Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence

Mining Query Subtopics from Questions in Community Question Answering

Yu Wu,[†*] **Wei Wu,**[‡] **Zhoujun Li,**[†] **Ming Zhou**[‡]
[†]State Key Lab of Software Development Environment, Beihang University, Beijing, China
[‡]Microsoft Research, Beijing, China
{wuyu,lizj}@buaa.edu.cn {wuwei,mingzhou}@microsoft.com

Abstract

This paper proposes mining query subtopics from questions in community question answering (CQA). The subtopics are represented as a number of clusters of questions with keywords summarizing the clusters. The task is unique in that the subtopics from questions can not only facilitate user browsing in CQA search, but also describe aspects of queries from a question-answering perspective. The challenges of the task include how to group semantically similar questions and how to find keywords capable of summarizing the clusters. We formulate the subtopic mining task as a non-negative matrix factorization (NMF) problem and further extend the model of NMF to incorporate question similarity estimated from metadata of CQA into learning. Compared with existing methods, our method can jointly optimize question clustering and keyword extraction and encourage the former task to enhance the latter. Experimental results on large scale real world CQA datasets show that the proposed method significantly outperforms the existing methods in terms of keyword extraction, while achieving a comparable performance to the state-of-the-art methods for question clustering.

Introduction

Community question answering (CQA) has become a popular platform for people to share their knowledge and learn from each other. Large CQA portals like Yahoo! Answers and Baidu Knows have integrated the functions of search engines and social media. They not only allow their users to search and browse existing content, but also encourage them to share and interact with others. Recently, Quora and Zhihu have emerged as a new type of CQA service. These web sites further enhance the social features of Yahoo! Answers and Baidu Knows with social links between users.

With the flourishing of CQA service, researchers have studied how to leverage the content in CQA to fulfill users' information needs. A typical way is to let users search existing questions. A lot of work has been done along this line (Jeon, Croft, and Lee 2005; Xue, Jeon, and Croft 2008; Cao et al. 2009). Although existing work mainly focuses on handling long question queries, recently, it was reported that many users are still used to issuing short queries on CQA portals. For example, in a one-day search log of Yahoo! Answers, 24% of queries were shorter than 4 words and were incomplete sentences (Wu et al. 2014). In contrast to long question queries, short queries are usually ambiguous and multi-faceted. Users who issue short queries may either have difficulty in formulating a full-fledged question or just want to browse some relevant questions and learn related knowledge. Therefore, for these queries, rather than throwing a bulk of questions with mixed topics to users, it is better to organize the returned questions into different clusters and summarize each cluster with one or several keywords. The clusters and keywords represent subtopics of a short query and can quickly guide users to what they are looking for.

In addition to facilitating browsing in question search, subtopics from questions also reflect users' informational intent and thus describe queries from a question-answer perspective. The information is particularly useful for navigational queries and transactional queries in web search. For example, "twitter" is a navigational query, and the top subtopics mined from query logs and top search results are "twitter.com," "login," "news," etc. However, even for these queries, people sometimes still want to seek useful information and learn knowledge about a specific aspect of the queries. Currently, they usually post questions in CQA portals and wait for others' help. Therefore, CQA is a valuable resource for capturing users' informational intent. From questions in Yahoo! Answers, we clearly observed subtopics for query "twitter" such as "technology," "celebrities," and "search." These subtopics clarify the aspects people would like to ask concerning "twitter" and are valuable for query suggestion and question recommendation in CQA search. Table 1 gives more examples for comparing subtopics mined from CQA, query logs, and top search results.

We propose mining query subtopics from questions in community question answering (CQA). The subtopics are represented as a number of clusters of questions with keywords summarizing the clusters. As far as we know, we are the first to study the problem of query subtopic mining in CQA. The task of subtopic mining consists of three subtasks: question retrieval, question clustering, and keyword extraction. For question retrieval, we employ the method proposed by Wu et al. (2014), which represents the state-of-the-art method in question retrieval for short queries.

[*]The work was done when the first author was an intern in Microsoft Research Asia.

