LETTER **Communicated by Zhoujun Li**

Facet Annotation by Extending CNN with a Matching Strategy

Bei Wu
wubeibetsy@gmail.com
MOEKLINNS Lab, Department of Computer Science and Technology, Xi'an Jiaotong University, 710049, China

Bifan Wei
weibifan@xjtu.edu.cn
National Engineering Lab of Big Data Analytics, School of Continuing Education, Xi'an Jiaotong University, 710049, China

Jun Liu
liukeen@xjtu.edu.cn
Guangdong Xi'an Jiaotong University Academy, 528300, China

Zhaotong Guo
gzhtcrystal@stu.xjtu.edu.cn
MOEKLINNS Lab, Department of Computer Science and Technology, Xi'an Jiaotong University, 710049, China

Yuanhao Zheng
zhengyuanhao@stu.xjtu.edu.cn
MOEKLINNS Lab, Department of Computer Science and Technology, Xi'an Jiaotong University, 710049, China

Yihe Chen
yiko.chen@mail.utoronto.ca
University of Toronto, Toronto, Ontario, M5S 1A1, Canada

Most community question answering (CQA) websites manage plenty of question-answer pairs (QAPs) through topic-based organizations, which may not satisfy users' fine-grained search demands. Facets of topics serve as a powerful tool to navigate, refine, and group the QAPs. In this work, we propose FACM, a model to annotate QAPs with facets by extending convolution neural networks (CNNs) with a matching strategy. First, phrase information is incorporated into text representation by CNNs with different kernel sizes. Then, through a matching strategy among QAPs and facet label texts (FaLTs) acquired from Wikipedia, we generate similarity matrices to deal with the facet heterogeneity. Finally, a three-channel CNN is trained for facet label assignment of QAPs. Experiments

Neural Computation **30, 1647–1672** (2018)
doi:10.1162/neco_a_01077

on three real-world data sets show that FACM outperforms the state-of-the-art methods.

1 Introduction

Community question answering (CQA) websites like Quora and StackOverflow become popular platforms for knowledge sharing, with plenty of question-and-answer pairs (QAPs).[1] A QAP consists of a question and a corresponding answer. Most CQA websites organize questions according to topics in specific knowledge domains (Wang, Gill, Mohanlal, Zheng, & Zhao, 2013), such as Data Structure, of which Binary Tree is a typical topic. An example of Quora questions, "What is a binary tree?" is assigned to three topics, including Binary Trees, Data Structures, and Algorithms.

Topic-based organization mode for QAPs cannot satisfy the demands of fine-grained knowledge acquisition. For example, a user who wants to know the definition of *binary tree* from Quora, posts a short query, like, "Binary tree, definition." Quora will return thousands of QAPs to the user. After statistical analysis on the top 10 questions (involving 1210 QAPs), we find that only 5% of QAPs are related to the definition of *binary tree*. It is time-consuming for users to search for their expected knowledge, and they may be reluctant to explore Quora again.

Facets of topics serve as a powerful tool to navigate, refine, and group the QAPs. A facet describes an important aspect of a specific topic in a knowledge domain, such as *definition*, *property*, *application*, or *example*.

An example of a topic facet-based organization mode in Binary Tree is shown in Figure 1. Facets organize a QAP into a triple (topic, facet, QAP), such as (Binary tree, definition, "What is a binary tree? A tree with not more than two branches at any branching point".). Using the facets, we can realize the faceted navigation in CQA websites. With faceted navigation, users do not rely on formulating precise keyword searches alone to find QAPs in CQA websites. Instead, they can enter broad searches and use the facets in a flexible way to refine the initial query. This gives confidence in being comprehensive and reduces uncertainty in information seeking in general; it also removes the frustration of finding no results.

In this work, we focus on facet annotation, which is to assign one or more facet labels to a given QAP. There are two challenges of facet annotation:

- Unlike normal short texts, QAPs are bound up with topics and contain plenty of topic phrases. Taking *Red-black tree* as an example, the combination of *red*, *black*, and *tree* represents a specific topic in a knowledge domain. In this case, *red*, *black*, and *tree* do not really retain their original meanings. Existing representation methods such as

[1] https://www.quora.com/; http://stackoverflow.com/.

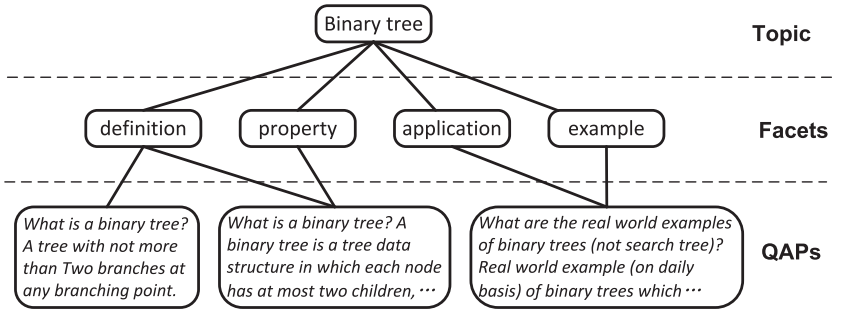


Figure 1: An example of a topic facet-based organization mode in *Binary tree*.

a bag-of-words model (Harris, 1954) and word embedding (Bengio, Ducharme, Vincent, & Jauvin, 2003) cannot effectively represent the phrase information such as topic phrases. However, it is a necessity to propose a proper representation method for the phrase information.

- The QAPs (such as the two following QAPs) describing the same facet of different topics contain notable differences in some statistic features like word distribution, which we call "facet heterogeneity" in this letter. Hence, it is necessary to acquire a model with a good generalization capability through small numbers of labeled data:
  1. Definition of Binary tree: "What is a binary tree? A tree with not more than two branches at any branching point."[2]
  2. Definition of Array: "What is an array? Array's a kind of data structure that can store a fixed-size sequential collection of elements of the same type."[3]

To overcome these challenges, we propose FACM, a model for facet annotation by extending convolutional neural network (CNN) with a matching strategy. First, an effective text representation method is developed to capture the phrase information of QAPs. The results are the input of the feature extraction pipeline. Second, a matching strategy is proposed to generate three-dimensional similarity matrices between QAPs and knowledge acquired from Wikipedia.[4] Finally, we adopt a three-channel CNN to implement the multiple facet label classification with three-dimensional similarity matrices.

There are some similarities and differences between our facet annotation task and the tagging task in data mining. Both tasks are to assign one

[2] https://www.quora.com/What-is-a-binary-tree.
[3] https://www.quora.com/What-is-an-array.
[4] https://en.wikipedia.org/.

or more labels to an object, and the purpose of assigning labels to content is to encode a higher-level abstraction. The facet label of facet annotation in our letter and the descriptors, or tags, of a tagging task in data mining are typically short textual labels. However, the tags of a tagging task usually refer to social tags, which are self-created without limitations on the content, number, and consistency of the tags. Facet labels of our task are closed vocabulary and assigned by professional annotators with some rules or limitations in some specific domains. In terms of the approach, the tagging task can be approached as either a series of binary classification tasks, one for each label, or a multiclass classification task. However, conventional classification approaches are unsuitable for our facet annotation/labeling task owing to facet heterogeneity.

We transform our task into a matching task between QAPs and external knowledge from Wikipedia rather than a direct classification task. A conventional matching model has been widely applied to the natural language inference (NLI) task (MacCartney, Galley, & Manning, 2008; Wang & Jiang, 2015; Liu, Huang, Zhang, & Huang, 2016; Mou et al., 2016; Liu, Sun, Lin, & Wang, 2016) and textual question answering (Q/A) task (Shen, Rong, Sun, Ouyang, & Xiong, 2015; Mollá & Gardiner, 2004; Hirschman & Gaizauskas, 2001; Carmel, Shtalhaim & Soffer, 2000; Andreas, Rohrbach, Darrell, & Klein, 2016). However, the problem of facet heterogeneity is not involved in the conventional matching model, thus leading to our own novel matching strategy.

