, answer number, the number of scores and the tag number in each dataset. In the raw data, tags were labeled by users, many tags look similar and redundant. We thus choose and combine the most popular top-x tags to fit our model. In our experiment, the value of x influences the model performance dramatically, and the tag number for each dataset is chosen according to the best experiment results (see Section IV-D).

Table I shows that the observed actual answer scores generated by user votings, which are extremely sparse in each dataset. The non-zero score numbers in each dataset account for only 0.097%, 0.106%, 0.041% and 0.062%, respectively.

b) Baseline methods: To evaluate the metadata-driven prediction performance, we compare MAGIC with an array of competing matrix factorization methods below. We exclude other non-MF models for comparison because it is incomparable from the statistical sense, and PF models have shown to fit sparse and large data well, which satisfies our problem.

- Non-negative matrix factorization (NMF) factorizes Q/A interaction matrix into two non-negative matrices in the low-dimensional space. The two low-dimensional matrices are updated alternatively by minimizing the KL-divergence between the actual and the estimated Q/A interaction matrices.
- Probability matrix factorization (PMF) models the actual interaction matrix as the product of two low-rank question and answer semantic intensity matrices by assuming the matching degree values in the actual matrix follow Gaussian distribution. PMF also places zero-mean spherical Gaussian priors on the two low-rank matrices.
- Poisson factorization (PF) replaces the Gaussian assumption with Poisson distribution. Accordingly, Gamma distribution is chosen as the priors for the convenience of variational inference.
- Hierarchical Poisson factorization (HPF) without involving Q/A metadata is a PF variant implemented per the work in [24] to capture more complicated interactions

[3]https://stackexchange.com/. Due to space limitation, we only report results on four randomly selected datasets. Since the datasets may likely follow the same distribution, similar experiments could be conducted on the other datasets.

TABLE II: The main POS tags and their weights (Note: the POS tags are from the NLTK package).

POS tag	description	weights
IN	preposition	0.1
JJ, JJR, JJS	adjectives	0.4
VB, VBD, VBG, VBN, VBP, VBZ	verbs	0.4
NN, NNS, NNP, NNPS	nouns	0.5
RB, RBR, RBS	adverb	0.3
Others	all the other tags e.g. pronouns	0

between questions and answers. HPF constructs a hierarchical Poisson factorization model by placing another Gamma prior layer on top of PF for the conjugation between Gamma distributions when shape parameters are known. Here, HPF does not involve the Q/A metadata to compare with MAGIC and test the impact of involving Q/A metadata on HPF modeling performance.

c) Parameter settings: The parameters in MAGIC are carefully tuned to mitigate the error accumulation in the further processes. In the Q/A interaction matrix completion, the hyper-parameters are set as $a_0 = c_0 = a' = c' = 0.3$. The number of latent components $L = 100$ (the same as [24]). The metadata hyper-parameters are set as $b' = d' = 0.1$ (the same as [28]). The threshold T in Eqn. (1) equals 5 and $\alpha_1 = \alpha_2 = 1$. For the answer keyword generation, we set POS tags with different weights as in Table II. We compare MAGIC with the above typical MF models by setting the same latent component number $L = 100$ for all models while keep the other hyper-parameters the same as their original settings.

B. Pre-processing

We adopt the common pre-processing pipeline in natural language processing to clean the raw data from StackExchange. The main pre-processing procedures are below.

