Information Sciences 537 (2020) 227–245

Contents lists available at ScienceDirect

Information Sciences

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

# An integrated retrieval framework for similar questions: Word-semantic embedded label clustering – LDA with question life cycle

Yue Liu [a,b,c,*], Aihua Tang [a], Zhibin Sun [d,*], Weize Tang [a], Fei Cai [a], Chengjin Wang [a]

[a] School of Computer Engineering and Science, Shanghai University, Shanghai 20444, China
[b] Shanghai Institute for Advanced Communication and Data Science, Shanghai University, Shanghai 20444, China
[c] Shanghai Engineering Research Center of Intelligent Computing System, Shanghai 200444, China
[d] Natural Resource Ecology Laboratory, Colorado State University, Fort Collins, CO 80523, USA

ARTICLE INFO

*Article history:*
Received 4 November 2019
Received in revised form 1 April 2020
Accepted 4 May 2020
Available online 26 May 2020

*Keywords:*
CQA
Question retrieval
Product life cycle
Semantic representation

ABSTRACT

Question retrieval is an extremely important research field in Community Question Answering (CQA). Most existing question retrieval methods depend on semantic analysis of questions, whose effectiveness suffers from the short texts of the noise words in the question corpus. In order to recommend the questions with more advanced knowledge to users, the influence of the questions' popularity should be considered during retrieving questions. To make retrieved questions with both similar semantics and high popularity, we propose an Integrated Retrieval Framework for Similar Questions named Word-semantic Embedded Label Clustering – LDA with Question Life Cycle (WELQLC-QR), consisting of Word-semantic Embedded Label Clustering – LDA (WEL) and Question Life Cycle Optimization Similar Question List Strategy (QLC). Firstly, WEL is proposed for question retrieval from the perspective of semantic matching. It not only overcomes the problem of over-generalization of the semantic information extracted by topic models when facing short questions with multi-levels labels, but also avoids the influence of noise vocabularies during semantic extracting of the questions. Then, based on the internal factors (i.e., the number of comments and answers to the question) and external factors (i.e., programming language ranking information) of questions, QLC constructs a popularity-predicted model to optimize the similar question set searched by WEL, making the final retrieval results both semantically similar and popular. Finally, experiments are conducted on CQADupStack dataset, and results show that the MRR@N of WELQLC-QR model has an average increase of 8.99%, 8.3%, 4.74% and 3.56% compared with that of L-LDA, LC-LDA, BM25 and Word2vec, respectively.



## 1. Introduction

Community-based Question and Answering services (CQA) (e.g., Stack Overflow and Baidu Zhidao) have gradually emerged as a popular way for people to access knowledge in recent years [1]. In CQA, a major way to acquire knowledge is question retrieval, which belongs to the field of Information Retrieval (IR) [2] and can search both the questions similar

* Corresponding authors at: School of Computer Engineering and Science, Shanghai University, Shanghai 20444, China (Yue Liu). Natural Resource Ecology Laboratory, Colorado State University, Fort Collins, CO 80523, USA (Zhibin Sun).

https://doi.org/10.1016/j.ins.2020.05.014
0020-0255/

to the queried ones from users and their answers [3]. One big challenge for the question retrieval is the lexical gap between questions [4], which means that two questions in different expressions could be similar in semantics [5], and thus pose difficulties for question retrieval methods to identify the similarity between them. Meanwhile, the popularity of a question (i.e., the recent visited times of the question) is equally essential to users, as it reveals whether the question can effectively solve current users' doubts. Some questions are difficult to attract users because they carry outdated knowledge and cannot answer users' doubts, while the questions with high popularity receive many attentions from CQA users because their carried knowledge can effectively solve users' problems. For example, there exist two questions: "How does Java implemented the HashMap?" proposed in 2013 and "HashMap Java implementation" proposed in 2017. These two questions are essentially similar in semantics because they are both related to the implementation of HashMap in Java programming language. As we know, the current version of Java is generally 1.8, and it is very different in the implementation of HashMap from the versions before 1.8. The first question is about HashMap of Java 1.6, which is outdated and cannot solve current users' doubts, while the second one is about HashMap of Java 1.8, which is popular and can effectively solve the doubts. In this sense, it requires question retrieval methods to find the question with high popularity to solve the users' problem more effectively. Thus, the motivation of this study is to construct an effective question retrieval methodology, which can retrieve the questions with both similar semantics and high popularity so as to enhance the user's experience in CQA.

Question retrieval models can be divided into two categories: statistical translation models and topic-based models. Statistical translation models learn the relevance between questions for the question retrieval. For example, IBM model1 translation model is a typical statistical translation model for question retrieval in CQA [6]. It estimates similarities based on the hypothesis of "parallel", which requires a fixed one-to-one correspondence among the words in two questions [7]. However, different questions are far from "parallel" in practice and asymmetric on their carried information [8], so this method could not achieve the best performance when facing "unparalleled" corpus. On the other hand, topic-based question retrieval models employ a topic model for retrieving similar questions without being constrained by the queried text forms [9], and thus they can make full use of semantic information contained in the questions when retrieving "unparalleled" questions. Zhou et al. [10] proposed Topic-enhanced Translation-based Language model to employ LDA to discover latent topic-based semantic knowledge, based on which similar questions were mined. Cai et al. [11] incorporated category information into the

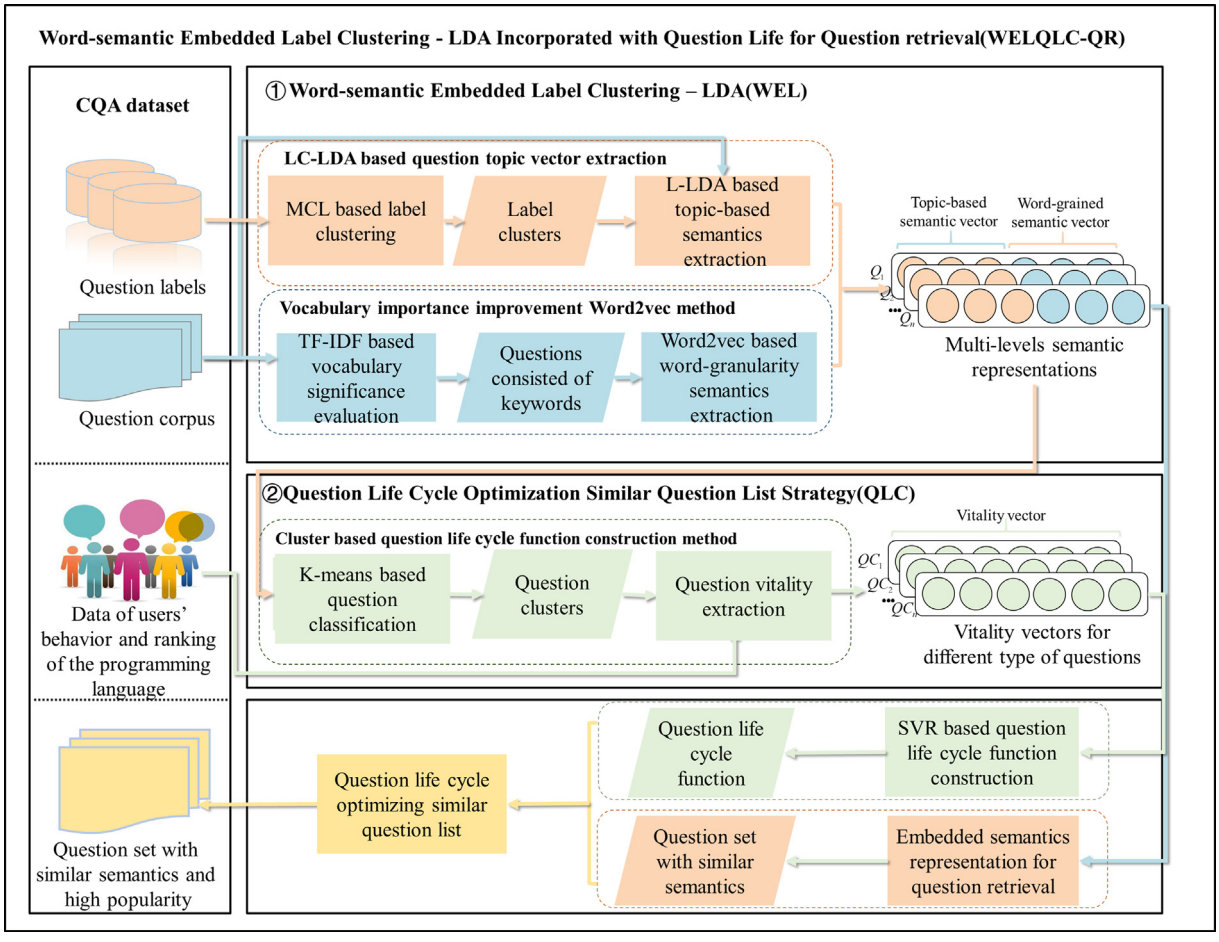


**Fig. 2.1.** The framework of WELQLC-QR.

process of extracting topic distribution using LDA so as to improve the performance of question retrieval. Chen et al. [3] employed LDA to make use of local and global semantic graph, and extracted the semantics of question for question retrieval. Assuming a question shares a same topic distribution with its answers, Ji et al. [12] learned topic vectors from question-answer pairs via LDA model, improving the accuracy of their retrieval method. Zhang et al. [13] proposed a supervised question–answer topic approach, which learned the topic of both question and answer using LDA for retrieving similar questions. These abovementioned works employ LDA to analyze the topics-based semantics of the question corpus for question retrieval. However, LDA relies on a word document co-occurrence matrix to extract its topic distribution [14]. When extracting the topics of short texts via LDA, the co-occurrence of words becomes sparse, which will lead to topic compulsory distribution, and thus the extracted topic vector cannot accurately represent the semantics of questions. Lately, Labeled-LDA model (L-LDA) [15], which defines a one-to-one mapping between LDA's latent topics and existing labels, was proposed to compensate the deficiency of LDA on topic distillation at a certain extent, because the classification results of L-LDA always belong to rather labels with determined and strong meanings than words weakly related to the corpus. Nevertheless, the labels in CQA usually show multi-levels features [16], which might cause over-fitting or over-generalization in the topic extraction via L-LDA, and thus it leads to inaccurate matching results in the question retrieval in CQA. Label Cluster-LDA (LC-LDA) proposed by Liu et al. [16] employed Markov Cluster Algorithm (MCL) [17] to cluster labels according to co-occurrence information between labels, to weaken the influence of multi-granularity labels during topic extraction. However, the topic vector extracted by LC-LDA is coarse-grained in semantics [18], leading to the generalization of question retrieval results. In order to comprehensively and adequately extract the semantics of questions, we can introduce a fine-grained model on the basis of topic-based question retrieval models. Word2vec [19] is a word-grained semantics extraction model compared with topic-based models, and can be utilized to encode questions into a low-dimensional continuous embedding space with word-level semantics. Chen et al. [5] proposed a novel framework which embeds the question contents extracted by Word2vec model and users' social interactions in order to learn the semantic representation of questions for question retrieval. Zhang et al. [20] transferred question contents into semantic representations using Word2vec, and constructed question correlation matrices to learn the similarity among questions. Shen et al. [21] integrated both translation model and Word2-vec for improving the accuracy of question representation and enhancing the effectiveness of question retrieval. Wang et al. [22] also proposed a hybrid semantic extraction approach utilizing the topic-based LDA incorporated with Word2vec to extract the semantics of texts from different granularities. The approach can improve the accuracy of text classification. Similarly, Zhang et al. [23] learned the distributed representation of questions using the combination of Word2vec and LDA for question retrieval. However, questions in CQA usually contain some noise vocabularies (e.g., "am", "we"), which have low correlation with the core semantics of questions. These almost unrelated words will result that the semantic vector extracted by Word2vec cannot accurately express the core semantics of the questions, leading to inaccurate question retrieval results in CQA, and thus should be removed during matching [24]. Othman et al. employed Term Frequency-Inverse Document Frequency (TF-IDF) for evaluating the importance of each word [25], so did Aggarwal et al. to find the keywords of texts [26], and then learned a meaningful vector representation of questions according to Word2vec. They treated the value of TF-IDF [27] as weighting coefficients when constructing the question semantic representation. However, the value of TF-IDF is usually much less than one, which might reduce the influence of keywords when constructing the question representation. Thus, how to further reduce the influence of noise vocabularies and highlight the impact of keywords for enhancing the accuracy of question retrieval is essential to this study.

