Information Sciences 545 (2021) 132–152

Contents lists available at ScienceDirect

Information Sciences

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

Community answer generation based on knowledge graph

Yongliang Wu [a], Shuliang Zhao [b,c,d,*]

[a] School of Mathematical Sciences, Hebei Normal University, Hebei 050024, China
[b] College of Computer and Cyber Security, Hebei Normal University, Hebei 050024, China
[c] Hebei Provincial Key Laboratory of Network and Information Security, Hebei 050024, China
[d] Hebei Provincial Engineering Research Center for Supply Chain Big Data Analytics & Data Security, Hebei 050024, China

ARTICLE INFO

Article history:
Received 22 January 2020
Received in revised form 28 June 2020
Accepted 27 July 2020
Available online 6 August 2020

Keywords:
Community question answering
Answer generation
Phrase embedding
Knowledge graph
Phrase mining

ABSTRACT

Community Question Answering (CQA) has become an indispensable way for modern people to share and acquire knowledge. It allows users to ask questions, which will be answered by experienced users enthusiastically. By recording user operation logs, CQA has accumulated a large amount of valuable and complex data. However, askers must wait (usually for a long time) until other expert users answer their questions on social platforms. This will seriously affect the user experience. In this paper, we propose a Community Answer Generation method based on the Knowledge Graph, called CAGKG, to generate natural language answers automatically. Firstly, we extract the core phrases of posts to represent their semantics relations. Then, we model the user's knowledge background based on their action records. Finally, we query knowledge entities in a knowledge graph based on user background and question semantics, then convert them into natural language answers. Besides, we proposed a Phrase-based Answers Semantic Similarity Evaluation indicator, called PASSE, which focuses on the semantic similarity between texts instead of literal matching. To the best of our knowledge, it is the first work that utilizes the user knowledge and text semantics to improve the performance of CQA. Experiments on four real datasets (Stack Overflow, Super User, Mathematics, and Quora) show that CAGKG is superior to the state-of-the-art question answering frameworks. Compared with other answer evaluation indicators, PASSE is a promising indicator for evaluating semantic similarity.

1. Introduction

Nowadays, community question answering (CQA) has become an essential way for people to acquire knowledge. People tend to ask questions and get answers in natural language rather than enter keywords to obtain a webpage list. Popular CQA platforms, including Stack Exchange and Quora, allow users to submit their questions, and then other users will answer the questions enthusiastically [4]. Askers can mark the accepted answer from many posts, or continue to comment for details. CQAs provide a highly convenient way to acquire knowledge. In recent years, they have accumulated enormous data about users, questions, and answers, which implies precious knowledge [6]. However, there is an inherent delay between askers submitting a question and the appearance of accepted answers [42]. Therefore, CQA needs some improvements to return answers quickly, thereby reducing the waiting time of users.

* Corresponding author at: College of Computer and Cyber Security, Hebei Normal University, Hebei 050024, China.
E-mail address: shuliangzhao@hebtu.edu.cn (S. Zhao).

https://doi.org/10.1016/j.ins.2020.07.077
0020-0255/

Community Answer Generation is becoming a promising research direction. It utilizes the user's community information to improve answer generation in CQA. To quickly provide answers to community questions, some researchers tried to route questions to possible answerers to speed up the correct answer [4,19]. They calculated the similarity of users and questions, then pushed the items to users with high matching degrees. Those plans effectively shorten the time for users to find questions, but they cannot determine the time for users to reply. Some researchers proposed to answer new questions based on real posts in CQA [10,41,42]. These methods extracted the characteristics of questions and answers to select the most likely answer to a new issue. They shorten the time to get answers, but the matching degree between real solutions and new questions is often humble. After all, there are rarely two identical questions in the world. Nowadays, some prevalent research begins to retrieve answers from existing large knowledge graphs [16,39,49]. Knowledge graphs have accumulated massive human-known knowledge, so they are the source of answers. These methods parse the user's posts (answers and questions) to obtain the core intention, then query relevant knowledge entities as answers [6]. However, most previous research extracts the intention based on words and splits phrase semantics, which leads to biases in question understanding. E.g., if the phrase "knowledge graphs" appears in a sentence, they will employ "knowledge" and "graphs" to train models for related answers. Meanwhile, answers in professional forms (knowledge entities or query subgraph) are difficult to accept by end-users directly.

Summarizing the previous research, we divide question answering systems into two categories: selecting existing answers or generating new answers. Our work focuses on the answer generation in case there are no related answers, or the answers do not match well. Since community questions depend not only on literal expression but also on the background of askers, it is a complex and challenging task to generate a standard answer. In the absence of similar questions or related answers, the knowledge-based approaches point us in a promising direction. They parse user questions and employ human-known knowledge as answers. However, we still face the following challenges in this research:

- How to understand questions accurately? In CQA, users post questions in natural language, even describe their scenarios and steps. Therefore, the primary task is to transform unstructured posts into structured representations. Most previous studies employed words or n-grams to represent texts [4]. They ignore phrase semantics, which leads to comprehension bias.
- How to model the user knowledge background? Users play an essential role in CQA. Their behavior builds entity relationships in CQA [41]. They are communicators of knowledge. The asker background indicates their expertise area, so it is an excellent supplementary description of the question. However, few studies focus on the user background, which affects the performance of CQA. E.g., Fig. 1 shows a question about file downloading. A user presents an answer describing the file downloading process, but it is not accepted. The key reason is that the answerer neglects that the asker is a Java developer.
- How to offer natural language answers? CQA is prevalent because people can get knowledge in a social way, rather than an elaborate page list, i.e., users hope to obtain answers in natural language. However, most research presents solutions in professional representations, such as query graphs [16] or triples [33]. They prevent users from directly understanding the answer semantics. So, CQA still faces many challenges in terms of natural language answer generation.
- How to evaluate the quality of the generated answers? The metric of matching between questions and answers is a very complex topic. It depends on many factors, such as user background, question semantics, personal preferences. There is no uniform experimental indicator in the existing CQA research. Most researchers employ the coverage of words or n-grams to measure the accuracy of answers, such as BLEU [29], ROUGE [23]. However, they only focus on literal matching, not on semantic similarity.

Following our previous research [43], we plan to employ phrases to improve the answer generation and evaluation in CQA. Firstly, we uniformly consider the relationship between question semantics and entities (users, question, and answer). Secondly, we model the user's knowledge background. Finally, we combine question semantics and user background with improving the answer generation of CQA. Besides, in our research, phrases are also used to optimize the answer evaluation process. In summary, the contributions of our work are as follows.

- We propose a Community Answer Generation method based on Knowledge Graph, called CAGKG. To the best of our knowledge, it is the first work to combine user background to generate community natural language answers. It creatively employs phrases to understand post semantics and model user background. Then we combine the question semantics and the user background knowledge to find relevant entities in knowledge graphs. Finally, we transform the extracted entities into natural language answers.
- To evaluate the accuracy of generated answers, we propose a novel Phrase-based Answers Semantic Similarity Evaluation index, called PASSE, which focuses on measuring the semantic similarity of texts. In this paper, it is used to calculate the semantic coverage of answers, instead of literal matching.
- Experiments demonstrate that CAGKG has excellent performance and efficiency. PASSE is also a promising semantic evaluation indicator.

The rest of our paper is structured as follows. Section 2 introduces the development of relevant work. In Section 3, we explain in detail the preliminary knowledge and lay a solid foundation for the follow-up research. Section 4 illustrates

Fig. 1. The schematic diagram of Community Question and Answer websites.

the implementation process of CAGKG and PASSE. Section 5 validates the effectiveness and efficiency of our framework through extensive experiments. In Section 6, we summarize our work and propose future research directions.

2. Related work

In this section, we review the research status of the most related fields.

2.1. Question answering

Question Answering (QA) can automatically give answers to natural language questions. It has gradually become the ancillary means for people to acquire knowledge. Many researchers have conducted in-depth work on QA. In [46], a knowledge-based question understanding method was proposed to find some answer related documents. Comprehensive experiments on a large scale of query logs verified its validity. Jia et al. [17] complemented the analysis of the temporal intent of Question and Answering over knowledge base. Hu et al. [16] employed the Knowledge Graph to answer natural language questions. It considered the ambiguity of natural language and proved its effectiveness through experiments. Wu et al. [42] proposed to combine the question subject and body to understand the question semantic for answer selection. In [12], a deep learning framework was employed to map question and candidate answers into a continuous space, solving complex problems by calculating their matching degree. Shin et al. [33] focused on the predicate constraint problem in the question answering system based on Knowledge Graph and generated a query graph based on the predicate to get answers. In [36], an automatic evaluation method was proposed for assessing the performance of question answering systems. Wang et al. [39] translated natural language questions into graph structure queries and employed knowledge graph embedding to solve the mapping between questions and answers. Zheng et al. [49] incorporated user-related information into the understanding of natural language issues and determined the answers in the knowledge base through user interaction. Lu et al. [26] proposed QUEST to obtain the textual resource for complex questions. Previous researchers have adequately demonstrated that knowledge graphs can effectively improve QA, but they still face two significant challenges. Firstly, the existing QAs only consider the relationship between questions and answers, ignoring the function of users. Secondly, the current QAs offer solutions in the form of documents or triples, which is not conducive for user understanding.

The rise of CQA eases the first challenge. More and more researchers are paying attention to the role of users in QAs. Figueroa [10] utilized the direct string matching of question title and body to obtain the related questions from the CQA datasets. Yan and Zhou [44] introduced a way of providing potential answerers to new questions. In [9], heterogeneous information networks were employed to represent complex entity relationships in CQA and integrated social information to accelerate related tasks. Figueroa [11] studied the hidden gender characteristics of users based on the posts. A new method for expert user prediction in CQA was produced in [28]. Le and Shah [19] integrated question content with user pro-

files to predict potential answerers. Users were modeled based on their submitted posts and incorporated into the answer selection process [41]. However, few researchers combine the Knowledge Graph to solve CQA tasks.

Table 1 lists the features of related work. We summarize the existing research. Firstly, many researchers have studied the question answering system based on the knowledge graph, but they do not pay attention to the role of users. Secondly, with the development of CQA, researchers begin to consider the role of users and employ user features to improve QA performance. However, most existing research ignores the user's background knowledge and utilizes existing answers to settle new questions.

In this paper, we plan to combine the user's knowledge background and the Knowledge Graph to improve the answer generation in CQA.

2.2. Natural language generation

The answers of some existing QAs are derived from the posts previously submitted by users [41,42,44], or related texts [10], or other structured data [45]. It results in users not being able to obtain the knowledge directly from the returned answers. Our research focuses on generating natural language answers to user questions. The most relevant research is natural language generation. Many predecessors have converted different structured data into natural language. Araki et al. [1] proposed an automatically generating text-questions and multiple-choice answers method. Phrase information was employed to generate a natural language summary of the product in [47]. Li et al. [22] built a model to produce category text based on the specified category characteristics. Table2Seq was a neural generative model that generated natural language sentences based on tables [3]. Wang and Wan [38] employed Mixture Adversarial Networks to generate sentimental texts. Zhu et al. [50] gave an efficient way to convert triples in Knowledge Graph into natural language text. These studies have laid a solid foundation for our research about answer generation.

