Guo SK, Wang SW, Li H *et al.* Multi-feature fusion based structural deep neural network for predicting answer time on Stack Overflow. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 38(3): 582-599 May 2023. DOI: 10.1007/s11390-023-1438-4

Multi-Feature Fusion Based Structural Deep Neural Network for Predicting Answer Time on Stack Overflow

Shi-Kai Guo[1] (郭世凯), *Member, CCF*, Si-Wen Wang[1, 2] (王思文), Hui Li[1, *] (李　辉), *Member, CCF*
Yu-Long Fan[1] (范玉龙), Ya-Qing Liu[1] (刘亚清), and Bin Zhang[3] (张　斌)

[1] *Information Science and Technology College, Dalian Maritime University, Dalian 116026, China*

[2] *Navigation College, Dalian Maritime University, Dalian 116026, China*

[3] *Computer Science and Technology College, Shandong Technology and Business University, Yantai 264005, China*

E-mail: shikai.guo@dlmu.edu.cn; wsw@dlmu.edu.cn; li_hui@dlmu.edu.cn; sumihui@dlmu.edu.cn
liuyaqing@dlmu.edu.cn; zhangb@sdtbu.edu.cn

Received March 15, 2021; accepted February 24, 2023.

Abstract Stack Overflow provides a platform for developers to seek suitable solutions by asking questions and receiving answers on various topics. However, many questions are usually not answered quickly enough. Since the questioners are eager to know the specific time interval at which a question can be answered, it becomes an important task for Stack Overflow to feedback the answer time to the question. To address this issue, we propose a model for predicting the answer time of questions, named Predicting Answer Time (i.e., PAT model), which consists of two parts: a feature acquisition and fusion model, and a deep neural network model. The framework uses a variety of features mined from questions in Stack Overflow, including the question description, question title, question tags, the creation time of the question, and other temporal features. These features are fused and fed into the deep neural network to predict the answer time of the question. As a case study, post data from Stack Overflow are used to assess the model. We use traditional regression algorithms as the baselines, such as Linear Regression, *K*-Nearest Neighbors Regression, Support Vector Regression, Multilayer Perceptron Regression, and Random Forest Regression. Experimental results show that the PAT model can predict the answer time of questions more accurately than traditional regression algorithms, and shorten the error of the predicted answer time by nearly 10 hours.

Keywords answer time, structural deep neural network, Stack Overflow, feature acquisition, feature fusion

1 Introduction

During the process of software development, developers often spend a large amount of time on searching for assistance in various ways, such as handbook querying, forum discussions, and online questions. Nowadays, the way of asking specific questions and getting targeted answers from online experts is generally considered to be the most effective way to find appropriate answers to technical questions[1]. Therefore, many online forums or platforms are emerged to provide this service.

Stack Overflow is one of the most famous and reliable online Community Question and Answer (CQA) exchanging knowledge and solving problems for developers[2, 3]. Some community users can post programming questions and technical questions, while others can easily find the corresponding posts according to their interests and demands. Furthermore, Stack Overflow is one of the largest CQAs for computer programming[4, 5]. All the records about questions are open source and available. These records are orga-

Regular Paper

This work was supported by the National Natural Science Foundation of China under Grant Nos. 61902050, 61602077 and 61672122, the China Postdoctoral Science Foundation under Grant No. 2020M670736, the Fundamental Research Funds for the Central Universities of China under Grant Nos. 3132019355 and 2020cxxmss14, and the High Education Science and Technology Planning Program of Shandong Provincial Education Department of China under Grant Nos. J18KA340 and J18KA385.

*Corresponding Author

nized as datasets, which contain "Posts", "Users", "Votes", "Comments", "PostHistory", "PostLinks", etc.①. Among them, the "Posts" datasets contain the most valuable information, including all the questions, answers, and their interactions. There are more than 15 million posts written by 8.5 million users with a total size of 15 GB in the Stack Overflow, which contains more than 500 programming languages[6]. The rich feature set of Stack Overflow has attracted the attention of many professional software developers, in which users can edit questions, answer questions, vote on the quality of answers, and comment on individual questions and answers. Besides, a growing number of users are sharing their programming algorithms, library technologies, and problems with programming through Stack Overflow[7]. The open datasets also can be used in a variety of ways to perform statistical analysis on the posted questions, evaluate the quality of questions and answers, and help the developer community to obtain better technical support[8].

When a developer posts a question, he (or she) is often eager to receive an answer as soon as possible. But many questions are usually not answered quickly enough because of various reasons. Therefore, it will release the questioners' anxiety through providing the specific time interval at which the question will be answered, which is named as "answer time" in this paper. However, the answer time of a question actually depends on many factors, including how the developer describes the question, whether the question is described in detail, how many tags are used to categorize questions, whether the question is recommended to the related developers[9, 10], how many developers are online and interested in the question, etc.[11]. One obvious drawback of Stack Overflow is that it does not have a clear expected answer time for the questions. As a result, the developers who posted questions do not know the specific answer time, and thus they may have to wait for a long time to get answers[12]. It is reported that 92% of the questions were answered on Stack Overflow, but the average answer time is about 24 days[13]. In other words, if someone posts a question, he (or she) may have to wait for about 24 days to receive an answer in average, because he (or she) does not know the specific time when the question will be answered, which causes the question not to be solved in a timely manner. Therefore, predicting the answer time of a question on Stack Overflow has become a challenging task.

In recent years, some machine learning techniques have been used for addressing this challenge. Previous work formulated the problem in different ways, and reported different accuracy measures in predicting the answer time. For example, Bhat *et al.*[12] formulated it as a classification problem of predicting 1) whether a given question will be answered in less than 16 minutes or not, and 2) whether a given question will be answered in less than or equal to one hour, or greater than or equal to one day. They studied multiple factors of questions on Stack Overflow and reported that popularity (i.e., the usage frequency of the tag) and the number of subscribers (i.e., how many users can answer the question containing the tag) played a key role in predicting the answer time of questions, which also proves the importance of the tags in predicting answer time. On this basis, Wu *et al.*[1] labeled the time into four different answer time groups, which are within one hour, one to four hours, four to 12 hours, and 12 hours or more. Then the datasets are used for training classification models (including Support Vector Machine, Random Forest Classifier, Logistic Regression, Decision Tree, Neural Network, Gaussian Naive Bayes, and K-Nearest Neighbors) and evaluating the classification accuracy of each algorithm. However, the researchers[12] all formulated the problem as a classification problem. They focused on whether the question will be answered within a specific time frame, rather than predicting the specific time interval when the question receives an acceptable answer.

In this work, we conduct a comprehensive study of the features of the question. We define a new problem formulation, which re-formulates the answer time prediction as a regression problem. Then we propose a new regression model named Predicting Answer Time (PAT) model. Specifically, we extract multiple text features and time features from the question, including the question description (*Body*), question title (*Title*), question tags (*Tags*), the creation time of the question (*Time-rate*), and question week feature (*Week*). Consequently, we use the Doc2vec model to convert text features into vectors. Then the normalization method is used to calculate the value of the time feature. We fuse them to get the new feature vector. Finally, we feed the new feature vector into the fully-connected neural network to predict the an-

①Stack Overflow dataset. https://archive.org/details/stackexchange, June 2021.

swer time of the question. We evaluate the performance of the PAT model by the relative error of the answer time. Finally, we assess the validity of the PAT model through experimental studies based on datasets of Stack Overflow.

The main contributions of this work are as follows.

1) Considering the practical implementation of Stack Overflow, we reconstruct the problem as a regression problem to accurately formulate the research question.

2) We propose a multi-feature fusion model based on a deep neural network, (i.e., the PAT model), to predict the answer time of questions on Stack Overflow.

3) We analyze and design features that may affect the answer time of questions. As a result, we identify a new feature set for predicting the answer time of questions. We experimentally prove that the PAT model outperforms Linear Regression, K-Nearest Neighbors Regression, Support Vector Regression, Multilayer Perceptron (MLP) Regression, and Random Forest Regression, in terms of the relative error of the answer time on Stack Overflow.

The remainder of this paper is organized as follows. Related work and motivation are discussed in Section 2. The design of the PAT model is described in Section 3. The experimental design and results are presented in Sections 4 and 5, respectively. The threats to validity are discussed in Section 6. The conclusions are given in Section 7.