Query: **Math**
CQA: Algorithm, Algebra, Probability
Query log: Game, Play ground, definition
Web Result: Online, Games, practice
Query: **New York**
CQA: Football, Restruant, Shopping
Query log: city, news, company
Web Result: City, News, hotel
Query: **Twitter**
CQA: Technology, Celebrities, Twitter Search
Query log: twitter.com, api, widget
Web Result: login, account, news
Query: **Steve Jobs**
CQA: Design, ipad, Leadership
Query log: Cancer, Biography, Quotes
Web Result: Biography, News, Ceo

Table 1: Subtopics from different sources

The challenges include how to group semantically similar questions and how to find keywords capable of summarizing the clusters. Existing methods for subtopic mining from query log and search results usually conduct clustering and keyword extraction in separate steps (Zeng et al. 2004; Wang et al. 2013). We propose formulating query subtopic mining as a non-negative matrix factorization (NMF) problem and further extend the model of NMF to incorporate question similarity into learning. Thus, question clustering and keyword extraction are conducted in a unified framework by simultaneously factorizing a tf-idf matrix and a question similarity matrix. Compared with existing methods, our method jointly optimizes question clustering and keyword extraction. The good clustering performance can enhance the performance of keyword extraction. We derive a projected gradient descent based algorithm that can efficiently solve the optimization problem and propose a heuristic but effective method that can estimate the number of subtopics from data for practical application.

We implemented our method with question similarity estimated using answers and categories in CQA and tested its performance on Quora and Zhihu data. We set up experiments on Quora and Zhihu because questions have rich topics annotated by users and thus we can easily build up large scale evaluation data sets without expensive and exhausting human annotation. We evaluated different subtopic mining methods on both question clustering and keyword extraction. The experimental results show that our method can significantly outperform existing methods for keyword extraction, while achieving a comparable performance with the state-of-the-art methods of question clustering.

Our contributions in this paper are three-fold: 1) proposal of mining query subtopics from questions in CQA; 2) formulation of the task as an NMF optimization problem, derivation of an efficient algorithm, and implementation with metadata in CQA; 3) empirical verification of the efficacy of the method on large scale CQA data.

Related Work

Existing research on identifying query subtopics can be categorized into two groups: mining from query logs (Beeferman and Berger 2000; Wen, Nie, and Zhang 2001; Craswell and Szummer 2007; Hu et al. 2012), and mining from top web search results (Zeng et al. 2004; Wang and Zhai 2007; Wang et al. 2013). In the former group, a lot of methods perform query clustering on a click-through bipartite graph and the clusters represent aspects of queries. For example, Hu et al. (2012) proposed clustering URLs and keywords that are either the prefix or suffix of a query. The clustering was conducted with similarity extracted from a click-through bipartite graph. The keywords together with the URLs in a cluster were taken as subtopics of a query. Instead of analyzing query logs, the latter group extracts aspects of queries from plain text. For example, Zeng et al. (2004) formalized search result clustering as a salient phrase ranking problem. Documents that share the same salient phrase were grouped together. The salient phrases associated with documents represent subtopics of a query. Recently, Wang et al. (2013) proposed clustering top search results and then extracting key phrases in the clusters as subtopics of queries. Our method is unique in that 1) it extracts subtopics from questions, and thus explains queries from a question-answering perspective; 2) it studies how to leverage the metadata of CQA to enhance clustering and keyword extraction of questions.

In this paper, we have formulated the query subtopic mining task as a non-negative matrix factorization (NMF) problem. The model of NMF, due to its good interpretability and superior performance, has been applied to many tasks like document clustering (Xu, Liu, and Gong 2003), document classification (Zhu et al. 2007), multi-document summarization (Wang et al. 2008), and subgroup identification (Mandayam-Comar, Tan, and Jain 2010). Various extensions such as GNMF (Cai et al. 2011) and symmetric NMF (Kuang, Park, and Ding 2012) were also developed and have proven to be effective in document clustering. As far as we know, our method is the first application of NMF to the subtopic mining task. By doing so, we jointly optimize question clustering and keyword extraction. To leverage metadata in CQA portals, we also extend the traditional NMF by simultaneously factorizing a tf-idf matrix and a question similarity matrix. Our model is similar to the one proposed by Zhu et al. (2007). The difference is that they perform classification and NMF is only used to learn document feature vectors for a classifier. We perform query subtopic mining, and both document representation and word representation are learnt through NMF and evaluated in question clustering and keyword extraction.

A Matrix Factorization Approach for Query Subtopic Mining

In this section, we elaborate our approach for mining query subtopics from questions in CQA. The task includes three subtasks: question retrieval for short queries, question clustering, and keyword extraction. For the first subtask, we employ the method proposed by Wu et al. (2014) as it represents the state-of-the-art method for short query question search. For the other two subtasks, one solution is to follow an existing method for subtopic mining from search results,

and conduct question clustering and keyword extraction in separate steps. The problem is that the correlation between the two subtasks is ignored. Intuitively, question clustering and keyword extraction should be coupled, and good clustering can enhance the performance of keyword extraction.