Furthermore, the abovementioned translation models and topic-based models retrieve similar questions from the perspective of semantic analysis. In fact, popularity is another essential factor for similar-question retrieval. It is known that popular questions tend to gain more views and answers [28], which means that the knowledge carried by the popular questions can be recognized by most people and thus can solve the doubts of users. However, outdated questions can hardly attract attentions of the users in CQA, which means the knowledge carried by them is ineffective and cannot solve the problems of queried users. Therefore, it is necessary to consider the factor of popularity during question retrievals. Liu et al. [28] proposed a binary classification model for predicting whether a question is popular according to the content of the question, relevant user's questions and answer records, and user's community information. This approach models the popularity of a question according to the statistical information of questions and users, but it does not consider the factor of time and the fluctuation of questions' popularity. Product life cycle theory (PLCT) provides an approach to describe the fluctuation of product popularity over time, which represents how the product popularity changes during product life (i.e., the process from initial design to the market and finally being eliminated by the market) [29]. According to the statistics of visits and views of websites, Althoff et al. [30] constructed a life cycle prediction model based on PLCT for analyzing and predicting the popularity of network topics. Based on information such as browsing and commenting on news stories of websites, Castillo et al. [31] proposed a PLCT-based model to analyze the popularity of website news stories, i.e., to predict the access of the stories in the future. According to item's basic information such as online time and item type, Liu et al. [32] constructed an SVR-based item popularity analysis model in Collaborative Filtering, which improves the performance of recommendation lists and provides more reasonable and popular recommendation lists for users. When it comes to the popularity analysis of CQA questions using PLCT, we need to focus on three key points: the determination of modeling target, the selection of factors, and integrating strategy. Firstly, there are lots of questions in CQA. It is time-consuming to build an individual life cycle model for each question, while a single life cycle model for all questions cannot accurately express the trend of question popularity. Therefore, we need to determine a modeling target at first. Sequentially, there are many factors associated with the popularities of questions in CQA, which include not only internal factors (e.g., the number of comments and answers) [28]

but also external factors (e.g., the popularity of the programming language belonging to the question in Stack Overflow, a CQA toward the programmer). The selection of these factors determines the accuracy and quality of a question life cycle model for describing the fluctuation of question popularity, and thus it is of great importance in the model construction. Additionally, how to integrate the constructed question life cycle model into the question retrieval and how to further make the retrieved questions with similar semantics and high popularity are also essential to this study.

In this paper, we will propose an Integrated Retrieval Framework for Similar Questions named Word-semantic Embedded Label Clustering – LDA with Question Life Cycle (WELQLC-QR), in which both topic-based and word-grained semantics are taken into consideration when retrieving similar semantic questions, and question popularity is constructed as a question life cycle function to optimize the retrieved results of semantic matching. Therefore, our proposed WELQLC-QR method makes final results with not only similar semantics but also high popularity. It is worth highlighting the following contributions of our work.

(1) To overcome the problem of over-generalization of extracted topics when facing short questions with multi-levels labels in CQA, we employ LC-LDA for topic-based semantics extraction. LC-LDA employs a graph clustering method to cluster labels so as to weaken the influence of multi-granularity labels during topic extraction using the improved LDA based model.
(2) To avoid the influence of noise vocabularies and highlight the keywords of a question, we proposed a vocabulary importance improved Word2vec method for word-grained semantics extraction. It evaluates the vocabulary significance for filtering the noise vocabularies in CQA questions and further obtaining word-grained representations.
(3) To effectively retrieve similar questions from the perspective of semantic matching, WEL, embedding word vectors into LC-LDA method, is proposed to comprehensively and adequately extract semantics of a question. It embeds the word vector into LC-LDA to achieve the multi-levels semantic representation of the question to enhance the accuracy of question retrieval.
(4) To make the retrieved results with similar semantics and high popularity, we proposed Question Life Cycle Optimization Similar Question List Strategy (QLC) for incorporating popularity into question retrieval. According to the internal factors (e.g., the number of comments and answers) and external factors (e.g., programming language ranking information) of a question, QLC constructed a question life cycle function for characterizing the fluctuation of the questions' popularity, which is then used for popularity optimization and makes final retrieved results both semantic similar and popular.
(5) Finally, we carry out comparative experiments along with benchmark models (e.g., L-LDA, BM25, Word2vec) and counterpart methods on a published Stack Overflow dataset to demonstrate the effectiveness of our proposed method.

The rest of this paper is organized as follows. Section 2 details the main idea and key techniques of proposed WELQLC-QR. In Section 3, we describe the experimental settings and report a variety of results to verify the superiority of the proposed framework. Conclusions and possible future researches are presented in Section 4.

## 2. Method

In order to make retrieved questions with both similar semantics and high popularity, we propose Word-semantic Embedded Label Clustering-LDA Incorporated with Question Life for Question (WELQLC-QR), which can be divided into the following two modules (see Fig. 2.1):

(1) Word-semantic Embedded Label Clustering-LDA (WEL) is proposed for question retrieval from the perspective of semantic matching. WEL first extracts the topic-based semantic representation of a question using LC-LDA, which utilizes Markov Cluster Algorithm (MCL) to cluster question labels according to the co-occurrence information between labels so as to weaken the influence of multi-granularity labels. And then it employs L-LDA to obtain the distribution of question topic with the combination of question corpus. Furthermore, WEL extracts a word-grained question representation using Vocabulary Importance Improvement Word2vec method. It evaluates the vocabulary significance in the question corpus using TF-IDF for filtering noise words and obtaining questions with keywords. It further achieves a word-grained semantic vector using Word2vec, reducing the influence of noise vocabularies and highlighting the impact of keywords. WEL finally embeds the word vector into LC-LDA to achieve a multi-level semantic representation of the question, based on which WELQLC-QR completes the question retrieve from the perspective of semantic matching and obtains the question set with similar semantics.
(2) Question Life Cycle Optimization Similar Question List Strategy (QLC) introduces popularity to optimize the semantic retrieved results of WEL, making the retrieved questions with both similar semantics and high popularity. QLC first employs K-Means to classify questions according to the question topic distribution extracted by LC-LDA. Based on question clusters, QLC uses the data of users' behavior and ranking of a programming language to extract question vitality vectors in different periods. Then QLC constructs a question life cycle function for characterizing the fluctuation of questions' popularity using SVR. The constructed question life cycle function is eventually used to optimize the similar question set searched by WEL, so that the final retrieval results are both semantically similar and popular.

### 2.1. Word-semantics embedded label clustering-LDA method

In order to comprehensively and adequately extract semantics of a question, we propose WEL to extract the semantic vector of the question by topic-based and word-grained methods. From the perspective of topic-based extraction, the proposed method introduces LC-LDA [16], which can distinguish the granularity of tags with the integration of Markov clustering [17], so that it can solve problems carried by the different granularity of tags in CQA and make the extracted results consistent with the core semantics of questions. Additionally, the vocabulary importance improvement Word2vec method, which can identify the weight of each word in a question and filter noise words, is then embedded into LC-LDA to extract word-grained semantic vectors. Compared with single topic-based models, it makes the extracted results contain more semantic information to improve the accuracy of question retrieval.

#### 2.1.1. LC-LDA based question topic vector extraction

WEL first extracts question topic distribution using LC-LDA, which can distinguish the granularity of tags with the integration of Markov clustering. Using LC-LDA can solve problems carried by the different granularity of tags in CQA and make the extracted results consistent with the core semantics of questions.

Before using L-LDA Model for topic discovery, LC-LDA Topic Model [16] applies Markov clustering (MCL) to cluster labels through co-expression among labels, reducing the impact of inaccurate labels and low-frequency labels on expert classification.

MCL is a graph clustering algorithm based on random walk proposed by Van Dongen et al. [16,17]. The core idea of MCL is that the nodes within each cluster are densely linked and the inter-cluster nodes are sparsely linked, and thus the probability of a traverser walking in the cluster is greater than that walking between adjacent clusters during its random walking from a node in the graph. In this paper, labels are clustered by using co-expression among labels, and the label with the highest co-expression number is clustered as the label cluster. We first define a probability transfer matrix for the random walk, named the label co-expression probability matrix $C$. Assume there are $L$ questions in CQA, then the element of $C$ is defined as Eq. (2-1).

$$c_{ij} = \sum_{l=1}^{L} c_i^l c_j^l \tag{2-1}$$

where $c_{ij}$ is the probability of labeling a same question with tag $t_i$ and $t_j$ in CQA, $c_i^l$ and $c_j^l$ indicate the number of $t_i$'s and $t_j$'s tagging to the *l-th* question, respectively.

The MCL-based label clustering algorithm alternates between "expansion" and "inflation" operations on the label co-expression probability matrix until the resulting matrix converges. The "expansion" operation refers to the product of the label co-expression probability matrix, as shown in Eq. (2-2).

$$C_{exp} = Expand(C) = CC = C^2 \tag{2-2}$$

where $C$ represents the probability transfer matrix, and $C_{exp}$ is the result of the "expansion". From the view of the random walk, the calculated $C_{exp}$ represents the probability that a walker stays at each node at the next moment.

The "inflation" operation is the process of stopping the "expansion" operation. It expands each element of the label co-expression probability matrix to a power of $r$ and then normalizes each column. When the label co-expression probability matrix $C$ converges, there is no transfer path between the labels in different clusters. Thus the final clustering result can be interpreted as follows: the presence of a non-zero value $c_{ij}$ in the $j$th column indicates that label $v_i$ is the attractor (i.e., the cluster center) of label $v_j$, and label $v_i$ and $v_j$ belong to a same class.

The cluster vector $\Lambda'^{(d)} = (l'_1, \cdots, l'_C)$ affiliated with labels after label clustering, where $l_k, l'_c \in \{0, 1\}$, is added as a new question label vector to the probability graph model of LC -LDA model (Fig. 2.2). The hollow circle in the figure represents the potential variable, the shaded circle represents the observable variable, the arrow represents the conditional dependency between two variables, the box represents the repeated sampling, and the variable at the bottom right of a box represents the times of repeated sampling. In the figure, $\alpha$ and $\beta$ are the Dirichlet super parameters for the topic and the word, respectively, $\eta$ is the Bernoulli super parameter for the label, $\phi$ and $\theta$ are the polynomial distribution of the word and the topic, respectively, $z$ represents the topic, $w$ means the observable word variable, and $\Lambda^{(d)} = (l_1, \cdots, l_K)$ represents the original label vector of the question.