2 Related Work and Motivation

2.1 Related Work

Prediction on the answer time for CQAs has attracted more and more attentions of scientific researchers from software engineering to artificial intelligence. Bhat *et al.*[12] studied multiple factors of questions on Stack Overflow and reported that popularity (the usage frequency of the tag) and the number of subscribers (how many users can answer the question containing the tag) play the key role in predicting the answer time of questions. Treude *et al.*[14] studied the questions on Stack Overflow, and reported that 72.30% of the questions have two to four tags. The tag can then reveal which topic the question belongs to, and developers can encode the questions with tags to allow navigation to their questions. On this basis, Goderie *et al.*[15] reported that the answer time of the questions could be predicted based on the features of question tags. They derived ideas from the model of Bhat *et al.* and presented three tag-related features associated with the answer time, namely the active user ratio of each tag (ASR), the responsive subscribers ratio for each tag (RSR), and the popularity level for each tag (PR). Then they classified the questions based on the tag's metrics and used the supervised learning algorithm K-nearest neighbors to calculate the expected answer time of questions.

As we know, the answer time may depend on whether the question is easy to be answered. Therefore, it is worthy to investigate which kinds of questions are easy or difficult to be answered. Teevan *et al.*[16] discussed the number of replied questions, the quality of the answers, and the speed of response on the Facebook. They studied the punctuation of the question, the number of clauses, and the scope of the questions. It is reported that a question with a single clause is more likely to receive a faster response, namely, the description of the question has an impact on the predicted answer time[16]. Arguello *et al.*[17] investigated the factors affecting the communication between individuals and online communities in various aspects, such as the ability and scale of group identification, the status of new users and their contributions, the rhetorical strategies for publishing content, the coherence of topics, and the semantic complexity. It is revealed that questions with unclear semantics, questions with complex topics, and questions with novice posters are not easily to be replied. Conversely, questions with simple language content or their posters with a greater degree of contribution are more likely to be replied.

On this basis, studies on answer time prediction for questions have been emerged. Dror *et al.*[18] presented a prediction method via multiple features to predict whether a question will be answered and how many answers the question will receive. The purpose of this prediction is to help the user re-express his/her question (if it is unlikely to be answered) and reduce the frustration of waiting for an answer. However, it does not consider when the question would be answered. Arunapuram *et al.*[19] studied the answer time based on more than two million question-and-answer threads, and discussed the distribution and relevance prediction of answer time for the questions on Stack Overflow. They produced the characteristics associated with the answer time through analyzing the length of the question title, keywords, punctuation, time of day, etc., and then employed a weighted average algorithm to predict the distribution range of the answer time. However, they only considered the impact of a single feature on the answer time.

Subsequently, Bhat *et al.*[12] formulated the answer time prediction problem as two separate classification tasks: 1) whether a given question will be answered in less than 16 minutes or not, and 2) whether a given question will be answered in less than or equal to one hour, or greater than or equal to one day. They reported that the tag features have an influence on predicting the answer time of the question. Wu *et al.*[1] conducted a comprehensive study on this basis, and labeled the time into four different answer time groups, which are within one hour, one to four hours, four to 12 hours, and 12 hours or more. They used a variety of classification algorithms for training and evaluated the performance of the algorithms through classification accuracy. Although many factors affecting the answer time of questions have been investigated in the previous studies, the features of the questions they considered are still not comprehensive. Thus we propose a new feature set to predict the answer time of questions, and take the prediction of the answer time as a regression task. The relative error of the answer time predicted by the model can be used for more intuitively understanding the answer time of questions.

2.2 Motivation

It is valuable to understand the answer time of a question on CQAs, because users are often eager to know the answer to the question. Most CQAs are not able to guarantee that users can receive satisfying answers to their questions on time, resulting in disappointment and frustration of users. Bhat *et al.*[20] reported that the answer time of about 37.7% questions on Stack Overflow is over one hour. Even worse, the answer time of 11.81% questions is longer than one day. It indicates that the answer time of questions is with a larger range of fluctuation. The above issues make it difficult for questioners to decide whether to switch focus to other parts of software development or to keep waiting for answers. This dilemma has brought great inconvenience for questioners to manage their time. Actually, the mechanism of providing users with an accurate time of answering their questions can not only help them manage their time reasonably, but also prompt them to rephrase their questions for obtaining answers faster.

Therefore, it is important to figure out the factors affecting the answer time of questions on CQAs, and then we can shorten the answer time of questions by adjusting the factors. These factors include changing the label of the question, shortening the content of the question, and posting a question at a specific time of day[21]. If CQAs provide the expected answer time of a question, it can help users better schedule their work hours and increase their productivity, and CQAs will also become more popular[22, 23]. At present, the studies on predicting the answer time of questions for CQAs, such as Stack Overflow, are still rare. Previous studies take the answer time prediction as a classification problem, in which the answer time is divided into several time intervals. The performance of the model is usually determined by the accuracy of the classification. These studies only predict whether the question will be answered within a specified time interval. However, users more expect to know the specific time when the question will be answered. Thus, the previous studies do not fundamentally solve the problem of predicting the answer time of questions for users.

In this work, the problem is converted to a regression task, in which the relative error of the answer time is used to measure the performance of the proposed model. Hence, users can also understand the answer time of questions more intuitively. That is the motivation of carrying out this study.

3 Proposed Framework

3.1 Problem Statement

Whether the answer to a question can be accepted by the users depends on the quality of the question and the answer. The accepted answers are chosen and studied in this work, because we can only obtain the necessary time stamps from them. Thus the answer time is defined as the time span between the point when a question is posted and the point when the question has an acceptable answer. Specifically, q_i denotes the i-th question, a_i denotes the acceptable answer for question q_i, and the answer time is defined as $T_i = t(a_i) - t(q_i)$, where $t(a_i)$ is the creation time of the acceptable answer and $t(q_i)$ is the creation time of the question. Therefore, we create a set of features $F = \{F_1, F_2, ..., F_n\}$ to predict the variable y_i.

3.2 Overview

In this subsection, we present the multi-feature fusion network based on the deep neural network to predict the answer time of questions, named Predicting Answer Time (PAT) model. It consists of two parts: 1) a feature acquisition and fusion model, and 2) a deep neural network model. In the feature acquisition and fusion model, it includes the extraction of multi-features and the fusion of multi-features. The entire framework is shown in Fig.1. We extract a variety of features from questions. These features are divided into two types, namely text features and time features. We extract the body, title, and tags of questions as text features, and the creation time and week features of the questions as time features. In the following, we use the Doc2vec model[24] to convert the text features of questions into vectors. Then the normalization method is used to convert each time feature of the question into a specific value. We expand the dimension to make it be a vector. Then we use the feature fusion to process these two types of vectors to obtain a new feature vector. In the deep neural network model, we feed the obtained new feature vector into the three-layer fully-connected neural network model to predict the answer time of questions.

3.3 Feature Acquisition and Fusion Model

3.3.1 *Multi-Feature Extraction*

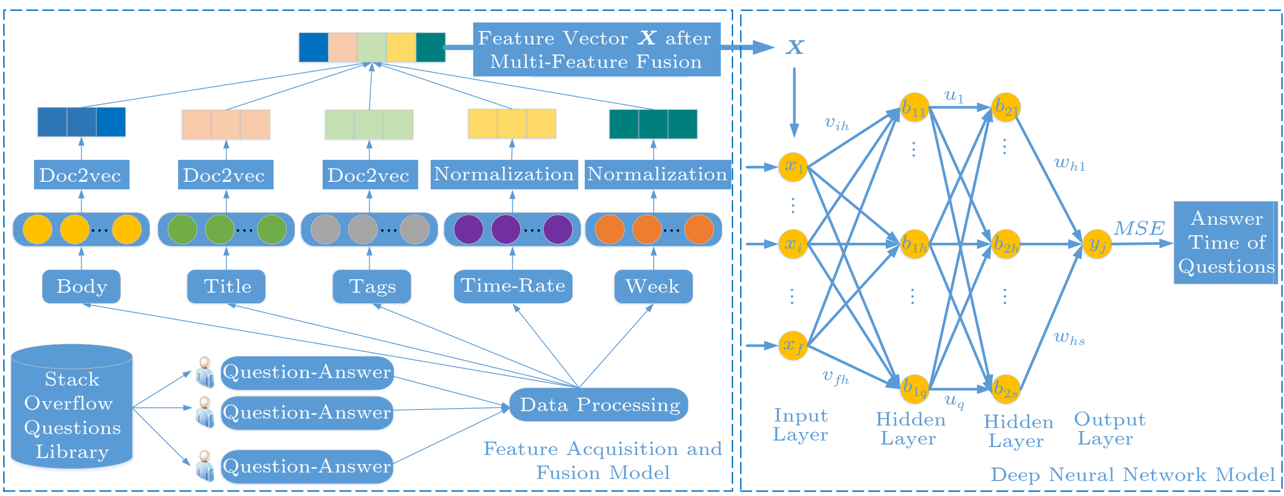
We conduct a comprehensive study for the questions on Stack Overflow and present a new feature set to predict the answer time of questions. Specifically, we extract text features and time features of questions as shown in Fig.1(a), where the text features include the body, title and tags of questions, and the time features include the creation time and week feature of questions. The mentioned features are listed below.