We have considered modeling question clustering and keyword extraction in a unified framework. Specifically, we formulate the two subtasks as a non-negative matrix factorization problem. The method simultaneously embeds questions and words into a vector space, and elements of the vectors indicate groups of questions and representative words of the groups. In this way, question clustering and keyword extraction are jointly optimized and the two subtasks interact with each other in the learning process. To leverage the meta data in CQA, we further extend the basic NMF model and incorporate question similarity estimated from the meta-data into learning. Then, vectors of questions and vectors of words are learnt by simultaneously factorizing a tf-idf matrix and a question similarity matrix.

We first introduce the formulation of our method, then we derive an efficient algorithm based on projected gradient descent (Lin 2007) to solve the optimization problem.

Problem Formulation

Suppose that we have a question set $\mathbb{Q}$ that is a subset of a CQA web site. Given a query q, we have some questions $\mathcal{Q} = \{Q_1, Q_2, \ldots, Q_n\}$ returned by the retrieval model. Suppose that there are k subtopics of q implied by $\mathcal{Q}$. Our goal is to recover the subtopics by a partition $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ of $\mathcal{Q}$. Each C_i is a cluster of questions and we summarize C_i by keyword w_i. The cluster-keyword pair (C_i, w_i) represents the i-th subtopic of q. Note that in this work, we assume that each question only belongs to one subtopic and only consider using one keyword to summarize each cluster. It is easy to extend this work to involve cases that involve one question with multiple subtopics and one cluster with multiple keywords. We leave these cases for future investigations. Let $\mathcal{W} = \{w_1, w_2, \ldots, w_m\}$ denote words appearing in $\mathcal{Q}$. We assume that the subtopics correspond to a subspace of $\mathbb{R}^k$, where $k \ll \min(m, n)$, and consider learning a vector $v_i = (v_{i1}, v_{i2}, \ldots, v_{ik})$ and a vector $u_i = (u_{i1}, u_{i2}, \ldots, u_{ik})$ in $\mathbb{R}^k$ to represent question Q_i and word w_i, respectively. Thus, $\mathcal{Q}$ and $\mathcal{W}$ can be represented as $V = (v_1^\top, v_2^\top, \ldots, v_n^\top)^\top$ and $U = (u_1^\top, u_2^\top, \ldots, u_m^\top)^\top$, respectively. Each element of U and V reveals the intensity of a question and a word in a subtopic and therefore we require U and V to be non-negative. The cluster-word pair (C_j, w_j) is defined as $C_j = \{q_i \mid \max_{1 \leqslant s \leqslant k} v_{is} = j\}$ and $w_j = p_s$ where $s = \arg\max_{1 \leqslant r \leqslant m} u_{rj}, \forall\, 1 \leqslant j \leqslant k$.

We aim to learn U and V in a unified framework. To this end, we further assume that there is an $m \times n$ matrix X measuring the similarity of words and questions, and an $n \times n$ matrix D measuring the similarity of questions. A common setup of X is using tf-idf as weights. We consider recovering U and V from X and D, and expect similar question-word pairs and question-question pairs are still similar in $\mathbb{R}^k$. This leads to the following optimization problem:

Algorithm 1: Optimization Algorithm for Problem (1)

Input: $\alpha \geqslant 0$, $\gamma_0 = 1$, $0.1 < \beta < 0.5$, $0 < \epsilon \ll 1$,
$0 < \sigma < 1$, $t = 1$, T = max iteration count
Initialize $U = U_1 \geqslant 0$, $V = V_1 \geqslant 0$
repeat
  $\gamma_t = 1$; $U^{'} = [U_t - \gamma_t[\frac{\partial O}{\partial U}]_t]_+$;
  if $O(U^{'}, V_t) - O(U_t, V_t) > \sigma \cdot trace[[\frac{\partial O}{\partial U}]_t^\top \cdot (U^{'} - U_t)]$ **then**
    repeat
      $\gamma_t = \gamma_t \cdot \beta$; $U^{'} = [U_t - \gamma_t[\frac{\partial O}{\partial U}]_t]_+$;
    until $O(U^{'}, V_t) - O(U_t, V_t) \leqslant \sigma \cdot trace[[\frac{\partial O}{\partial U}]_t^\top \cdot (U^{'} - U_t)]$ ***or*** $U^{'} = U_t$;
  end
  $U_{t+1} = [U_t - \gamma_t[\frac{\partial O}{\partial U}]_t]_+$, $V^{'} = [V_t - \gamma_t[\frac{\partial O}{\partial V}]_t]_+$
  if $O(U_{t+1}, V^{'}) - O(U_{t+1}, V_t) > \sigma \cdot trace[[\frac{\partial O}{\partial V}]_t^\top \cdot (V^{'} - V_t)]$ **then**
    repeat
      $\gamma_t = \gamma_t \cdot \beta$; $V^{'} = [V_t - \gamma_t[\frac{\partial O}{\partial V}]_t]_+$;
    until $O(U_{t+1}, V^{'}) - O(U_{t+1}, V_t) \leqslant \sigma \cdot trace[[\frac{\partial O}{\partial V}]_t^\top \cdot (V^{'} - V_t)]$ ***or*** $V^{'} = V$;
  end
  $V_{t+1} = [V_t - \gamma_t[\frac{\partial O}{\partial V}]_t]_+$, $t = t + 1$;
until $t > T$ ***or*** $(|||[\frac{\partial O}{\partial U}]_t||_F < \epsilon \cdot |||[\frac{\partial O}{\partial U}]_1||_F$ ***and*** $|||[\frac{\partial O}{\partial V}]_t||_F < \epsilon \cdot |||[\frac{\partial O}{\partial V}]_1||_F)$;
Output: U_t, V_t