2.3. Question answer evaluation

Another attractive issue in QA is the answer evaluation. Predecessors have proposed some evaluation methods for the answer-question matching. Ramakrishnan et al. [30] proposed an evaluation method for the visual question answering system. Sorokin and Gurevych [34] presented a set of answer examples for evaluating knowledge-based QA systems. Hassanzadeh et al. [15] studied an evaluation method of unconstrained reasoning questions. In the QA evaluation process, Ribeiro et al. [31] not only paid attention to the individual questions but also their logical relationship. The more commonly used answer evaluation methods are derived from other NLP tasks, such as machine translation. BLEU is used initially to evaluate the performance of machine translation, and it relies on word coverage between generated text and reference text [29]. ROUGE is often used for the evaluation of text summaries, which rely on the word sequences overlap ratio between generated summary and the ideal text created by humans [23]. They have been used for many QAs [3,24], but they ignore the integrity of semantic units, which limits the QA performance.

2.4. Phrase mining and phrase embedding

Most QA methods are based on words, which limits the understanding of questions [6,32]. The semantic of texts is based on phrases [25,40,48]. Researchers have conducted related research. Liu et al. [25] considered the evaluation of the phrase quality in phrase mining. Mikolov et al. [27] mapped the phrases into a continuous vector space through a combination of word embedding, which laid a solid foundation for the application of phrases in other NLP tasks. Hashimoto and Tsuruoka [14] analyzed the impact of phrase combinations on phrase embedding. Kim et al. [18] compared the effects of different text representations on text classification. Phrases have been shown to improve multiple NLP tasks. Li et al. [20] combined phrases with topic representations. Zhang et al. [48] used phrases to improve multilingual question retrieval. Li et al. [21] employed phrases to improve topic consistency issues. Eriguchi et al. [8] proved that phrases could improve the performance of machine translation. Wang et al. [40] represented question semantics based on phrases. Some researchers have tried to use phrases to promote the performance of QA. Hasan et al. [13] found phrase embedding improved diagnostic inferencing for clinical question answering system. Datla et al. [5] employed phrases in question decomposition and answer generation to improve the open domain real-time question answering. Wang and Nyberg [37] encoded the question into phrase vectors to calculate the relevance of questions and answers. The above research lays a solid foundation for QA tasks, and we plan to improve the answer generation and answer evaluation based on phrases.

3. Preliminaries

In this section, we introduce the relevant definitions and symbolic representations.

Definition 1. Phrase is defined as a fixed-order word sequence. In Computational Linguistics, phrases represent the semantics of texts [25,43], so we express the semantics of questions and answers by extracting core phrases. A sentence is formulated as $s = [p_1, p_2..]$, where p_i denotes the ith core phrase.

Table 1
Comparison of related work features.

Features	[16,17,33,39,46,49]	[12,36]	[10,42]	[26]	[9,44]	[11,19,28]	[41]	Our method
Knowledge graph	Yes	Yes	No	No	No	No	No	Yes
Semantics	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes
User feature	No	No	No	No	Yes	Yes	Yes	Yes
User background	No	No	No	No	No	No	Yes	Yes
Selecting existing answers	No	No	Yes	No	Yes	No	Yes	No
Natural language answer	No	No	No	No	No	No	No	Yes

Table 2
Statistics of CQA datasets.

Corpus	Mathematics	Super User	Stack Overflow	Quora
Size	179.2M	837.7M	14.05G	12.5M
Number of Questions	1.1M	412 K	18M	15 K
Number of Answers	1.5M	599 K	27M	18 K
Number of Users	555 K	773 K	11M	10 K
Number of Phrases	171.3 K	112.4 K	3.6M	8.3 K

Definition 2. Phrase embedding aims to find a function that maps the phrase space to a multi-dimensional vector space for representing the phrase similarity [43]. The idea of phrase embedding comes from Skip-gram. In our work, we employ a phrase information network to show the similarity among phrases.

Definition 3. Heterogeneous information network (HIN) is a graph with multiple node types or multiple edge types. It is often used to represent complex relationships among multiple entities. In this paper, we employ HIN to represent the entity-relationship in CQA, which contains different types of entities and relationships.

Definition 4. Network schema is a meta template for HIN and contains the semantics of different relationships. Fig. 2 shows the network schema of CQA. It contains four entity types (User, Question, Answer, and Phrase) and seven entity relationships. Q-U relationship represents the behavior of posting or browsing a question. Q-A relationship denotes the action of a user submitting or browsing answers. Q-P and A-P relationships indicate the relationship between posts (answers or questions) and their core phrases. The similarity among phrases is shown in P-P relationship. U-P relationship means the user's knowledge background. Q-A relationship is the applicability between a question and an answer.

Definition 5. Knowledge Graph stores complex structured or unstructured data in the form of triples. It shows the connection between different entities. E.g., ⟨'Michael Jordan', 'Career', 'Basketball player'⟩ means "Michael Jordan is a basketball player.".

Definition 6. Community Question Answering based on Knowledge Graph aims to generate natural language answers based on the user's background and the knowledge graph.

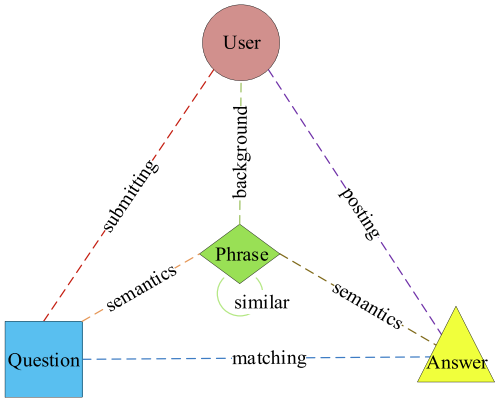


Fig. 2. The network schema of CQA.

4. Community answer generation based on knowledge graph

In this section, we introduce the Community Answer Generation method based on Knowledge Graph, called CAGKG. Fig. 3 briefly illustrates the steps of CAGKG through an example.

- Firstly, we define the similarity between different types of entities and construct a heterogeneous information network to display the complicated relationship in CQA. Fig. 3(a) offers an example of HIN, which contains complex relationships between multiple entities (include u_1, u_2, q_1, q_2, q_3, a_1, and a_2).
- Secondly, we extract core phrases to represent the semantic of questions and answers, then calculate the similarity of phrases. Fig. 3(a) and (b) show some instance relationships. E.g., the connection between q_1 and p_1 indicates that p_1 is a core phrase of q_1. The association between p_1 and p_2 denotes their similarity.
- Subsequently, we model the user background by their posts. E.g., according to related posts (a_1 and q_2) of u_2, its background knowledge is p_2.
- Then, we combine the question semantic and the user background to obtain related entities and generate natural language answers. E.g., when a new question q_3 is raised by u_2, we find the related entities by user knowledge p_2 and question semantic p_3, shown in Fig. 3(c). The new answer a_3 is generated for q_3 based on retrieved entities.
- Finally, we employ semantic coverage to evaluate the quality of the generated answers.

4.1. The Heterogeneous Information Network for Community Question Answering

In this section, we formulate fundamental entity relationships. There are three basic entities (User, Question, Answer) and three related relationships (U-Q, U-A, Q-A) in CQA.

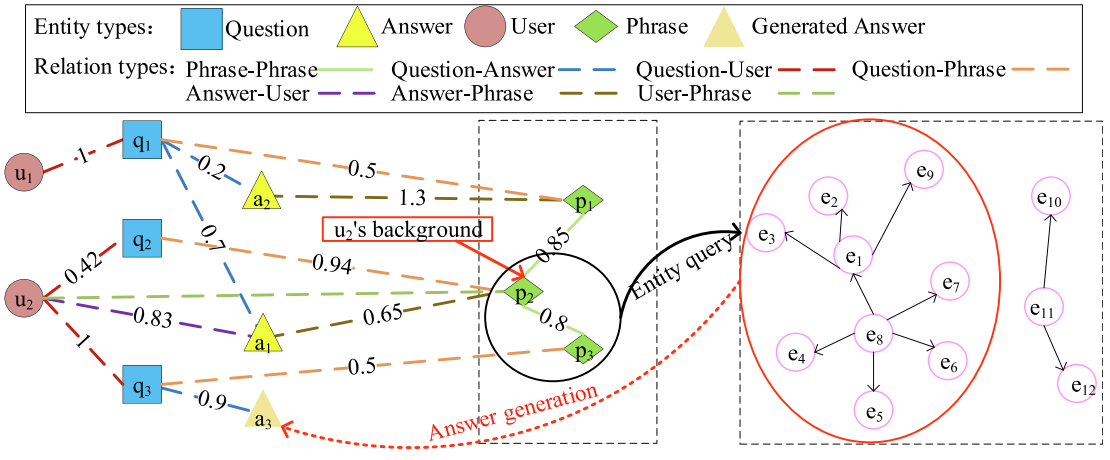
U-Q relation reflects the user's behavior of posting or browsing questions. If a user posts a question, it means that the user is strongly related to this question, denoted as 1. Otherwise, It is denoted as the frequency of the user browsing the question *browsefreq*(u, q), divided by the user's total views *totalbrowsefreq*(u), $u \in U, q \in Q$. It is shown in Eq. (1).

$$Sim_{UQ}(u,q) = \begin{cases} 1 & q \text{ is posted by } u \\ \frac{browsefreq(u,q)}{totalbrowsefreq(u)} & \text{otherwise} \end{cases}. \tag{1}$$

U-A relation represents the action of posting or liking answers. Users have a close relationship with answers submitted by themselves, denoted as 1. If a user likes an answer, their relationship is denoted as the reciprocal of the liked number of the answer *likefreq*(a), $a \in A$, shown in Eq. (2).

$$Sim_{UA}(u,a) = \begin{cases} 1 & a \text{ is posted by } u \\ \frac{1}{likefreq(a)} & u \text{ likes } a \end{cases}. \tag{2}$$

Q-A relation indicates the relevance of questions and answers. If the asker marks an answer as accepted, their relevance is defined as 1. Otherwise, the relevance is defined as its liked number divided by the liked number of the most popular answer. It is denoted as Eq. (3).



(a) Heterogeneous information network for CQA (b) Phrase information network (c) Common knowledge graph

Fig. 3. The architecture of CAGKG.

$$Sim_{QA}(q,a) = \begin{cases} 1 & a \text{ is the accepted answer of } q \\ \frac{likefreq(a)}{\max_{a' \in A}(likefreq(a'))} & a' \text{ is } a \text{ answer of } q \end{cases}. \quad (3)$$

According to the above definition, we construct a HIN to represent the entity-relationship in CQA, shown as Fig. 3(a).

4.2. Question and answer understanding model

Algorithm 1: Phrase-based Question-Answer Understanding Model
Input: Question-Answer Corpus $C = [s_1, s_2 ... s_{|C|}]$.
Output: Phrase representation of corpus $C = (s_1, ps_1), (s_2, ps_2) ... (s_{|C|}, ps_{|C|})$;
Phrase base with similarity *PB*.
1: for s_i in C do//Traverse the parse tree of each text to get the *PB*
2: $t_i \leftarrow GenerateParseTree(s_i)$//Utilize PCFG to build parsing trees
3: $ps_i \leftarrow$ traverse(t_i))//Traverse t_i and get a phrase set ps_i
4: $PB.add(ps_i)$//Add phrase set ps_i to *PB*
5: end for
6: $PhraseEmbedding(PB)$//Obtain phrase embedding, shown in Fig. 5.
7: return C, *PB*

The goal of understanding questions and answers is to obtain their core semantics. Phrases in natural language represent complete semantics. In this section, we propose the Phrase-based Question-Answer Understanding Model, which utilizes a parsing tree to extract core phrases and gets the relationship among phrases. Its overall process is shown in Algorithm 1. Input is a question–answer dataset, which offers a training corpus for phrase extracting. There are two aspects of the output: posts (questions or answers) with their phrase representation, shown in Fig. 3(a) and (b); a phrase information network representing phrase relations, shown in Fig. 3(b). Algorithm 1 consists of three parts: Parse Tree Generation, Phrase-based Question-Answer Understanding, and Phrase Embedding.