1) *Body Feature* (*Body*). It refers to the description of the question. The body of a question expands the summary provided by its title. The text should be well-written, engaging, and informative, and contains properly formatted sentences[25].

2) *Title Feature* (*Title*). The title is equivalent to a summary of the question. Since many Stack Overflow members may create content of a question that mismatches the title, we also need to consider this feature.

3) *Tags Feature* (*Tags*). Tags reflect related topics of the questions, and some tags may appear in the same questions[26]. Tags are the words or phrases that can highlight the main topics of the questions. They can also be used to help users rapidly identify interesting or self-related questions[26]. The posters have to specify the tag when creating the question on Stack Overflow. Specifically, each question must be labeled with one to five tags. With the help of tags, all the questions can be categorized clearly.

The purpose of using subject tags on Stack Overflow is to target questions to specific users. For example, a developer could label a tag "Java" when he or she posts a question with the topic of Java, so that developers who are interested in Java or usually answer Java-related questions can view it more quickly.



(a) (b)

Fig.1. Overall framework of the PAT model. (a) Feature acquisition and fusion model. (b) Deep neural network model. u_q denotes the input weight of the q-th node of the first hidden layer to the second hidden layer.

Therefore, it is possible to make the questions to be answered faster through adjusting the factors which directly impact the answer time of the questions. For instance, Arguello *et al.*[17] suggested that the answer time can be shortened through cross posting messages. Besides, Arunapuram *et al.*[19] reported that using more specific tags (for example, using visual-studio-2010/2008 and ruby-on rails-3 instead of visual-studio and ruby-on-rails, respectively) can greatly reduce the answer time.

4) *Creation Time of the Question* (*Time-rate*). This is the time stamp for a question to be posted. We use the creation time of the question to determine and predict how long it will take for the question to get an acceptable answer. The creation time of questions may be in the morning, noon, or evening. Avrahami *et al.*[27] reported that developers answer questions more actively in the morning and at noon, compared with their performance in the afternoon. Therefore, the creation time of the question is a feature that needs to be considered for predicting the answer time of the questions.

We extract the number of hours, minutes, and seconds of the question creation time through the built-in time function of Python. The time-rate feature $time_{\rm feature}$ can be expressed by

$$time_{\rm feature} = 3\,600 \times hours + 60 \times minutes + seconds,$$

where *hours*, *minutes* and *seconds* denote the number of hours, minutes, and seconds of the question creation time, respectively.

5) *Week Feature* (*Week*). This represents in which day of the week is the question posted. It is known that the number of created questions could be different for each day of a week. For instance, the number of new questions may be relatively small on Monday, and the answer time could be relatively long, because many Stack Overflow users are busy in working. Conversely, the number of new questions may be also small on Sunday, but the answer time may be shorter, because many Stack Overflow users could rest at home. The week feature of a question could be extracted from the creation time of the question, through the built-in time function of Python. The values are enumerated by $weekday \in \{1, 2, 3, 4, 5, 6, 0\}$, in which the elements denote Monday to Sunday, respectively.

In summary, there are two types of question features. The first type is the textual features, including the *Body* feature, the *Title* feature and the *Tags* feature. The second type is the time features, including the *Time-rate* feature and the *Week* feature. We convert the text features of the question into vectors through the Doc2vec model, and use the normalization method to convert the time features into vectors. Then we use the feature fusion method to fuse them into a new feature vector.

3.3.2 Multi-Feature Fusion

First, for the text features of questions, we use the Doc2vec model to convert the processed text sequence of questions into a high-dimensional vector, that is, the *Body* feature, the *Title* feature, and the *Tags* feature. Each paragraph is represented by a unique vector, which is named a paragraph vector. Each word is also represented by a unique vector, named word vector. We concatenate the paragraph vectors and word vectors, and then average the integrated vectors to get a new vector, which is used to predict the next word in the paragraph. This paragraph vector can also be considered as a word. It acts as a memory unit of the context or topic of this paragraph. Thus this method is generally named as Distributed Memory Model of Paragraph Vectors (PV-DW)[24]. The PV-DW method slides and samples fixed-length words from one paragraph at a time, and takes one of them as the predicted word and the other words as the input word. Here we set the embedding vector dimensions of the *Body* feature, *Title* feature and *Tags* feature to 50, 20 and 5, respectively.

The process of summarizing the Doc2vec model consists of two main steps. First, in the training stage, the word vector, the parameters of the softmax function, and the paragraph vector are obtained from the training data. Each paragraph has a unique paragraph vector $\boldsymbol{d}$, and each word has a unique word vector $\boldsymbol{w}$. More formally, given a sequence of training words $w_1, w_2, ..., w_T$ and a sequence of training paragraphs $d_1, d_2, ..., d_T$, the objective of the Doc2vec model is to maximize the average logarithmic probability of the sentence vector and the word vector by softmax, that is, if

$$\frac{1}{T}\sum_{t=k}^{T-k} \log p(d_t|d_{t-k}, ..., d_{t+k}, w_t|w_{t-k}, ..., w_{t+k}),$$

has a maximum value, we have

$$p(d_t|d_{t-k}, ..., d_{t+k}, w_t|w_{t-k}, ..., w_{t+k}) = \frac{e^{y_{w_t},\, y_{d_t}}}{\sum_i e^{y_i}},$$

where y_i is the output value of word i before normalization. The output of the Doc2vec model is

$$y = b + U\boldsymbol{h}(d_{t-k}, ..., d_{t+k}, w_{t-k}, ..., w_{t+k}; \boldsymbol{d}, \boldsymbol{w}),$$

where U and b are softmax parameters, and $\boldsymbol{h}$ is constructed by a concatenation or average of word vectors extracted from $\boldsymbol{w}$ and paragraph vectors extracted from $\boldsymbol{d}$. The paragraph vector $\boldsymbol{d}$ is also trained while training the word vector $\boldsymbol{w}$. After the training is finished, the vectorized representation of the paragraph is also included.

Second, in the inference stage, the new paragraph vector is obtained by the gradient drop method, and the values of $\boldsymbol{w}$, U, and $\boldsymbol{h}$ remain constant.

Finally, we obtain the feature vectors of *Body*, *Title*, and *Tags* through the Doc2vec model.

For the time features of the question, we use the normalization method to get the feature values. We obtain the eigenvalues of the *Time-rate* and *Week* features through the following formula; thus we have

$$time_{\text{time-rate}} = time_{\text{feature}}/(3\,600 \times 24),$$

and

$$time_{\text{week}} = weekday/7,$$

where $time_{\text{time-rate}}$ and $time_{\text{week}}$ denote the *Time-rate* eigenvalue and *Week* eigenvalue after normalization, respectively. We convert them into vectors by expanding the dimension.

We use the feature fusion algorithm to fuse the textual feature vector and the time feature vector of the question as follows. The frequently-used feature fusion methods include concatenation, element-wise addition and element-wise multiplication[28]. Since concatenation can combine feature vectors of different dimensions, we use concatenation for feature fusion in this work. Let $\boldsymbol{tf}_1$ be the *Body* feature vector, $\boldsymbol{tf}_2$ be the *Title* feature vector, $\boldsymbol{tf}_3$ be the *Tags* feature vector, $\boldsymbol{tf}_4$ be the *Time-rate* feature vector, and $\boldsymbol{tf}_5$ be the *Week* feature vector. Then we could express high-level feature vector $\boldsymbol{X}$ after combination by (1),

$$\boldsymbol{X} = \boldsymbol{tf}_1 \circ \boldsymbol{tf}_2 \circ \boldsymbol{tf}_3 \circ \boldsymbol{tf}_4 \circ \boldsymbol{tf}_5 = (\boldsymbol{tf}_1, \boldsymbol{tf}_2, \boldsymbol{tf}_3, \boldsymbol{tf}_4, \boldsymbol{tf}_5), \quad (1)$$

where $\circ$ denotes the concatenation operator. Then the new feature vector can be fed into the neural network model to predict the answer time of questions.

3.4 Deep Neural Network Model

Neural networks simulate many interconnected processing units that resemble abstract versions of neurons[29-32]. The processing units are usually distributed in different layers. Typically, a neural network includes three parts. The first part is an input layer that contains the units representing the input fields; the second part includes one or more hidden layers; the third part is an output layer which contains a unit or units representing the target fields. The units in a neural network are connected with varying connection strengths (or weights). The input data is sent to the first layer, and then the corresponding values are propagated from each neuron to every neuron in the next layer. Finally, the result will be delivered from the output layer.