$$\begin{aligned} \arg\min_{U,V} \quad & ||X - UV^\top||_F^2 + \alpha||D - VV^T||_F^2, \quad (1) \\ \text{s.t.} \quad & U \geqslant 0,\ V \geqslant 0, \end{aligned}$$

where $||\cdot||_F$ is the Frobenius norm of a matrix, and $\alpha \geqslant 0$ acts as a trade-off between the two factorization items.

In Problem (1), question clustering and keyword extraction are performed in a unified framework by simultaneously learning V and U. Ideally, we expect to learn a good representation of questions from D and X and let the representation of questions enhance the learning of word representation by the minimization of $||X - UV^\top||_F^2$. This idea is verified in our experiments.

Problem (1) is a natural extension of the existing non-negative matrix factorization (NMF) approaches. Specifically, when $\alpha = 0$, our method degenerates to the standard NMF method (Xu, Liu, and Gong 2003), while when α goes to $+\infty$, it becomes the symmetric NMF method recently proposed by Kuang et al. (2012). Therefore, we actually apply the idea of NMF to subtopic mining and, to the best of our knowledge, we are the first to formulate the problem of subtopic mining as a matrix factorization problem.

Algorithm

The object function (1) is non-convex with respect to both V and U. Therefore, it is difficult to find a global minimum. To solve this problem, we derive an efficient algorithm based on the projected gradient descent framework in (Lin 2007). The details are described in Algorithm 1.

In Algorithm 1, we denote the objective function of (1) as $O(U, V)$, and the gradients of U and V in the t-th iteration are given by

$$[\frac{\partial O}{\partial U}]_t = -2XV_t + 2U_tV_t^\top V_t$$

$$[\frac{\partial O}{\partial V}]_t = -2X^\top U_{t+1} + 2V_tU_{t+1}^\top U_{t+1} - 4\alpha DV_t + 4\alpha V_tV_t^\top V_t,$$

where U_t and U_{t+1} represent matrix U in the t-th and $t+1$-th iteration, and V_t represents V in the t-th iteration. $[\cdot]_+$ is a projection function that forces the entries of U and V to be non-negative by shrinking the negative ones to zero.

The algorithm iteratively optimizes U and V by stepping toward the negative of the gradients and projecting the updates to the non-negative cone. Since k is the number of subtopics and much smaller than the number of words m and the number of questions n, the time complexity is $O(nkm)$ for updating U, and $O(nkm + n^2k)$ for updating V. Thus, the overall time complexity of Algorithm 1 is no more than $O(\max(m, n)^2k)$.

Experiment

We conducted experiments to test the proposed method on question clustering and keyword extraction.

Data Sets

To avoid expensive and exhausting human annotation, we propose automatically generating evaluation data from questions with topics edited by users in CQA portals. Specifically, we have considered Quora [1] and Zhihu [2], because questions on these sites have key phrases as topics edited by users. We first crawled $600,000$ questions from Quora and $60,000$ questions from Zhihu, and separately indexed the two data sets. Then, we collected 2000 English queries and 1000 Chinese queries from query logs of a commercial search engine, and employed the model proposed in (Wu et al. 2014) to get the top 1000 questions for each query. After that, we collected all the topics in the returned results and organized questions into the topics. For the questions with multiple topics, we randomly picked one topic. In this manner, we obtained several question groups so that each group has a topic. The question groups associated with the topics represent subtopics of the queries. Since all baseline methods, including our method, can only learn subtopics from question texts, we removed the groups in which none of the words in the topic appear in the questions. To further filter noise in the topics, we removed groups with less than 5 questions for Quora data and groups with less than 3 questions for Zhihu data. Queries with less than 3 groups of questions were also removed. After pre-processing, we obtained 1287 queries for Quora data and 713 queries for Zhihu data. On average, each query in Quora had 7.01 topics, and each topic had 16.49 questions. The numbers for Zhihu data were 7.00 and 6.52, respectively. 64% topics in Quora and 68% topics in Zhihu consisted of a single word or a single word plus the query. Therefore, we only considered extracting keywords as subtopics. If a topic consisted of more than one word (except the query), we checked if the extracted keyword was contained by the topic. This can reduce the complexity of the experiments and ensures that the comparisons are fair for all the methods. For Quora data, after stemming and removing stop words, on average there were 431.38 words for each query, while for Zhihu data, after conducting Chinese word segmentation, on average there were 276.19 words for each query.