Parse Tree Generation is dedicated to transforming a natural language sentence into a syntactic parse tree whose subtrees denote sentence constituents, shown in Fig. 4. The input includes PCFG[1], denoted as G, which is a set of grammatical rules. It is denoted as $G = (N, \Sigma, R, S, P)$. S is a start point, which usually expresses the sentence. N is a non-terminal symbol set, which includes all possible candidate phrases. The parse tree employs phrase types as non-leaf nodes for displaying the sentence structure. E.g. "the largest city" is a noun phrase, so it is displayed "NP" in the parse tree, shown in Fig. 4. All type tags come from Penn Treebank II.[2] Σ denotes the terminal symbol set, i.e., leaf nodes consist of words. R is a set of production rules. Each $r \in R$ denotes a generation rule, i.e. $r : A \rightarrow B$, where A indicates a non-terminal node, B means a combination of elements derived from $(\Sigma \cup N)$. E.g., there is a rule "NP = NP + PP", by which "the largest city in the United States" can be divided into "the largest city" and "in the United States". A non-terminal node may have multiple divisions, so P includes the probability of all generation rules, denoted as $P(B|A)$. When a sentence comes, we first initialize a parse tree with the root node S whose value is the content of the sentence. Then we find the rule which has the highest partition probability, i.e. $\underset{B' \in (\Sigma \cup N)}{\operatorname{argmax}} P(B'|S)$. We repeat this step to generate each divided subtree iteratively. Finally, a complete parse tree is constructed.

Phrase-based Question-Answer Understanding extracts core phrases of each post. For a post s_i, we first generate its corresponding syntactic parse tree t_i. We define an auxiliary queue to traverse t_i. Firstly, the root node of t_i is put into the queue. Then we sequentially dequeue the head element and enqueue its direct sub-nodes at the same time. When the queue is empty, t_i is traversed. Each node of the parse tree is tagged according to grammar rules so that we can obtain the node content with specified tags. In our research, we only obtain NP (noun phrase) in the corpus. After the traversal process, we get the phrase representation for each post and integrate all phrases in a phrase base. Then we expand two relationships (P-Q and P-A) on the heterogeneous information network in Section 4.1. If p is a core phrase of a post (question q or answer a), their relationship is denoted as the semantic coverage of the phrase on the post, shown in Eqs. (4) and (5), where $\text{freq}_q(p)$ and $\text{freq}_a(p)$ denote the number of times p appears in the corresponding post, $|a|$ and $|q|$ represent the total number of phrases in the post.

$$Sim_{PQ}(p,q) = \frac{\text{freq}_q(p)}{|q|}, \quad p \in P, \ q \in Q. \quad (4)$$

[1] https://catalog.ldc.upenn.edu/LDC99T42
[2] http://www.surdeanu.info/mihai/teaching/ista555-fall13/readings/PennTreebankConstituents.html

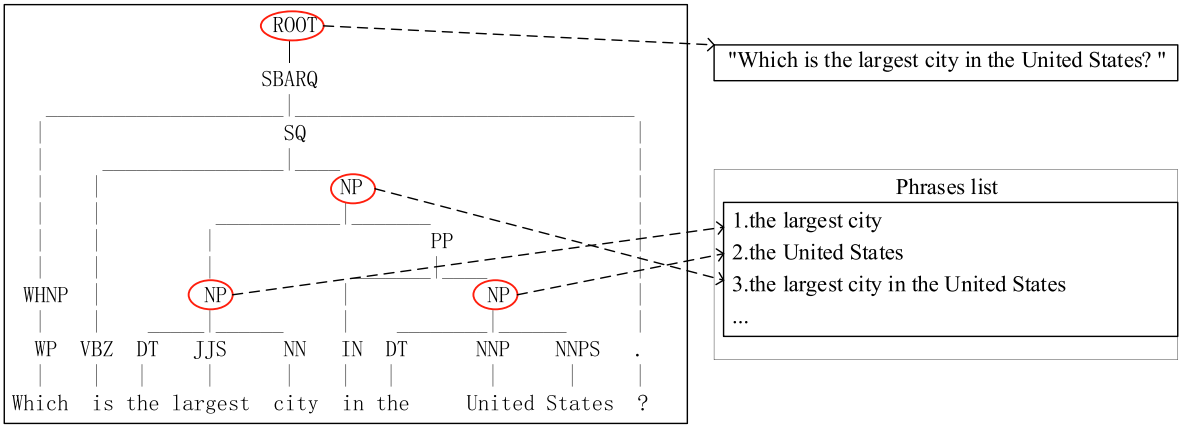


Fig. 4. The structure diagram of the parse tree.

$$Sim_{PA}(p,a) = \frac{\text{freq}_a(p)}{|a|}, \quad p \in P,\ a \in A. \tag{5}$$

Phrase Embedding takes advantage of the co-occurrence phrase pairs in posts to train a neural network for the phrasal vectorized representation. The training process of phrase embedding is inspired by Word2Vec. We train phrase vectors by Skip-Phrase, which predicts context phrases by giving a specific phrase [43]. Skip-Phrase is a three-layer neural network, shown in Fig. 5. The input layer receives a one-hot form of the central phrase. The number of nodes in the hidden layer is equivalent to the dimension of the embedded vector. Our previous research focused on several phrase embedding methods and the determination of related parameters [43]. The vector dimension is proportional to the accuracy of semantics and the training complexity. According to previous research [27,43], we set the dimension of the embedded vector to 300. The output layer contains one-hot representations of all context phrases. Then we iterate the training process until the epoch achieves 5. The connection weight from the input layer to the hidden layer is the embedded representation of the corresponding phrase. Based on the phrase embedding process, each phrase is mapped to a vector in the multi-dimensional space. If phrase p_1 and p_2 are given, their corresponding vectors are $\mathbf{p_1}$ and $\mathbf{p}_2$. Their similarity is defined as Eq. (6), where $||\mathbf{p_1}||$ and $||\mathbf{p}_2||$ denote the norm of the related vector, p_{1i} and p_{2i} represents the value of a particular dimension.

$$\text{Sim}(p_1,p_2) = \frac{\mathbf{p_1}\cdot\mathbf{p_2}}{||\mathbf{p_1}||\times||\mathbf{p_2}||} = \frac{\sum_{i=1}^{|\mathbf{p_1}|} p_{1i}\times p_{2i}}{\sqrt{\sum_{i=1}^{|\mathbf{p_1}|}(p_{1i})^2}\times\sqrt{\sum_{i=1}^{|\mathbf{p_2}|}(p_{2i})^2}}. \tag{6}$$

4.3. User knowledge background model

The most prominent feature of CQA is the introduction of users, who establish the relationship between questions and answers through their behavior. However, most existing research does not consider the knowledge background of askers, which leads to question understanding deviation. In this section, we propose a user knowledge model that obtains the user's background based on their past posts (questions and answers). Based on Section 4.2, we have established the relationships

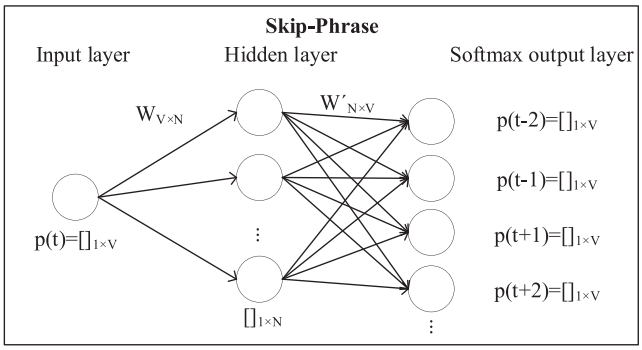


Fig. 5. The neural network structure of Skip-Phrase.

between posts and phrases. So, we formulate the user knowledge as Eq. (7). $URP(u)$ contains all user-related phrases of u. It comes from all posts associating with the user. E.g., if user u is related to question q (a submitting-relationship or liking-relationship), i.e. $Sim_{UQ}(q,u) > 0$, and phrase p is a core phrase of question q, i.e. $Sim_{PQ}(p,q) > 0$, the phrase p belongs to the user-related phrases of u, i.e. $p \in URP(u)$. Then we utilize the phrase whose embedding is closest with $URP(u)$ to represent the user's knowledge $UK(u)$. It is shown in Eq. (7), where PB denotes the phrase base, $\text{count}(p_j)$ counts the frequency of the phrase p_j appeared in $URP(u)$. The similarity of phrases comes from Eq. (6).

$$\begin{aligned} UK(u) &= \underset{p_i \in PB}{\text{argmax}} \left(\sum_{p_j \in URP(u)} (\text{count}(p_j) \times Sim(\mathbf{p_i}, \mathbf{p_j})) \right), \\ URP(u) &= [p | \exists a, Sim_{UA}(a,u) > 0 \wedge Sim_{PA}(p,a) > 0] \\ &\cup [p | \exists q, Sim_{UQ}(q,u) > 0 \wedge Sim_{PQ}(p,q) > 0]. \end{aligned} \tag{7}$$

Fig. 6 shows an example of user knowledge acquisition. It assumes that user u is associated with one question and three answers. We extract the user-related phrases from the post list and count the frequency of each phrase. Then we find a phrase from PB, which is the most relevant with $URP(u)$, to represent the user's knowledge by Eq. (7). Fig. 6 is only to illustrate the process of acquiring user knowledge, so it does not give the complete phrase set and embedding vector values.

4.4. Answer generation model

In this section, our goal is to find relevant knowledge entities based on the user's background and question semantics, then generate the natural language answer, shown in Fig. 3(c). It is implemented in Algorithm 2. Its input includes a knowledge graph consisting of triples, a new question from a user, and the user knowledge from Section 4.3. The output is a natural language answer that matches the question. The implementation of Algorithm 2 is divided into two parts.

In the first stage, we query matching triples in the Knowledge Graph based on the user's background and question semantics by SPARQL. When user u raises a new question q, we obtain its user background $\text{UK}(u)$ by Eq. (7). Algorithm 1 represents q with a phrase set $[p_1, p_2..]$. In lines 3–7 of Algorithm 2, we find the triples related to the question in the scope of user knowledge. Different from previous research, the user's knowledge determines the search range and improves the matching accuracy. The improved query statement adds user knowledge as a query condition, thereby improving efficiency. Fig. 7 gives a sample SPARQL statement. The result of this stage is a set of triples relating to the question.