Given a set of tagged questions, as for problem $m$, the word vector $\vec{w}_m$, the original label vector $\Lambda_m$ and the label cluster vector $\Lambda'_m$ with $K$ dimensions are observable variables. $\vec{\alpha}$ and $\vec{\beta}$ are the given Dirichlet priori parameters based on experience. Topic distribution $\vec{\theta}_m$, word distribution $\phi_{z_{m,n}}$, topic vector $\vec{z}_m$ and Bernoulli super parameter $\eta_m$ of the label cluster are unknown implicit variables. Because $\Lambda$ and $\Lambda'$ are observable, $\eta_m$ can be isolated from other parameters of the model. Thus, LC-LDA model only needs to estimate $\vec{\theta}_m$, $\phi_{z_{m,n}}$ and $\vec{z}_m$. This study uses Gibbs Sampling to estimate the potential parameters of LC-LDA model by the following steps.

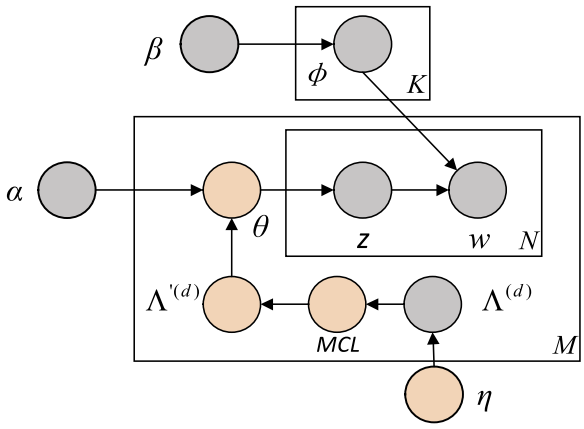


**Fig. 2.2.** Probability diagram model of LC-LDA.

(1) Topic $z^{(0)}$ is assigned to each word in the question randomly at initial time.
(2) Count the number of feature words that appear under each topic $z$ and the number of words on topic $z$ that appear in each question. The posterior probability $p(z_i = k|z_{\neg i}, w)$ of topics $z_i$ is calculated at each round. That is, eliminating the topic allocation of the current word, calculating the topic distribution of the current word based on that of other words, and randomly selecting a new topic $z^{(j)}$ for the word based on this probability distribution.
(3) Repeat Step (2) to update the topic distribution of the next word, until both the topic distribution $\overrightarrow{\theta}$ under each question and the word distribution $\phi$ under each topic are convergent.

The clustered labels $\Lambda'^{(d)}$ of the question are observed. Given $\Lambda'^{(d)}$, the labeling prior $\eta$ is separated from the rest of model. Therefore, the model is the same as LDA model [15], except for the constraint that the topic prior $\alpha$ is combined with the clustered labels $\Lambda'^{(d)}$. The posterior topic probability for word $w$ in the document is evaluated via Eq. (2-3) [33].

$$p(z_i = k|z_{\neg i}, w) \propto \frac{\mathrm{n}_{k,\neg i}^{(w_i)} + \beta_{w_i}}{\sum_{t=1}^{V} \mathrm{n}_{\neg i,k}^{(\cdot)} + \beta^T \mathbf{1}} \cdot \frac{(\mathrm{n}_{k,\neg i}^{(m)} + \alpha_k)}{\sum_{t=1}^{V} \mathrm{n}_{\neg i}^{(m)} + \alpha^T \mathbf{1}}, \quad k \in \overrightarrow{\lambda}^{(m)} \tag{2-3}$$

where $z_i$ represents the topic variable of the $i$th word, $\neg i$ means the items other than the $i$th item, $n_k^{w_i}$ represents the times of word $w_i$ appearing in the $k$th topic, $\beta_{w_i}$ represents the Dirichlet priori of word $w_i$, $\mathrm{n}_{\neg i,k}^{(\cdot)}$ represents the sum of dimensions, $\mathbf{1}$ means the vector with the same value of one, $n_k^m$ represents the times of the $k$th topic appearing in the $m$th question, $\alpha_k$ represents the Dirichlet priori of the $k$th topic, and $k \in \overrightarrow{\lambda}^{(m)}$ indicates the limitation of the label factor on the distribution of question topics.

From the above analysis, the other two parameters to be estimated are $\overrightarrow{\theta}$ and $\phi$. When Gibbs Sampling converges, these two parameters can be calculated by Eq. (2-4) and (2-5).

$$\phi_{k,w_i} = \frac{n_{k,\neg i}^{(w_i)} + \beta_{w_i}}{\sum_{t=1}^{V} \mathrm{n}_{\neg i,k}^{(\cdot)} + \beta^T \mathbf{1}} \tag{2-4}$$

$$\theta_{m,k} = \frac{n_{k,\neg i}^{(m)} + \alpha_k}{\sum_{k=1}^{K} \mathrm{n}_{\neg i}^{(m)} + \alpha^T \mathbf{1}} \tag{2-5}$$

where $\phi_{k,w_i}$ represents the probability that word $w_i$ belongs to topic $k$, and $\theta_{m,k}$ indicates the probability that question $q^m$ belongs to topic $k$.

Given a set of questions $Q$ in CQA and a new online question $q$, where $Q$ includes all questions asked by respondents and questioners, the category distribution $Z^{(q^m)} = \left(z_1^{(q^m)}, z_2^{(q^m)}, \cdots, z_K^{(q^m)}\right)$ in $K$ topics of question $q^m$ can be extracted based on the above Step (1), (2) and (3) to discover question topics by LC-LDA. $z_i^{(q^m)}$ represents the probability that $q^m$ belongs to topic $i$.

*2.1.2. TF-IDF based vocabulary significance evaluation*

After the extraction of question topics, WEL then tries to extract the word granularity semantic representation of the question. However, there often exist noise words in CQA questions which would influence the effectiveness of semantic rep-

resentation. Therefore, WEL utilizes Term Frequency-Inverse Document Frequency (TF-IDF) method to evaluate the vocabulary significance and filter the noise words.

TF-IDF is a popular weighting technique for information retrieval and data mining [27]. It evaluates how important a word is to a text set or a corpus file by counting the times of the word appearing in the text. It considers not only the times a word appearing in a file, but also how often it appears in the entire corpus. Contrary to the traditional idea that higher occurrence frequency means more importance, TF-IDF comprehensively considers how often words appear in text and corpus. Its main idea is that when a word appears more frequently in the corpus, it indicates that the word is most likely a regular word (e.g., "the", "a" and "is"), which does not show the uniqueness of the text very well. However, if a word or phrase appears frequently in an article and rarely in the corpus, it is considered to have a good category distinction and is appropriate for text retrieval.

Therefore, for the set of questions $Q_m$ with similar topics in the previous section, TF-IDF algorithm is used in this section to calculate the vocabulary importance of each word in $Q_m$. According to Huang et al. [34], vocabulary importance is defined as follows.

**Definition 2. −1 Vocabulary Importance:** Vocabulary Importance refers to the importance of each word in expressing the semantics of text.

The steps for TF-IDF to calculate the vocabulary importance are as follows.

(1) For each question $q$ in the set of questions $Q_m$, the Term Frequency (TF) value of the word in $q$ is calculated first. TF refers to that each word $t$ is given a weight based on the times that it appears in question $q$ (Eq. (2-6)).

$$tf(t,q) = \frac{c_{t,d}}{N_q} \tag{2-6}$$

where $c_{t,d}$ represents the times that word $t$ appears in question $q$, $N_d$ means the total number of words in question $q$, and $N$ is the total number of questions in the question set. The higher is the TF value, the more important of word $t$ is.

(2) The IDF value of word $t$ is evaluated through Eq. (2–7).

$$idf(t) = \log(\frac{N}{n_t + 1}) \tag{2-7}$$

where $n_t$ represents the number of questions containing word $t$, and the "+1" introduced in the denominator is a smoothing factor to prevent the extreme scenario that a rarely used word never appears in the entire set of questions. Similarly, the higher is the IDF value, the higher the importance of the word $t$ is.

(3) As a result, the vocabulary importance of word $t$ is evaluated through Eq. (2–8).

$$tf - idf(t) = tf(t,q) \times idf(t) \tag{2-8}$$

Therefore, for the set of words $T_q = (t_1, t_2, \cdots, t_K)$ in question $q$, where $K$ is the number of words, the corresponding vector of vocabulary importance is $TF - IDF_q = (tf - idf(t_1), \cdots, tf - idf(t_K))$. We further delete 20% words with the lowest TF-IDF value in the question to filter noise words and avoid their influence. As a result, the question set after filtering noise words $Q'_m = \{ q'_1, q'_2, ..., q'_m \}$ is achieved.

*2.1.3. Vocabulary importance improvement Word2vec*

Through TF-IDF method, WEL filters noise vocabularies and obtains the questions consisting of words with high importance, based on which WEL then extracts word-grained representation using Word2vec.

Word2vec language model is an efficient tool developed by Google in 2013 to convert words into real-value vectors. Not only can it be efficiently trained in dictionaries of millions of orders of magnitude and hundreds of millions of data sets, but also it provides trained results of word vectors, which can measure the similarity between words very well [19].

In Word2vec training model, language models, including Skip-Gram model and Continuous Bag-of-Words (CBOW) model, are used to effectively train word vectors [35,36]. Both models contain input, output and mapping layers, but the functionalities of these two models are different. Skip-Gram model is given a word and then predicts the appearance probability of the word in the context, while CBOW model proceeds along the opposite direction. CBOW model uses the words in the context to predict the appearance probability of the current word. In terms of improving training efficiency, Word2vec provides two optimizations, Herachy SoftMax (HS) and Negative Sampling (NS). Since the training of Skip-Gram model is similar to that of CBOW model, CBOW + HS model is introduced here to obtain the intermediate result of word vectors trained in the set of questions. The output layer is the vector of target word $w_i$, which can be represented as $V_{\bar{w}} = (\bar{x_1}, \bar{x_2}, \cdots, \bar{x_l})$. The input layer is vector $V_w$ in the context of target word $w_{(i)}$.

$$V_w = \{V_{w(i-c)}, V_{w(i-c+1)}, \cdots, V_{w(i+c)}\}, V_{w(i\ -\ c+j)} = (x_1, x_2, \cdots, x_l) \tag{2-9}$$

where undetermined parameter $c$ represents the size of a window, and $x_i$ is the $i$th vector of word $w_{(i)}$. Additionally, Bag-of-words is employed, and word $w_{(i\ -\ c\ +\ j)}$ is transferred to a one-hot vector before model training.