To sum up, our contributions are threefold:

- As far as we know, we are the first to formalize and investigate the problem of facet annotation for QAPs corresponding to specific topics. Facet annotation is to assign one or more facet labels to a given QAP to provide a facet to realize the faceted navigation in CQA websites.
- We propose FACM, a model for facet annotation by extending CNN with a matching strategy. FACM uses a novel text representation method based on CNNs with three convolutional kernels to incorporate the phrase information at three levels. Then it combines the matching strategy to match QAPs with facet label texts (FaLTs) into facet annotation to cope with the facet heterogeneity.
- To validate our proposed model, comprehensive experiments are conducted on three real-world data sets from Quora in different knowledge domains: Data Structure, Data Mining, and Computer Network. Experimental results show that the FACM model outperforms the highly-cited and popular methods.

This letter is organized as follows. Section 2 reviews the related work. In section 3, we define the problem of facet annotation formally and describe FACM with a detailed description. In section 4, we evaluate FACM on three

data sets and analyze the results. Section 5 concludes and offers directions for future research.

2 Related Work

Facet annotation of QAPs means assigning one or more facet labels to a given QAP in CQA. It targets short text classification and subtopic mining.

2.1 Short Text Classification. Short text classification assigns labels from a predefined taxonomy to short texts, such as questions, instant messages, tweets, and comments (Chen, Jin, & Shen, 2011). Short text classification can be divided into two categories: rule based and machine learning based.

Rule-based approaches classify short texts using a straightforward method of a set of predefined heuristic rules. Some earlier research employs manually handcrafted rules (Hull, 1999; Prager, Radev, Brown, & Coden, 1999) to classify questions from TREC-8 (Voorhees, 1999). More recently, Haris and Omar (2012) adopted various types of interrogative words and analyzed the cognitive levels of Bloom's taxonomy to classify each question through the development of rules. The rule-based approaches perform well in terms of accuracy in a specific domain. However, they suffer from time-consuming and laborious human efforts using handcrafted rules, as well as low recall in different domains.

Machine learning–based approaches classify short texts to predefined categories using automatically extracted features. Li and Roth (2002) proposed a hierarchical sparse network of winnows with bag of words features for short text classification in questions. In the experiments carried out by Zhang and Lee (2003), the support vector machine (SVM) combining bag-of-words performs well in short question classification with finely grained categories, such as abbreviation and expansion. Sriram, Fuhry, Demir, Ferhatosmanoglu, and Demirbas (2010) classified the short texts to a predefined set of generic classes such as News, Events, Opinions, Deals, and Private Messages, with a small set of domain-specific features extracted from tweet authors' profiles and texts. Li, Su, Chen, and Yuan (2017) introduced a semi-supervised question classification method based on ensemble learning. Recently, Kim (2014) used a CNN trained on pretrained word vectors for sentence-level classification.

In short, the heterogeneity of predefined taxonomy from different topics, like facet heterogeneity in our task, is ignored in most approaches to short text classification. Furthermore, most approaches also ignore phrase information.

2.2 Subtopic Mining. Subtopic mining has gained much attention in recent years, which is to identify more ne-grained keywords with topics (Hu et al., 2012). Subtopics can be extracted from query logs (Hu et al., 2012, Wu,

Wu, Li, Zhou, 2015), anchor texts (Dang, Xue, & Croft, 2011), and document corpus (Allan & Raghavan, 2002). The approaches of subtopic mining can be categorized into two groups: graph based and feature based.

Graph-based approaches are often used in subtopic mining for query intent mining. Beeferman and Berger (2000) treated the query log as a bipartite graph where clusters are interpreted as subtopics covering diverse queries. Kong and Allan (2013) proposed a supervised approach based on a graphical model to distinguish query facets from noisy candidates. Yu and Ren (2014) proposed mining query subtopics through clustering on a modifier graph. Modifiers are the co-appearing terms with a given query that can explicitly specify user's interests. Yi, Zhang, and Wei (2016) integrated the nonnegative sparse latent semantic analysis (LSA) model with a subtractive clustering algorithm to mine subtopics from the search behavior tripartite graph.

Feature-based approaches mine subtopics based on the features extracted from the documents. Matsumoto, Takamura, and Okumura (2005) trained a classifier on an annotated corpus using frequent subpatterns corresponding to dependency relations as features. Zheng, Fang, Cheng, and Wang (2012) measured the similarity between two patterns by the divergence and grouped the similar patterns together to model one query subtopic. Wang, Qian et al. (2013) adopted query-independent features like the readability of subtopic candidate strings. Ullah, Shajalal, Chy, and Aono (2016) introduced multiple query-dependent and query-independent features for computing balancing relevancy and novelty to mine the subtopics. Manabe and Tajima (2016) utilized the hierarchical structure feature of headings in documents to generate diversified rankings of subtopics of keyword queries. A heading describes the topic of its associated block and the hierarchical descendant blocks of the block.

Graph-based approaches focus on the dependence structure among texts and ignore some semantic information of independent texts. Feature-based approaches focus on the feature of documents, whereas our QAPs are ascribed to a kind of short text whose features are sparse. Both kinds of approaches mine the strings of subtopics such as city and person, which often appear in corpora. The approaches for subtopic mining may fail to achieve satisfactory results owing to the absence of facet label strings in QAPs. The detailed statistics are shown in table 2 in section 3.4.

2.3 Text Matching. We transform our annotation task into a matching task between question-and-answer pairs (QAPs) and knowledge from Wikipedia. Our task is, in a sense, related to the text matching task.

Text matching models the relevance or similarity of a pair of texts in many natural language processing tasks (Liu, Qiu, Chen, & Huang, 2016), such as text classification, question answering, and machine translation. Recently, deep learning has been rising as a substantial interest in text semantic matching and has achieved great progress. The approaches of text matching

can be categorized into two groups: word-by-word matching and sentence encoding based.

Word-by-word matching approaches focus on calculating the matching score for pairwise words from two sentences. For example, Salton and Buckley (1988) computed the word-by-word similarities between two texts. Shen et al. (2015) fed the word-by-word cosine similarity matrix into a deep architecture to calculate the matching probability between the question and each candidate answer. Pang et al. (2016) constructed a matching matrix whose entries represent the similarities between words.

Sentence encoding–based approaches predict relationships between two corresponding sentences through mapping each sentence to a vector (embedding). Bromley, Guyon, LeCun, Säckinger, and Shah (1994) proposed an artificial siamese neural network, consisting of two identical subnetworks joined at their outputs, to compare two patterns and calculate the cosine similarity. The siamese architecture has been applied to different tasks as a nonlinear similarity function (Lu & Li, 2013; Hu, Lu, Li, & Chen, 2014). Hu et al. (2014) proposed convolutional neural network (CNN) models to match two sentences by adapting the convolutional strategy to combine the hierarchical modeling of individual sentences and the patterns of their matching. Rocktäschel, Grefenstette, Hermann, Kočiskỳ, and Blunsom (2015) proposed a neural model that reads two sentences to determine entailment using long short-term memory (LSTM) units for sentence matching. Shen, Zhang, Henao, Su, and Carin (2017) proposed deconvolutional networks as the sequence decoder in a latent-variable model to match natural language sentences.

An appropriate matching approach is needed to capture the rich semantic interaction information between two texts in the matching process. However, both word-by-word matching and sentence encoding–based approaches discard some semantic information, such as phrase information.

To capture the phrase information of a sentence in QAPs, we uses the convolutional layer to obtain three-level text representation corresponding to three-level phrase information in sequence. Based on three-level text representation, a matching strategy is proposed to generate three-dimensional similarity matrices between QAPs and knowledge acquired from Wikipedia.