Algorithm 2: User background-based answer generation
Input: Knowledge Graph $KG = [tri_1, tri_2...]$;
  The user u who posted the question;
  A new question $q = [p_1, p_2..]$.
Output: Natural language answer *TextAnswer*.
1: *resultTri* ← []
2: for p_i in q do
3: for tri_j in KG and in the domain of $\text{UK}(u)$ do
4: if tri_j is associated with p_i then
5: $resultTri.add(tri_j)$
6: end if
7: end for
8: for tri_i in *resultTri* do
9: $oneSent \leftarrow TripleToText(tri_j)$
10: $TextAnswer.add(oneSent)$
11: end for
12: end for
13: Return *TextAnswer*

In the second stage, we employ Triple-to-Text [50] to convert the matched triples into natural language sentences iteratively. It first replaces the entities in the RDF triples with corresponding types to reduce the vocabulary size. E.g. the triple ⟨'Bill Gates', 'founder', 'Microsoft Corporation'⟩ is pre-processed as ⟨PERSON, founded, CORPORATION⟩. Then it employs commas to separate values within an RDF triple, and semicolons to separate different RDFs. Finally, the Seq2Seq model [35] is used to encode the preprocessed triples and generate natural language sentences. The training goal of the Seq2Seq model is to get a generator G_θ and a discriminator M_ϕ. Given a preprocessed sequence pair $\{ (X, Y) \}$, G_θ is defined as a chain product of probability distribution of the next token y_t under the condition of the input sequence X and prefix $y_{<t}$, shown in Eq. (8).

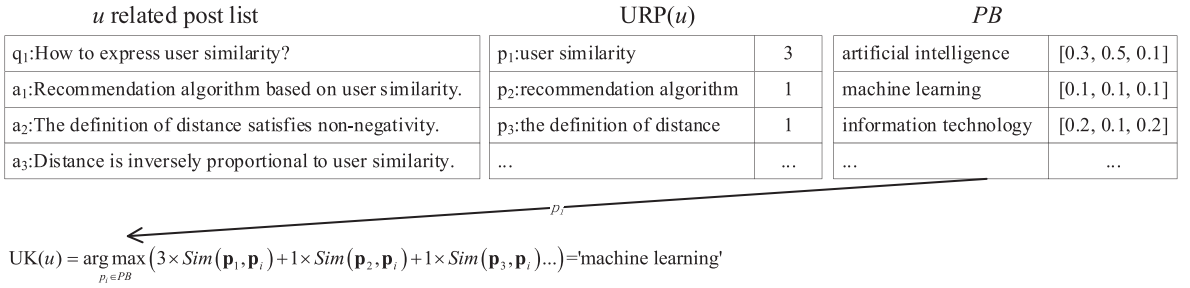


Fig. 6. An example of user knowledge acquisition.

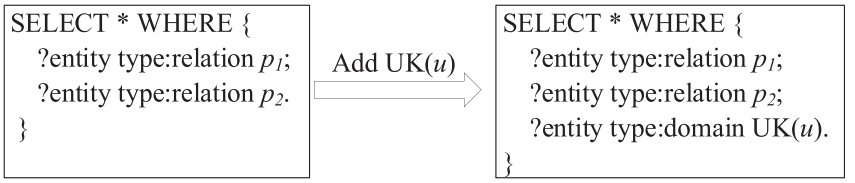


Fig. 7. A SPARQL sample based on user background.

$$G_\theta(Y|X) = \prod_{t=1}^{T} g_\theta(y_t|y_{<t}, X). \tag{8}$$

M_ϕ is also modeled as a chain product of conditional distributions, denoted as Eq. (9). The goal of M_ϕ is to minimize the inverse KL divergence $KL(G_\theta||M_\phi)$, so the objective function is in Eq. (10), where $(X, Y) \sim G_\theta$ is the entity pair, X is sampled from conditional context and Y is the output of the generator.

$$M_\phi(Y|X) = \prod_{t=1}^{T} m_\phi(y_t|y_{<t}, X). \tag{9}$$

$$J_G(\theta) = KL(G_\theta||M_\phi) = E_{(X,Y)\sim G_\theta}\left[\log \frac{G_\theta(Y|X)}{M_\phi(Y|X)}\right]. \tag{10}$$

The final objective function is obtained by bringing Eqs. (8) and (9) into Eq. (10), denoted as Eq. (11). We use G_θ to generate natural language answers from the queried entity triples.

$$J_G(\theta) = E_{(X,Y)\sim G_\theta}\left[\sum_{t=1}^{T}(\log g_\theta(y_t|y_{<t}, X) - \log m_\phi(y_t|y_{<t}, X))\right]. \tag{11}$$

4.5. Community answer generation method based on knowledge graph

In this section, we summarize the overall structure of the CAGKG in Algorithm 3. Its pipeline is divided into six parts. Firstly, we employ a HIN to represent complex entity relationships in CQA. Then we parse the posts and represent them by phrases, according to Algorithm 1. Then we get the phrase similarity according to Skip-Phrase. To more accurately understand the question, we also model the background of the asker. We combine the user background and question semantics to query relevant knowledge in the knowledge graph. Finally, these triples are transformed into a natural language answer to the new question.

Algorithm 3: Community Answer Generation based on Knowledge Graph

1: Build a HIN for CQA in Section 4.1.
2: Get parse trees of the posts by Algorithm 1.
3: Obtain the phrase base and the phrase embedding by Algorithm 1.
4: Model the user's background based on history posts in Eq. (7).
5: Retrieve relevant knowledge entities in the user knowledge by Algorithm 2.
6: Generate natural language answers based on relevant entities by Algorithm 2.

4.6. Phrase-based answers semantic similarity evaluation

Previous studies have shown some indicators, such as BLEU and ROUGE, to measure the coverage of words or word sequences between texts [23,29]. However, the cover rate of words does not represent the semantic similarity of texts directly. For example, in a machine translation task, the same piece of text can have many different translation forms with non-overlapping word sequences. E.g. "New York is the most populous city in the United States." and "The most crowded city in the USA is the Big Apple city." have the same meaning but few overlapping word sequences. In QAS, a question may contain multiple standard answers. E.g., when a user posts the question "How to improve spoken English?". The best answer may be "Continuous practice of spoken pronunciation." or "Try to speak English as much as possible.".

In this section, we present a Phrase-based Answers Semantic Similarity Evaluation indicator, called PASSE. That is given two texts: $T_1 = \{ p_{11}, p_{12}, ..p_{1n} \}$ and $T_2 = \{ p_{21}, p_{22}, ..p_{2m} \}$. p_{ij} denotes the jth phrase in the text i. n and m are the length of texts. The relation between a phrase and a piece of text is defined as the similarity between p and p_i which is the closest to p in T. It is shown in Eq. (12). The phrase similarity has been obtained in Section 4.2. Then the semantic similarity between texts is defined as Eq. (13). Its first part shows the weighted similarity of T_1 to T_2. It is the semantic coverage of T_1 to T_2 like Precision. The second part has the same processing and equivalent to Recall.

$$Sim(p, T) = \max_{p_i \in T, i=1,2,...} \mathrm{Sim}(p, p_i). \tag{12}$$

$$\mathrm{PASSE}(T_1, T_2) = \frac{\sum_{p_i \in T_1, i=1,2,...n} Sim(p_i, T_2)}{n} \times \frac{\sum_{p_j \in T_2, j=1,2,...m} Sim(p_j, T_1)}{m}. \tag{13}$$

PASSE can be used for measuring the semantic overlap of two texts. There are two main ideas for its application. The first one is to evaluate the performance of answer generation algorithms from the perspective of semantic matching. The second one is to determine whether new answers need to be generated (usually comparing a threshold with the relevance of existing answers and questions). QA corpus gives standard answers (accepted answers) to questions, so we think the right answers should have the same semantics with them. In this paper, we employ it to calculate the semantic coverage of generated answers and reference answers. E.g., given baseline methods M1 and M2, we obtain their predicted answers. For method M1, we calculate PASSE of each generated answer and the corresponding reference answer, and finally average all results to obtain PASSE_{M1}. Similarly, we get PASSE_{M2}. If $\mathrm{PASSE}_{M1} > \mathrm{PASSE}_{M2}$, we believe that the answers generated by M1 are closer to reference answers in terms of semantics than M2, and vice versa. Regarding the second usage, we will conduct in-depth research in future work.

PASSE can also be applied to the evaluation of other NLP tasks. For example, it can evaluate the similarity between manual results and computer-generated texts in machine translation. In the process of summary generation, it may measure the correlation between the original abstract and machine-generated text. In our work, we employ it to evaluate the semantic overlap between generated answers and reference answers.

5. Experiments

5.1. Dataset and knowledge graph

Our experiments are based on question answering datasets and typical Knowledge Graphs. The former provides real question–answer pairs to train models and evaluate generated answers. We obtain three prevalent CQA datasets from Stack Exchange, consisting of Stack Overflow for computer programming questions, Super User for computer enthusiasts and power users, and Mathematics for math-related questions. These datasets are real-time updated, so we obtain the data from Stack Exchange Data Dump.[3] Meanwhile, we also randomly crawled 15,000 questions from the most ten popular spaces on Quora.[4] Then, we grabbed relevant entity information from the page of each question, including the user name of the asker, posting history of the asker, the highest-rated answer, user name of the answerer, posting history of the answerer. These crawled questions and related entities constitute the fourth data set, called Quora. To compare their entity information, we count the number of questions, answers, and users. In this paper, we employ phrases to supplement the entity-relationship, so we extract phrases and count the number of phrases in different datasets separately. The number of phrases is proportional to the count of posts approximately, so Stack Overflow has the most phrases. The details of the CQA datasets are shown in Table 2.

Knowledge Graph is a broad knowledge base that stores a vast number of entities and relationships in triples. In our experiments, comparison algorithms generate candidate answers base on two Knowledge Graphs. Freebase is a large collaborative knowledge base derived from the contributions of community members.[5] It often acts as an external knowledge in NLP tasks. DBpedia stores structured knowledge from Wikipedia in the form of entity relationships.[6] Table 3 shows the statistics of the Knowledge Graphs.

[3] https://archive.org/details/stackexchange
[4] https://github.com/scku/Quora-Crawler

The goal of our research is to query the relevant triples from KG and convert them into natural language answers. Both questions and knowledge bases have domain attributes, so the best solution is to find triples in the most relevant knowledge base. We employ the co-occurrence ratio of phrases to show the intersection between each KG and the datasets, shown in Table 4. E.g., 29.86% of extracted phrases from Quora appeared in Freebase. Through comparative analysis, we found that the phrase coverage of DBpedia is higher than Freebase, and the main reason is that Freebase has stopped updating in 2016. In real CQA applications, the choice of knowledge base depends on many factors, such as domain knowledge and response time. In our experiment, we compared the generated answers by the baseline methods in both two knowledge graphs.

5.2. Baselines and evaluation indicators

To assess the effectiveness and efficiency of CAGKG, we selected six excellent studies as baseline methods, which focus on the answer selection or generation in QA. The answers from all baseline methods are optimized based on knowledge graphs.