- Data cleaning: We extract questions, answers, tags and scores of Q/A pairs from the dataset with Beautiful Soup 4, which is a Python package for pulling data out of HTML and XML files. Some abbreviations have been restored, and special symbols such as hyphens have been removed from sentences.
- Truncation: In CQA, a question usually contains two parts: the question title and the question body. A question title can be seen as the brief summary of the question body, and the question body is the detailed explanation. We directly concatenate these two parts. Due to the different lengths of sentences, we truncate the long sentences and set the maximum length as 50 words for questions and 100 words for answers. These settings are based on the Q/A statistics, questions are usually shorter than answers, and most valuable information often appears in the fore part of answers and the question titles.
- Stemming and lemmatization: They reduce the inflectional forms of a word to a common base form. However, stemming usually chops off the ending part of a word in a crude heuristic process, while lemmatization handles this more properly by using a vocabulary and morphological analysis of words. Hence, we use lemmatization to keep the word integrity as much as possible.
- Punctuation, POS tagging, stopword removal and tokenization: Punctuation and stopwords are removed by the NLTK package. Further, words are labeled with different POS tags to describe their different levels of importance.
- Removing both the most common and uncommon words: To generate more valuable answer keywords, the first 1% high frequency words and the last 1% low frequency words are removed from the corpus. These words may contain less information but much noise, which influences the quality of answer keyword generation in our model.

Further, as shown in Table I, the answer scores are very sparse. Although PF is good at dealing with the sparse matrix efficiently, its prediction accuracy suffers from the value sparsity. To address the sparsity and its challenge to PF, we combine the Jaccard similarity between a Q/A pair based on their keywords with the original answer scores per Eqn. 1 and further normalize them to form the Q/A interaction matrix (valued in $[0, 1]$). In fact, introducing the Jaccard similarity also brings about two benefits: on one hand, the keyword-based Q/A Jaccard similarity compensates for the word-level similarity (which reflects the lower level similarity between words than the sentence similarity reflected by user voting scores); on the other hand, the Jaccard similarity slightly reduces the sparsity of the answer scores-based Q/A interaction matrix as validated by the experiments. In combining the word-level Jaccard similarity and sentence-level answer score similarity by Eqn. (1), we need to respect and handle the different meanings and scales of Jaccard similarity and the observed scores. To alleviate their possible contradiction and ensure their consistency in their combination, the explicit numerical semantics are weakened in the similarity matrix, where a positive score indicates that an answer has certain matching degree with a question, while the true score value is discarded (same as in [24]). As a result, we produce a Q/A interaction matrix for further statistical modeling.

C. Prediction comparison

To the best of our knowledge, this is the first attempt to compute and predict the Q/A matching degree and to recommend answer keywords for given questions by hierarchical PF modeling to cope with the so-called 'cold start' problem that often troubles MF methods. The evaluation mainly tests the prediction ability of MAGIC w.r.t. two metrics: prediction accuracy and Root Mean Square Error (RMSE).

We use 80% data for training and 20% for testing by random splitting. The prediction accuracy of MAGIC against all baselines is reported in Fig. 3, and at the same time the RMSE to evaluate the prediction error is presented in Fig. 4. Fig. 3 shows that MAGIC obviously outperforms the other

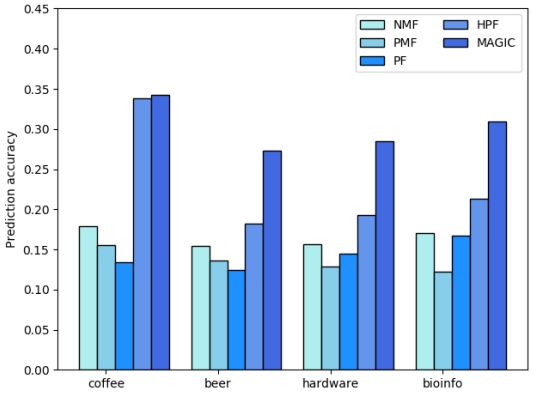

Fig. 3: Comparison of the prediction accuracy.

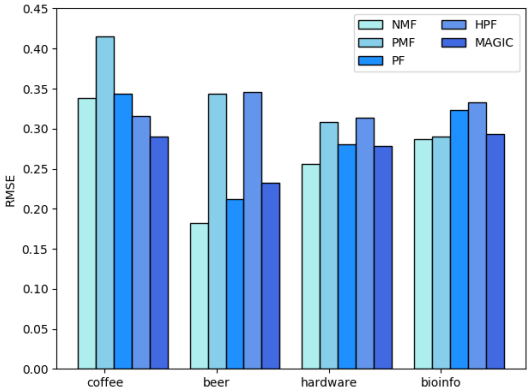

Fig. 4: Comparison of the prediction RMSE.