3 FACM Model

In this section, we present FACM, a model proposed to automatically annotate facet labels of QAPs by extending CNN with a matching strategy. We first define the problem of facet annotation and then describe the schematic framework of the FACM model with detailed descriptions of each stage.

The a custom symbol system for this model that we developed in this letter is in Table 1.

Table 1: A Custom Symbol System.

Item	Set	Element	Num	Comment
Topic	T	t_j	m	Such as *Binary Tree*, *Array*, and *Red-Black Tree*
Facet	L	l_k	r	Such as *definition*, *property*, and *application*
QAP	Q	q_i	n	q_i denotes a QAP corresponding to a specific topic $t_j \in T$
	RQ	rq_i		rq_i is the text representation of q_i.
FaLT(t_j)	F_j	f_{jk}	r	f_{jk} denotes a FaLT of l_k corresponding to t_j
	RF_j	rf_{jk}		rf_{jk} is the text representation of f_{jk}
FaLT(total)	$F = \bigcup_{t_j \in T} F_j$	f_{jk}	$r \times m$	F is a universal set of FaLTs of all facets from all topics
	$RF = \bigcup_{t_j \in T} RF_j$	rf_{jk}		RF is a text representation set corresponding to F

3.1 Problem Definition. Our target is to assign one or more facet labels to a given QAP.

Formally, $Q = \{q_1, q_2, \ldots, q_i, \ldots, q_n\}$ denotes a QAP set, related to topic set $T = \{t_1, t_2, \ldots, t_j, \ldots, t_m\}$. $q_i = (w_1, w_2, \ldots, w_h, \ldots, w_s)$ denotes a QAP with topic $t_j \in T$, where w_h is the hth word. $L = \{l_1, l_2, \ldots, l_k, \ldots, l_r\}$ denotes a facet set consisting of r facets, such as definition, property, and application. Let $F_j = \{f_{j1}, f_{j2}, \ldots, f_{jk}, \ldots, f_{jr}\}$ denote a FaLT set consisting of r FaLTs corresponding to topic t_j. Each FaLT f_{jk} describes a facet l_k of topic t_j. We generate a matching function $\Phi : Q \times F \mapsto \{0, 1\}$, $F = \bigcup_{t_j \in T} F_j (1 \leqslant j \leqslant m)$.

Based on the matching function $\Phi : Q \times F \mapsto \{0, 1\}$ and $F \mapsto L$, our work aims at generating a multilabel classification function $\widetilde{\Phi} : Q \times L \mapsto \{0, 1\}$.

3.2 Framework. The schematic framework of FACM is shown in Figure 2. In practice, a QAP can be assigned with one or more facets. For simplicity, we consider the framework of a QAP and a facet together. We transform the task of automatic facet annotation into multiple binary classification problems.

3.2.1 Text Representation. The short natural language texts of QAPs are embedded into three vector spaces with a convolution layer and a max-pooling layer. Each QAP is represented as three ordered lists indicating the phrase information at three levels: unigram, bigram, and trigram. Each well-ordered list of word vectors at a level forms a matrix, and three matrices of three levels are the output of this stage.

3.2.2 Text Matching. A matching strategy is used to construct three-dimensional similarity matrices to deal with facet heterogeneity. We

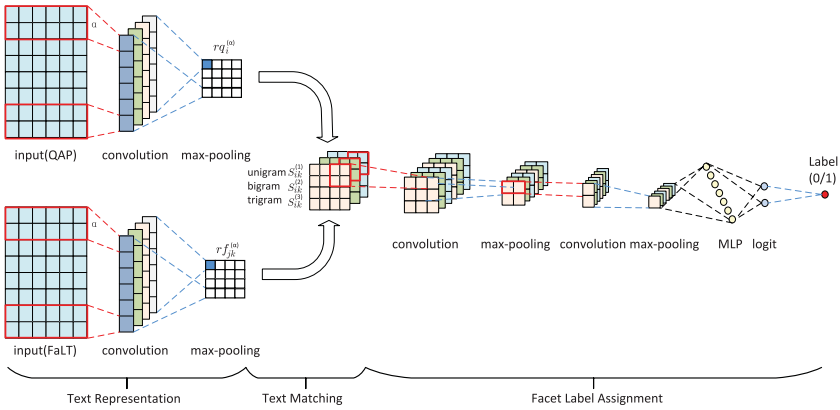


Figure 2: The schematic framework of FACM.

Figure 3: An example of a simplified Wikipedia page.

introduce a text corresponding to a specific facet of a topic from a Wikipedia page, which we call facet label text (FaLT) in this letter (see Figure 3). For example, a FaLT describing the facet definition of topic Binary Tree, is extracted from a Wikipedia page with the title of *Binary Tree*. A FaLT is also represented as three matrices with different levels of phrase information. Then three-dimensional similarity matrices are constructed by similarity measures between a QAP and a FaLT at different levels through a matching strategy.

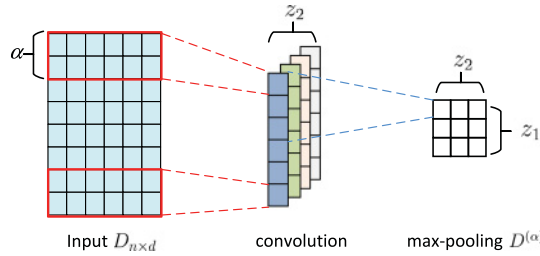


Figure 4: Text representation with different kernels $C_{\alpha\times d}$ gaining phrase information in different levels. $\alpha \in \{1, 2, 3\}$ denotes the number of words for combining and the height of kernels. $\alpha = 1$: unigram representation $D^{(1)}$ with one-word window; $\alpha = 2$: bigram representation $D^{(2)}$; $\alpha = 3$ with two-word window: trigram representation $D^{(3)}$ with three-word window.

3.2.3 Facet Label Assignment. The three-dimensional similarity matrices are combined and fed to a three-channel CNN, consisting of two convolution layers, two max-pooling layers, and a multilayer perceptron layer for facet label assignment served as a binary classification.

3.3 Text Representation. Effectively representing the embedding phrase information in QAPs is a challenging task. A word embedding matrix $M \in \mathbb{R}^{(|V|\times d)}$ is pretrained by an unlabeled large corpus from English Wikipedia 2015, where $|V|$ is the size of the vocabulary and d is the dimension of the vector space. Each word vector (a row in M) represents a word in the corpus. For QAP $q_i = (w_1, w_2, \ldots, w_h, \ldots, w_s)$, we build a matrix $D_i \in \mathbb{R}^{(s\times d)}$ containing a sequential information of words, which can also be represented as an ordered list of word vectors $(a_1, a_2, \ldots, a_h, \ldots, a_s)^\intercal$, in which a_h is the word vector of w_h.

To more precisely capture the information of a sentence in QAPs, the phrase information of words is taken into account in our model by using the proposed convolutional layer with different kernels $C_{\alpha\times d}$ ($\alpha \in \{1, 2, 3\}$), shown in Figure 4.

The proposed text representation model contains a convolution layer and a max-pooling layer. At the convolution level, three kinds of kernels are adopted to conduct different convolution operations and capture different phrase information in different size of phrases.

3.3.1 Convolutional Layer. As illustrated in Figure 4, the convolution operates on vertical sliding windows of words (variable width $\alpha \in \{1, 2, 3\}$ and length d) to gain phrase information in different levels, and vertical stride is l. $D_{s\times d}$ denotes the sentence input matrix, in which s is the length of sentence and d denotes the dimension of a word vector. The convolution unit for a feature map of type f with different α is

Table 2: Statistics of Three Data Sets in Different Knowledge Domain.