- AGEQ [10] can automatically find similar questions in the CQA through the title and content of questions. Experiments show that it has certain advantages in the retrieval of related questions, but it does not consider the impact of user knowledge on the issue.
- ANLQSM [16] employs subgraph matching over Knowledge Graph to answer natural language questions. It considers question disambiguation in the process of answering queries. Experiments show that it improved in accuracy and query performance.
- UIA-LSTM-CNN [41] solves answer selection in CQA by proposing a deep attention neural network. It expresses local importance by intra-sentence attention and incorporates user information into the answer selection. Experiments show that it is superior to other baseline methods in answer selection.
- QUEST [26] is an unsupervised method for answering complex questions. It computes similarity joins using partial results from different documents. Experiments show that it is superior to other baseline methods for answering complex questions.
- INLQA [49] is a data-driven approach for answering natural language questions based on the Knowledge Graph. It considers user interaction in the process of question understanding. Experiments show that it outperforms the existing question answering methods.
- CAGKG-WOUK is the version of CAGKG without user knowledge. Its overall process is the same as CAGKG, but only the question semantics is used for querying triples in the knowledge graph. In the comparative experiments, we employ its results to evaluate the role of user knowledge on CAGKG.

Most QA evaluation indicators come from machine translation or automatic text summary. We list common evaluation indicators and give a brief introduction as follows.

- BLEU: It utilizes the number of co-occurring n-grams to calculate the matching degree between generated texts and standard texts.
- NIST [7]: It first calculates the amount of information of n-gram in reference texts. Then, it calculates the relevance of texts based on the weighted sum of n-grams.
- ROUGE-N: It refers to the overlap of n-grams between the generated text and reference text. E.g., ROUGE-1 is the overlap of unigrams. ROUGE-2 focuses on the overlap of bigrams.
- ROUGE-L: It uses LCS (Longest Common Subsequence) to measures the longest matching sequence of words. It does not require consecutive matches but in-sequence matches, that reflect sentence-level word order.
- ROUGE-W: It improves ROUGE-L by giving greater weight to consecutively matched word sequences.
- ROUGE-S: It uses Skip-Bigram to generate discontinuous word pairs for measuring the word order at the sentence level.
- ROUGE-SU: It employs Skip-Bigram and unigram together to calculate co-occurrence statistics for evaluating the relevance of texts.
- METEOR [2]: It introduces WordNet to supplement the thesaurus and measures the fluency of sentences through chunk alignment.

The above indicators could evaluate the relevance between texts from different granularities and different aspects. They are suitable for different research tasks. Researchers should choose appropriate evaluation indexes according to the goal of specific tasks and the feature of indicators. Table 5 briefly summarizes the features of different evaluation indicators. Based on previous research [40,50], our baseline methods mainly focus on the semantics similarity of texts based on n-grams or phrases, so we select BLEU and ROUGE-N to compare Precision and Recall of baselines.

BLEU (Bilingual Evaluation Understudy) measures the similarity of natural language texts. It is regularly used to evaluate the semantic relevance between machine-generated text and original text, so many natural language processing tasks (such

[5] https://developers.google.com/freebase
[6] https://wiki.dbpedia.org/develop/datasets

Table 3
Statistics of knowledge graphs.

Knowledge Graph	Freebase	DBpedia
Number of Triples	1.9 billion	13 billion
Number of Entities	2.3M	6.6M
Number of Relations	3.1M	8.5M

Table 4
The intersect between each KG and datasets.

Corpus	Mathematics	Super User	Stack Overflow	Quora
Freebase	32.16%	38.52%	39.47%	29.86%
DBpedia	35.83%	40.62%	43.38%	32.15%

as machine translation, question answering, dialogue system) choose it as an evaluation indicator. It has a value between 0 and 1, which indicates the similarity between the texts: 1 means that the semantics are entirely identical, and vice versa [29]. The modified n-gram precision of answers is defined as Eq. (14). Considering the effect of the answer length on Precision, we introduce BP (Brevity Penalty) in Eq. (15) and obtain the calculation formula for BLEU in Eq. (16). In our experiments, we employed $N \leqslant 4$ and uniform weights $w_n = 1/N$.

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{ngram \in C} Count_{clip}(ngram)}{\sum_{C' \in \{Candidates\}} \sum_{ngram' \in C'} Count(ngram')}. \tag{14}$$

$$\mathrm{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leqslant r \end{cases}. \tag{15}$$

$$\mathrm{BLEU} = \mathrm{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right). \tag{16}$$

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is frequently employed to evaluate the difference between machine-generated text and standard text [23]. It is based on the co-occurrence probability of n-gram to examine the adequacy of the semantic coverage of two texts. It is defined as Eq. (17). In our experiment, we employ it to evaluate the similarity between the generated answer and the reference answer which is called the accepted answer coming from CQA corpus. Its numerator indicates the number of n-grams that appear in the generated answer and the reference answer simultaneously. Its denominator indicates the number of n-grams in the reference answer. The value range of N is defined from 1 to 4.

$$\mathrm{ROUGE\text{ - }N} = \frac{\sum_{S \in \{ReferenceAnswer\}} \sum_{ngram \in S} Count_{match}(ngram)}{\sum_{S \in \{ReferenceAnswer\}} \sum_{ngram \in S} Count(ngram)} \tag{17}$$

5.3. Experiment setup

Our experiments are implemented by python 3.7. Our experimental equipment configuration is Intel Core (TM) i7-5500U, 16G RAM, Windows operating system. The setting associated with our framework is as follows. For each corpus, we extract all core phrases in the training set to form the phrase base and train the phrase embedding for subsequent tasks. The detailed process refers to Section 4.2. Following our previous research [43], we set the number of nodes in the hidden layer to 300, which is the dimension of the embedding space. Before comparative experiments, four datasets are obtained and preprocessed, e.g., removing images or non-character information from posts (questions and answers), filtering out the posts with lower ratings, deleting users who have not submitted any post. In the four datasets, each question may contain multiple answers. Asker has marked the best reply as the accepted answer, which is called the reference answer in our experiment. The accepted answers may not the most professional answers (e.g., including non-professional terminology or redundant description), but they apply to users' questions. In the evaluation phase, we will calculate the similarity between generated answers and accepted answers to compare the performance of the baseline methods.

Knowledge graphs (including Freebase and DBpedia) are presented as triples, e.g., ⟨E1, R, E2⟩, where E1, R, E2 may be specific attribute value or refer to another triple entity. If anyone of E1, R, and E2 is empty, or its associated object cannot be found, we call it the incomplete triple. In the preprocessing stage, we remove incomplete triples in Freebase and DBpedia

Table 5
The characteristics of evaluation indicators.

Feature	Weighted	Continuity	Text coverage	Dependency	Granularity	Precision	Recall
BLEU	No	Yes	all	–	n-gram	Yes	–
NIST	Yes	Yes	all	–	n-gram	Yes	–
ROUGE-N	No	Yes	all	–	n-gram	–	Yes
ROUGE-L	No	No	partial	–	LCS	–	–
ROUGE-W	Yes	No	partial	–	LCS	–	–
ROUGE-S	No	No	partial	–	Skip-Bigram	–	–
ROUGE-SU	No	No	all	–	Skip-Bigram + unigram	–	–
METEOR	No	No	all	WordNet	unigram	–	–

to avoid meaningless matching in the entity retrieve. During the experiment, we select three evaluation indicators (BLEU, ROUGE, and PASSE) to compare the baseline methods. In each round of experiments, we perform baseline methods and obtain their predicted answers. Then, we compare the predicted answers with the corresponding reference answers and calculate the average indicators for evaluation. To avoid randomness, we divide the experimental datasets by 2–8 ratios and repeat ten times to get the final average result.

5.4. Experiment analysis of CAGKG

Experiment results are shown in Tables 6 and 7. In different knowledge bases, we compared various indicators of baseline methods and conducted an in-depth analysis of the following aspects.

- We compare the performance of the baseline methods on different datasets. Tables 6 and 7 show the performance of different baseline methods on four datasets. CAGKG achieves the best performance. There are two critical reasons. Firstly, CAGKG uses core phrases to associate knowledge base entities, which improves the accuracy of the generated answers. Secondly, the user's knowledge avoids interference from similar phrases in different fields. AGEQ uses the keywords of new questions to find relevant questions with accepted answers [10]. It ignores the diversity between different issues. Thus, some resulting answers only match part of the question. ANLQSM converts natural language questions into semantic query graphs and finds answers from RDF resources [16]. It ignores the role of user knowledge in CQA, causing the question to match relevant knowledge in other fields. UIA-LSTM-CNN considers the role of user information in the answer selection process [41], but obtained answers are only derived from the existing answers, which results in a lower degree of matching. QUEST retrieves documents related to questions from a text corpus based on words [26]. So, if the related documents cannot be matched entirely based on the literal word, or the question is beyond the scope of documents, the answers cannot be obtained, which affects its QA performance. INLQA considers text semantics in the question understanding stage but ignores the background knowledge of askers [49], which limits the experimental performance. From the above analysis, we conclude that CAGKG employs phrases to represent question semantics and user backgrounds, so it achieves better experimental results.
- We analyze the impact of different knowledge graphs on the baseline methods. It is found that the baseline methods perform differently in distinct knowledge graphs. Based on Table 3, we believe that the extended knowledge graph can promote the performance of the answer generation.
- To evaluate the role of user knowledge in CAGKG, we focus on the performance of CAGKG-WOUK. The experimental results of CAGKG-WOUK are almost superior to all baseline methods, except CAGKG. Its algorithm flow is the same as CAGKG, except that it removes user knowledge when querying the knowledge graph. The key reason is that it utilizes phrases to represent question semantics, thus ensuring performance. Comparing the experimental results of CAGKG and CAGKG-WOUK, we find that the former is slightly better than the latter, which is affected by the user knowledge. The conclusion is that the phrase representation of question semantics and the user's background both have a positive effect on CAGKG, and the former plays a more significant role.
- In our comparative experiments, we employ three indicators (BLEU, ROUGE, and PASSE) to evaluate the baseline methods. By comparing experimental results, we find that both BLEU and ROUGE reflect the similar relationship between texts, i.e., the larger their value, the more relevant the generated answer and the reference answer. However, experimental results about BLEU and ROUGE of different baseline methods are very similar. E.g., the ROUGE of UIA-LSTM-CNN and QUEST are almost equal in Mathematics. The main reason is that BLEU and ROUGH only focus on the literal match between words and n-grams, which ignores the semantics of generated answers. To illustrate the impact of semantic relevance to user evaluation, we calculated the average correlation between 3000 questions selected from each corpus and their worst-evaluated answers (answers with the fewest likes). From Table 8, we find that even the worst-rated answers still have certain similarities to the questions if we evaluate by BLEU and ROUGE. However, PASSE gives a meager correlation between the worst-rated answers and the questions, which is consistent with real user evaluation. To sum up, PASSE can evaluate the question answering system more effectively from the perspective of semantic matching.

Table 6
Experimental results obtained on different datasets in Freebase. Bold marks the best performance for each indicator.

Methods	Mathematics			Super User			Stack Overflow			Quora		
	BLEU	ROUGE	PASSE	BLEU	ROUGE	PASSE	BLEU	ROUGE	PASSE	BLEU	ROUGE	PASSE
AGEQ	0.570	0.457	0.315	0.538	0.422	0.312	0.572	0.452	0.332	0.487	0.412	0.306
ANLQSM	0.615	0.435	0.338	0.590	0.432	0.326	0.593	0.469	0.350	0.503	0.423	0.322
UIA-LSTM-CNN	0.592	0.471	0.355	0.584	0.442	0.340	0.627	0.462	0.357	0.491	0.418	0.331
QUEST	0.635	0.470	0.347	0.620	0.458	0.331	0.660	0.450	0.353	0.505	0.421	0.335
INLQA	0.654	0.488	0.351	0.627	0.458	0.360	0.688	0.473	0.367	0.514	0.427	0.348
CAGKG-WOUK	0.667	0.485	0.435	0.636	0.461	0.454	0.705	0.485	0.502	0.519	0.429	0.473
CAGKG	**0.698**	**0.490**	**0.519**	**0.654**	**0.468**	**0.498**	**0.711**	**0.493**	**0.539**	**0.523**	**0.431**	**0.483**

Table 7
Experimental results obtained on different datasets in DBpedia. Bold marks the best performance for each indicator.