In this work, we use a three-layer fully connected neural network model to predict the answer time of a question. The input of the fully-connected neural network model is the new feature vector $\boldsymbol{X}$ obtained in Subsection 3.3. The structure of the neural network includes an input layer, two hidden layers and an output layer, where the nodes in each layer receive input from the previous layer, and the output of the nodes in the previous layer is the input of the next layer. The activation function of the first three layers is ReLU. The inputs to each node are combined using a weighted linear combination. Finally, the answer time of the question is obtained through the sigmoid function.

As shown in Fig.1(b), the input data $\boldsymbol{X} = \{x_1, x_2, ..., x_i, ..., x_f\}$ is given, where the number of neurons is f. The neurons $b_{11}, b_{12}, ..., b_{1h} ..., b_{1q}$ are in the first hidden layer, where the number of neurons is q. Then $v_{1h}, ..., v_{fh}$ is the input weight of the corresponding nodes of the first hidden layer, with a bias value of b'. The neurons $b_{21}, b_{22}, ..., b_{2h}, ..., b_{2s}$ are in the second hidden layer, where the number of neurons is s. The values $u_1, ..., u_q$ are the input weights of the corresponding nodes of the second hidden layer, with a bias value b''. Additionally, y_j is the true value of the answer time of the question, and $w_{h1}, ..., w_{hs}$ are the input weights of the output nodes, with a bias value b'''.

The computation process of the neural network is described as follows. α_h is the input value of the h-th neuron in the first hidden layer, and then we have

$$\alpha_h = \sum_{i=1}^{f} v_{ih} x_i + b'.$$

The output value α_{oh} of the h-th neuron in the first hidden layer can be calculated by $\alpha_{oh} = \varphi(\alpha_h)$, where $\varphi(x)$ is the sigmoid activation function,

$$\varphi(x)=\frac{1}{1+\mathrm{e}^{-x}}.$$

Its derivative function is

$$\varphi(x)'=\varphi(x)(1-\varphi(x)).$$

${\alpha_h}'$ is the input value of the h-th neuron in the second hidden layer, and it is expressed as

$${\alpha_h}'=\sum_{h=1}^{q}u_h\alpha_{oh}+b''.$$

${\alpha_{oh}}'$ is the output value of the h-th neuron in the second hidden layer, and thus we have

$${\alpha_{oh}}'=\varphi({\alpha_h}').$$

β_j is the input value of the output neuron, and thus

$$\beta_j=\sum_{j=1}^{s}w_{hj}{\alpha_{oh}}'+b'''.$$

Therefore, the predicted value $\hat{y}_j$ of the neural network is

$$\hat{y}_j=\varphi(\beta_j).$$

The loss of the neural network E is defined as

$$E=\frac{1}{2}(\hat{y}_j-y_j)^2.$$

Based on the loss function, the updated formula of weights is deduced. After the training of the neural network model, the weights of the neural network update are shown in (2), (3) and (4).

$$\bar{w}_{hj}=w_{hj}-\eta\frac{\partial E}{\partial w_{hj}}=w_{hj}-\eta\left(\frac{\partial E}{\partial\hat{y}_j}\times\frac{\partial\hat{y}_j}{\partial\beta_j}\times\frac{\partial\beta_j}{\partial w_{hj}}\right), \quad (2)$$

$$\begin{aligned}\bar{u}_h&=u_h-\eta\frac{\partial E}{\partial u_h}\\&=u_h-\eta\left(\frac{\partial E}{\partial\hat{y}_j}\times\frac{\partial\hat{y}_j}{\partial\beta_j}\times\frac{\partial\beta_j}{\partial{\alpha_{oh}}'}\times\frac{\partial{\alpha_{oh}}'}{\partial{\alpha_h}'}\times\frac{\partial{\alpha_h}'}{\partial u_h}\right), \quad (3)\end{aligned}$$

$$\begin{aligned}\bar{v}_{ih}&=v_{ih}-\eta\frac{\partial E}{\partial v_{ih}}\\&=v_{ih}-\eta\left(\frac{\partial E}{\partial\hat{y}_j}\times\frac{\partial\hat{y}_j}{\partial\beta_j}\times\frac{\partial\beta_j}{\partial{\alpha_{oh}}'}\times\right.\\&\left.\frac{\partial{\alpha_{oh}}'}{\partial{\alpha_h}'}\times\frac{\partial{\alpha_h}'}{\partial\alpha_{oh}}\times\frac{\partial\alpha_{oh}}{\partial\alpha_h}\times\frac{\partial\alpha_h}{\partial v_{ih}}\right), \quad (4)\end{aligned}$$

where $\bar{w}_{hj}$ is the connection weight between the hidden layer and the output layer after the update, $\bar{u}_h$ is the connection weight between the hidden layers after the update, $\bar{v}_{ih}$ is the connection weight of the input layer and the hidden layer after the update, η is the learning rate, and the updating method of the offset value is the same as that of the connection weight.

4 Experimental Design

4.1 Experimental Dataset

In order to extract the data from Stack Overflow, we start with the file named posts.xml from the Stack Overflow data dump, which contains all the user posts (i.e., questions and answers) on Stack Overflow[2]. The detailed information of the posts is shown in Table 1. Firstly, we select the first 100 000 questions from Stack Overflow in 2013. In order to ensure the timeliness of the data, we append 376 685 and 372 075 questions posted on January 2020 and February 2020 from the Stack Exchange website[2]. Secondly, all of the questions without AcceptedAnswerId will be removed in the further pre-processing, and the remaining questions have acceptable answers. Thirdly, we eliminate the question data of the HTML and other rich-text tags in the question description, because these tags contain some useless information that can increase the prediction error. Fourthly, we remove the question data with the answer time of more than 400 000 seconds (i.e., more than about four days) to avoid excessive time variance that could affect the experimental results. Finally, we get three datasets for experiments, which are the questions in 2013, January 2020, and February 2020, respectively. The statistics of these three datasets are listed in Table 2.

After data pre-processing, each dataset is divided into a training set and a test set at a ratio of 9:1. As a result, 29 332 questions are randomly sampled from the 2013 dataset as the training set. Similarly, we can get 57 177 and 55 619 questions from the January 2020 and February 2020 datasets for training, respectively. The datasets and the related codes for experiments can be found in Github[3].

4.2 Experimental Setup

In the process of encoding, the Doc2vec model is employed, in which the embedding vector dimension of the *Body* feature is set to 50, the embedding vector dimension of the *Title* feature is set to 20, the embedding vector dimension of the *Tags* feature is set to

[2]StackExchange API. http://data.stackexchange.com/stackoverflow, June 2021.
[3]Code and datasets. https://pan.baidu.com/s/1r9yea5jRENqJeDBfAPvOJg. Extraction code: 9gza. Oct. 2021.

Table 1. Attribute Information and Values of a Post

Name	Description
ID	ID of the post
PostTypeId	Type of post: if PostTypeId = 1, it means this is a question; if PostTypeId = 2, it means this is an answer
AcceptedAnswerId	The ID of the relevant acceptable answer post for the question post (it exists only when PostTypeId = 1)
ParentId	The ID of the related question post for the answer post (it exists only when PostTypeId = 2)
CreationDate	The creation time of the post
Score	Average score by the viewers for the post
ViewCount	Total number of views for the post
Body	Description of the post (body)
OwnerUserId	ID of the post owner
OwnerDisplayName	Username of the post owner
LastEditorUserId	ID of the person who last edited the post
LastEditorDisplayName	Username of the person who last edited the post
LastEditDate	Date when the post is last edited
LastActivityDate	Date when the status of the post is last changed
Title	Title of the post (it exists only when PostTypeId = 1)
Tags	Tags of the post (it exists only when PostTypeId = 1)
AnswerCount	Number of answers for the question post (it exists only when PostTypeId = 1)
CommentCount	Number of comments for the post
FavoriteCount	Number of people who like the post (it exists only when PostTypeId = 1)
ClosedDate	Date when the post is closed

Table 2. Statistics for the Three Datasets on Stack Overflow

Dataset	Number of Questions	Number of Answers	Number of Question-Answer Pairs After Pre-Processing
2013	100 000	675 611	32 592
January 2020	376 685	1 048 575	63 530
February 2020	372 075	846 646	61 799

5, and the embedding vector dimensions of the other time features are set to 1. We use a fully-connected network with three hidden layers, in which the number of neurons in the first layer is 100, the number of neurons in the second layer is 200, the number of neurons in the third layer is 100, and the activation function of the hidden layer is ReLU. We also use the Dropout method for randomly excluding some neurons during each training to avoid overfitting of the neural network, and it further improves the effect of the prediction phase. The Dropout parameter is set to 0.8. The activation function of the output layer is sigmoid.