Domain	Number of QAP	Number of QAP_1	$\frac{\text{Number of QAP}^a}{\text{Number of QAP}_1}$ (%)
Data Structure	35,076	4,309	12.28
Data Mining	12,723	2,205	17.33
Computer Network	13,081	1,490	10.77

[a]#QAP is the number of total QAPs; $\#QAP_1$ is the number of QAPs containing facet labels; and $\frac{\#QAP}{\#QAP_1}$ is the proportion of the QAPs containing facet labels.

$$d_i^{(\alpha,f)} \overset{def}{=} d_i^{(\alpha,f)}(D_{s\times d}) = \sigma(\mathbf{w}^{(\alpha,f)} D_{(i:i+\alpha-1)\times d} + b^{(\alpha,f)}), \quad f = 1, 2, \ldots, z_2, \tag{3.1}$$

in which z_2 is the number of feature maps. The matrix form of convolution unit is $D_i^{(\alpha)} \overset{def}{=} D_i^{(\alpha)}(D_{s\times d}) = \sigma(\mathbf{W}^{(\alpha)} D_{(i:i+\alpha-1)\times d} + \mathbf{b}^{(\alpha)})$, where

- $d_i^{(\alpha,f)}(D_{s\times d})$ gives the output of type-f feature map for location i with width α.
- $\mathbf{w}^{(\alpha,f)}$ is the parameters for f with width α, and matrix form $\mathbf{W}^{(\alpha)} \overset{def}{=} [\mathbf{w}^{(\alpha,1)}, \mathbf{w}^{(\alpha,2)}, \ldots, \mathbf{w}^{(\alpha,z_2)}]$.
- $\sigma(\cdot)$ is the Relu activation function.
- $D_{(i:i+\alpha-1)\times d}$ concatenates the vectors of α (width of sliding window) words from the sentence input matrix $D_{s\times d}$ for the convolution at location i.

3.3.2 Max-Pooling Layer. To reach a fixed size ($z_1 \times z_2$) matrix representation $D^{(\alpha)}$ after different convolution operations with different convolutional kernels $C_{\alpha\times d}$, we take a max-pooling in every ξ-unit vertical window with the vertical stride β for every feature map of type f, calculated as follows.

$$\beta = \lfloor \frac{s}{z_1} \rfloor, \tag{3.2}$$

$$\xi = \beta + (s \mod z_1), \tag{3.3}$$

$$d_i^{(\alpha,f)} = \max\left(z_{\beta i-\beta+1}^{(\alpha,f)}, z_{\beta i-\beta+2}^{(\alpha,f)} \cdots, z_{\beta i-\beta+\xi}^{(\alpha,f)}\right), \quad i = 1, 2, \ldots, z_1. \tag{3.4}$$

z_1 is a self-set hyperparameter.

3.4 Text Matching. We conduct a statistical analysis on three real-world data sets of the number of QAPs in which facet labels appear. The results shown in Table 2 indicate that labels of facets rarely appear in the texts of QAPs.

It is very time-consuming to manually annotate topic facets of QAPs because of the absence of facet labels and facet heterogeneity of topics. To reduce the massive manual work of facet annotation owing to facet heterogeneity, we propose a matching strategy to establish the matching relationship between QAPs and FaLTs and generate three-dimensional similarity matrices.

Let $F_j = \{f_{j1}, f_{j2}, \ldots, f_{jk}, \ldots, f_{jr}\}$ denote a FaLT set consisting of r FaLTs corresponding to topic t_j. A FaLT set $F = \bigcup_{t_j \in T} F_j (1 \leqslant j \leqslant m)$ is acquired by our developed web crawler.

After text representation described in section 3.3, a QAP representation set $RQ = \{rq_1, rq_2, \ldots, rq_i, \ldots, rq_n\}$ is generated corresponding to QAP set Q. $rq_i = (rq_i^{(1)}, rq_i^{(2)}, rq_i^{(3)})$ denotes three matrix representations of QAP q_i, where $rq_i^{(\alpha)} (\alpha \in \{1, 2, 3\})$ is the QAP representation of α-level phrase information. A FaLT representation set $RF_j = \{rf_{j1}, rf_{j2}, \ldots, rf_{jk}, \ldots, rf_{jr}\}$ represents FaLT set F_j, and $RF = \bigcup_{t_j \in T} RF_j (1 \leqslant j \leqslant m)$ is a universal set of FaLT representations. $rf_{jk} = (rf_{jk}^{(1)}, rf_{jk}^{(2)}, rf_{jk}^{(3)})$ denotes three matrix representations of FaLT f_{jk}, where $rf_{jk}^{(\alpha)} (\alpha \in \{1, 2, 3\})$ is the FaLT representation of α-level phrase information.

For each rq_i and rf_{jk} corresponding to the same topic t_j, the three-dimensional similarity matrix S_{ik} is generated and formulated as follows.

$$S_{ik} = (S_{ik}^{(1)}, S_{ik}^{(2)}, S_{ik}^{(3)}), \tag{3.5}$$

$$S_{ik}^{(\alpha)} = \cos(rq_i^{(\alpha)}, rf_{jk}^{(\alpha)}) \quad (\alpha \in \{1, 2, 3\}). \tag{3.6}$$

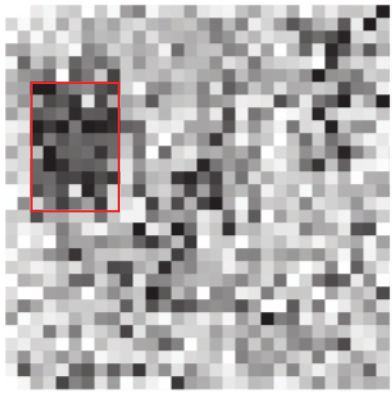
An element s_{uv} in the matrix $S_{ik}^{(\alpha)}$ is calculated as follows:

$$s_{uv} = \frac{\vec{w}_u \cdot \vec{w}_v^{\mathsf{T}}}{||\vec{w}_u|| ||\vec{w}_v||}. \tag{3.7}$$

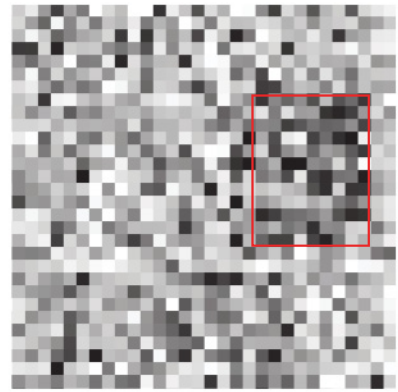
$\vec{w}_u$ is the uth$(1 \leqslant u \leqslant z_1)$ row of $rq_i^{(\alpha)}$, and $\vec{w}_v$ is the vth$(1 \leqslant v \leqslant z_1)$ row of $rf_{jk}^{(\alpha)}$.

$S_{ik} = (S_{ik}^{(1)}, S_{ik}^{(2)}, S_{ik}^{(3)})$ is calculated by cosine similarity of three matrix representations rq_i of QAP q_i and three matrix representations rf_{jk} of FaLT f_{jk} corresponding to the same topic t_j. A three-dimensional similarity matrix set $\mathbb{S}_i = \{S_{i1}, S_{i2}, \ldots, S_{ik}, \ldots, S_{ir}\}$ is produced for each QAP q_i corresponding to topic t_j, as the input of facet label assignment.

3.5 Facet Label Assignment. We discover local correlations in the similarity matrices by visual analysis. Figure 5 shows examples of similarity matrices for Binary Tree. "Local correlation" refers to degree of a QAP matching with a FaLT in local region. The similarity pattern at the top left



(a) **QAP:** *What is a binary tree? A tree with not more than two branches at any branching point.* **FaLT:** *A binary tree is a tree data structure in which each node has at most two children, which are referred to as the left child and the right child.*



(b) **QAP:** *What is a binary tree? A binary tree is a tree data structure in which each node has at most two children, which are referred to as the left child and the right child.* **FaLT:** *A binary tree is a tree data structure in which each node has at most two children, which are referred to as the left child and the right child.*

Figure 5: Examples of visual similarity matrices $S^{(1)}$. Deeper color represents a higher value of similarity. (For QAPs, see https://www.quora.com/What-is-a-binary_tree. For FaLTs, see https://en.wikipedia.org/wiki/Binary_tree.)