Methods	Mathematics			Super User			Stack Overflow			Quora		
	BLEU	ROUGE	PASSE	BLEU	ROUGE	PASSE	BLEU	ROUGE	PASSE	BLEU	ROUGE	PASSE
AGEQ	0.595	0.479	0.338	0.56	0.426	0.339	0.596	0.478	0.335	0.494	0.435	0.343
ANLQSM	0.622	0.458	0.363	0.604	0.445	0.35	0.614	0.498	0.373	0.517	0.457	0.358
UIA-LSTM-CNN	0.618	0.474	0.371	0.603	0.465	0.366	0.64	0.492	0.383	0.507	0.442	0.364
QUEST	0.66	0.497	0.356	0.621	0.474	0.353	0.663	0.455	0.363	0.512	0.458	0.372
INLQA	0.677	0.49	0.369	0.627	0.466	0.37	0.692	0.481	0.368	0.527	0.447	0.385
CAGKG-WOUK	0.708	0.493	0.495	0.673	0.485	0.475	0.709	0.502	0.541	0.574	0.452	0.497
CAGKG	**0.721**	**0.499**	**0.521**	**0.681**	**0.494**	**0.51**	**0.721**	**0.515**	**0.556**	**0.584**	**0.468**	**0.502**

In summary, we verified the effectiveness of CAGKG, and deeply analyzed the effect of dataset, knowledge graph, and user knowledge on the experimental results. Moreover, PASSE is a promising valuation indicator for semantic matching.

5.5. Performance analysis of CAGKG

To verify the efficiency of CAGKG, we compared the testing time of different baseline methods on four datasets, shown in Fig. 8. Overall, CAGKG has certain advantages under various comparison conditions. We analyze the experimental results from the following aspects and draw relevant conclusions.

- Fig. 8 shows that CAGKG consumes less testing time than other baselines. The critical factor is that it employs phrases for knowledge base matching, which improves accuracy and reduces matching time. ANLQSM matches the RDF resources through query graphs. It takes much time to eliminate question ambiguity in query graph matching, which affects its efficiency. AGEQ and QUEST get answers from existing texts or posts, which have huge volumes, so they get lower operating efficiency. UIA-LSTM-CNN employs a mixed attention model to learn the relationship between questions and answers, which requires more time to obtain the local importance of words. INLQA is a data-driven approach and gains user features through multiple rounds of user interaction, which affects the operating efficiency. Also, most baselines are based on word matching, which costs more matching time. The above analysis demonstrates that CAGKG is superior to the baseline methods in terms of testing time.
- By comparing the experimental results from the perspective of different knowledge graphs, we find that the baseline methods generally perform higher efficiency on Freebase than on DBpedia. The main reason is that the larger knowledge graph improves algorithm accuracy, but consumes more running time simultaneously to match or optimize answers.
- To verify the role of user knowledge on CAGKG, we compare the results of CAGKG and CAGKG-WOUK. The former is more efficient than the latter. The primary time consumption of both is question semantic expression and knowledge graph retrieval. User knowledge significantly improves the matching efficiency of question semantics in the knowledge graph, and a small number of accurately matched entities also saves answer generation time.

In summary, CAGKG is superior to the baseline methods in terms of efficiency.

5.6. Phrase dimensional analysis

The foundation of CAGKG is to utilize phrases to represent question semantics. The vector dimension is an essential parameter for phrase embedding, so this section analyzes its effect on CAGKG from the aspects of the accuracy and running time.

Fig. 9 shows the trend of PASSE as the phrase embedding dimension changes. It is found that PASSE is proportional to the phrase dimension both on Freebase and DBpedia. The essential reason is that high-dimensional phrase vectors represent the

Table 8
The similarity evaluation of questions and worst rated answers in different data sets.

Corpus	Mathematics	Super User	Stack Overflow	Quora
BLEU	0.333	0.356	0.368	0.281
ROUGE	0.268	0.254	0.256	0.274
PASSE	0.087	0.079	0.093	0.091

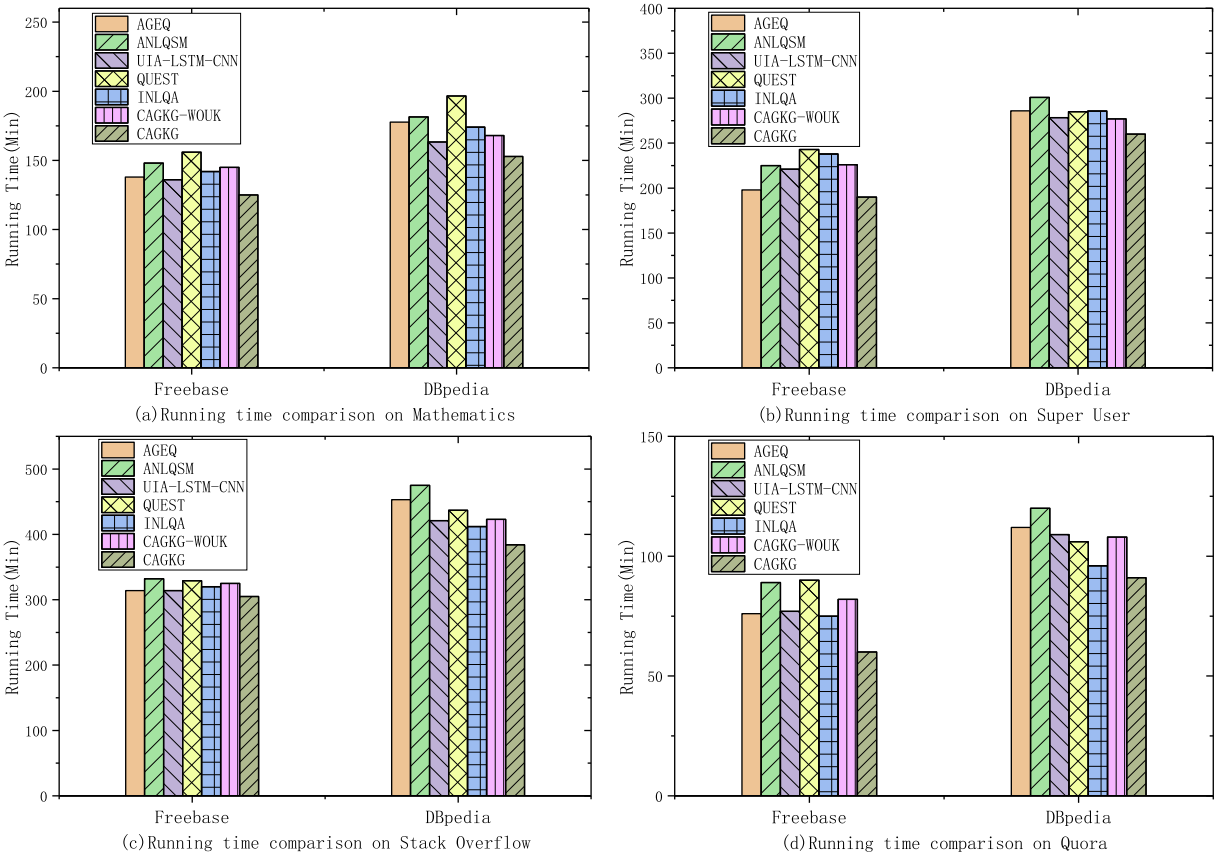


Fig. 8. Testing time comparison of different baseline methods.

question semantics more accurately, thus improving the performance of CAGKG. However, they are not linear, and as the dimension increases, the performance grows slowly. Experimental results show that before the dimension reaches 300, the promotion rate is higher. After that, it is relatively flat. Therefore, setting the phrase dimension to 300 can reflect the performance of CAGKG to a certain extent.

The increase of the embedding dimension will raise computational complexity. Fig. 10 shows the relation between CAGKG's running time and the embedding dimension. Both on Freebase and DBpedia, the rising of the embedding dimension reduces the operating efficiency because most of the essential functions of CAGKG (such as question-semantic extraction, user knowledge representation, and phrases embedding) are based on phrases. The experimental results show that the running time growth rate changes slower before the dimension reaches 300. We speculate that the increase of the embedding dimension not only affects the computational complexity but also bring more machine load.

Based on the experimental results, we set 300 as the phrase embedding dimension, which is a compromise between performance and running time.

5.7. Performance analysis of PASSE

Computational efficiency is an essential index of text similarity calculation. In our research, PASSE is used to calculate the similarity between the generated answer and the reference answer (the accepted answer). Therefore, we conducted some experiments in terms of the number of processed answer-pairs and the frequency of basic unit comparisons (word to word, n-gram to n-gram, or phrase to phrase) within a fixed time.

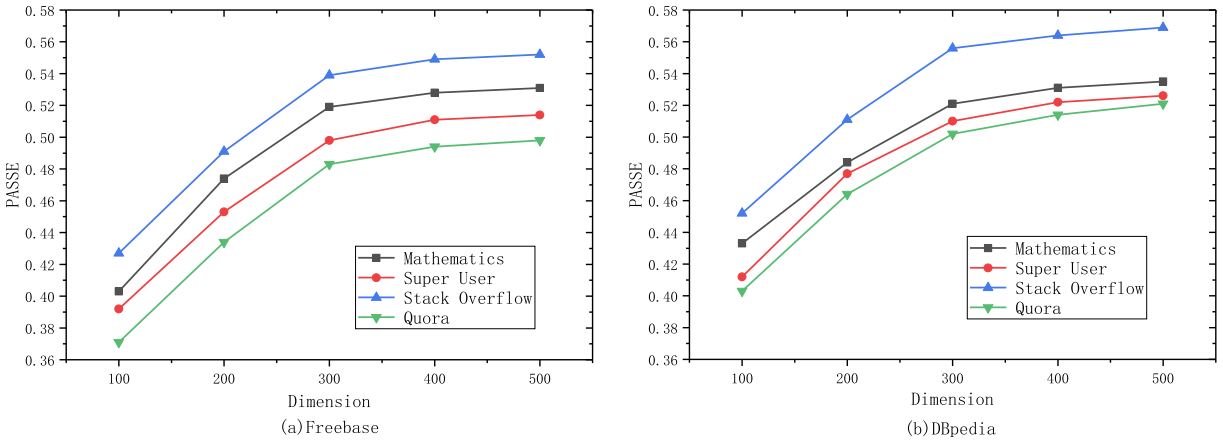


Fig. 9. PASSE comparison of different Phrase embedding dimension.