In the output layer of CBOW + HS model, it changes from the SoftMax layer of traditional neural network word vector language models to a binary Huffman Tree. All internal nodes of the Huffman Tree are similar to the neurons in the previous hidden layer of the neural network, where the word vector of the root node is the output vector $v_{\bar{w}}$ of the hidden layer, and all leaf nodes $leaf = (w_1, w_2, \cdots, w_V)$ constitute the entire vocabulary, which are similar to the neurons in the SoftMax output layer of the neural network. The direction of the Huffman Tree from root node to child node is based on a binary logic regression method, which determines whether the data of each node is moved to the left sub-tree or the right one by calculating its moving probability through a sigmoid function (Eq. (2-10)).

$$P(d_j^w|v_{\bar{w}}, \theta_{j-1}^w) = \sigma(v_{w^-}^T \theta_{j-1}^w) = \begin{cases} \frac{1}{1+e^{-v_{\bar{w}}^T \theta_{j-1}^w}}, d_j^w = 0 \\ 1 - \frac{1}{1+e^{-v_{\bar{w}}^T \theta_{j-1}^w}}, d_j^w = 1 \end{cases} \tag{2-10}$$

where $d_j^w$is the Huffman Code of node $j$, and $\theta$ is the model parameter corresponding to node $j-1$ according to the training data. If node $j$ is the left child node of the parent node of node $j$, its Huffman Code is 1. And it is coded as 2 when it is the right child node of the parent node of node $j$. Then its corresponding formula is selected to calculate its moving probability. The maximum-likelihood function is employed for the construction of loss function, which is defined as

$$L = \prod_{j=2}^{l_w} P(d_j^w|v_{\bar{w}}, \theta_{j-1}^w) \tag{2-11}$$

where $l_w$ is the total number of leaf nodes in the Huffman Tree. Then random gradient ascending method is used to iteratively update the $v_{\bar{w}}$ and $\theta_{j-1}^w$ in the likelihood function $L$. In summary, the following are the steps for the establishment of CBOW + HS model.

(1) Establish a Huffman tree based on the corpus-training sample of text set $Q_m$.
(2) Randomly initialize the model parameter $\theta$ and the word vector $v_{w(i)}$ of all internal nodes of the Huffman tree.
(3) For each sample $(w(i), v_{w(i)})$ in the training set, proceed the following:
(4) Firstly, use Eq. (2–9) to calculate the output vector $v_{\bar{w}} = (\bar{x_1}, \bar{x_2}, \cdots, \bar{x_l})$ of hidden layer by averaging the sum of the context word vector of input layer, which is also the word vector of the root node of the Huffman Tree.
(5) When the word vector of the root node is transferred to the Huffman Tree, gradient ascending method is used to update the context word vector $\bar{x_i}\,(i = 1, 2, \ldots, 2c)$ and the model parameter $\theta_{j-1}^w$via

$$\theta_{j-1}^w = \theta_{j-1}^w + \eta(1 - d_j^w - \sigma(v_{w^-}^T \theta_{j-1}^w))v_{w^-}^T \tag{2-12}$$

and

$$v_{w^-}^T = v_{w^-}^T + \eta \sum_{j=2}^{l_w} (1 - d_j^w - \sigma(v_{w^-}^T \theta_{j-1}^w))\theta_{j-1}^w, \quad i = 1, 2, \ldots, c \tag{2-13}$$

where $\eta$ is the step size of gradient ascending method.

(6) If the gradient converges, then the gradient iteration ends and the algorithm ends. Otherwise, go back to step (4).

When the gradient converges, one can obtain the word vector for each word of noise-word-filtered question set $Q'_m$ and question $q'$. The vector of the question is the average of the sum of each word vector in the noise-word-filtered question (Eq. (2-14)).

$$\overrightarrow{V}'_{q^{(m)}} = \frac{\sum_{i=1}^{n} \overrightarrow{v}_{w(i)}}{n} \tag{2-14}$$

where $n$ is the number of words in the question. Thus the final word importance filtered word-grained vector of the question can be represented as $\overrightarrow{V}_{q^{(m)}}{}' = (\bar{x}_1^{(q^m)}, \bar{x}_l^{(q^m)}, \cdots, \bar{x}_l^{(q^m)})$, where $\bar{x}_l^{(q^m)}$ denotes the $l$th dimension average value over all the $l$th dimension of important words in question $q^m$.

*2.1.4. Embedded semantics representation for question retrieval*

WEL obtains question topics through LC-LDA and question representations through vocabulary importance improvement Word2vec method, based on which WEL then constructs the embedded semantics representation for each question and completes the question retrieval.

Through the LC-LDA based question topic vector extraction method and vocabulary importance improvement Word2vec method, we can extract the topic-based vector $Z^{(q^m)} = \left(z_1^{(q^m)}, z_2^{(q^m)}, \cdots, z_K^{(q^m)}\right)$ and the word-grained vector $\overrightarrow{V_{q^{(m)}}}' = (\bar{x}_1^{(q^m)}, \bar{x}_l^{(q^m)}, \cdots, \bar{x}_l^{(q^m)})$ of questions. In order to accurately retrieve similar questions, we embed word-semantics into Label Clustering – LDA through the combination of these two different granularities question representations, and formulate a comprehensive question semantic description vector $Q_{q^{(m)}} = (\bar{x}_1^{(q^m)}, \bar{x}_l^{(q^m)}, \cdots, \bar{x}_l^{(q^m)}, z_1^{(q^m)}, z_2^{(q^m)}, \cdots, z_K^{(q^m)})$. The new question $q$ is then used to calculate its semantic vector similarity with question $q$' in problem set $Q$ (Eq. (2-15)).

$$Sim_{Z_{q,q'}} = \frac{\sum_{l=1}^{L} x_{q,l} \cdot x_{q',l}}{\sqrt{\sum_{l=1}^{L} x_{q,l}^2}\sqrt{\sum_{l=1}^{L} x_{q',l}^2}} + \frac{\sum_{k=1}^{K} z_{q,k} \cdot z_{q',k}}{\sqrt{\sum_{k=1}^{K} z_{q,k}^2}\sqrt{\sum_{K=1}^{K} z_{q',k}^2}}, q' \in Q \tag{2-15}$$

where $L$ is the length of the question vector, and $x_{q,l}$ is the first value of the problem vector of question $q$. $Sim_{Z_{q,q'}}$ is the similarity between problem $q$ and $q'$, which is calculated by cosine similarity. Then, after sorting the resulting score set $Sim_Q = (Sim_{z_{q,q1}}, \cdots, Sim_{z_{q,qn}})$ of the topic similarity of questions, we achieve the top $m$ question set $Q_m$ in the score of semantic similarity.

The above is the process of using Word-semantics Embedded Label Clustering – LDA method to discover a question set semantically similar to the questions to be solved.

### 2.2. Question life cycle optimizing similar question list strategy

Existing semantic-analysis-based methods often neglect the influence of questions' popularity in question retrieval, resulting in the fact that questions with outdated knowledge are recommended to users. In this section, we propose Question Life Cycle Optimizing Similar Question List Strategy, the second component of WELQLC-QR, to make retrieval results with both similar semantics and high popularity.

Based on product life cycle theory, we propose QLC to incorporate popularity into question retrieval. In practice, there are a large number of questions in CQA. Building an individual life cycle model for each question is time-consuming, while a single life cycle model for all questions cannot accurately express the trend of question popularity. Therefore, we need to determine the modeling target first. Moreover, there are many factors associated with the popularity of questions in CQA, which include both internal factors (e.g., the number of comments and answers) and external factors (e.g., the popularity of the programming language belonging to the question in Stack Overflow, a CQA toward the programmer). The selection of these factors determines the accuracy and quality for question life cycle models to describe the fluctuation of question popularity, and thus it is of great importance during model construction. At the same time, how to integrate it into question retrieval and make retrieved questions both similar in semantics and highly popular is also essential to this study. Therefore, in this section, we propose a question life cycle function construction method. This method classifies questions into different categories according to semantics, then comprehensively considers internal and external factors to construct a problem life cycle curve for each category of questions, and finally optimizes the list sequences of similar questions from the previous section.

#### 2.2.1. Definition and depiction of question life cycle

Before the construction of question life cycle, we first present the definition and depiction of question life cycle. Question life cycle is defined based on traditional product life cycle theory.

**Definition 2. −1 Question Life Cycle (QLC)**: QLC refers to the process of answering, browsing and commenting on new questions from the first time they are uploaded online.

General question life cycle consists of four stages of development: introduction period, growth period, maturity period, and recession period. Modeling question life cycle is essential in the task of time series model construction. There are two major considerations: the determination of intervals and the computation of effective interval lengths [37]. According to the method from Liu et al. [16], we analyze the number of respondents in a Stack Overflow question and answer community in different time periods, as shown in Fig. 2.3. By analyzing the behavior of users in CQA, we set the time unit in the horizontal directions to 16 min, i.e., one unit time is 16 min. When a question is just online, it is at the begging of its introduction period. During this period, the number of times that the question is viewed or answered by others is increasing, and the curve growth rate is slow. However, as this kind of questions gradually becomes popular in the community, the slope of the curve increases, which means that the number of respondents gradually increases and the question begins to be changed from introduction period to growth period, reflecting the change in the number of response in the period from 12 to 19 in Fig. 2.3. When such questions have been answered or commented by interested users in most communities, the life cycle of the question is in maturity period, during which the number of curve responses does not increase any longer. Finally, when users think that the question has a satisfactory solution and is no longer attractive, the responsiveness curve of the problem begins to tilt downwards, and it comes to the decline phase of problem life cycle (i.e., recession period).

Question life cycle is not only affected by the type of a question, but also by many other factors, such as the upload time of the question and whether the content of the question matches the direction of mainstream interest in the community.

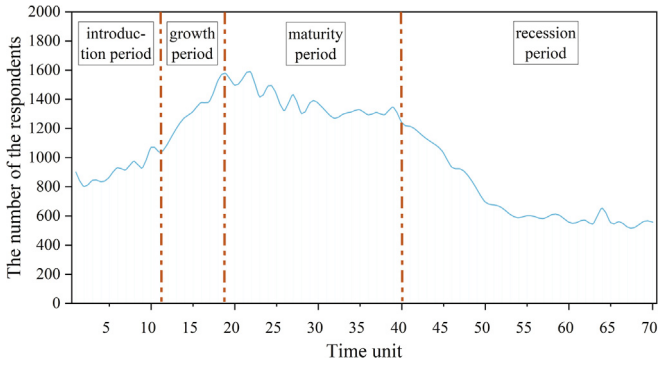


**Fig. 2.3.** Curve of the number of responses of one certain category.

Therefore, researchers use appropriate regression analysis methods [38] to construct life cycle models based on the development patterns of specific problems. For example, in order to solve the problem that is difficult to predict the decline point of the life cycle curve of footwear products, Liu et al. [38] proposed a footwear product demand forecasting system that combines wavelet transform with polynomial fitting based on artificial bee colony algorithm.

Although these proposed life cycle models are based on the characteristics of marketing and retail management, they tried to determine the quantitative relationship among two or more variables that affect the popularity of items based on the prevalence of product, which is the essential idea of regression analysis. The premise of life cycle regression analysis on problems is to obtain the popularity of questions at different times. The popularity of a question at different times can be replaced with the vitality of the question. Question vitality (QV) is defined as the following.

**Definition 2.** −**2**: Question vitality refers to the quantitative popularity of a question at one time point throughout the life cycle of the question.