In the process of training, the optimization process uses mean square error as the loss function, and the optimizer uses AdamOptimizer to adjust the model parameters during the training process dynamically[33]. Additionally, the learning rate is set to 0.01. This method could make the model achieve better convergence by dynamically adjusting the learning rate. Linear Regression[34], K-Nearest Neighbors Regression[35], Support Vector Regression[36], MLP Regression[37], and Random Forest Regression[38] are employed as the baseline algorithms. We build the system using a Python library scikit-learn for training with default parameter settings④.

4.3 Evaluation Metrics

As mentioned in Subsection 3.1, the answer time of a question is defined as the time span between the creation time of an acceptable answer and the creation time of the question. Thus we normalize the answer time interval and convert it to a value between 0 and 1, and the actual answer time after normalization is

$$Y_i = (T_i - time_{\min})/(time_{\max} - time_{\min}).$$

Since we select the questions where their answer time is within four days, the maximum time $time_{\max}$ is 400 000 seconds, and the shortest time $time_{\min}$ is set to 0 second by default.

We use Mean Square Error (MSE) as an indicator to evaluate the performance of the PAT model. Assuming the predicted value is $y = \{y_1, y_2, ..., y_n\}$ and the true value is $Y = \{Y_1, Y_2, ..., Y_n\}$, MSE is defined by (5).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - Y_i)^2. \tag{5}$$

Aiming at characterizing the error of the predicted answer time more clearly, the relative error of the answer time within four days, which is between the time predicted by the PAT model and the actual answer time, is used for measuring the performance of the PAT model. Specifically, the relative error of the answer time is defined as MSE', where $MSE' = \sqrt{MSE} \times 400\,000/3\,600$. The unit of MSE' is hour.

5 Experimental Results

In this section, the experimental results are dis-

④Python library. http://scikit-learn.org/, June 2021.

cussed in relation to the specific research questions (RQs).

5.1 RQ1: Can PAT Model Better Predict the Answer Time of Questions?

In this research question, we plan to explore whether the PAT model has improved its prediction effect on predicting the answer time of questions of Stack Overflow, compared with previous regression algorithms. Similar to the comparison experiments conducted by Burlutskiy *et al.*[11], we compare the PAT model with some traditional regression algorithms, such as Linear Regression[34], *K*-Nearest Neighbors Regression[35], Support Vector Regression[36], MLP Regression[37], and Random Forest Regression[38]. Previous studies show that these classic regression algorithms play an important role in data analysis, function fitting, and time series prediction. We record the experimental results of each regression algorithm. In contrast to the experimental results, we observe whether the PAT model is superior to the traditional algorithms in predicting the answer time of questions. For the baseline models, we use the same features as the PAT model to make predictions, and extract the *Body*, *Title*, *Tags*, *Time-rate* and *Week* features of the question. Table 3 shows the values of relative error MSE' for the answer time of the PAT model and baseline regression models for three datasets. The unit of error in the table is hour.

Table 3. Values of Relative Error MSE' for Answer Time (h) for Three Datasets

Model	Dataset		
	2013	January 2020	February 2020
Linear Regression	15.533 570	19.709 493	18.801 359
K-Nearest Neighbors Regression	16.545 609	20.354 959	19.860 232
Random Forest Regression	16.673 747	23.814 190	19.860 628
Support Vector Regression	16.539 869	19.559 889	19.323 035
MLP Regression	15.923 066	33.428 693	19.027 602
PAT model	**5.597 671**	**5.500 320**	**5.499 918**

We can see from Table 3 that the values of relative error MSE' of the PAT model are much smaller than those of traditional regression models, and thus the PAT model performs better in predicting the answer time of questions for the three datasets. The optimal performance is marked out in bold in Table 3. Besides, it can be seen from Table 3 that the gap of the prediction error for different datasets is very small. Therefore, it reveals that the prediction ability of the PAT model is stable for different datasets.

Among the baseline models, the best prediction models are Linear Regression and Support Vector Regression. For the dataset in 2013, the prediction error reaches 15.533 570 hours and 16.539 896 hours, and it is about three times of the error of the PAT model. In other words, given a question, the error of the answer time predicted by the PAT model is about 5.5 hours compared with the actual answer time of the question, while the best result of traditional regression models is around 16 hours. Therefore, the PAT model shortens the error by nearly 10 hours.

5.2 RQ2: How Does a Single Feature Extracted from a Question Affect the Prediction of Answer Time?

In this subsection, five experiments are carried out for exploring the impact of the features on predicting the answer time of questions. We aim to figure out the most important feature of the questions. In each experiment, one feature is removed, namely, only the remaining four features are used as the input. The experimental results obtained are compared with the experimental result of the PAT model which considers all the features. Through the above experimental results, we observe the impact of each feature on the performance of the PAT model and identify the most important feature for predicting the answer time of questions. Table 4 shows the values of relative error MSE' between the predicted values and the actual values of the answer time of questions after removing a feature from the question. The bold indicates the minimum error predicted by the model. The first column represents the features we use, and the second column represents the features that are not considered.

For the three datasets, it can be seen from Table 4 that the results containing all the features (i.e., *Body*, *Title*, *Tags*, *Time-rate*, *Week*) are the optimal (the error of the answer time is about six hours), while the results after removing the *Body* feature are the worst. Therefore, for the problem of predicting the answer time of questions, we need to consider as many features as possible, and each feature has a certain impact on the answer time of questions. Besides, it also reveals that the *Body* feature is the most important feature, because the *Body* feature represents the description of the question, which is the most informative one among all features. The clarity or am-

Table 4. Values of Relative Error MSE' for Answer Time (h) for PAT Model after Removing a Feature

Features of Used Questions	Feature Removed from the Questions	Dataset		
		2013	January 2020	February 2020
Body, *Title*, *Tags*, *Week*, *Time-rate*	None	**5.597 671**	**5.500 320**	**5.500 320**
Title, *Tags*, *Week*, *Time-rate*	*Body*	6.401 961	6.290 593	6.356 981
Body, *Tags*, *Week*, *Time-rate*	*Title*	5.611 655	5.508 178	5.506 810
Body, *Title*, *Week*, *Time-rate*	*Tags*	5.598 784	5.511 303	5.505 734
Body, *Title*, *Tags*, *Week*	*Time-rate*	5.631 999	5.537 991	5.523 851
Body, *Title*, *Tags*, *Time-rate*	*Week*	5.614 476	5.525 467	5.522 045

biguity of the question description will directly affect the answer time of the question. Also, it can be seen from Table 4 that the *Time-rate* feature and the *Week* feature are relatively more important than the other features. In other words, the creation time of the question and the day of the week for the posted questions are important for the answer time.

To analyze the impact of the *Week* feature on the performance of the PAT model in a more fine-grained way, we record the number of the questions for each day of a week, and the average answer time of the questions for the three datasets. Fig.2 shows the number of questions posted and the average answer time of questions in each day of the week for the three datasets, where Fig.2(a) is the number of questions posted in each day of the week, and Fig.2(b) is the average answer time of questions in each day of the week. It can be seen from Fig.2(a) that the number of questions decreases significantly on weekends, and it even reaches one-half to one-third of the peak. Thus the result shows that only a few people posted questions on the weekends. For questions in 2013 and January 2020, it can be seen from Fig.2(b) that the average answer time is the shortest during the weekend. For the questions in February 2020, there are more questions posted on weekends than in January 2020, but the average answer time of questions in February is less than that in January, indicating that the data fluctuates greatly. For the three datasets, although there are few questions on weekends, the average answer time of questions per day is not much different. It can be seen from Fig.2 that the *Week* feature can affect the answer time of questions, which is an effective feature to predict the answer time of questions.

Then we analyze the number of questions and the average answer time of questions in each hour for the three datasets, in order to study the impact of each time period of the day on the answer time of questions in more detail. It can be seen from Fig.3(a) that the number of questions is normally distributed and peaks in the 14–16 time period. However, it can be seen from Fig.3(b) that the average answer time of questions does not change significantly during this time period. Therefore, it is necessary to explore whether the hour of the day has an effect on the answer time of a question in the following research.