[1] https://www.quora.com/

[2] http://www.zhihu.com/

Implementation of Question Similarity

To implement our method, we need to specify matrices X and D in (1). We set X a matrix with word tf-idf as weights, and normalized each column of X to norm 1. For question similarity, we considered three features: 1) cosine of tf-idf vectors of questions. This might generate some overlap between X and D, but has been proven effective in the experiments; 2) Question-answer topic model proposed in (Ji et al. 2012). We trained the topic models using 1 million question-answer pairs collected from Yahoo! Answers and Baidu Knows, and calculated question-question similarity in a topic space; 3) Explicit semantic analysis (ESA) proposed in (Gabrilovich and Markovitch 2007). In ESA, texts are explicitly represented as weighted vectors of predetermined concepts from Wikipedia and the cosine of the vectors is calculated as the similarity of text fragments. Inspired by ESA, we leveraged the predefined categories of CQA sites to calculate question similarity in Quora and Zhihu. In fact, many CQA sites have category systems (e.g., Yahoo! Answers, see https://answers.yahoo.com/dir/index) with categories that are similar to the concepts of Wikipedia. We took the root categories of Yahoo! Answers for Quora data and root categories of Baidu Knows for Zhihu data as concepts, and trained naive Bayes classifiers to obtain the vector representations of questions. We considered the category systems of Yahoo! Answers and Baidu Knows, because they are the largest English and Chinese CQA sites, respectively. Question similarity then was calculated as the cosine of the vectors of categories. The three features were linearly combined by learning a Ranking SVM model (Herbrich, Graepel, and Obermayer 1999) from extra training data created using another 2000 English queries and 1000 Chinese queries. Questions in the training data were grouped in the same way as the evaluation data, and those in the same group were regarded as being similar.

We created D by combining the three features. By doing so, we actually incorporated the information of answers and categories in CQA into learning. We also observed that D is dense but most of its elements have small values. Therefore, to filter noise and enhance efficiency, we further shrank the elements of D by considering their nearest neighbors. Formally, D was defined as

$$D_{ij} = \begin{cases} 1 & \text{if } q_i \in NN_p(q_j) \text{ or } q_j \in NN_p(q_i) \\ 0 & \text{other,} \end{cases} \quad (2)$$

where D_{ij} is the element of D in the i-th row and j-th column, and $NN_p(q_i)$ means the p nearest neighbors of question q_i found by the similarity function. p is a parameter that needs tuning.

Evaluation Metrics

For question clustering, we followed the example of most document clustering methods and employed accuracy (AC) as an evaluation measure. Given a question q_i, suppose that the cluster label in the evaluation data is a_i, and the predicted cluster label by the algorithm is l_i, then AC is calculated as follows:

$$AC = \frac{\sum_{i=1}^{n} \sigma(a_i, map(l_i))}{n}, \quad (3)$$

where n is the total number of questions under a query. $\sigma(x, y) = 1$ if $x = y$, otherwise it is equal to 0. $map(l_i)$ is a function that maps every predicted label l_i to an optimal label in the evaluation data by the best match theory (Plummer and Lovász 1986).

For keyword extraction, if the extracted keyword exactly matched the topic or it was contained by the topic (words in queries and stop words were excluded), we judged the extraction correct. Then, we calculated the ratio of correct extractions and averaged it over queries to evaluate keyword extraction methods.

Baselines

We considered two groups of baselines: 1) first conducting question clustering, and then extracting a keyword for each cluster; An example is the method proposed by Wang et al. (2013); 2) first extracting keywords from the whole question set, and then organizing questions into clusters based on the keywords. A representative is the method proposed by Zeng et al. (2004).