Because most of the existing methods for calculating the vitality of items are based on the internal factors of items (such as the number of purchases in the user's unit time), the main direction of their environment is often neglected. Therefore, this study considers question vitality calculation mainly from the following two aspects.

(1) Internal factors. Internal factors mainly include the number of answers and comments. According to the number of respondents and that of visits within a unit time, the attention degree of users in CQA can be directly reflected, which also indicates the popularity of the question.
(2) External factors. External factors are mainly the programming language ranking of the question. Since questions on Stack Overflow are mostly IT-related technical problems of programming, the more popular is the technology, the more people will learn it. Thus, more and more related problems will appear on Stack Overflow. We use TIOBE programming language leaderboard [39] to reflect the mainstream interest in CQA. TIOBE programming language leaderboard is an indicator of the popularity of programming languages, and it is updated monthly. Its ranking is based on the number of experienced programmers, courses and third-party vendors on the Internet. Rankings are calculated using well-known search engines such as Google, MSN, Yahoo!, Wikipedia, YouTube, and Baidu.

As the type, body length and topic popularity of each question are different, the length of each question's life cycle is also different. Therefore, determining the length of question life cycle is also very important for characterizing the question life cycle besides considering the factors that reflect the question vitality. The length of question life cycle is determined in experimental settings section.

*2.2.2. Cluster-based question life cycle function construction*

After the definition of question life cycle, QLC constructs a cluster-based question life cycle function by considering internal information.

Given a question set $Q$ in CQA, we use LC-LDA introduced in Section 2.1.1 to extract the probability distribution of topic $q$ for each question $Z^{(q)} = \left(z_1^{(q)}, z_2^{(q)}, \cdots, z_K^{(q)}\right)$ in a question set. Because building life cycle functions for each question is extremely complicated, K-means clustering [40] is used to cluster questions into different categories according to the topic distribution of the questions. The basic idea of K-means clustering algorithm is to find $N$ cluster centers $\{c_1, c_2, \cdots, c_N\}$, so that the sum of the squares of the distances between each data point and its nearest cluster center is the smallest, where $c_i$ is the $i$th cluster center, and the distance function is described in Eq. (2-16).

$$Dis(q_i, q_j) = \frac{\sum_{n=1}^{N} z_n^{q_i} \cdot z_n^{q_j}}{\sqrt{\sum_{n=1}^{N} (z_n^{q_i})^2} \cdot \sqrt{\sum_{n=1}^{N} (z_n^{q_j})^2}}, \tag{2-16}$$

where $Dis(q_i, q_j)$ represents the distance between question $q_i$ and question $q_j$ across all subjects, and $z_k^{q_i}$ is the probability that question $q_i$ belongs to the $k$th subject. Then, according to the two aspects of calculating question vitality, question life cycle time $T$ is equally divided into $n$ time units in the order of time. The vitality vector $P_i$ of the $i$th question cluster is constructed to represent the life cycle trend of the question cluster (Eq. (2-17)).

$$P_i = (p_{i1}, p_{i2}, ..., p_{in}), \tag{2-17}$$

where $p_{im}$ denotes the activity value of the $i$th problem cluster at the $m$th time unit (Eq. (2-18)).

$$p_{im} = \frac{u_{im} + s_{im}}{num_i}, \tag{2-18}$$

where $u_{im}$ is the number of times that the questions belonging to the $i$th cluster are answered in the $m$th unit time, and $s_{im}$ is the number of times that the questions belonging to the $i$th cluster are commented in the $m$th unit time. With the above methods, the vitality vector set $P = (P_1, \ldots, P_b)$ of each question cluster in the question and answer community can be obtained, where $b$ is the number of question clusters. Since the number of question clusters after clustering in the previous section is small, there often exist some noises, such as outliers. The traditional curve fitting based on least squares is sensitive to these noisy points, so under-fitting often occurs. In order to avoid the under-fitting problem, a quadratic fitting or a higher-order fitting is generally used, and high-order terms are added after a first-order linear equation, which then could cause over-fitting. On the other hand, Least Squares Support Vector Machines (LSSVM) [41] can solve problems with small sample size, and has the characteristics of minimum over-fitting and strong generalization. Therefore, this study uses LSSVM to energize each problem cluster, whose learned vectors are used to construct a problem lifecycle function set $I = (I_1, \ldots, I_b)$.

*2.2.3. Question life cycle optimizing similar question list*

After the construction of the question life cycle model using external information, QLC further introduces external factors in order to accurately describe the popularity of questions in the current CQA environment. Finally, QLC introduces a constructed popularity model into question retrieval to optimize the similar question list and make retrieved results with similar semantics and high popularity.

The programming language ranking information involved in a question is treated as the external information and utilized when using life cycle function to calculate the vitality of problem at a certain moment. TIOBE programming language ranking information for the question is shown in Table 2.1.

The ranking scores for the programming language rankings are discrete, and the ranking scores of languages differ a lot. For example, in January 2019, "Java" language ranks 1, while "Objective-C" language ranks 9. Therefore, in order to narrow the gap between language rankings, a language ranking correction function (Eq. (2-19)) is proposed to improve the problem programming language ranking score.

$$rank' = \log\left(\frac{1}{1 + rank}\right), \tag{2-19}$$

where *rank* represents the rank of the programing language of a question. As a result, the higher is the *rank*, the higher *rank'* is.

Therefore, the vitality value $v_q$ for question $q$ at time $t$ is defined as

$$v_q = I_q(t) \cdot rank', \quad I_q \in I \tag{2-20}$$

Finally, we use $v_q$ to optimize the similarity score of similar question set $Q_m$ obtained in Section 2.2.2 via Eq. (2-21).

**Table 2.1**
TIOBE programming language ranking information in January 2019 [39].

| Jan 2019 | Jan 2018 | Change | Programming Language |
|---|---|---|---|
| 1 | 1 | | Java |
| 2 | 2 | | C |
| 3 | 4 | ↑ | Python |
| 4 | 3 | ↓ | C++ |
| 5 | 7 | ↑ | JavaScript |
| 6 | 6 | | C# |
| 7 | 5 | ↓ | PHP |
| 8 | 9 | ↑ | SQL |
| 9 | – | ↑ | Objective-C |

$$v_q = \alpha \cdot Sim_{v_{q,q'}} + (1-\alpha) \cdot v_q, \tag{2-21}$$

where $\alpha$ is a correction factor to determine the final weights of the question semantic similarity score and the question activity score. Through ranking the similar question set $Q_m$ obtained in Section 2.1.4 using the calculated score, we can achieve a final similar question set $Q'_m$, which contains questions with both similar semantics and high popularity.

## 3. Experiments

In order to evaluate the effectiveness of our proposed framework for question retrieval, we will conduct a series of comparative experiments on a realistic CQA dataset.

### 3.1. Experimental dataset

The CQADupStack dataset collected by Doris Hoogeveen from the University of Malborne is employed for experiments in this study [42]. The dataset consists of records of users' interaction from August 2008 to September 2014 in the Programmers Forum of StackExchange. Furthermore, these records can be divided into three different parts: questions, answers and comments. They can be used to construct questions life cycle and analyze users' behaviors in CQA from different perspectives.

When modeling the life cycle of questions, our method needs to count both the number of times that a question is answered by others and the number of comments. However, not all questions are answered or commented by users. Thus, to facilitate the analysis, we removed those unresolved questions. The final data set contains records of 31,034 questions, 120,772 answers and 308,695 comments from 16,014 users.

Additionally, in order to verify the effectiveness of the methods of similar question retrieval, 234 questions with similar question labels are chosen for testing and other 29,623 questions are used for training the model.

### 3.2. Evaluation settings

#### 3.2.1. Evaluation metrics of question cluster

In the question clustering in Section 2.2.2, as K-means clustering algorithm needs to set the number of clustering manually, Calinski-Harabaz (CH) index [43] (Eq. (3-1)) is used to calculate and evaluate the cohesion of clustering, so as to obtain the best clustering result.

$$CH(k) = \frac{tr(B_k)}{tr(W_k)} \frac{m-k}{k-1}, \tag{3-1}$$

where $m$ is the number of questions that need to be clustered in the questions set, $k$ denotes the number of clusters, $B_k$ denotes the covariance matrix between different categories, and $W_k$ denotes the covariance matrix of data within a category. When $W_k$ is smaller and $B_k$ is greater, CH index is greater, indicating that better clustering result is achieved.

#### 3.2.2. Evaluation metrics of similar question retrieval

P@N (precision at position N) and MRR (mean reciprocal rank) are used to evaluate the effectiveness of methods for question retrieval.

(1) P@N

P@N is a common metrics to evaluate the effectiveness of questions retrieval in CQA. For a queried question set $Q$, P@N (Eq. (3-2)) refers to the average accuracy of the first $N$ similar problems matched by the question retrieval method.

$$P@N = \frac{1}{|Q|} \sum_{q \in Q} \frac{N_q}{N}, \tag{3-2}$$

where $N_q$ represents the number of correct similar questions retrieved according to question $q$ in $N$ questions. In question retrieval, users usually only focus on the first several top searched results, so P@1, P@5, P@10 and P@15 are used for the evaluation in experiments.

(2) MRR

MRR is a statistical metrics used in information retrieval to evaluate a list of possible similar problems generated by a question, and the problems are sorted by their probabilities of correctness (note: the correctness refers to the similarity score of the retrieval question). For queried question set $Q$, its MRR (Eq. (3-3)) is calculated by taking the reciprocal of the ranking of the correct similar questions among the first $N$ retrieved questions as its accuracy, and then averaging it across all the queried questions.

$$MRR = \frac{1}{|Q|}\sum_{q\in Q}\frac{1}{Rank_q} \tag{3-3}$$

Similar to P@N, the number of similar question lists is 1, 5, 15 and 20. Their MRR expressions are MRR@1, MRR@5, MRR@10 and MRR@15, respectively.

### 3.3. The other methods for comparisons

In order to evaluate the performance of the WELQLC-QR proposed in this study, we set up a few comparison models according to two perspectives.

1) Benchmark models: The following six state-of-art models are used to validate the superiority in prediction accuracy and time efficiency of the proposed WELQLC-QR.
   - BM25: BM25 is a benchmark search model for Lucene that retrieves similar questions by calculating the correlation between questions [44]. It first separates a queried question into a set of words, and then calculates the relevance score between each separated word of the set and that of the questions in the corpus. Finally, all correlation scores are weighted and summed to obtain the correlation score between the queried problem and the problem in each corpus, so as to find the most relevant questions.
   - L-LDA-QR: L-LDA defines a one-to-one mapping between LDA's latent topics and existing labels, and it is used to compensate for the deficiency of LDA on topic distillation at a certain extent. In this study, L-LDA is used to extract question topics by introducing label information of CQA. The extracted topics are then utilized for question retrieval via similarity calculation methods.
   - LC-LDA-QR: Based on L-LDA topic model, this question retrieval method integrates tag cluster information after clustering question tags to obtain the topic distribution of questions, avoiding the over-fitting caused by low-frequency tags. Finally, the topic distribution is used to calculate the similarity between the queried questions and the historical questions in the corpus.
   - WET-QR: Word embedding with TF-IDF method for question retrieval (WET-QR) was proposed by Othman et al. [25]. This method treats TF-IDF value as the weight to construct question representations using Word2vec, which is different from our TW-QR method.
   - RCM-QR: Recurrent and convolutional model for question retrieval (RCM-QR) was proposed by Tao et al. [45]. RCM-QR employs a framework of encoder and decoder to map a question into semantic representations. Then question retrieval experiments are carried out based on RCM-QR.
   - Word2vec-QR: This method employs Word2vec to vectorize all the words of questions, and then sums each word vector to a question vector. Based on question vectors, the method calculates the similarity between the questions so as to obtain the similar question set to the query question.
2) Counterpart models: There are three models to verify the effectiveness of each part of the proposed framework.
   - Vocabulary importance improvement Word2vec method (TW-QR): In this proposed method, we employed TF-IDF to filter noise words, deleting 20% words with the lowest TF-IDF value in a question. Based on the filtered question corpus, we embed the question to the semantic representation through averaging the Word2vec-extracted vectors of the remaining words.
   - WEL: This counterpart model retrieves the questions from the perspective of semantics. It integrates LC-LDA, Word2vec model and TF-IDF, which is describe in Section 2.3.1.
   - WELQLC-QR: This method is the semantic question retrieval method based on question life cycle proposed in this study.