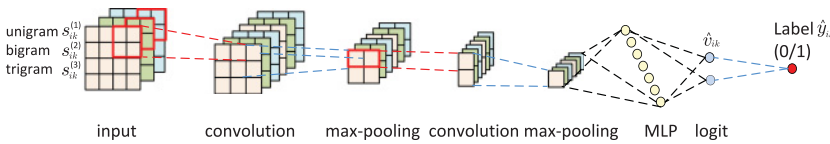


Figure 6: Three-channel CNN for facet label assignment.

corner in Figure 5 indicates a local correlation between a QAP representation matrix and a FaLT representation matrix.

We also find that the similarity pattern may appear in any part of similarity matrices. For example, similarity patterns are shown at different locations in Figures 5a and 5b.

Inspired by local correlations of similarity patterns, FACM uses a three-channel CNN to combine phrase information of three levels for the facet label assignment, shown in Figure 6. This three-channel CNN uses the three-dimensional similarity matrices as the input and consists of two convolution layers (*conv*1, *conv*2), two max-pooling layers (*pool*1, *pool*2), and a multilayer perceptron layer (*mlp*).

For each three-dimensional similarity matrix S_{ik} between QAP q_i and FaLT f_{jk}, the representation $\hat{v}_{ik} = (P^0_{ik}, P^1_{ik})$ of the three-channel CNN is computed by

$$\hat{v}_{ik} = logit(CNN'(S_{ik}; \theta)), \tag{3.8}$$

where $\theta = \{W_1, W_2, W_{mlp}\}$ denotes the parameters of $conv1$, $conv2$, and mlp respectively and $CNN'(S_{ik}; \theta)$ is the output of the mlp layer. Afterward, a logistic function $logit()$ with two output values is applied to obtain the probability P^1_{ik} of predicting (q_i, f_{jk}) as 1 and P^0_{ik} as 0. (q_i, f_{jk}) is a pair of QAP q_i and FaLT f_{jk} corresponding to same topic t_j.

At the last hop, the result of the three-channel CNN is $\hat{y}_{ik} \in \{0, 1\}$, which is computed by equation 3.9:

$$\hat{y}_{ik} = \underset{y \in \{0,1\}}{argmax}(P^y_{ik}). \tag{3.9}$$

In equation 3.9, $\hat{y}_{ik} = 1$ denotes that QAP q_i can match with FaLT f_{jk} corresponding to facet label $l_k \in L(1 \leqslant k \leqslant r)$ of topic t_j. Namely, QAP q_i can be assigned with facet label l_k; otherwise q_i cannot be assigned with l_k.

The proposed model is trained in a supervised manner by minimizing the loss function, which is formulated by

$$loss = -\sum_{i=1}^{n}\sum_{k=1}^{r} y_{ik} \log(P^1_{ik}), \tag{3.10}$$

where $y_{ik} \in \{0, 1\}$ denotes whether the QAP q_i matches the FaLT f_{jk} of the facet label l_k, both corresponding to topic t_j.

4 Experiments

We evaluate FACM on the three data sets from different knowledge domains: Data Structure, Data Mining and Computer Network. The three data sets are described in section 4.1. Experimental settings and experimental results are discussed in sections 4.2 and 4.3, respectively.

4.1 Data Sets and Evaluation Metrics.

4.1.1 Data Sets. We construct three data sets of different knowledge domains owing to the of appropriate data sets for FACM evaluation. Each data set consists of two parts: QAPs and FaLTs.

According to all topics of the three knowledge domains, we acquire QAPs from Quora automatically from our web crawler. Each QAP is represented as a word sequence of a question and a corresponding answer.

Table 3: Statistics of Three Data Sets in Different Knowledge Domains.

Data Set	Number of Topics	Number of QAPs	Number of FaLTs
Data Structure	193	35,076	1,930
Data Mining	94	12,723	940
Computer Network	84	13,081	840

We recruited seven graduate students majoring in computer science as annotators to construct data sets on the three knowledge domains of Data Structure, Data Mining, and *Computer Network*. Each QAP in the three domain data sets was annotated with one or more facet labels from the predefined candidate facet label set T ={definition, property, application, algorithm, example, type, history, method, implementation, operation}. The annotators learned the strict annotation guidelines with annotated examples before annotation. Pairs of annotators are asked to label the same data set independently, and the seventh annotator resolved annotation disagreements during group meetings. The labeling processes of judges are considered dependable if they have high consistency. The pairwise agreement established by two annotators is measured with Fleiss's kappa (Fleiss & Cohen, 1973). The agreement coefficient must be at least 0.61, which is described as "substantial agreement" by Landis and Koch (1977). The kappa coefficient of our annotation is 0.78, which means our annotations are reliable.

According to all the topics of the three knowledge domains, we acquire FaLTs from Wikipedia automatically by our developed web crawler. We have developed software to automatically parse the contents of Wikipedia pages, obtaining the headwords of items in the contents (e.g., *definition*), and extracting corresponding texts linked to the items as FaLTs.

The statistics for the three data sets are shown in Table 3. In our experiments, 70% of the data sets is used as a training set, 10% is used as a validation set, and the other 20% is used as a test set.[5]

4.1.2 Evaluation Metrics. The *Macro_F*1 score of multilabel forms is selected as the main metric, which is a widely used evaluation metric and computed as follows:

$$Macro\_P = \frac{1}{m}\sum_{i=1}^{m} P_i, \tag{4.1}$$

[5]The data are available at https://github.com/xjtu-BeiWu/FN_Dataset.

Table 4: Coverage Proportion of QAPs with Different *s*.

	s			
Coverage Proportion (%)	0–100	0–200	0–300	0–400
Data Set				
Data Structure	79.58	97.08	98.33	99.58
Data Mining	78.00	96.33	98.98	99.54
Computer Network	80.99	98.91	99.06	99.72

$$Macro\_R = \frac{1}{m}\sum_{i=1}^{m} R_i, \tag{4.2}$$

$$Macro\_F1 = \frac{2 \times Macro\_P \times Macro\_R}{Macro\_P + Macro\_R}. \tag{4.3}$$

In equations 4.1 to 4.3, m is the number of facets, and P_i and R_i are the precision and recall scores of ith label respectively.

A t-test is used to determine whether the performance difference is statistically significant.

4.2 Experimental Settings.

4.2.1 Text Representation. The 50-dimensional vectors for word embedding are pretrained on 12.2G corpus from English Wikipedia 2015 by the word2vec utility (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013).

Since the size of the input of CNN should be fixed, it is necessary to select an appropriate s value. It is known that the sizes of the lengths of QAPs are variable, and Table 4 shows the number and proportion of QAPs covered by different lengths s. From the table, it is observed that $s = 200$ has a high coverage proportion and is less costly of memory. Thus, we tile or trim our QAPs into size $s = 200$ to make sure that a QAP is represented as a 200×50 matrix. We report z_2 feature maps with $\alpha \times 50$ kernels. In general, a phrase is not more than three words, so we adopt $\alpha \in \{1, 2, 3\}$. According to equations 3.1 to 3.3, the kernel size of max-pooling and the stride of max-pooling are preset values of z_1. A simple grid search is used to estimate the value of z_1 and z_2, and we select $z_1 = 30$ and $z_2 = 30$.

4.2.2 Facet Label Assignment. Convolution layer *conv1* has 32 feature maps with 5×5 kernels, and convolution layer *conv2* has 64 feature maps with 5×5 kernels. The max pool sizes for *pool1* and *pool2* both are 2×2. The dropout rate is 0.5 (Hinton, Srivastava, Krizhevsky, Sutskever, & Salakhutdinov, 2012). These hyperparameters are selected through grid search on a validation set.