First clustering, and then keyword extraction: In addition to the hierarchical agglomerative clustering (HAC) method in (Wang et al. 2013), we also implemented kernel k-means (Dhillon, Guan, and Kulis 2004), spectral clustering (Ng, Jordan, and Weiss 2001), NMF (Xu, Liu, and Gong 2003), GNMF (Cai et al. 2011), and symmetric NMF (SymNMF) (Kuang, Park, and Ding 2012) as clustering methods. Based on the clustering results, we implemented typical keyword extraction methods, including TFIDF, TextRank (Mihalcea and Tarau 2004) with window size 2, and the recently proposed topical PageRank (TPR) (Liu et al. 2010). In TPR, we used the question-answer topic model trained from the 1 million CQA dataset to estimate the topic distributions of words.

First keyword extraction, and then clustering: we implemented the method proposed by Zeng et al. (2004). In our experiments, we trained a keyword ranker using the data with which we learned the question similarity. To make a fair comparison, we further clustered the questions that could not be grouped by the method of Zeng et al. by calculating the similarity of the questions with the centroid of each cluster and assigning the questions to the most similar cluster.

Evaluation Results

With both Quora and Zhihu data, we randomly set initializations for kernel k-means, spectral clustering, NMF, GNMF, SymNMF, and our method 5 times, and reported the average results of the 5 runs. For our method, we followed existing methods (Cai et al. 2011; Kuang, Park, and Ding 2012) and implemented Algorithm 1 with $\beta = 0.1$, $\sigma = 0.01$, $\epsilon = 10^{-4}$, and $T = 200$. α in Equation (1) and the number of nearest neighbors p for D needed tuning. Therefore, we randomly split the evaluation data into a validation set and a test set with a ratio of $1 : 3$, and selected α from $\{0.1, 1, 10, 100\}$ and p from 1-10 of the validation set. The best choice for α was 100 with both the Quora and Zhihu data, while with the Quora data the best p was 7, and with the Zhihu data the best p was 4. With these settings, we compared different methods on the test set.

	Quora	Zhihu
The method of Zeng et al. (2004)	0.383	0.410
Kernel k-means	0.516	0.410
HAC	0.434	0.488
Spectral clustering	**0.602**	**0.586**
NMF	0.520	0.519
GNMF	0.579	0.564
SymNMF	**0.606**	**0.592**
Our method	**0.607**	**0.594**

Table 2: Evaluation on question clustering

	Quora	Zhihu
SymNMF+TFIDF	0.369	0.305
SymNMF+TextRank	0.312	0.257
SymNMF+TPR	0.229	0.162
Our method+TFIDF	0.370	0.307
Our method+TextRank	0.311	0.257
Our method+TPR	0.232	0.167
The method of Zeng et al. (2004)	0.380	0.279
Our method	**0.398**	**0.318**

Table 3: Evaluation on keyword extraction

Table 2 shows the evaluation results of question clustering. Our method, SymNMF, and spectral clustering performed comparably well on both data sets, and the three methods significantly outperformed the other clustering methods (t-test with p value < 0.01). It is not surprising that the proposed method is comparable with SymNMF, as they leverage the same information in a similar way for clustering. The difference is that our method can also leverage the good performance of clustering to enhance the recognition of keywords, as will be seen later. The method from Zeng et al. (2004) organized questions based on the common keywords they share. In practice, however, questions under the same subtopic do not necessarily share common words. That is why the method performs badly on question clustering, even though we have conducted post-processing as a remedy.

We took SymNMF as the best performing clustering method and combined it with different keyword extraction methods. Table 3 shows the evaluation results on keyword extraction. We conducted a statistical test (t-test), and the results show that our method significantly outperformed all

other methods (p value < 0.01). In the first group of baselines, although SymNMF is powerful at grouping questions, question clustering and keyword extraction have to be optimized separately. The power of SymNMF only weakly influenced keyword extraction. On the other hand, we checked the learnt U and V in (1), and found that our method not only grouped semantically similar questions together through V, but also weighted the questions in the group according to their topic concentration. The questions that are more topically concentrated had larger weights, and these questions further highlighted the salient words they contain. This explains why our method performs significantly better on keyword extraction than SymNMF plus a bunch of state-of-the-art keyword extraction methods, even though they achieved comparable performance on clustering. The advantage of joint optimization of question clustering and keyword extraction was further verified by the performance drop when only treating our method as a clustering method and combining it with other keyword extraction methods. Topical PageRank performs worse than TFIDF and the TextRank. This may stem from the noise in CQA data for learning the topic space. The method proposed by Zeng et al. (2004) learns a supervised ranker for keyword extraction, but ignores the useful local information from clusters. Therefore, it is worse than our method.