5.3 RQ3: How Does the Hour of the Day Affect the Answer Time of a Question?

To explore the impact of the hour in a day for the answer time of questions, we analyze the number of

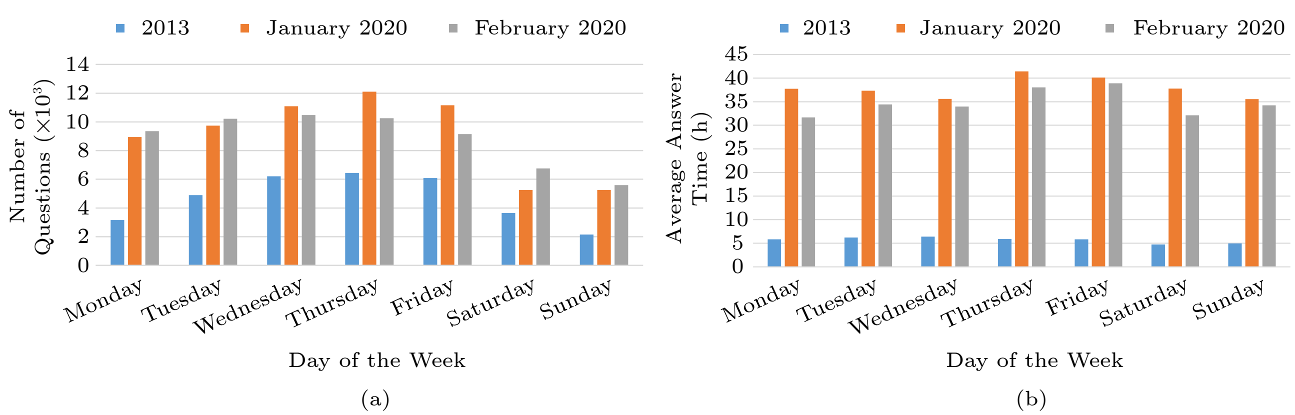
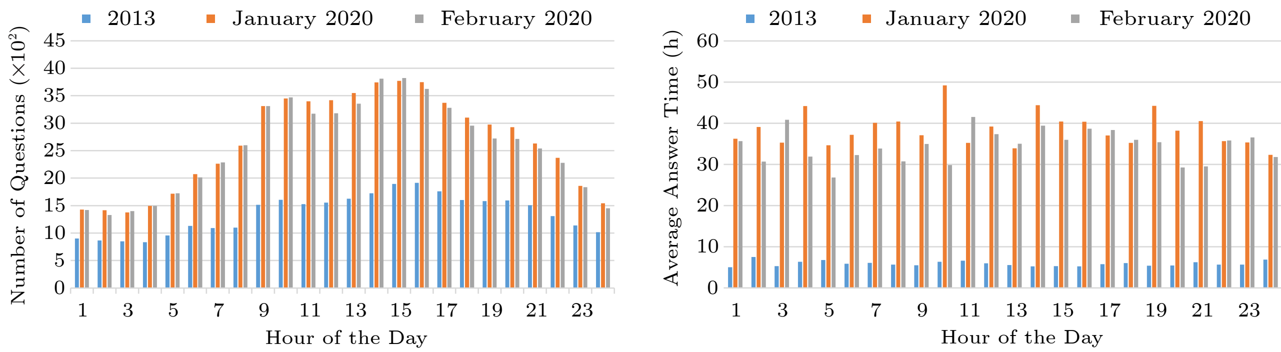


Fig.2. (a) Number of questions posted and (b) average answer time for questions in each day of the week for the three datasets.



(a) (b)

Fig.3. (a) Number of questions posted and (b) average answer time for questions in each hour for the three datasets.

questions and the average answer time in each hour and each day in the above data analysis (as shown in Fig.3). We extract a new time feature from question data (named the *Weekall* feature) for representing which hour of the day the question was posted. We get the *Weekall* feature by

$$time_{\text{week}} = h_o/24,$$

where h_o is the hour extracted from the creation time of the question. We expand the *Weekall* feature value into a vector by expanding the dimension, and fuse it with other feature vectors through the feature fusion model to form a new feature vector. Then, the feature set used for predicting the answer time of the question includes *Body*, *Title*, *Tags*, *Time-rate*, *Week*, and *Weekall* features.

We design the following two cases of comparisons for the three datasets. Table 5 shows the values of relative error MSE' for the answer time in these two cases: without the *Weekall* feature, and with the *Weekall* feature. The optimal results are marked in bold.

It can be seen from Table 5 that the values of relative error MSE' for the answer time with all the features (*Body*, *Title*, *Tags*, *Time-rate*, *Week*, *Weekall*) are the smallest, which is the most obvious for the questions of January 2020. Therefore, the *Weekall* feature has a positive effect on predicting the answer time of questions. Furthermore, we could use a new feature set (including the *Body* feature, *Title* feature, *Tags* feature, *Time-rate* feature, *Week* feature, and *Weekall* feature) to predict the answer time of questions.

Table 5. Values of Relative Error MSE' for Answer Time (h) after Adding *Weekall* Feature for Three Datasets

Features of Used Questions	Dataset		
	2013	January 2020	February 2020
Body, *Title*, *Tags*, *Week*, *Time-rate*	5.597 671	5.500 321	5.499 918
Body, *Title*, *Tags*, *Week*, *Time-rate*, *Weekall*	**5.593 785**	**5.478 284**	**5.497 325**

In the following, we study the *Tags* feature of the question and analyze the number of questions with each specified tag. Figs.4–6 show the number of ques-

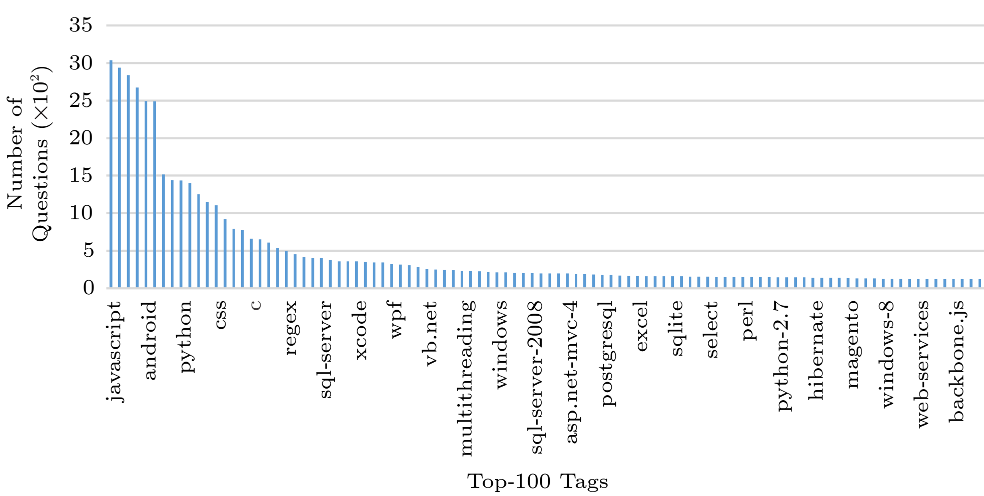


Fig.4. Number of questions with top-k (k = 100) tags for dataset 2013.

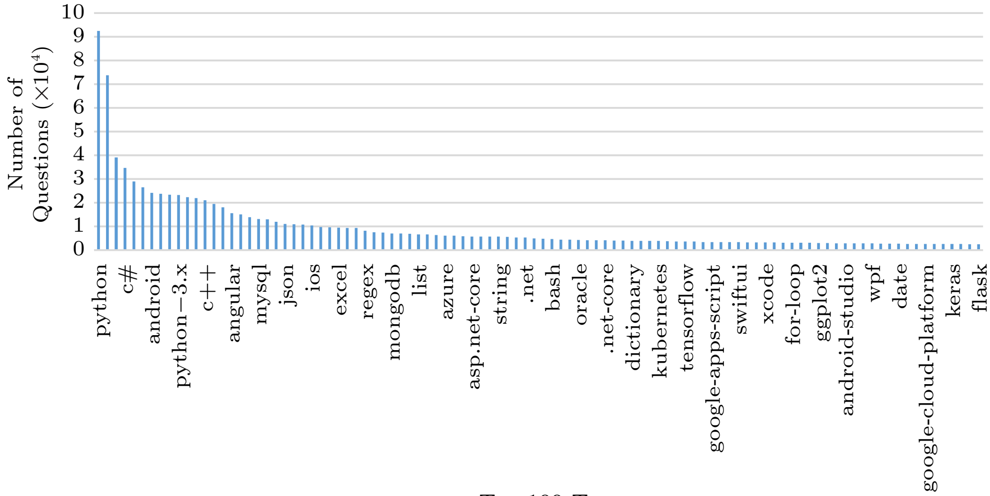


Fig.5. Number of questions with top-k (k = 100) tags for dataset January 2020.