### 3.4. The procedure of experiment and parameter setting

The procedure of experiments is introduced as the following. First, MCL is utilized for tags clustering according to the co-occurrence probability matrix of the tags of questions. Based on the clustering results, L-LDA algorithm is used to extract the topic distribution of questions and answers, and the questions are classified according to their topic vectors. Additionally, we employ the Word2vec coupled with TF-IDF to construct the semantic vectors of historical questions and the queried question. Based on the question vectors, we can find the set of similar questions by calculating the similarity between the queried question and the questions in the corpus. At last, based on the question sets with similar topics, we use the life cycle function based on LSSVM to obtain the current vitality value of each question, so as to update the ranking scores of each question in the similar question set. It can achieve the final rank of the similar questions, which integrates question life cycle.

There are some parameters in the proposed method: Dirichlet super parameters $\alpha, \beta$ in Eq. (2-3), cluster number $k$ in Eq. (2-16), question life cycle time $T$ in Eq. (2-17), and correction factor $\alpha$ in Eq. (2-21). According to the analysis of Liu et al. [16], we set the Dirichlet super parameters $\alpha$ as 0.5 and $\beta$ as 0.1. Similarly, we set the correction factor $\alpha$ as 0.11 according to the analysis of Liu et al. [16]. Additionally, how to determine the number of clusters $k$ is introduced in Section 2.3.4. When it comes to determining question life cycle time $T$, we count the number of respondents of the questions in the CQADupStack dataset for different unit time (1800 s) intervals. Its statistical results are shown in Fig. 3.1. It can be seen that for the first half

hour right after launching a question, the number of answers has increased over time, and the peak of the number of respondents has been reached during this period. However, in the next half hour, the number of respondents starts to decrease. By the end of the time, few people answer the question. When this period of time is over, the question is outside of the first several pages in CQA questions. Thus, it is difficult for other users to browse or answer the question, which is already in the recession of problem life cycle. The purpose of this study is to retrieve questions that are popular and semantically similar to the problem to be solved. Therefore, the problem life cycle trend after one day is not necessary to be taken into consideration in this study. In summary, we set the length of question life cycle $T$ as one day.

### 3.5. Experimental results and analysis

K-means clustering algorithm is used as one part of the life cycle curve of different types of questions, and the number of clusters affects the experimental results of WELQLC-QR question retrieval model. Therefore, the experiments in this study first compare the scores of CH indicator under different cluster numbers. After determining the best number of clusters, WELQLC-QR is compared with benchmark models and a derived model to verify the validity of the proposed WELQLC-QR method.

#### 3.5.1. Determination of the best number of clusters and simulation of the question life cycle

In Table 3.1 are shown the CH index scores after clustering 29,866 questions using K-means clustering algorithm under different cluster numbers. When the number of clusters is 7, the CH index score at this time reaches the highest, which is 19774.81 and 32.85% higher than the second highest score (the number of clusters at this time is 8). It indicates that the question clusters have the best cohesion and achieve the best clustering effect when the number of clusters is 7. Thus, we set the $k$ as 7 in this study.

After the number of clusters is confirmed, we curve the life cycle of question. Curve results of seven categories are shown in Fig. 3.3a and b. The horizontal axis of the figures represents the unit time node. As described in Section 2.3.3, each unit time represents half an hour. The vertical axis represents the question vitality value, which is calculated by Eq. (2-18).

It can be seen from Fig. 3.2a and b that the change of question life cycle curves can be divided into two types. The first type of change is that when a question is published in CQA, the question activity gradually rises. After reaching the apex, the problem vitality begins to decline rapidly until it fluctuates above 0 (e.g., Fig. 3.2a). According to statistics, we find that

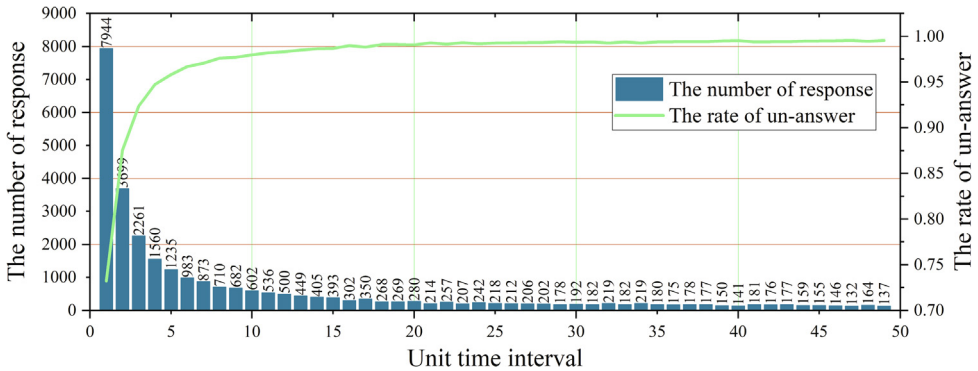


**Fig. 3.1.** The number of responses and the rate of un-answer in different time intervals.

**Table 3.1**
Table of CH index score.

| Number of clusters | CH index score |
|---|---|
| 3 | 18241.44 |
| 4 | 17230.58 |
| 5 | 17010.03 |
| 6 | 17951.71 |
| 7 | 19774.81 |
| 8 | 19147.87 |
| 9 | 18904.84 |
| 10 | 18548.64 |
| 11 | 18093.26 |
| 12 | 18695.88 |
| 13 | 18625.88 |

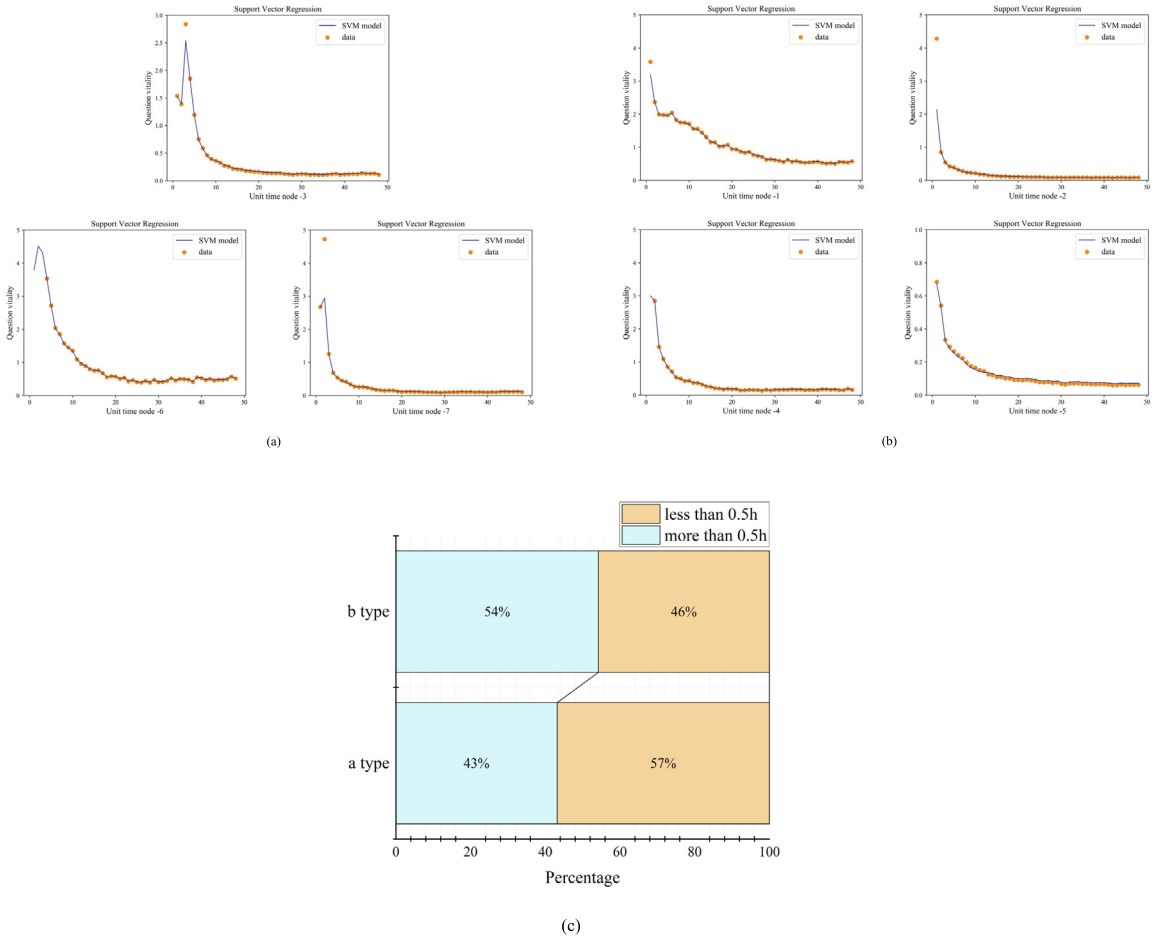


**Fig. 3.2.** (a) The life cycle curve goes up and then goes down. (b) The life cycle curve continues to decline. (c) Answer time distribution of questions belonging to a and b respectively.

66.78% questions of this type tend to have the best answers after more than half an hour, which is shown in the left panel of Fig. 3.2c. The reason of this fluctuation is that the initial answer does not answer the user's question perfectly, so it attracts other users to solve the problem. Until the best answer to the problem occurs, other users rarely or do not continue to comment or answer the question. Another type of change is that when a question is published in CQA, the vitality of the question has reached the apex, and then it begins to show a downward trend until the problem vitality fluctuates above 0 (e.g., Fig. 3.2b). Similarly, according to statistics, we find that 56.31% questions of this type tend to achieve the best answer less than half an hour, which is shown in the left panel of Fig. 3.2c. The reason of this type of life cycle fluctuations is that the initial answer has already answered users' question, so the attractiveness of the question to others begins to decline gradually after the initial answer. In summary, it is necessary to take question life cycle into consideration during question retrieval.

### *3.5.2. Comparisons between bench mark models and WELQLC-QR*

To evaluate the performance of the proposed method, we conducted comparative experiments on the metrics mentioned above. In Tables 3.2 and 3.3 are shown the evaluation results on P@N and MRR@N.