Discussions

Besides the comparison with existing methods, there are some other problems worth investigation. First, in addition to performing clustering, NMF and GNMF can also jointly optimize question clustering and keyword extraction. Therefore, it is interesting to check if our method can outperform these two models when all of them perform joint optimization. Second, to filter noise, we shrank the values of D and used a sparse matrix in the experiments. The preprocessing can enhance efficiency, but we do not know if the sparse D is better than the original dense D in terms of performance. Finally, we leveraged three features to construct D. A natural question is how these features contributed to the final results. Table 4 reports some of the comparison results, which is an attempt to answer the questions above. From the fourth row, the results represent the comparison of the proposed method with D using partial features and all features. Note that to highlight the differences, we reported the percent deviation of different methods from our method on the two tasks. From the table, you can see 1) our extension to NMF brought performance improvement on both question clustering and keyword extraction. This demonstrates that D is useful for both question clustering and keyword extraction. Moreover, although GNMF and our method (1) leveraged the same information, the performance difference verified that our extension is more effective for the task of subtopic mining. 2) the decrease in D with nearest neighbors can really make the learning method robust to noise. 3) with the similarity estimated from the answers and categories of CQA, the performance on clustering improved. This is reasonable since the two features can capture the semantic similarity of questions, and can group the questions from the same topic but that share few common words.

	clustering		**keyword**	
	Quora	Zhihu	Quora	Zhihu
NMF	-8.7	-7.5	-3.6	-4.7
GNMF	-2.8	-2.7	-1.5	-2.9
Dense D	-4.0	-3.7	-3.4	-4.5
TFIDF	-3.2	-2.9	+1.3	+1.1
TFIDF+Topic Model	-1.2	-1.4	+0.5	+0.5
TFIDF+Category	-0.9	-1.2	+0.6	+0.4

Table 4: Discussions on the implementation of our method

Algorithm 2: A heuristic method for selecting k

Input: $k_0 = 1$, $k_1 = \sqrt{n/2}$, t,$i = 1$
$d_0 = InnerDis(k_0)$;$d_1 = InnerDis(k_1)$;
if $d_0/d_1 < t$ **then**
  repeat
    $i = i + 1$; $k_i = k_{i-1} + 1$; $d_i = Inner(k_i)$;
  until $d_0/d_i > t$;
end
else
  repeat
    $i = i + 1$; $k_i = k_{i-1} - 1$; $d_i = Inner(k_i)$
  until $d_0/d_i < t$;
end
Output: $k = k_i$

On the other hand, with these questions, noise for keyword recognition was also introduced, which is why the more features we used for constructing D, the worse results we got for keyword extraction.

In practice, unlike the experiments, the number of subtopics k is usually unknown. To make our method applicable, we propose a heuristic method to estimate k. Formally, we define the distance between two questions Q_i and Q_j as $Distance(i, j) = 1 - Sim(i, j)$, where $Sim(i, j)$ is the similarity of Q_i and Q_j calculated using the linear combination of the three features. Then, suppose that there are k groups of questions, we define the inner group distance as $InnerDis(k) = \frac{1}{k}\sum_{1 \leqslant l \leqslant k} \frac{1}{n_c^2} \sum_{Q_i, Q_j \in C_l} Distance(i, j)$, where C_l is the l-th group and n_c denotes the number of questions in C_l. We propose an iterative method to obtain the optimal k. Specifically, for each query, we suppose that $k_0 = 1$ at the beginning. In other words, all questions belong to one cluster. We select a k by making $InnerDis(k_0)/InnerDis(k)$ close to a threshold t. If the ratio is too large, then we reduce k, otherwise we enlarge k. The threshold can be obtained by asking human labelers to manually group questions for a small query set and averaging $InnerDis(k_0)/InnerDis(k_q)$ over queries, where k_q is the number of question groups for query q decided by humans. Algorithm 2 summarizes the method.

To test our idea, we estimated the threshold t using the validation set (it is 1.15) and checked how the automatically selected k is different from the one obtained based on human annotated topics on the test set. Table 5 reports the results, in which Avg is the average of absolute deviation of k over

absolute dev	Avg	0	1	2	$\geqslant 3$
Quora	1.89	15.9%	36.4%	22.4%	25.3%
Zhihu	2.32	10.6%	29.4%	23.7%	36.3%

Table 5: Performance of k-selection algorithm

queries, and the other columns give the ratio of queries that have the corresponding absolute deviation of k. We can see that on more than 70% of queries from Quora and 60% of queries from Zhihu, the absolute deviation is no more than 2. The results demonstrate the effectiveness of the heuristic method.

Conclusion

We propose mining query subtopics from questions in CQA. Unlike existing methods, we apply the techniques of NMF to the task and perform joint optimization of question clustering and keyword extraction. Experiments on large-scale Quora data and Zhihu data show that our method can significantly outperform existing methods for keyword extraction, while achieving a comparable performance with the state-of-the-art methods for question clustering.