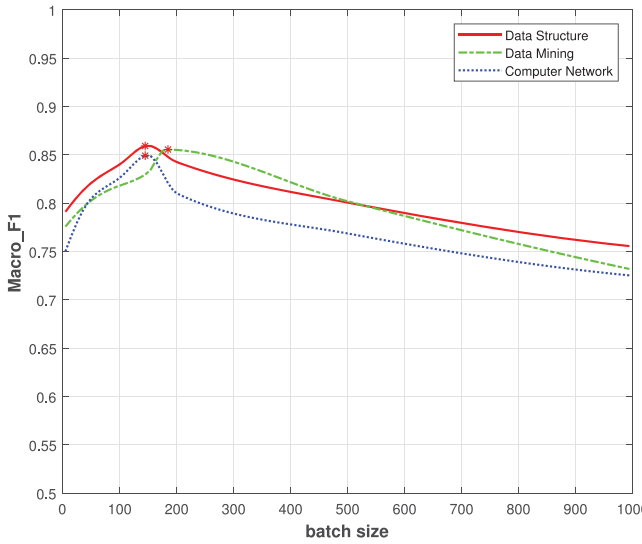


Figure 7: *Macro_F*1 with different batch sizes of three data sets.

FACM is learned with mini-batch Adam (Kingma & Ba, 2014) and implemented with TensorFlow (Abadi et al., 2016). To investigate the effect of batch size on the experimental results, the other hyperparameters remain fixed, such as the base learning rate $\lambda = 0.001$, the exponential decay rate for the first-moment estimates $\beta_1 = 0.9$, the exponential decay rate for the second-moment estimates $\beta_2 = 0.999$, and numerical stability $\epsilon = 0.00001$. The training time on both data sets is limited to a maximum of 600 epochs. The results with different batch sizes are shown in Figure 7.

The results indicate that our model performs better with a mini-batch (about 150–200 in size), which can be easily parallelized on single machine with multicores. Considering the memory capacity, we select batch size $bs = 150$ finally.

The parameters of our model are shown in Table 5.[6]

4.2.3 Baseline Approaches. We develop four baseline experiments: CNN-static based on a highly cited and up-to-date approach, SVM + bag of words based on a highly cited and state-of-the-art approach, SVM + word embedding for excluding incomparability and rule-based approach based on an up-to-date approach.

- CNN-static. This approach, proposed by Kim (2014), was selected as the first baseline experiment. Kim trained a simple CNN with one

[6] The codes are available at https://github.com/xjtu-BeiWu/FACM-model.

Table 5: The Parameters of Our Model.

Parameter	Value	Parameter	Value
s	200	Kernel size ($pool1$, $pool2$)	2×2
d	50	Dropout	0.5
α	{1,2,3}	Optimizer	Adam
z_1	30	Base learning rate λ	0.001
z_2	30	First-moment estimates β_1	0.9
Kernel size ($conv1$, $conv2$)	5×5	Second-moment estimates β_2	0.999
Number of feature maps ($conv1$)	32	Numerical stability ϵ	0.00001
Number of feature maps ($conv2$)	64	Batch size bs	150

layer of convolution using the top of word vectors as input. Word vectors are obtained from an unsupervised neural language model to classify sentences. To exclude incomparability of consequence owing to different input features, we chose CNN-static with 50-dimensional pretrained vectors from word2vec utility (Mikolov et al., 2013).

- SVM + bag of words. Zhang and Lee (2003) carried out experiments by an SVM with bag-of-words, which performs well in short question classification with finely grained categories. The method is used as the baseline because it is popular and effective in short text classification.
- SVM + word embedding. In order to exclude incomparability of the consequence owing to different input features, we conducted an experiment using an SVM with word embedding, where features were the average value of word embeddings (Liu et al., 2015).
- Rule-based. Haris and Omar (2012) developed a series of rules to analyze the cognitive levels of Bloom's taxonomy for each question, which improves the precision of the result in short text classification. We used the application level rules of the method by using the syntactic structure of each QAP and developed others on our own.

In addition, in all the experiments using the SVM, we used LibSVM and the radial basis function kernel with default parameters $C = 1.0, \epsilon = 10^{-3}, \gamma = 1/\sigma^2$.

4.3 Experimental Results and Analysis. Table 6 summarizes the performances of our proposed model FACM and other baseline approaches. FACM consistently improves the *Macro_F*1 scores compared with all baseline approaches on the three data sets, respectively, and improves *Macro_P*, *Macro_R* scores compared with CNN-static and SVM + bag of words. SVM + word embedding does not yield better performance than all baselines and FACM, partly because word embedding, the distributed representation, is not appropriate for the SVM. Rule-based with high-quality

Table 6: Comparisons of FACM with Baseline Approaches.

Metric	Method				
	CNN-Static	SVM + Bag of Words	SVM + Word Embedding	Rule-Based	FACM
Data Structure					
Macro_P (%)	83.54	83.21	60.92	**93.78**	85.08
Macro_R (%)	82.28	78.46	73.53	60.25	**86.86**
*Macro_F*1 (%)	82.91	80.77	66.63	73.37	**85.96**
Data Mining					
Macro_P (%)	82.09	84.58	61.05	**85.52**	83.60
Macro_R (%)	80.13	76.63	74.85	62.56	**82.72**
*Macro_F*1 (%)	81.10	80.41	67.25	72.26	**83.16**
Computer Network					
Macro_P (%)	80.54	82.12	65.18	83.75	**85.66**
Macro_R (%)	83.33	81.25	56.52	68.68	**84.29**
*Macro_F*1 (%)	81.91	81.68	60.54	75.47	**84.97**

*FACM versus CNN-static significantly different ($p < 0.05$).
Note: The numbers in bold are the highest value of each metric.

hand-crafted rules possess high *Macro_P*, even exceeding 93% in Data Structure. In our experiment, we made used strict rules to ensure high-quality labels for our QAP. However, it led to a lower *Macro_R* rate. FACM improved about 10% of *Macro_F*1 scores through the improvement of *Macro_R* scores. It indicates that FACM performs well in text representation with phrase information and generalization capability.

To further describe the models' generalization capability in different topics of the same knowledge domain, we compared the standard deviation σ of *Macro_F*1 among all methods. σ can be formulated as

$$\sigma = \sqrt{\frac{1}{r}\sum_{i=1}^{r}(F_i - \overline{F})}, \tag{4.4}$$

where r is the number of topics in a specific domain. F_i is the *Macro_F*1 in ith topic and $\overline{F}$ is the *Macro_F*1 in domain.

Figure 8 shows that standard deviation σ of Rule-based is the highest because of the rules with a strong dependency on topics. The standard deviation of SVM + bag of words is very close to that of SVM + word embedding, because both of them adopt the same classifier, SVM. Meanwhile, σ of FACM is the lowest among all methods, which indicates that FACM implements facet annotation with a good generalization capability in different topics.

In order to further evaluate the effectiveness of each part of the model, an experiment was conducted between FACM and uncompleted FACM

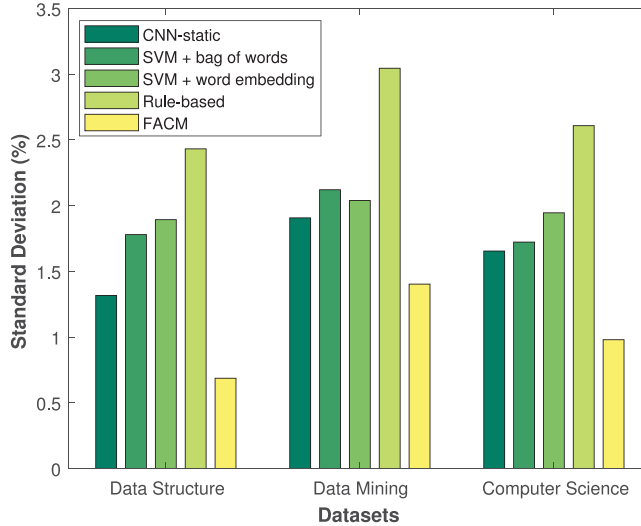


Figure 8: Comparisons of FACM with baseline approaches in standard deviation.

without certain parts. We used PI, PII, and PIII to represent the three parts, respectively, of FACM text representation, matching strategy, and facet label assignment. Considering the realizability, we selected the following three contrast methods.