To verify the effectiveness of each part of the proposed method, we conduct comparative experiments between the counterpart models and WELQLC-QR, which combines question life cycle and semantic matching methods. Their results are shown in Fig. 3.3, which reveals that with increasing number of retrieved questions *N*, the MRR and P@N of various question retrieval methods gradually decrease. When the lexical importance extracted by TF-IDF model is incorporated into question vector as a weight, the accuracy of TW-QR model is 2.12% lower than that of Word2vec in P@1 and MRR@1. However, with the increase of number *N*, the P@N and MRR@N of TW-QR model increase by 15.1% and 8.88% compared with that of Word2vec method, respectively. It shows the importance of the weights of the words extracted by TF-IDF. Through comparing WT-QR with WET-QR, it can be found that the P@N and MRR@N of TW-QR are higher than those of WET-QR. They have the gaps of 1.88% and 1.48%, respectively, indicating that it is effective to filter noise words using TF-IDF.

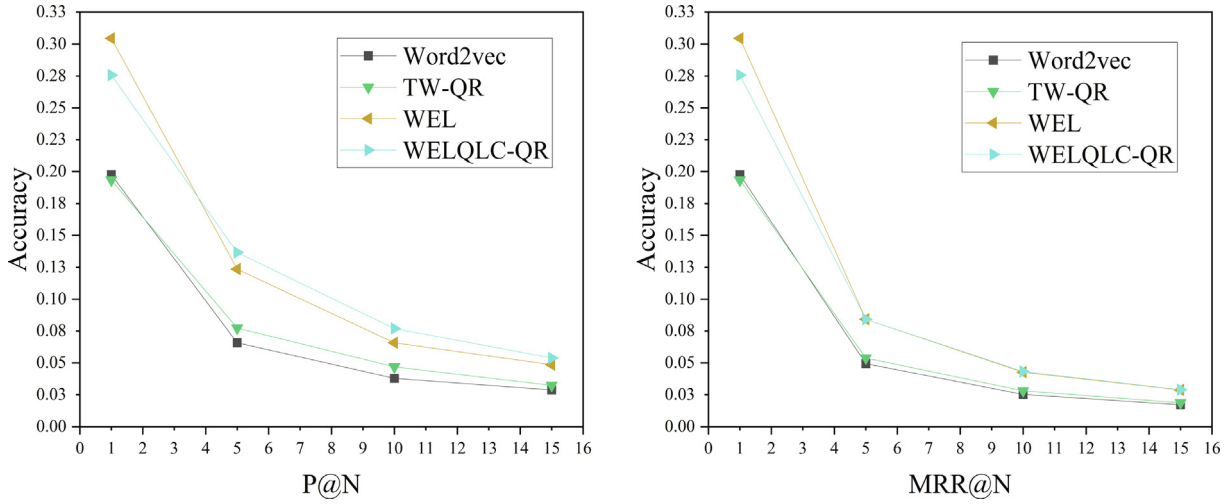


**Fig. 3.3.** P@N and MRR@N of WELQLC-QR and other models.

**Table 3.2**
P@N of different methods for question retrieval.

| Methods | P@1 | P@5 | P@10 | P@15 |
|---|---|---|---|---|
| Word2vec | 0.1975 | 0.0658 | 0.0379 | 0.0288 |
| TW-QR | 0.1934 | 0.0773 | 0.0469 | 0.0324 |
| WET-QR | 0.1535 | 0.0656 | 0.0291 | 0.0266 |
| RCM-QR | 0.0683 | 0.0372 | 0.0248 | 0.0198 |
| L-LDA | 0.0411 | 0.0224 | 0.0172 | 0.0136 |
| LC-LDA | 0.0532 | 0.0312 | 0.0161 | 0.0147 |
| BM25 | 0.1700 | 0.0660 | 0.0380 | 0.0061 |
| WEL | 0.3045 | 0.1235 | 0.0658 | 0.0486 |
| WELQLC-QR | 0.2757 | 0.1366 | 0.0770 | 0.0538 |

**Table 3.3**
MRR@N of different methods for question retrieval.

| Methods | MRR@1 | MRR@5 | MRR@10 | MRR@15 |
|---|---|---|---|---|
| Word2vec | 0.1975 | 0.0494 | 0.0253 | 0.0172 |
| TW-QR | 0.1934 | 0.0538 | 0.0281 | 0.0188 |
| WET-QR | 0.1535 | 0.0437 | 0.0213 | 0.0164 |
| RCM-QR | 0.0683 | 0.0263 | 0.0108 | 0.0084 |
| L-LDA | 0.0411 | 0.0125 | 0.0102 | 0.0084 |
| LC-LDA | 0.0532 | 0.0234 | 0.0123 | 0.0112 |
| BM25 | 0.1700 | 0.0462 | 0.0237 | 0.0025 |
| WEL | 0.3045 | 0.0842 | 0.0427 | 0.0288 |
| WELQLC-QR | 0.2757 | 0.0842 | 0.0432 | 0.0290 |

Compared with TW-QR model, WEL model first classifies question sets in CQA. Then, it filters the question sets with inconsistent topics and uses TW-QR model for similar question retrieval. It can be seen from Fig. 3.3 that the accuracy of TW-QR model is increased by 33.97% and 35.38%, respectively. It indicates the necessity to filter inconsistencies in topics during question retrieval.

When $N = 1$, the P@N and MRR@N of WELQLC-QR are reduced by 10.44%, compared with WEL model. Because the similar question labels of the dataset are assigned according to the similarities of question semantics. Thus, when retrieving the most similar question, semantics-matching-based WEL performs better than WELQLC-QR, which further takes popularity into consideration on the basis of WEL. Although the P@1 and MRR@1 of WELQLC-QR are a bit lower than those of WEL, the retrieved result of WELQLC-QR is not only effective but also meaningful for the users in CQA. For example, a CQA user proposed the question with the title of "What is the best book to prepare for a Java interview?" in 2011-10-18. The most similar question retrieved by WEL is the question with the title of "Best Java book you have read so far", which is the labeled similar question in the experimental dataset and can be viewed as a correct retrieval. However, this retrieved question was proposed in 2008-09-16, which means that it contains the knowledge of three years ago, resulting in the fact that the retrieved result of WEL may not solve the questioner's doubts. On the other hand, WELQLC-QR retrieved the question titled by "How to prepare yourself

for programming interview questions?", which was proposed in 2010-9-9. Although the retrieved question of WELQLC-QR is not the labeled question in the dataset, it is with similar semantics with the proposed question. More importantly, the retrieved question of WELQLC-QR is more popular than that of WEL, and it is more meaningful to users. Additionally, when number $N$ is 5, 10 or 15, the P@N of WELQLC-QR model is 10.6%, 17% and 10.7% higher than that of WEL model, respectively. It reveals that the question retrieval model, which is based on recommending some semantically similar questions and coupling popularity of the question, will allow users to obtain more meaningful similar questions.

In Fig. 3.4 is shown a comparison between the question-life-cycle-based question retrieval method (WELQLC-QR) and the benchmark models (L-LDA, LC-LDA, BM25, Word2vec, RCM). As number $N$ increases, the MRR@N and P@N of the baseline model are also gradually reduced. It can be found that the MRR@N and P@N of WELQLC-QR model have an average increase of 4.74% and 6.58%, compared with those of BM25 method. Additionally, the MRR@N and P@N of WELQLC-QR are higher than those of RCM-QR, which increase 9.825% and 7.935%, respectively. The reason of RCM-QR with such poor performance is that the number of samples is not enough for RCM to effectively extract question representations. Compared with L-LDA and LC-LDA model based on the similarity of question topic, the evaluation index of WELQLC-QR model is much higher than that of these two models. Compared with Word2vec method, the MRR@N and P@N of WELQLC-QR model have increased by an average of 3.57% and 5.33%, respectively. It indicates that WELQLC-QR model proposed in this study is better than other benchmark models in terms of the accuracy of question retrieval and the ranking of similar question.

The time complexity of these algorithms is shown in Fig. 3.5. We can find that BM25, Word2vec, TW-QR, WET-QR have low time-consuming on training and spend less than 100 s. L-LDA, LC-LDA, WEL and WELQLC-QR have higher cost for train-

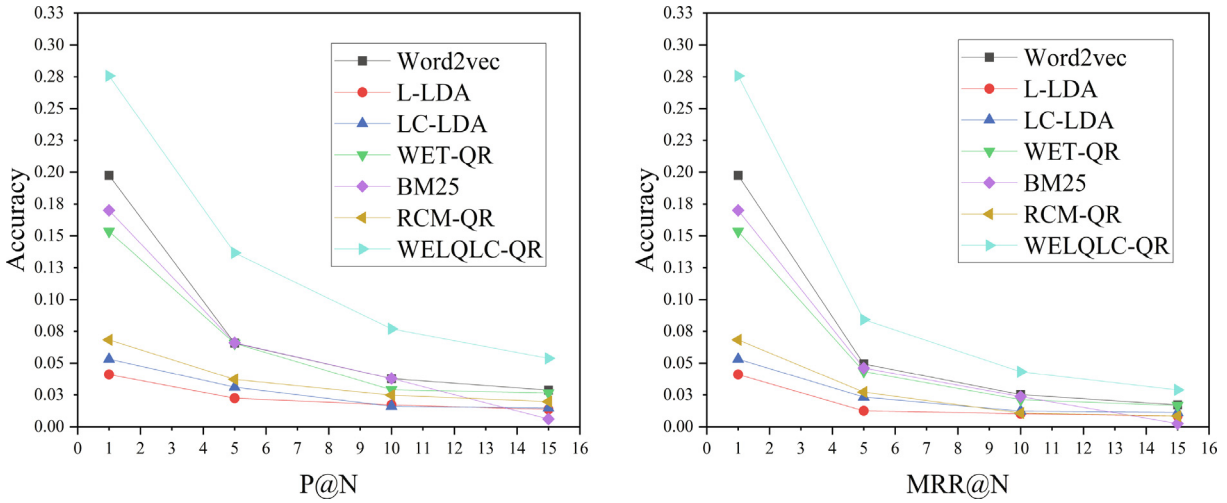


**Fig. 3.4.** P@N and MRR@N of WELQLC-QR and different bench mark models.

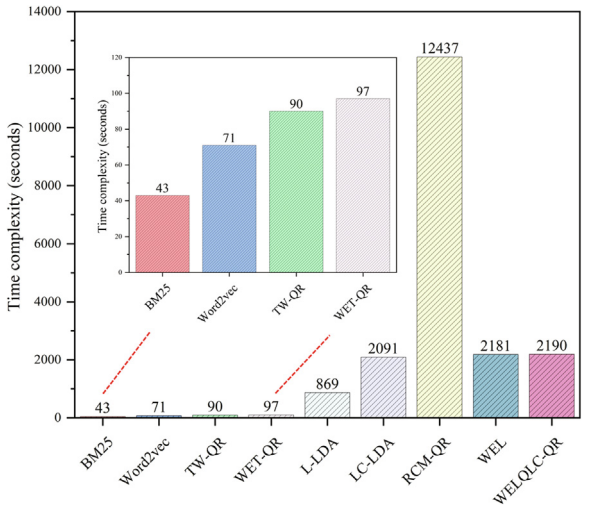


**Fig. 3.5.** The time complexity of each question retrieval method.

ing algorithm, spending more than 1000 (even 2000) seconds. This is because the base algorithms of L-LDA and MCL need to keep iterating until they converge. Meanwhile, the cost of RCM-QR is the highest and far exceeds that of other methods.