Acknowledgment

This work was supported by NSFC (Grand Nos. 61170189, 61370126, 61202239), the Research Fund for the Doctoral Program of Higher Education (Grand No. 20111102130003), the Fund of the State Key Laboratory of Software Development Environment (Grand No. SKLSDE-2013ZX-19), and Microsoft Research Asia Fund (Grand No. FY14-RES-OPP-105).

References

Beeferman, D., and Berger, A. 2000. Agglomerative clustering of a search engine query log. In *CIKM'00*, 407–416.

Cai, D.; He, X.; Han, J.; and Huang, T. S. 2011. Graph regularized nonnegative matrix factorization for data representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 33(8):1548–1560.

Cao, X.; Cong, G.; Cui, B.; Jensen, C. S.; and Zhang, C. 2009. The use of categorization information in language models for question retrieval. In *CIKM'09*, 265–274. ACM.

Craswell, N., and Szummer, M. 2007. Random walks on the click graph. In *SIGIR'07*, 239–246.

Dhillon, I. S.; Guan, Y.; and Kulis, B. 2004. Kernel k-means: spectral clustering and normalized cuts. In *SIGKDD'04*, 551–556. ACM.

Gabrilovich, E., and Markovitch, S. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI'07*, volume 7, 1606–1611.

Herbrich, R.; Graepel, T.; and Obermayer, K. 1999. Large margin rank boundaries for ordinal regression. *NIPS'99* 115–132.

Hu, Y.; nan Qian, Y.; Li, H.; Jiang, D.; Pei, J.; and Zheng, Q. 2012. Mining query subtopics from search log data. In *SIGIR'12*, 305–314.

Jeon, J.; Croft, W. B.; and Lee, J. H. 2005. Finding similar questions in large question and answer archives. In *CIKM'05*, 84–90.

Ji, Z.; Xu, F.; Wang, B.; and He, B. 2012. Question-answer topic model for question retrieval in community question answering. In *CIKM'12*, 2471–2474.

Kuang, D.; Park, H.; and Ding, C. H. 2012. Symmetric nonnegative matrix factorization for graph clustering. In *SDM*, volume 12, 106–117. SIAM.

Lin, C.-J. 2007. Projected gradient methods for nonnegative matrix factorization. *Neural computation* 19(10):2756–2779.

Liu, Z.; Huang, W.; Zheng, Y.; and Sun, M. 2010. Automatic keyphrase extraction via topic decomposition. In *EMNLP'10*, 366–376.

Mandayam-Comar, P.; Tan, P.-N.; and Jain, A. K. 2010. Identifying cohesive subgroups and their correspondences in multiple related networks. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, volume 1, 476–483. IEEE.

Mihalcea, R., and Tarau, P. 2004. Textrank: Bringing order into texts. In *EMNLP'04*, volume 4, 275.

Ng, A. Y.; Jordan, M. I.; and Weiss, Y. 2001. On spectral clustering1 analysis and an algorithm. *NIPS'01* 14:849–856.

Plummer, M. D., and Lovász, L. 1986. *Matching theory*. Elsevier.

Wang, X., and Zhai, C. 2007. Learn from web search logs to organize search results. In *SIGIR'07*, 87–94.

Wang, D.; Li, T.; Zhu, S.; and Ding, C. 2008. Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, 307–314. ACM.

Wang, Q.; nan Qian, Y.; Song, R.; Dou, Z.; Zhang, F.; Sakai, T.; and Zheng, Q. 2013. Mining subtopics from text fragments for a web query. *Inf. Retr.* 16(4):484–503.

Wen, J.-R.; Nie, J.-Y.; and Zhang, H.-J. 2001. Clustering user queries of a search engine. In *WWW'01*, 162–168.

Wu, H.; Wu, W.; Zhou, M.; Chen, E.; Duan, L.; and Shum, H.-Y. 2014. Improving search relevance for short queries in community question answering. In *WSDM'14*, 43–52.

Xu, W.; Liu, X.; and Gong, Y. 2003. Document clustering based on non-negative matrix factorization. In *SIGIR'03*, 267–273.

Xue, X.; Jeon, J.; and Croft, W. B. 2008. Retrieval models for question and answer archives. In *SIGIR'08*, 475–482.

Zeng, H.-J.; He, Q.-C.; Chen, Z.; Ma, W.-Y.; and Ma, J. 2004. Learning to cluster web search results. In *SIGIR'04*, 210–217.

Zhu, S.; Yu, K.; Chi, Y.; and Gong, Y. 2007. Combining content and link for classification using matrix factorization. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 487–494. ACM.