- *Word embedding + PIII*. To verify the effectiveness of text representation of FACM (PI), we conducted the experiment based on Word embedding + PIII. Considering the characteristics of input features, we replaced the three-channel CNN with a single-channel CNN in PIII. In other words, CNN-static was adopted as the first contrast method.
- *PI+PIII*. To verify the necessity of matching strategy (PII), we conducted the experiment based on PI+PIII. The QAPs are embedded into three vector spaces indicating the three-level phrase information by our text representation method, generating three matrices. Taking these three matrices as the input, we used a three-channel CNN to assign facet label.
- *PI+SVM*. To judge whether the text representation method (PI) we proposed in FACM is appropriate for other classifiers and verify the appropriateness of facet label assignment of FACM (PIII), we conducted the experiment based on PI+SVM. The QAPs are embedded into three vector spaces indicating the three-level phrase information by our text representation method, generating three matrices. Directly connecting the three matrices and taking the result as input features, we adopt the SVM as a classifier.

Table 7: Comparisons of *Macro_F*1 between FACM and Uncompleted FACM without Certain Parts.

	Method			
Macro_F1	Word Embedding+PIII	PI+PIII	PI+SVM	PI+PII+PIII
Data Set				
Data Structure	82.91	83.54	75.06	**85.96**
Data Mining	81.10	82.65	68.28	**83.16**
Computer Network	81.91	83.27	76.24	**84.97**

Note: The numbers in bold are trhe highest values.

The results are shown in Table 7. The *Macro_F*1 score of PI+PII+PIII is higher than word embedding+PIII, which indicates that the method of text representation (PI) we proposed outperforms existing representation methods. The *Macro_F*1 score of PI+PII+PIII (FACM) is higher than PI+PIII, which indicates that the stage of matching strategy (PII) is necessary for facet annotation task. The low *Macro_F*1 scores of PI+SVM indicate that the features obtained from our representation method are not proper for the SVM, which also illustrates the necessity of the facet label assignment method (P) we proposed in FACM.

To evaluate the effectiveness of our embedding approach, we compared our FACM with three embedding-based approaches:

- #TagSpace is a CNN that learns feature representations for short textual posts using hashtags as a supervised signal to rank hashtags with a minimum of task-specific assumptions and engineering (Weston, Chopra, & Adams, 2014).
- Bi-LSTM presents two separate parallel recurrent nets (forward and backward) to obtain the contextual information (Graves & Schmidhuber, 2005).
- **C-LSTM** uses a CNN to extract a sequence of higher-level phrase representations and feeds the phrase representation into an LSTM to obtain the sentence representation (Zhou, Sun, Liu, & Lau, 2015).

To ensure the comparability of FACM with other three approaches, we choose the dimension of feature vector $d = 50$. The comparison results, shown in Table 8, demonstrate that FACM outperforms the other three approaches on the same data set. Part of the reason is that the three approaches do not take facet heterogeneity into account. Furthermore, #TagSpace does not yield better performance than FACM because the hashtags in it are frequent in texts, whereas our facet labels rarely appear in QAPs. The *Macro_F*1 of Bi-LSTM is lower than that of FACM because Bi-LSTM does not encode the phrase information effectively.

Table 8: Comparisons of *Macro_F*1 between FACM and Three Embedding-Based Approaches.

	Method			
Macro_F1	#TAGSPACE	Bi-LSTM	C-LSTM	FACM
Data Set				
Data Structure	72.31	82.13	83.65	**85.96**
Data Mining	73.34	79.67	80.23	**83.16**
Computer Network	69.97	80.15	82.79	**84.97**

Note: The numbers in bold are highest values.

Table 9: Comparisons of *Macro_F*1 between Three-Dimensional Similarity Matrices Input and Other Similarity Matrix Combination Inputs.

	Method						
Macro_F1	SI	SII	SIII	SI+SII	SI+SIII	SII+SIII	SI+SII+SIII
Data Set							
Data Structure	83.05	84.08	82.77	84.50	83.19	84.21	**85.96**
Data Mining	81.51	82.43	82.09	82.56	81.02	82.85	**83.16**
Computer Network	82.94	83.54	83.12	83.81	83.43	83.98	**84.97**

Note: The numbers in bold are the highest values.

To evaluate the effectiveness of the three input similarity matrices in CNNs, we conducted a comparison experiment in CNNs with the inputs of different matrix combinations. We used SI, SII, and SIII to represent the three similarity matrices, respectively. Considering the realizability, we selected CNNs with different channel numbers depending on the number of input matrices, such as the one-channel CNN for one-dimensional similarity matrix. The comparison results are in Table 9.

The comparison results demonstrate that FACM with three-dimensional similarity matrices outperforms the models with the input of other similarity matrix combinations on the same data set.

5 Conclusion

We present FACM, a neural network model that annotates facets of QAPs by extending CNN with a matching strategy. The FACM model incorporates phrase information into QAP representation and combines a matching strategy to tackle the massive manual work of data annotation owing to facet heterogeneity. A three-channel CNN is trained as a binary classifier to implement facet label assignment. Results of comprehensive experiments indicate that FACM outperforms the highly cited and popular methods related to facet annotation. In the future, we would like to apply a

dynamic number of facets corresponding to a specific topic into the facet annotation task and incorporate the hierarchical structure of facets into facet annotation.

Acknowledgments

This work was supported by National Key Research and Development Program of China, grant 2016YFB1000903; Science and Technology Planning Project of Guangdong Province, China, grant 2017A010101029; National Natural Science Foundation of China grants 61672419, 61672418, 61532004, and 61532015; Innovative Research Group of the National Natural Science Foundation of China, grant 61721002; Ministry of Education Innovation Research Team, IRT_17R86; Project of China Knowledge Centre for Engineering Science and Technology; and MOE Research Center for Online Education Funds under grants 2016YB165 and 2016YB166.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., . . . Ghemawat, S. (2016). *Tensorflow: Large-scale machine learning on heterogeneous distributed systems*. arXiv:1603.04467.

Allan, J., & Raghavan, H. (2002). Using part-of-speech patterns to reduce query ambiguity. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 307–314). New York: ACM.

Andreas, J., Rohrbach, M., Darrell, T., & Klein, D. (2016). *Learning to compose neural networks for question answering*. arXiv:1601.01705.

Beeferman, D., & Berger, A. (2000). Agglomerative clustering of a search engine query log. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 407–416). New York: ACM.

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, *3*, 1137–1155.

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., & Shah, R. (1994). Signature verification using a "siamese" time delay neural network. In J. D. Cowan, G. Tesauro, & J. Alspector (Eds.), *Advances in neural information processing systems, 7* (pp. 737–744). Cambridge, MA: MIT Press.

Carmel, D., Shtalhaim, M., & Soffer, A. (2000). Eresponder: Electronic question responder. In O. Etzion & P. Scheuermann (Eds.), *Cooperative information systems* (pp. 150–161). New York: Springer.

Chen, M., Jin, X., & Shen, D. (2011). Short text classification improved by learning multi-granularity topics. In *Proceedings of the AAAI International Joint Conference on Artificial Intelligence* (pp. 1776–1781). Menlo Park, CA: AAAI.

Dang, V., Xue, X., & Croft, W. B. (2011). Inferring query aspects from reformulations using clustering. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management* (pp. 2117–2120). New York: ACM.

Fleiss, J. L., & Cohen, J. (1973). The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and Psychological Measurement*, *33*(3), 613–619.