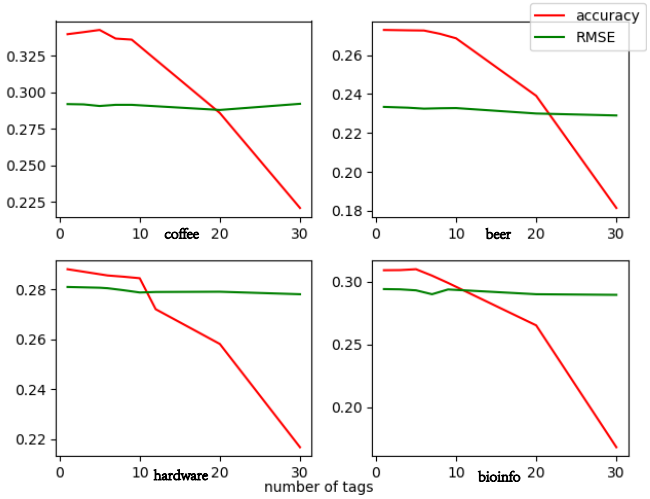

Fig. 5: Performance change with different tag numbers.

Q: I ground some coffee beans this morning, how long can I store ground coffee in the fridge?	
Answer (Score: 14)	keywords generated by MAGIC
Coffee should never be stored in the fridge. Coffee will absorb smells and flavors in your refrigerator. …Proper storage of coffee is to put your beans into an airtight container, and store around 25°C out of sunlight.	coffee, flavor, extract, absorb, put, compound, brew, ground, bean, whether, undesired, sunlight, store, refrigerator, proper, never, major, fridge, **degassing**, contribute, container, component, **cabinet**, airtight, **ziplock**

Fig. 6: Answer keywords generated for an existing question.

MF/PF models and makes the highest improvement (nearly 9%) over HPF, which is the second best-performing model without involving Q/A metadata. Fig. 4 shows that MAGIC also achieves relatively small RMSE. By taking both the prediction accuracy and RMSE into account, MAGIC performs best when compared with NMF, PMF, PF and HPF.

The MAGIC performance improvement is driven by two aspects. On one hand, hierarchical structure makes PF more flexible. The layer-by-layer abstraction helps the model to capture more underlying and general sentence features. On the other, Q/A metadata (i.e., the Q/A tagging information) contributes a lot to the performance improvement. Compared with the HPF without metadata, Fig. 3 shows that the MAGIC prediction accuracy dramatically increases after introducing the tagging information as metadata into MAGIC. This is quite reasonable, since there is a high probability that a question and an answer match to some extent if they share many same tags.

D. Selection of the tag number

Since many tags labeled by users in the raw data are similar and redundant, deciding how many valuable tags to combine in MAGIC is a non-trivial task. Here, we conduct a series of experiments to figure out how the tag number influences the performance of MAGIC. Fig. 5 shows that the variation of tag number incurs dramatic impact on the performance of MAGIC. Taking both the prediction accuracy and RMSE into consideration, the tag number is chosen according to the best experimental results on each dataset and the corresponding tag numbers are shown in Table I.

E. Case study

As mentioned before, MAGIC is a combination of answer generation and retrieval-based Q/A methods. We present two case studies below to demonstrate the accuracy and results of MAGIC for automatic answer keyword generation.

question	I like sweet Nespresso coffee. What's the temperature and humidity to store it?	How does black beer taste?
keywords generated by MAGIC	**air**, flavor, **bag**, place, **seal**, **container, dark, cool,** case, **vacuum, dry,** change, **storage,** bitterness, cause, acidity, **sweetness**	beer, alcohol, **flavor, cold,** temperature, **bitterness,** cause, content, release, **mix,** post, dilute, ethanol, chemical, **strong**

Fig. 7: Answer keywords generated for new questions.