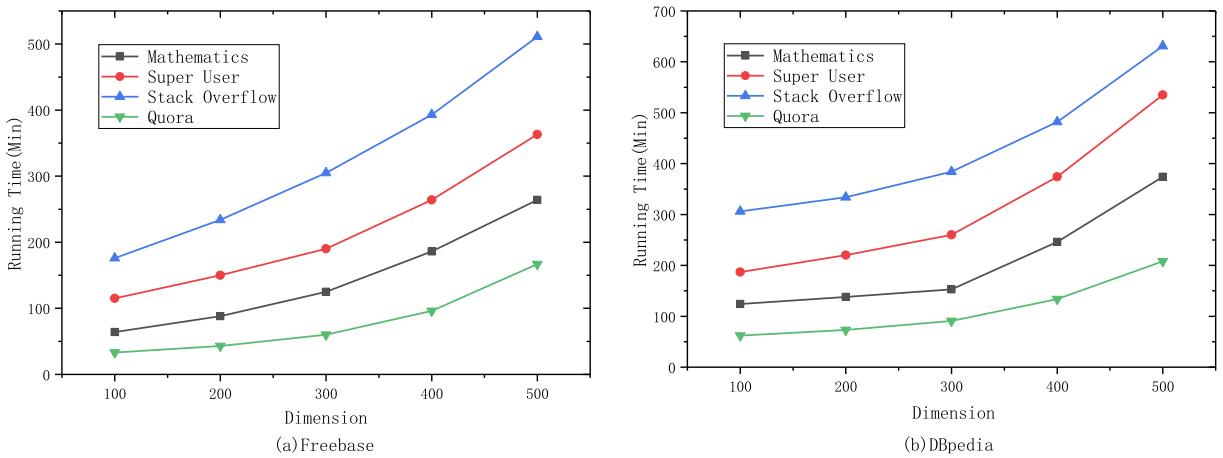


Fig. 10. Testing time comparison of different Phrase embedding dimension.

We compare PASSE with BLEU and ROUGE, based on 30,000 answer pairs, consisting of generated answers by CAGKG and the reference answers in Stack Overflow. Fig. 11 shows the answer contrast performance in the same period. In general, their efficiency is stable because the cumulative number of compared answer-pairs and the comparison frequency of essential elements (words, n-grams, and phrases) both are linearly related to the running time. Experimental results display that PASSE takes less time to complete the goal, shown in Fig. 11(a), which shows that its comparing effectiveness is ahead of BLEU and ROUGE. To explore its essence, we also counted the comparison number of essential elements in different indicators, shown in Fig. 11(b). We find that the comparison frequency of PASSE is far less than the others. BLEU and ROUGE utilize words and n-grams as comparison units. However, PASSE compares in phrases, which is the key to complete the task with less time and fewer comparison frequencies. In summary, PASSE decreases the comparison frequency, so it has certain efficiency advantages in terms of text semantic evaluation.

5.8. Study cases

Based on Sections 5.1 and 5.3, we introduce a study case based on actual CQA data, shown in Table 9. It contains three typical questions, selected from the Stack Overflow corpus. The answers generated by CAGKG have the same meaning as the accepted answers. In Question 1, CAGKG combined the user's background ('Java language') and question semantics ('comparing string') to query the Knowledge Graph. The generated answer covers all the semantics of the accepted answer. Question 2 contains multiple question statements that involve various knowledge entities. CAGKG generates multiple query statements and converts the queried knowledge entities into a unified answer. Question 3 is a multi-statement problem with interdependence, so CAGKG generates a set of dependent query statements. The generated answer also matches the accepted

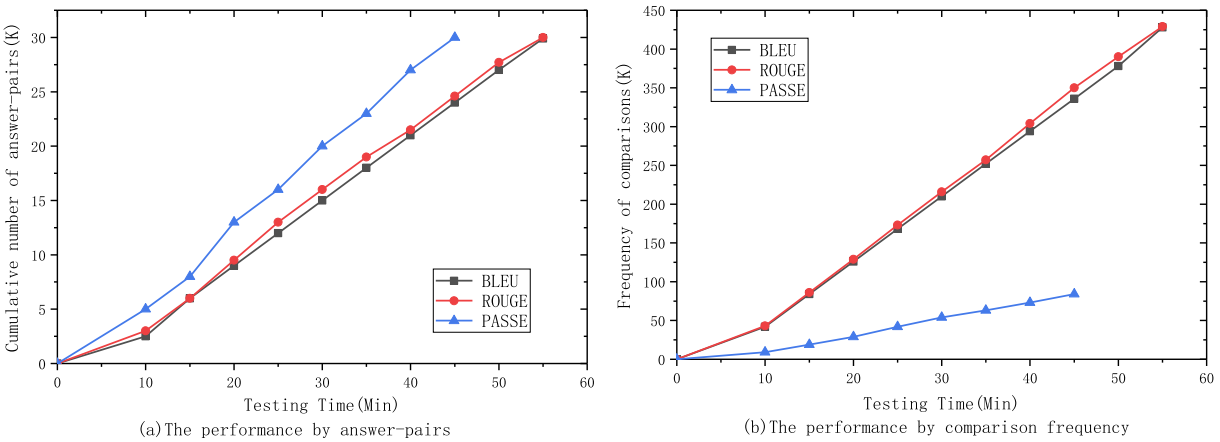


Fig. 11. Efficiency comparison of BLEU, ROUGE, and PASSE.

answer semantically, but the answer form is relatively single. Therefore, CAGKG alleviates the issue of generating natural language answers to a certain extent.

5.9. Discussion

As we mentioned in Section 1, CQA faces some challenges, including question understanding, natural language answer generation, and answer evaluation. For question understanding, CAGKG employs phrases to represent the question semantics and combines user knowledge to constrain the field of the issue. According to the question semantics, we retrieve the related entities from knowledge graphs and convert them into natural language answers. Also, we propose PASSE for evaluating the answer similarity from a semantic perspective.

Experimental results indicate that CAGKG leads baseline methods in terms of generated answers accuracy and running efficiency. The main reason comes from two aspects. Firstly, CAGKG utilizes phrases in extracting the semantics of questions, which promotes the development of question understanding from word granularity to phrase granularity. User background knowledge constrains the question domain, which helps to understand the question accurately and improve retrieval efficiency. Secondly, comparing with word-based baseline methods, CAGKG utilizes fewer phrase matching queries to retrieve the Knowledge Graph, which saves running time effectively. E.g., given a question about "Relational database system", word-based methods need to query triples based on each word and overlapping n-grams repeatedly. CAGKG retrieves the phrase from the knowledge graph just once. These findings confirm that phrases can promote the research on question understanding and help other researchers to design better CQA solutions.

Our experiments also show that PASSE has certain advantages in the evaluation of semantic similarity. It uses phrase coverage instead of text matching to improve the semantic similarity and reduce the comparison frequency of essential elements. PASSE promotes the development of text similarity evaluation from literal matching to semantic relevance and points out the direction for semantic evaluation of NLP tasks.

Everything has two sides. Objectively speaking, CAGKG is based on phrases, so it requires more pre-processing work (such as phrase mining and phrase embedding). Besides, the calculation process of PASSE is more complicated than word matching. Some unresolved issues may be our future research direction.

- Answer selection and answer generation are two existing CQA routes. When new questions are coming, if there are very similar questions, it is wise and efficient to choose existing answers. Otherwise, answer generation becomes more sensible. We will research the matching threshold between questions and the existing answers, which determines the way to obtain answers.
- It is valuable to analyze the impact of user background on answers deeply. Our work proves that user background can promote the question understanding from a holistic perspective. However, there may be some exceptional cases where it has the opposite conclusion. E.g., users ask some questions that have nothing to do with their background.
- CQA allows users to provide many answers for a question, and then the asker can choose the optimal response. Therefore, if we produce multiple answers with the same semantic, it may further promote user satisfaction. The diversity of natural language generation will be a promising direction.
- Our research employs the degree of semantic similarity between generated answers and accepted answers as the evaluation criterion. It makes sense to analyze the feature of generated answers and human answers deeply.

Table 9
Community answer generation process and some study cases.

Question 1	How do I compare strings?
User background	Java language
Query statement	<comparing string, function, X >&&<X, *any relation*, Java language >
Match triples	<'==',function, comparing the reference of string objects ><'equals()',function, comparing the value of string objects >
Accepted answer	== tests for reference equality (whether they are the same object).equals() tests for value equality (whether they are logically "equal").
Generated answer	The function of '==' is comparing the reference of string objects. The function of 'equals()' is comparing the value of string objects.
Question 2	How to define a Class? How to set an Interface?
User background	Java language
Query statement	<Class, define, X >&&<X, *any relation*, Java language><Interface, define, Y>&&<Y, *any relation*, Java language>
Match triples	<Class, format as, "public class classname"><Interface, format as, "public interface interfacename">
Accepted answer	In the Java language, the simplest form of a class definition is class < class_name > .The interface is defined as interface < interface_name > .
Generated answer	The format of Class is "public class classname". The format of Interface is "public interface interfacename".
Question 3	What are the common network software architectures? What is the difference between their functions?
User background	Software engineering
Query statement	<network software architectures, include, X >&&<X, *any relation*, Software engineering ><X, equal, Y>&&<Y, *any relation*, Software engineering >
Match triples	<network software architecture, compose, BS and CS><BS, equal, Browser/Server><CS, equal, Client/Server>
Accepted answer	They are BS and CS. BS is short for Browser and Server. The user accesses the server through a browser. CS is short for Client and Server. Users need to access the server through a compatible client.
Generated answer	The network software architectures include BS and CS. BS is equal to Browser/Server. CS is equal to Client/Server.

6. Conclusions

To sum up, we propose CAGKG, which employs phrases to represent question semantic and generates natural language answers based on the knowledge graph. Firstly, we traverse the parsing tree to extract core phrases of posts and learn the similarity among phrases by co-occurrence frequency. Then we model the user's background through semantic analysis of past posts. Finally, we query related entities in the knowledge graph through the user background and the question semantics, then convert the matching entities into natural language answers. To evaluate generated answers, we propose the PASSE, which employs phrase overlay to evaluate text similarity from a semantic perspective. Experiments show that CAGKG outperforms baseline methods in BLEU and ROUGE. By analyzing the experimental results, we find that PASSE can measure the semantic similarity of texts effectively. In summary, CAGKG employs phrases to represent post semantics and user knowledge, which effectively improves the performance of the answer generation. PASSE is also a promising semantic evaluation indicator.

In the future, we plan to (1) improve CAGKG to generate complex answers, e.g., the summary answer or the detailed explanation; (2) propose an answer selecting approach based on phrase-fuse heterogeneous information network and combine CAGKG to form a comprehensive CQA solution; (3) apply PASSE in other NLP tasks for semantic evaluation, such as social analysis and dialogue system; (4) extend text generation based on knowledge graph to other NLP tasks, e.g., machine translation and text summary.

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.

Acknowledgments

This research is partially supported by The National Social Science Fund of China No. 18ZDA200, No. 13&ZD091 and The Key R & D project of Hebei Province No. 20370301D. Shuliang Zhao is the corresponding author.