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks, 18*(5), 602–610.

Haris, S. S., & Omar, N. (2012). A rule-based approach in Bloom's taxonomy question classification through natural language processing. In *Proceedings of the 7th International Conference on Computing and Convergence Technology* (pp. 410–414). Piscataway, NJ: IEEE.

Harris, Z. S. (1954). Distributional structure. *Word, 10*(2–3), 146–162.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). *Improving neural networks by preventing co-adaptation of feature detectors*. arXiv:1207.0580.

Hirschman, L., & Gaizauskas, R. (2001). Natural language question answering: The view from here. *Natural Language Engineering, 7*(4), 275–300.

Hu, B., Lu, Z., Li, H., & Chen, Q. (2014). Convolutional neural network architectures for matching natural language sentences. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems, 27* (pp. 2042–2050). Red Hook, NY: Curran.

Hu, Y., Qian, Y., Li, H., Jiang, D., Pei, J., & Zheng, Q. (2012). Mining query subtopics from search log data. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 305–314). New York: ACM.

Hull, D. A. (1999). Xerox TREC-8 question answering track report. In *Proceedings of Text Retrieval Conference* (pp. 743–752). Washington, DC: Department of Commerce and National Institute of Standards and Technology.

Kim, Y. (2014). *Convolutional neural networks for sentence classification*. arXiv:1408.5882.

Kingma, D., & Ba, J. (2014). *Adam: A method for stochastic optimization*. arXiv:1412.6980.

Kong, W., & Allan, J. (2013). Extracting query facets from search results. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 93–102). New York: ACM.

Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics, 33*, 159–174.

Li, X., & Roth, D. (2002). Learning question classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics* (vol. 1, pp. 1–7). Stroudsburg, PA: ACM.

Li, Y., Su, L., Chen, J., & Yuan, L. (2017). Semi-supervised learning for question classification in CQA. *Natural Computing, 16*(4), 567–577.

Liu, P., Qiu, X., Chen, J., & Huang, X. (2016). Deep fusion LSTMS for text semantic matching. In *Proceedings of the 54th Meeting of the ACL* (pp. 1034–1043). Stroudsburg, PA: ACL.

Liu, X., Gao, J., He, X., Deng, L., Duh, K., & Wang, Y.-Y. (2015). Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In *Proceedings of the North American Chapter of the Association for Computational Linguistics–Human Language Technologies* (pp. 912–921). Stroudsburg, PA: ACL.

Liu, Y., Sun, C., Lin, L., & Wang, X. (2016). *Learning natural language inference using bidirectional LSTM model and inner-attention*. arXiv:1605.09090.

Liu, Z., Huang, D., Zhang, J., & Huang, K. (2016). Research on attention memory networks as a model for learning natural language inference. In *Proceedings of the Conference on Empirical Methods on Natural Language Processing* (pp. 18–24). Stroudsburg, PA: ACL.

Lu, Z., & Li, H. (2013). A deep architecture for matching short texts. In C. J. C. Burges, L. Bottou, M. Welling, G. Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems, 26* (pp. 1367–1375). Red Hook, NY: Curran.

MacCartney, B., Galley, M., & Manning, C. D. (2008). A phrase-based alignment model for natural language inference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (pp. 802–811). Straudsburg, PA: ACL.

Manabe, T., & Tajima, K. (2016). Subtopic ranking based on hierarchical headings. *Webist, 2*, 121–130.

Matsumoto, S., Takamura, H., & Okumura, M. (2005). Sentiment classification using word sub-sequences and dependency sub-trees. In *Proceedings of the Pacific-Asia Conference on Knowlege Discovery and Data Mining* (vol. 5, pp. 301–311). New York: Springer.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems, 26* (pp. 3111–3119). Red Hook, NJ:

Mollá, D., & Gardiner, M. (2004). Answerfinder-question answering by combining lexical, syntactic and semantic information. In *Australasian Language Technology Workshop* (pp. 9–16). N.p.

Mou, L., Men, R., Li, G., Xu, Y., Zhang, L., Yan, R., & Jin, Z. (2016). Natural language inference by tree-based convolution and heuristic matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* (pp. 130–136). Stroudsburg, PA: ACL.

Pang, L., Lan, Y., Guo, J., Xu, J., Wan, S., & Cheng, X. (2016). Text matching as image recognition. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence* (pp. 2793–2799). Menlo Park, CA: AAAI.

Prager, J., Radev, D., Brown, E., & Coden, A. (1999). The use of predictive annotation for question answering in TREC8. In *Proceedings of 8th Text Retrieval Conference* (pp. 399–409). Washington, DC: Department of Commerce and National Institute of Standards and Technology.

Rocktäschel, T., Grefenstette, E., Hermann, K. M., Kočiskỳ, T., & Blunsom, P. (2015). *Reasoning about entailment with neural attention*. arXiv:1509.06664.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing and Management, 24*(5), 513–523.

Shen, D., Zhang, Y., Henao, R., Su, Q., & Carin, L. (2017). *Deconvolutional latent-variable model for text sequence matching*. arXiv:1709.07109.

Shen, Y., Rong, W., Sun, Z., Ouyang, Y., & Xiong, Z. (2015). Question/answer matching for CQA system via combining lexical and sequential information. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence* (pp. 275–281). Menlo Park, CA: AAAI.

Sriram, B., Fuhry, D., Demir, E., Ferhatosmanoglu, H., & Demirbas, M. (2010). Short text classification in twitter to improve information filtering. *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 841–842). New York: ACM.

Ullah, M. Z., Shajalal, M., Chy, A. N., & Aono, M. (2016). Query subtopic mining exploiting word embedding for search result diversification. In S. Ma, J.-R. Wan, Y. Liu, Z. Dou, M. Zhang, Y. Chang, & X. Zhao (Eds.), *Information retrieval technology* (pp. 308–314). New York: Springer.

Voorhees, E. M. (1999). The TREC-8 question answering track report. *Trec*, *99*, 77–82.

Wang, G., Gill, K., Mohanlal, M., Zheng, H., & Zhao, B. Y. (2013). Wisdom in the social crowd: An analysis of quora. In *Proceedings of the 22nd International Conference on World Wide Web* (pp. 1341–1352). New York: ACM.

Wang, Q., Qian, Y., Song, R., Dou, Z., Zhang, F., Sakai, T., & Zheng, Q. (2013). Mining subtopics from text fragments for a web query. *Information Retrieval*, *16*(4), 484–503.

Wang, S., & Jiang, J. (2015). *Learning natural language inference with LSTM*. arXiv:1512.08849.

Weston, J., Chopra, S., & Adams, K. (2014). # tagspace: Semantic embeddings from hashtags. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* (pp. 1822–1827). Stroudsburg, PA: ACL.

Wu, Y., Wu, W., Li, Z., & Zhou, M. (2015). Mining query subtopics from questions in community question answering. In *Proceedings of the Association for the Advancement of Artificial Intelligence* (pp. 339–345). Menlo Park, CA: AAAI.

Yi, D., Zhang, Y., & Wei, B. (2016). Query subtopic mining via subtractive initialization of non-negative sparse latent semantic analysis. *Journal of Information Science and Engineering, 32*(5), 1161–1181.

Yu, H.-T., & Ren, F. (2014). Subtopic mining via modifier graph clustering. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining* (pp. 337–347). New York: Springer.

Zhang, D., & Lee, W. S. (2003). Question classification using support vector machines. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval* (pp. 26–32). New York: ACM.

Zheng, W., Fang, H., Cheng, H., & Wang, X. (2012). Diversifying search results through pattern-based subtopic modeling. *International Journal on Semantic Web and Information Systems, 8*(4), 37–56.

Zhou, C., Sun, C., Liu, Z., & Lau, F. (2015). *A C-LSTM neural network for text classification*. arXiv:1511.08630.

Received August 30, 2017; accepted January 16, 2018.