Fig. 6 shows a question existing in the Q/A corpus and its only answer available is listed in the left column. The score given to this answer is 14, which means it is a high-quality answer. The right column consists of keywords recommended by MAGIC, where the red-colored words are the new and related words. They do not appear in the actual answer but do show a strong relevance to the given question. This example shows that MAGIC can leverage existing question by summarizing and selecting the keywords that capture more diversity of answers.

In addition, Fig. 7 shows the answer keywords recommended by MAGIC for two new questions. The bold-face keywords are closely related to their questions. For example, in the first question, it asks mainly about how to store coffee. Accordingly, MAGIC generates keywords 'container', 'cool', 'dry', and 'dark', etc., to be included in its answers. These keywords correspond well to keywords 'temperature' and 'humidity' in the question. Further, for the taste of black beer in the second question, MAGIC generates keywords 'flavor', 'bitterness', and 'strong', etc. These keywords present a general description of the black beer taste.

We further visualize the POS channel in Fig. 8. POS weights are shown; the darker the colour, the greater the weight. More concretely, for the question *Why won't the can change the taste of beer?*, our model generates answer keywords from two candidate answers: *The can keeps out light so contents never become tainted.* in Fig. 8a and *You can change the taste of beer by strong light.* in Fig. 8b. Stopwords (grey areas in Fig. 8) are first removed from both the question and answers. Obviously, answer (a) is of a higher quality than answer (b). However, from the viewpoint of exact matching, answer (b) may receive a higher matching degree with the question because it shares more keywords than with answer (a), although most of the shared keywords are meaningless. To address this problem, the POS channel gives different weights to each type of POS (part of speech) shown in Table II. In the answer sentences, the content words like *light* convey more information than verbs like *become* and prepositions like *out*. Accordingly, we define the order of weights of POS as follows: nouns $>$ verbs, and adjectives $>$ adverbs $>$ prepositions $>$ other POS.

Different weights in the POS channel not only affect the filtering of words but play an important role in distinguishing polysemous words. The word *can* appears in both answers but with different POSs and meanings. In answer (a), *can* is a noun, while it is a modal in answer (b). If POS tags are not considered, the model would treat them as the same and give them the same weights (without differing noun from modal). In the two sentences, in fact, *can* is more important as a noun in answer (a) than as a modal in answer (b). Therefore, the answer keywords generated for this question should contain the word *can*, which is a noun rather than a modal.

		Why	won't	the	can	change	the	taste	of	beer
		NN			VB			NN		NN
	The									
NN	can				1.0	0.9		1.0		1.0
VB	keeps				0.9	0.8		0.9		0.9
RB	out				0.8	0.7		0.8		0.8
NN	light				1.0	0.9		1.0		1.0
	so									
NNS	contents				1.0	0.9		1.0		1.0
RB	never				0.8	0.7		0.8		0.8
VB	become				0.9	0.8		0.9		0.9
JJ	tainted				0.9	0.8		0.9		0.9

(a) Good answer.

		Why	won't	the	can	change	the	taste	of	beer
		NN			VB			NN		NN
	You									
	can									
VB	change				0.9	0.8		0.9		0.9
	the									
NN	taste				1.0	0.9		1.0		1.0
	of									
NN	beer				1.0	0.9		1.0		1.0
	by									
JJ	strong				0.9	0.8		0.9		0.9
NN	light				1.0	0.9		1.0		1.0

(b) Bad answer.

Fig. 8: This result shows more valuable words can be selected by using the POS channel. Stopwords are first removed. The values and corresponding colors in cells represent the different POS weights of the answer words. In this example, the question is *Why won't the can change the taste of beer?*. A good answer (a) and a bad answer (b) are given to this question.