## 4. Conclusion and future work

Question retrieval is essential for the users in CQA to acquire knowledge. In this study, we proposed an Integrated Retrieval Framework for Similar Questions named WELQLC-QR, which is capable of retrieving the question with both similar semantics and high popularity. We developed Word-semantics Embedded Label Clustering-LDA method (WEL) for question retrieval from the perspective of semantic matching, which can comprehensively and adequately extract semantics of the question from both word-grained and topic-based views. Additionally, Question Life Cycle Optimization Similar Question List Strategy (QLC) was proposed for incorporating popularity into question retrieval, taking both the internal factors (e.g., the number of comments and answers) and external factors (e.g., programming language ranking information) of the question into consideration to construct a popularity model. Experiments were conducted on a realistic CQA dataset from Stack Overflow to evaluate the effectiveness of the framework. Results show that the MRR@N of WELQLC-QR model has an average increase of 8.99%, 8.3%, 4.74% and 3.56% compared with that of L-LDA, LC-LDA, BM25 and Word2vec, respectively. Similarly, the P@N of WELQLC-QR model has an average increase of 11.22%, 10.69, 6.57% and 5.32% compared with that of L-LDA, LC-LDA, BM25 and Word2vec, respectively.

Meanwhile, this study leads to several interesting directions for future works. Word2vec is employed to extract word representation in this study, but fixed representation might cause the problem of Word Sense Disambiguation (WSD). It would be interesting to take some actions to reduce WSD, so as to enhance retrieval accuracy [46]. Additionally, the essential factor influencing question retrieval is the quality of questions, part of which includes semantics similarity and question popularity. It is necessary to extend popularity to question quality, so as to make the retrieved question with similar semantics and high quality. Finally, it is of relevance to apply the proposed framework to other information retrieval fields such as text matching.

## CRediT authorship contribution statement

**Yue Liu:** Conceptualization, Methodology, Validation, Formal analysis, Writing - original draft, Writing - review & editing, Supervision. **Aihua Tang:** Methodology, Software, Validation, Data curation, Writing - original draft, Writing - review & editing. **Zhibin Sun:** Conceptualization, Methodology, Formal analysis, Writing - original draft, Writing - review & editing. **Weize Tang:** Software, Validation, Writing - review & editing. **Fei Cai:** Software, Formal analysis, Resources, Data curation, Writing - original draft. **Chengjin Wang:** Writing - original draft.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

This work is supported by the State Key Program of National Nature Science Foundation of China (Grant No. 61936001). Additionally, we appreciate the High Performance Computing Center of Shanghai University, and Shanghai Engineering Research Center of Intelligent Computing System (No. 19DZ2252600) for providing the computing resources and technical support.

## References

[1] D. Hoogeveen, L. Wang, T. Baldwin, et al, Web forum retrieval and text analytics: a survey, Found. Trends Inf. Retr. 12 (1) (2018) 1–163.
[2] A. Berger, J. Lafferty, Information retrieval as statistical translation, ACM SIGIR Forum, ACM, New York, NY, USA, 2017, 51 (2), 219–226.
[3] L. Chen, J.M. Jose, H. Yu, et al., A semantic graph based topic model for question retrieval in community question answering, in: Proceedings of the ninth ACM International Conference on Web Search and Data Mining, 2016, pp. 287–296.
[4] L. Sang, M. Xu, S.S. Qian, et al, Multi-modal multi-view Bayesian semantic embedding for community question answering, Neurocomputing 334 (2019) 44–58.
[5] Z. Chen, C. Zhang, Z. Zhao, et al, Question retrieval for community-based question answering via heterogeneous social influential network, Neurocomputing 285 (2018) 117–124.
[6] R.C. Moore, Improving IBM word alignment Model 1, in: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, 21–26 July, 2004, Barcelona, Spain, DBLP, 2004.
[7] X. Xue, J. Jeon, W.B. Croft, Retrieval models for question and answer archives, in: International Acm Sigir Conference on Research & Development in Information Retrieval. ACM, 2008.
[8] G. Zhou, T. He, J. Zhao, et al., Learning continuous word embedding with metadata for question retrieval in community question answering, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, vol. 1, Long Papers, 2015, pp. 250–259
[9] M.J. Carman, F. Crestani, M. Harvey, et al, Towards query log-based personalization using topic models, in: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, 2010, pp. 1849–1852.

[10] T.C. Zhou, C.Y. Lin, I. King, et al., Learning to Suggest Questions in Online Forums, in: AAAI Conference on Artificial Intelligence, DBLP, 2011.
[11] L. Cai, G. Zhou, K. Liu, et al, Learning the latent topics for question retrieval in community qa, in: Proceedings of 5th International Joint Conference on Natural Language Processing, 2011, pp. 273–281.
[12] Z. Ji, F. Xu, B. Wang, et al, Question-answer topic model for question retrieval in community question answering, Acm International Conference on Information & Knowledge Management, ACM, 2012.
[13] K. Zhang, W. Wu, H. Wu, et al, Question retrieval with high quality answers in community question answering, in: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, 2014, pp. 371–380.
[14] T. Griffiths, Gibbs sampling in the generative model of latent dirichlet allocation, 2002.
[15] D. Ramage, D. Hall, R. Nallapati, et al., Labeled LDA: a supervised topic model for credit attribution in multi-labeled corpora, in: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1, Association for Computational Linguistics, 2009, pp. 248–256.
[16] Y. Liu, A.H. Tang, F. Cai, et al, Multi-feature based Question-Answerer Model Matching for predicting response time in CQA, Knowl.-Based Syst. 182 (2019) 10479–10491.
[17] D. Van, Graph clustering by flow simulation, Cambridge UK, 2001.
[18] V. Vargas-Calderón, J.E. Camargo, H. Vinck-Posada, Event detection in Colombian security Twitter news using fine-grained latent topic analysis. arXiv preprint arXiv:1911.08370, 2019.
[19] X. Rong, word2vec parameter learning explained, arXiv preprint arXiv:1411.2738, 2014.
[20] W.E. Zhang, Q.Z. Sheng, Z. Tang, et al, Related or duplicate: distinguishing similar CQA questions via convolutional neural networks, in: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, 2018, pp. 1153–1156.
[21] Y. Shen, W. Rong, N. Jiang, et al., Word embedding based correlation model for question/answer matching, Thirty-First AAAI Conference on Artificial Intelligence, 2017.
[22] Z. Wang, L. Ma, Y. Zhang, A hybrid document feature extraction method using latent Dirichlet allocation and word2vec, in: 2016 IEEE First International Conference on Data Science in Cyberspace (DSC), IEEE, 2016, pp. 98–103.
[23] K. Zhang, W. Wu, F. Wang, et al, Learning distributed representations of data in community question answering for question retrieval, in: Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, 2016, pp. 533–542.
[24] J.T. Lee, S.B. Kim, Y.I. Song, et al, Bridging lexical gaps between queries and questions on large online Q&A collections with compact translation models, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2008, pp. 410–418.
[25] N. Othman, R. Faiz, K. Smaïli, Using word embeddings to retrieve semantically similar questions in community question answering, ISGA 1 (1) (2018) hal-01873748.
[26] A. Aggarwal, C. Sharma, M. Jain, et al., Semi supervised graph based keyword extraction using lexical chains and centrality measures, Comput. Sist. 22 (4) (2018) 1307–1315.
[27] J. Ramos, Using tf-idf to determine word relevance in document queries, in: Proceedings of the First Instructional Conference on Machine Learning, 2003, 242, pp. 133–142.
[28] T. Liu, W.N. Zhang, L. Cao, et al, Question popularity analysis and prediction in community question answering services, PloS One 9 (5) (2014) e85236.
[29] R. Vernon, International investment and international trade in the product cycle, Int. Execut. 8 (4) (1966), 16-16.
[30] T. Althoff, D. Borth, J. Hees, et al., Analysis and forecasting of trending topics in online media streams, in: Proceedings of the 21st ACM international conference on Multimedia, ACM, Barcelona, Spain, 2013, pp. 907–916.
[31] C. Castillo, M. El-Haddad, J. Pfeffer, et al., Characterizing the life cycle of online news stories using social media reactions, in: Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing, ACM, Baltimore, MD, United states, 2014, pp. 211–223.
[32] Y. Liu, F. Cai, et al, Item life cycle based collaborative filtering, J. Intell. Fuzzy Syst. (2018) 2743–2755.
[33] T.L. Griffiths, M. Steyvers, Finding scientific topics, Proc. Nat. Acad. Sci. 101 (Suppl. 1) (2004) 5228–5235.
[34] C.H. Huang, Y. Jian, H. Fang, A text similarity measurement combining word semantic information with TF-IDF method, Chin. J. Comput. 34 (5) (2011) 856–864.
[35] D. Guthrie, B. Allison, W. Liu, et al., A closer look at skip-gram modelling, LREC, 2006, pp. 1222–1225.
[36] Y. Zhang, R. Jin, Z.H. Zhou, Understanding bag-of-words model: a statistical framework, Int. J. Mach. Learn. Cybern. 1 (1–4) (2010) 43–52.
[37] C. Gupta, A. Jain, D.K. Tayal, et al, ClusFuDE: forecasting low dimensional numerical data using an improved method based on automatic clustering, fuzzy relationships and differential evolution, Eng. Appl. Artif. Intell. 71 (2018) 175–189.
[38] Y. Liu, W.W. Ju, et al., Demand forecasting for footwear products using wavelet transform and Artificial Bee Colony algorithm optimized Polynomial Fitting, in: Proceedings of the International Conference on Natural Computation. 2016. Zhangjiajie, China, IEEE, 2016, pp. 1146–1150.
[39] https://www.tiobe.com/tiobe-index/.
[40] A.K. Jain, Data clustering: 50 years beyond K-means, Pattern Recogn. Lett. 31 (8) (2010) 651–666.
[41] H. Wang, D. Hu, Comparison of SVM and LS-SVM for regression, in: 2005 International Conference on Neural Networks and Brain, IEEE, 2005, 1, pp. 279–283.
[42] D. Hoogeveen, K.M. Verspoor, T. Baldwin, CQADupStack: a benchmark data set for community question-answering research, in: Australasian Document Computing Symposium (ADCS), ACM, Parramatta, NSW, Australia, 2015, a3.
[43] Harabasz Caliński, A dendrite method for cluster analysis, Commun. Stat. Theory Methods 3 (1) (1974) 1–27.
[44] C.X. Zhai, A study of smoothing methods for language models applied to ad hoc information retrieval, in: Proceedings of International ACM Sigir Conference on Research and Development in Information Retrieval, 2001, New Orleans, LA, United States, ACM, 2001, pp. 334–342.
[45] T. Lei, H. Joshi, R. Barzilay, et al., Semi-supervised question retrieval with gated convolutions, arXiv preprint arXiv:1512.05726, 2015.
[46] S. Vij, A. Jain, D. Tayal, et al, Fuzzy logic for inculcating significance of semantic relations in word sense disambiguation using a WordNet graph, Int. J. Fuzzy Syst. 20 (2) (2018) 444–459.