References

[1] J. Araki, D. Rajagopal, S. Sankaranarayanan, S. Holm, Y. Yamakawa, T. Mitamura, Generating questions and multiple-choice answers using semantic analysis of texts, in: 26th International Conference on Computational Linguistics, ACL, 2016, pp. 1125–1136

[2] S. Banerjee, A. Lavie, METEOR: an automatic metric for MT evaluation with improved correlation with human judgments, in: Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, 2005, pp. 65–72.
[3] J.-W. Bao, D. Tang, N. Duan, Z. Yan, M. Zhou, T. Zhao, Text generation from tables, IEEE/ACM Trans. Audio Speech Lang. Process. 27 (2019) 311–320, https://doi.org/10.1109/TASLP.2018.2878381.
[4] X. Cheng, S. Zhu, S. Su, G. Chen, A multi-objective optimization approach for question routing in community question answering services, IEEE Trans. Knowl. Data Eng. 29 (2017) 1779–1792, https://doi.org/10.1109/TKDE.2017.2696008.
[5] V.V. Datla, T.R. Arora, J. Liu, V. Adduru, S.A. Hasan, K. Lee, A. Qadir, Y. Ling, A. Prakash, O. Farri, Open domain real-time question answering based on asynchronous multiperspective context-driven retrieval and neural paraphrasing, in: Proceedings of The Twenty-Sixth Text REtrieval Conference, 2017
[6] D. Diefenbach, V. López, K. Singh, P. Maret, Core techniques of question answering systems over knowledge bases: a survey, Knowl. Inf. Syst. 55 (2018) 529–569, https://doi.org/10.1007/s10115-017-1100-y.
[7] G. Doddington, Automatic evaluation of machine translation quality using n-gram co-occurrence statistics, in: Proceedings of the Second International Conference on Human Language Technology Research, 2002, pp. 138–145.
[8] A. Eriguchi, K. Hashimoto, Y. Tsuruoka, Incorporating source-side phrase structures into neural machine translation, Comput. Linguist. 45 (2019) 267–292, https://doi.org/10.1162/coli_a_00348.
[9] H. Fang, F. Wu, Z. Zhao, X. Duan, Y. Zhuang, M. Ester, Community-based question answering via heterogeneous social network learning, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI, 2016, pp. 122–128
[10] A. Figueroa, Automatically generating effective search queries directly from community question-answering questions for finding related questions, Expert Syst. Appl. 77 (2017) 11–19, https://doi.org/10.1016/j.eswa.2017.01.041.
[11] A. Figueroa, Male or female: what traits characterize questions prompted by each gender in community question answering?, Expert Syst. Appl. 90 (2017) 405–413, https://doi.org/10.1016/j.eswa.2017.08.037.
[12] Z. Hao, B. Wu, W. Wen, R. Cai, A subgraph-representation-based method for answering complex questions over knowledge bases, Neural Netw. 119 (2019) 57–65, https://doi.org/10.1016/j.neunet.2019.07.014.
[13] S.A. Hasan, Y. Ling, J. Liu, O. Farri, Using neural embeddings for diagnostic inferencing in clinical question answering, in: Proceedings of The Twenty-Fourth Text REtrieval Conference, 2015
[14] K. Hashimoto, Y. Tsuruoka, Adaptive joint learning of compositional and non-compositional phrase embeddings, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL, 2016, pp. 205–215, doi:10.18653/v1/p16-1020
[15] O. Hassanzadeh, D. Bhattacharjya, M. Feblowitz, K. Srinivas, M. Perrone, S. Sohrabi, M. Katz, Answering binary causal questions through large-scale text mining: an evaluation using cause-effect pairs from human experts, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI, 2019, pp. 5003–5009, doi:10.24963/ijcai.2019/69
[16] S. Hu, L. Zou, J.X. Yu, H. Wang, D. Zhao, Answering natural language questions by subgraph matching over knowledge graphs, IEEE Trans. Knowl. Data Eng. 30 (2018) 824–837, https://doi.org/10.1109/TKDE.2017.2766634.
[17] Z. Jia, A. Abujabal, R.S. Roy, J. Strötgen, G. Weikum, TEQUILA: temporal question answering over knowledge bases, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, ACM, 2018, pp. 1807–1810. doi:10.1145/3269206.3269247
[18] D. Kim, D. Seo, S. Cho, P. Kang, Multi-co-training for document classification using various document representations: TF-IDF, LDA, and Doc2Vec, Inf. Sci. 477 (2019) 15–29, https://doi.org/10.1016/j.ins.2018.10.006.
[19] L.T. Le, C. Shah, Retrieving people: identifying potential answerers in community question-answering, J. Assoc. Inf. Sci. Technol. 69 (2018) 1246–1258, https://doi.org/10.1002/asi.24042.
[20] B. Li, B. Wang, R. Zhou, X. Yang, C. Liu, CITPM: a cluster-based iterative topical phrase mining framework, in: International Conference on Database Systems for Advanced Applications, Springer, 2016, pp. 197–213. doi:10.1007/978-3-319-32025-0_13
[21] B. Li, X. Yang, R. Zhou, B. Wang, C. Liu, Y. Zhang, An efficient method for high quality and cohesive topical phrase mining, IEEE Trans. Knowl. Data Eng. 31 (2019) 120–137, https://doi.org/10.1109/TKDE.2018.2823758.
[22] Y. Li, Q. Pan, S. Wang, T. Yang, E. Cambria, A generative model for category text generation, Inf. Sci. 450 (2018) 301–315, https://doi.org/10.1016/j.ins.2018.03.050.
[23] C.-Y. Lin, ROUGE: a package for automatic evaluation of summaries, in: Proceedings of the Workshop on Text Summarization Branches Out, ACL, 2004, pp. 74–81.
[24] C. Liu, S. He, K. Liu, J. Zhao, Curriculum learning for natural answer generation, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI, 2018, pp. 4223–4229. doi:10.24963/ijcai.2018/587
[25] J. Liu, J. Shang, C. Wang, X. Ren, J. Han, Mining quality phrases from massive text corpora, in: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, ACM, 2015, pp. 1729–1744. doi:10.1145/2723372.2751523
[26] X. Lu, S. Pramanik, R.S. Roy, A. Abujabal, Y. Wang, G. Weikum, Answering complex questions by joining multi-document evidence with quasi knowledge graphs, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2019, pp. 105–114. doi:10.1145/3331184.3331252
[27] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Annual Conference on Neural Information Processing Systems, MIT Press, 2013, pp. 3111–3119
[28] M. Neshati, Z. Fallahnejad, H. Beigy, On dynamicity of expert finding in community question answering, Inf. Process. Manag. 53 (2017) 1026–1042, https://doi.org/10.1016/j.ipm.2017.04.002.
[29] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: a method for automatic evaluation of machine translation, in: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, ACL, 2002, pp. 311–318. doi:10.3115/1073083.1073135
[30] S.K. Ramakrishnan, A. Pal, G. Sharma, A. Mittal, An empirical evaluation of visual question answering for novel objects, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2017, pp. 7312–7321. doi:10.1109/CVPR.2017.773
[31] M.T. Ribeiro, C. Guestrin, S. Singh, Are red roses red? Evaluating consistency of question-answering models, in: Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL, 2019, pp. 6174–6184. doi:10.18653/v1/p19-1621
[32] A.A. Shah, S.D. Ravana, S. Hamid, M.A. Ismail, Accuracy evaluation of methods and techniques in Web-based question answering systems: a survey, Knowl. Inf. Syst. 58 (2019) 611–650, https://doi.org/10.1007/s10115-018-1203-0.
[33] S. Shin, X. Jin, J. Jung, K.-H. Lee, Predicate constraints based question answering over knowledge graph, Inf. Process. Manag. 56 (2019) 445–462, https://doi.org/10.1016/j.ipm.2018.12.003.
[34] D. Sorokin, I. Gurevych, Interactive instance-based evaluation of knowledge base question answering, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, ACL, 2018, pp. 114–119. doi:10.18653/v1/d18-2020
[35] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, in: Advances in Neural Information Processing Systems, MIT Press, 2014, pp. 3104–3112.
[36] A. Sydorova, N. Pörner, B. Roth, Interpretable question answering on knowledge bases and text, in: Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL, 2019, pp. 4943–4951, doi:10.18653/v1/p19-1488
[37] D. Wang, E. Nyberg, CMU OAQA at TREC 2017 LiveQA: a neural dual entailment approach for question paraphrase identification, in: Proceedings of the Twenty-Sixth Text REtrieval Conference, 2017.
[38] K. Wang, X. Wan, Automatic generation of sentimental texts via mixture adversarial networks, Artif. Intell. 275 (2019) 540–558, https://doi.org/10.1016/j.artint.2019.07.003.
[39] R. Wang, M. Wang, J. Liu, W. Chen, M. Cochez, S. Decker, Leveraging knowledge graph embeddings for natural language question answering, in: Database Systems for Advanced Applications – 24th International Conference, Springer, 2019, pp. 659–675. doi:10.1007/978-3-030-18576-3_39.

[40] S. Wang, Z. Wei, Z. Fan, Y. Liu, X. Huang, A multi-agent communication framework for question-worthy phrase extraction and question generation, in: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI, 2019, pp. 7168–7175. doi:10.1609/aaai.v33i01.33017168
[41] J. Wen, H. Tu, X. Cheng, R. Xie, W. Yin, Joint modeling of users, questions and answers for answer selection in CQA, Expert Syst. Appl. 118 (2019) 563–572, https://doi.org/10.1016/j.eswa.2018.10.038.
[42] W. Wu, X. Sun, H. Wang, Question condensing networks for answer selection in community question answering, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL, 2018, pp. 1746–1755. doi:10.18653/v1/P18-1162
[43] Y. Wu, S. Zhao, W. Li, Phrase2Vec: phrase embedding based on parsing, Inf. Sci. 517 (2020) 100–127, https://doi.org/10.1016/j.ins.2019.12.031.
[44] Z. Yan, J. Zhou, Optimal answerer ranking for new questions in community question answering, Inf. Process. Manag. 51 (2015) 163–178, https://doi.org/10.1016/j.ipm.2014.07.009.
[45] M. Yang, W. Tu, Q. Qu, W. Zhou, Q. Liu, J. Zhu, Advanced community question answering by leveraging external knowledge and multi-task learning, Knowl. Based Syst. 171 (2019) 106–119, https://doi.org/10.1016/j.knosys.2019.02.006.
[46] S. Yang, L. Zou, Z. Wang, J. Yan, J.-R. Wen, Efficiently answering technical questions – a knowledge graph approach, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI, 2017, pp. 3111–3118
[47] N. Yu, M. Huang, Y. Shi, X. Zhu, Product review summarization by exploiting phrase properties, in: 26th International Conference on Computational Linguistics, ACL, 2016, pp. 1113–1124
[48] W. Zhang, Z. Ming, Y. Zhang, T. Liu, T.-S. Chua, Capturing the semantics of key phrases using multiple languages for question retrieval, IEEE Trans. Knowl. Data Eng. 28 (2016) 888–900, https://doi.org/10.1109/TKDE.2015.2502944.
[49] W. Zheng, H. Cheng, J.X. Yu, L. Zou, K. Zhao, Interactive natural language question answering over knowledge graphs, Inf. Sci. 481 (2019) 141–159, https://doi.org/10.1016/j.ins.2018.12.032.
[50] Y. Zhu, J. Wan, Z. Zhou, L. Chen, L. Qiu, W. Zhang, X. Jiang, Y. Yu, Triple-to-Text: Converting RDF Triples into High-Quality Natural Languages via Optimizing an Inverse KL Divergence, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2019, pp. 455–464. doi:10.1145/3331184.3331232