V. Conclusion and Future Work

This paper introduces a novel model, multi-aspect Gamma-Poisson matrix completion (MAGIC), to automatically generate answer keywords for existing and new questions in topic-specific community question answering. To the best of our knowledge, no existing work has incorporated hierarchical Poisson factorization into CQA to address Q/A sparsity, Q/A tags as metadata, and the answer scores rated by users to learn Q/A relations and similarity. We show the Q/A tagging information further greatly improves the prediction accuracy of the Poisson factorization model and the integrity and diversity of answer keywords recommended by our model.

This work only focuses on the keyword generation for CQA for two-fold reasons. On one hand, keywords convey the main semantic information and thus play an important role in a sentence. On the other hand, in CQA, two different questions are less likely to share exactly the same answer unless they are redundant. While our model shows the modeling flexibility and addresses the 'cold start' problem in matrix factorization models, it does not generate a complete answer for a given or new question, which will be our further work.

ACKNOWLEDGMENT

This work is partially sponsored by the Australian Research Council Discovery grant DP190101079.

REFERENCES

[1] X. Cheng, S. Zhu, S. Su, and G. Chen, "A multi-objective optimization approach for question routing in community question answering services," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 9, pp. 1779–1792, 2017.

[2] A. Azzam, N. Tazi, and A. Hossny, "Text-based question routing for question answering communities via deep learning," in *Proceedings of the Symposium on Applied Computing*. ACM, 2017, pp. 1674–1678.

[3] X. Zhou, B. Hu, Q. Chen, and X. Wang, "Recurrent convolutional neural network for answer selection in community question answering," *Neurocomputing*, vol. 274, pp. 8–18, 2018.

[4] L. Nie, X. Wei, D. Zhang, X. Wang, Z. Gao, and Y. Yang, "Data-driven answer selection in community QA systems." *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 6, pp. 1186–1198, 2017.

[5] X. Yang, M. Wang, W. Wang, M. Khabsa, and A. Awadallah, "Adversarial training for community question answer selection based on multi-scale matching," in *AAAI*, 2019.

[6] C. Tao, W. Wu, C. Xu, Y. Feng, D. Zhao, and R. Yan, "Improving matching models with contextualized word representations for multi-turn response selection in retrieval-based chatbots," *arXiv preprint arXiv:1808.07244*, 2018.

[7] L. Cao, "Coupling learning of complex interactions," *Information Processing & Management*, vol. 51, no. 2, pp. 167–186, 2015.

[8] Z. Ji, F. Xu, B. Wang, and B. He, "Question-answer topic model for question retrieval in community question answering," in *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 2012, pp. 2471–2474.

[9] G. Zhou, T. He, J. Zhao, and P. Hu, "Learning continuous word embedding with metadata for question retrieval in community question answering," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, vol. 1, 2015, pp. 250–259.

[10] J. Jeon, W. B. Croft, and J. H. Lee, "Finding similar questions in large question and answer archives," in *Proceedings of the 14th ACM international conference on Information and knowledge management*. ACM, 2005, pp. 84–90.

[11] K. Zhang, W. Wu, H. Wu, Z. Li, and M. Zhou, "Question retrieval with high quality answers in community question answering," in *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*. ACM, 2014, pp. 371–380.

[12] Q. H. Tran, V. Tran, T. Vu, M. Nguyen, and S. B. Pham, "JAIST: Combining multiple features for answer selection in community question answering," in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 2015, pp. 215–219.

[13] X. Zhang, S. Li, L. Sha, and H. Wang, "Attentive interactive neural networks for answer selection in community question answering," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[14] Y. Belinkov, M. Mohtarami, S. Cyphers, and J. Glass, "VectorSLU: A continuous word vector approach to answer selection in community question answering systems," in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 2015, pp. 282–287.

[15] Y. Shen, W. Rong, Z. Sun, Y. Ouyang, and Z. Xiong, "Question/answer matching for CQA system via combining lexical and sequential information," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[16] B. Li and I. King, "Routing questions to appropriate answerers in community question answering services," in *Proceedings of the 19th ACM international conference on Information and knowledge management*. ACM, 2010, pp. 1585–1588.