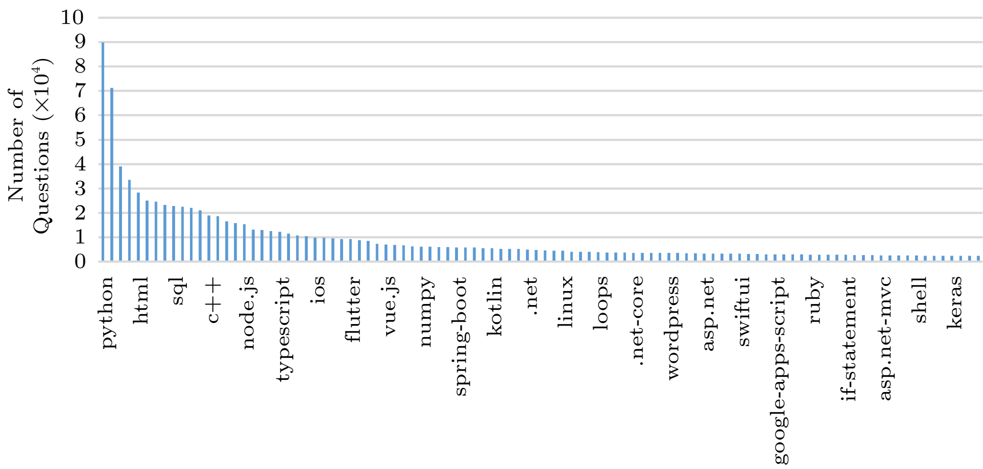


Fig.6. Number of questions with top-k (k = 100) tags for dataset February 2020.

tions containing each of the top-100 tags for the three datasets. Due to the limitations on space in this paper, 100 tags cannot be fully displayed. But we can still see the trend in the number of questions containing each of the top-100 tags. The greater the number of questions containing a tag, the more active the tag. Then we aim to figure out whether the activity of tags impacts on the answer time of questions, and whether the questions with active tags have shorter answer time. It can be seen from Figs.4–6 that the number of questions containing the top-10 tags is the largest, and the top-10 tags are active. Additionally, the number of questions containing the top 60–100 tags is small, and these tags are inactive. Therefore, we choose the questions with the top-50 tags for further study.

5.4 RQ4: How Does the Tag Activity Affect the Prediction of Answer Time?

To study the effect of tag activity on predicting the answer time of questions, we first select questions with the top-k (k = 10, 20, 50) tags as a test set for the three datasets. In the previous experiments, we used all the processed question data to train and test according to a ratio of $9:1$. In this experiment, our training set contains all of the processed question data, and the test set is questions with the top-k (k = 10, 20, 50) tags. As concluded in Subsection 5.3, the *Weekall* feature can improve the performance of the PAT model, and thus it is added in the next experiment. We extract the *Body*, *Title*, *Tags*, *Time-rate*, *Week*, and *Weekall* features from the questions as the input of the deep neural network model to analyze the performance of the PAT model under different test sets. Table 6 shows the values of relative error $MSE^{'}$ of the PAT model in predicting the answer time of questions for three datasets. The first column is the test data, where the optimal result is marked in bold.

Table 6. Values of Relative Error MSE' for Answer Time (h) Under the Top-k (k = 10, 20, 50) Test Sets

Test Set	Dataset		
	2013	January 2020	February 2020
Questions with top-10 tags	**5.559 093**	5.522 607	**5.515 567**
Questions with top-20 tags	5.567 227	**5.518 562**	5.517 117
Questions with top-50 tags	5.576 814	5.526 546	5.525 517

It can be seen from Table 6 that the values of relative error $MSE^{'}$ of predicted answer time are the smallest, which is 5.515 567 hours, when using questions with the top-10 tags as the test set for the February 2020 dataset. When the questions with top-20 tags are used as the test set, the value of relative error MSE' of the predicted answer time is 5.518 562 hours for the January 2020 dataset. Therefore, the activity of tags impacts the performance of the PAT model, which makes the model produce better prediction results on test sets with top-10 and top-20 tags than on the test set with top-50 tags. The results also reveal that the PAT model is more effective on test sets with active tags. It also suggests that labeling a popular tag can make it easier to catch one's attention and get the answers, if a user plan to ask a question for advice on Stack Overflow.

5.5 RQ5: How Does an Active Dataset Affect the Prediction of Answer Time?

In order to explore the impact of active datasets on the performance of the PAT model, we use all questions with the top-k (k = 10, 20, 50) tags to predict the answer time of the questions for the three datasets. We take questions with top-10 tags, questions with top-20 tags, and questions with top-50 tags as the datasets, and then divide them separately into the training set and the test set according to a ratio of 9 : 1. Next, we extract the *Body*, *Title*, *Tags*, *Timerate*, *Week*, and *Weekall* features of the questions as the input to the deep neural network model, and train the model to predict the answer time of questions. Finally, we record the results of the three experiments separately. Table 7 shows the values of the relative error MSE' for the answer time on questions with top-10 tags, top-20 tags, and top-50 tags for the three datasets, and the optimal results are marked in bold.

Table 7. Values of Relative Error $MSE^{'}$ for Answer Time (h) for PAT Model Under Different Datasets

Test Set	Dataset		
	2013	January 2020	February 2020
Questions with top-10 tags	**5.526 785**	5.501 368	5.479 926
Questions with top-20 tags	5.553 070	5.494 471	**5.475 987**
Questions with top-50 tags	5.556 707	**5.487 882**	5.498 280

It can be seen from Table 7 that the prediction performance by using the questions with top-k (k = 10, 20, 50) tags as the dataset is better than that by using all the questions for the 2013 dataset, January 2020 dataset and February 2020 dataset. It reveals that the performance of the PAT model can be improved by using questions with only active tags for experiments. However, there are also differences for datasets in different periods. It can be seen from Table 7 that there is no direct relationship between the activity of tags and the answer time of the question. In other words, it is not true that the more the questions that contain active tags, the shorter the answer time of the question. Actually, the answer time of questions fluctuates considerably. Therefore, we should not only consider *Tags* features, but also consider multiple features comprehensively to get the feature set of the questions.

5.6 RQ6: How Does the Activity of a Single Tag Affect the Prediction of Answer Time?

To explore the impact of a single specific active tag on the performance of the PAT model, we use the questions containing the top-10 tags as the training set, and questions with a single tag in the top-10 tags as the test set for the three datasets. We investigate the impact of a single tag on predicting the answer time of questions.

As before, we extract the *Body*, *Title*, *Tags*, *Timerate*, *Week*, and *Weekall* features from the questions as the input to the deep neural network model, and predict the answer time of questions through model training. Table 8 shows the values of relative error MSE' for the answer time predicted by the PAT

Table 8. Values of Relative Error MSE' for Answer Time (h) of Questions with Top-10 Tags

Dataset	Tags of Questions in Test Set	Values of Relative Error MSE'
2013	javascript	5.535 771
	java	5.503 521
	php	5.494 104
	c#	5.512 172
	android	5.544 136
	jquery	5.501 302
	html	5.523 103
	python	5.528 663
	ios	5.570 634
	c++	**5.490 342**
January 2020	python	5.496 512
	javascript	5.516 533
	java	5.501 913
	c#	5.507 224
	html	5.488 374
	reactjs	5.499 581
	android	5.531 280
	r	**5.449 537**
	php	5.500 238
	python-3.x	5.538 648
February 2020	python	5.536 867
	javascript	5.503 452
	java	5.493 337
	c#	5.496 660
	html	5.506 823
	r	**5.427 288**
	reactjs	5.506 511
	php	5.495 022
	sql	5.475 149
	android	5.514 656

model using questions with top-10 tags as the training set and questions with individual tags as the test set for the three datasets. The optimal results are marked in bold.

It can be seen from Table 8 that the questions with the "c++" tag perform the best in predicting the answer time of questions, with an error of 5.490 342 hours, followed by the questions with the "php" tag, with an error of 5.494 104 hours for the data of 2013. For the data of January 2020 and February 2020, the values of relative error MSE' for the answer time are the smallest for the questions with the "r" tag. Compared with the results of experiment for RQ1, the prediction performance has been improved. Therefore, the category of the tag can also impact on predicting the answer time of questions. Additionally, it can be seen that the questions with "c++" and "r" tags have smaller prediction error than those with other tags.

6 Threats to Validity

Internal Validity. A threat to the internal validity is the user status of Stack Overflow. The degree of user contribution, that is, the honor status, will possibly impact on the answer time of questions. Some users may have a great contribution to Stack Overflow, with more badges and honor, which will increase the probability of their questions being quickly answered. Therefore, the answer time of their questions is relatively short. But for some novice users, the answer time of their questions may be longer. Due to the uncertainty of the user group of Stack Overflow, there is also uncertainty in the answer time of the questions.

Construct Validity. In addition, the difference between the question data in different periods is large on Stack Overflow, which leads to the difference of experimental results. Subsequent studies could start from the aspect of data imbalance. When using the Doc2vec model for text vectorization, the default embedding vector dimension is 100 dimensions. As known, using default parameter settings may lead to insufficient dimensions or redundancy of dimensions. For example, when it is set to 100 dimensions, it can be seen that using 100 dimensions is redundant. We verify the performance of the model using embedding vectors of different dimensions through experiments. The results show that the dimension change of the embedding vector has little influence on the performance of the model. Therefore, we set up an appropriate embedding vector dimension through experimental analysis in order to save space, time, and cost.