[17] F. Riahi, Z. Zolaktaf, M. Shafiei, and E. Milios, "Finding expert users in community question answering," in *Proceedings of the 21st International Conference on World Wide Web*. ACM, 2012, pp. 791–798.

[18] S. Chang and A. Pal, "Routing questions for collaborative answering in community question answering," in *2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013)*. IEEE, 2013, pp. 494–501.

[19] I. Srba, M. Grznar, and M. Bielikova, "Utilizing non-QA data to improve questions routing for users with low QA activity in CQA," in *Proceedings of the 2015 IEEE/ACM international conference on advances in social networks analysis and mining 2015*. ACM, 2015, pp. 129–136.

[20] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, p. 788, 1999.

[21] Y. Chen, H. Zhang, R. Liu, Z. Ye, and J. Lin, "Experimental explorations on short text topic mining between LDA and NMF based schemes," *Knowledge-Based Systems*, vol. 163, pp. 1–13, 2019.

[22] Y. Wu, W. Wu, Z. Li, and M. Zhou, "Mining query subtopics from questions in community question answering." in *AAAI*, 2015, pp. 339–345.

[23] X. Gong, F. Wang, and L. Huang, "Weighted NMF-based multiple sparse views clustering for web items," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2017, pp. 416–428.

[24] P. Gopalan, J. M. Hofman, and D. M. Blei, "Scalable recommendation with hierarchical poisson factorization." in *UAI*, 2015, pp. 326–335.

[25] P. K. Gopalan, L. Charlin, and D. Blei, "Content-based recommendations with poisson factorization," in *Advances in Neural Information Processing Systems*, 2014, pp. 3176–3184.

[26] P. Gopalan, F. J. Ruiz, R. Ranganath, and D. Blei, "Bayesian nonparametric poisson factorization for recommendation systems," in *Artificial Intelligence and Statistics*, 2014, pp. 275–283.

[27] D. Liang, J. W. Paisley, D. Ellis *et al.*, "Codebook-based scalable music tagging with poisson matrix factorization." in *ISMIR*, 2014, pp. 167–172.

[28] T. D. T. Do and L. Cao, "Metadata-dependent infinite poisson factorization for efficiently modelling sparse and large matrices in recommendation." in *IJCAI*, 2018, pp. 5010–5016.

[29] ——, "Coupled poisson factorization integrated with user/item metadata for modeling popular and sparse ratings in scalable recommendation," *AAAI2018*, pp. 1–7, 2018.

[30] ——, "Gamma-poisson dynamic matrix factorization embedded with metadata influence," in *Advances in Neural Information Processing Systems*, 2018, pp. 5829–5840.

[31] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Advances in neural information processing systems*, 2008, pp. 1257–1264.

[32] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Foundations Trends in Machine Learning*, vol. 1, no. 12, pp. 1–305, 2008.

[33] H. Chen, F. X. Han, D. Niu, D. Liu, K. Lai, C. Wu, and Y. Xu, "Mix: Multi-channel information crossing for text matching," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 110–119.

[34] G. Salton and C. T. Yu, "On the construction of effective vocabularies for information retrieval," *Acm Sigplan Notices*, vol. 10, no. 1, pp. 48–60.

[35] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, "Stochastic variational inference," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 1303–1347, 2013.

Qing Liu is a PhD student at the Faculty of Engineering and Information Technology, University of Technology Sydney, Australia. Her research interests include Natural Language Processing, statistical models, deep Bayesian networks, in addition to general interest in data science especially machine learning.

Trong Dinh Thac Do is a Research Fellow at the Faculty of Engineering and IT, University of Technology Sydney, Australia. He has got a PhD degree in Machine Learning and Artificial Intelligence. His research interests include statistical models, machine learning, Bayesian networks, and deep learning.

Longbing Cao has got one PhD in Intelligent Sciences and another in Computing Sciences. He is a professor at the University of Technology Sydney. His major research interests include data science, machine learning, data mining, artificial intelligence, behavior informatics, and their enterprise applications.