7 Conclusions

In this paper, we took the task of predicting the answer time as a regression task, and found the feature set that affects the answer time of questions. We combined feature fusion and the deep neural network method to propose a PAT model to predict the specific answer time of questions. For a question post, the specific answer time can be directly predicted through the PAT model. The user can decide to choose another solution or continue to wait for an acceptable answer based on the prediction result of the model, which can help the user arrange time better. We conducted extensive experiments using real datasets from Stack Overflow, and experimental results showed that the PAT model can well predict the answer time of questions and outperforms the state-of-the-art models.

The PAT model can help users manage their time effectively by using the given exact time estimate for answering their questions. It can also encourage users to rephrase their inquiries in order to get answers more quickly. As a result, users could get prompt and satisfactory answers to their questions, while CQA can attract more users because of the improved user experience.

In a follow-up study, we plan to improve the PAT model through replacing the neural network model with another efficient model, such as the BERT model. We can also combine our feature acquisition and fusion model with the traditional regression model to get a better performance through model improvement and parameter optimization processes.

References

[1] Wu D, Johnson S, Foster C, Li E, Elmiligi H, Rahman M. Improving response time prediction for Stack Overflow questions. In *Proc. the 10th IEEE Annual Information Technology, Electronics and Mobile Communication Conference*, Oct. 2019, pp.786–791. DOI: 10.1109/IEMCON.2019.8936252.

[2] Lopez T, Tun T T, Bandara A, Levine M, Nuseibeh B, Sharp H. An investigation of security conversations in Stack Overflow: Perceptions of security and community involvement. In *Proc. the 1st International Workshop on Security Awareness from Design to Deployment*, May 2018, pp.26–32. DOI: 10.1145/3194707.3194713.

[3] Wang W, Malik H, Godfrey M W. Recommending posts concerning API issues in developer Q&A sites. In *Proc. the 12th IEEE/ACM Working Conference on Mining Software Repositories*, May 2015, pp.224–234. DOI: 10.1109/MSR.2015.28.

[4] Yanovsky S, Hoernle N, Lev O, Gal K. One size does not fit all: A study of badge behavior in Stack Overflow. *Journal of the Association for Information Science and Technology*, 2021, 72(3): 331–345. DOI: 10.1002/asi.24409.

[5] Mondal S, Rahman M M, Roy C K. Can issues reported at Stack Overflow questions be reproduced? An exploratory study. In *Proc. the 16th IEEE/ACM International Conference on Mining Software Repositories*, May 2019, pp.479–489. DOI: 10.1109/MSR.2019.00074.

[6] Tabassum J, Maddela M, Xu W, Ritter A. Code and named entity recognition in StackOverflow. In *Proc. the 58th Annual Meeting of the Association for Computational Linguistics*, Jul. 2020, pp.4913–4926. DOI: 10.18653/v1/2020.acl-main.443.

[7] Zhang J X, Jiang H, Ren Z L, Chen X. Recommending APIs for API related questions in Stack Overflow. *IEEE Access*, 2017, 6: 6205–6219. DOI: 10.1109/ACCESS.2017.2777845.

[8] Pan W F, Ming H, Chang C K, Yang Z J, Kim D K. ElementRank: Ranking Java software classes and packages using a multilayer complex network-based approach. *IEEE Trans. Software Engineering*, 2021, 47(10): 2272–2295. DOI: 10.1109/TSE.2019.2946357.

[9] Ai J, Su Z, Li Y, Wu C X. Link prediction based on a spatial distribution model with fuzzy link importance. *Physica A: Statistical Mechanics and Its Applications*, 2019, 527: 121155. DOI: 10.1016/j.physa.2019.121155.

[10] Su Z, Zheng X L, Ai J, Shang L H, Shen Y M. Link prediction in recommender systems with confidence measures. *Chaos*, 2019, 29(8): 083133. DOI: 10.1063/1.5099565.

[11] Burlutskiy N, Fish A, Ali N, Petridis M. Prediction of users' response time in Q&A communities. In *Proc. the 14th IEEE International Conference on Machine Learning and Applications*, Dec. 2015, pp.618–623. DOI: 10.1109/ICMLA.2015.190.

[12] Bhat V, Gokhale A, Jadhav R, Pudipeddi J, Akoglu L. Min(e)d your tags: Analysis of question response time in StackOverflow. In *Proc. the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, Aug. 2014, pp.328–335. DOI: 10.1109/ASONAM.2014.6921605.

[13] Rahman M M, Roy C K. An insight into the unresolved questions at Stack Overflow. In *Proc. the 12th Working Conference on Mining Software Repositories*, May 2015, pp.426–429. DOI: 10.1109/MSR.2015.55.

[14] Treude C, Barzilay O, Storey M A. How do programmers ask and answer questions on the web? (NIER track). In *Proc. the 33rd International Conference on Software Engineering*, May 2011, pp.804–807. DOI: 10.1145/1985793.1985907.

[15] Goderie J, Georgsson B M, Graafeiland B V, Bacchelli A. ETA: Estimated time of answer predicting response time in Stack Overflow. In *Proc. the 12th Working Conference on Mining Software Repositories*, May 2015, pp.414–417. DOI: 10.1109/MSR.2015.52.

[16] Teevan J, Morris M R, Panovich K. Factors affecting response quantity, quality, and speed for questions asked via social network status messages. In *Proc. the 5th International Conference on Weblogs and Social Media*, Jul. 2011. DOI: 131.107.65.14

[17] Arguello J, Butler B S, Joyce E, Kraut R, Ling K S, Rosé C, Wang X Q. Talk to me: Foundations for successful individual-group interactions in online communities. In *Proc. the 2006 International Conference on Human Factors in Computing Systems*, Apr. 2006, pp.959–968. DOI: 10.1145/1124772.1124916.

[18] Dror G, Maarek Y, Szpektor I. Will my question be answered? Predicting "question answerability" in community question-answering sites. In *Proc. the 2013 European Conference on Machine Learning and Knowledge Discovery in Databases*, Sept. 2013, pp.499–514. DOI: 10.1007/978-3-642-40994-3_32.

[19] Arunapuram P, Bartel J W, Dewan P. Distribution, correlation and prediction of response times in Stack Overflow. In *Proc. the 10th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing*, Oct. 2014, pp.378–387. DOI: 10.4108/icst.collaboratecom.2014.257265.

[20] Bhat V, Gokhale A, Jadhav R, Pudipeddi J, Akoglu L. Effects of tag usage on question response time. *Social Network Analysis and Mining*, 2015, 5(1): Article No. 24. DOI: 10.1007/s13278-015-0263-3.

[21] Mi Q, Gao Y J, Keung J, Xiao Y, Mensah S. Identifying textual features of high-quality questions: An empirical study on Stack Overflow. In *Proc. the 24th Asia-Pacific Software Engineering Conference*, Dec. 2017, pp.636–641. DOI: 10.1109/APSEC.2017.77.

[22] Remígio J, Aragão F, Souza C, Costa E, Fechine J. Question's advisor—A Wizard interface to teach novice programmers how to post "better" questions in Stack Overflow. In *Proc. the 19th International Conference on Enterprise Information Systems*, Apr. 2017, pp.471–478. DOI: 10.5220/0006389504710478.

[23] Kowalik G, Nielek R. Senior programmers: Characteristics of elderly users from Stack Overflow. In *Proc. the 8th International Conference on Social Informatics*, Nov. 2016, pp.87–96. DOI: 10.1007/978-3-319-47874-6_7.

[24] Le Q V, Mikolov T. Distributed representations of sentences and documents. In *Proc. the 31st International Conference on Machine Learning*, Jun. 2014, pp.1188–1196.

[25] Gupta R, Reddy P K. Towards question improvement on knowledge sharing platforms: A Stack Overflow case study. In *Proc. the 2017 IEEE International Conference on Big Knowledge*, Aug. 2017, pp.41–48. DOI: 10.1109/ICBK.2017.25.

[26] Lezina G E, Kuznetsov A M. Predict closed questions on StackOverflow. In *Proc. the 9th Spring Researchers Colloquium on Databases and Information Systems*, May 2013, pp.10–14.

[27] Avrahami D, Fussell S R, Hudson S E. IM waiting: Timing and responsiveness in semi-synchronous communication. In *Proc. the 2008 ACM Conference on Computer Supported Cooperative Work*, Nov. 2008, pp.285–294. DOI: 10.1145/1460563.1460610.

[28] Li K, Zou C Q, Bu S H, Liang Y, Zhang J, Gong M L. Multi-modal feature fusion for geographic image annotation. *Pattern Recognition*, 2018, 73: 1–14. DOI: 10.1016/j.patcog.2017.06.036.

[29] Borovykh A, Oosterlee C W, Bohté S M. Generalization in fully-connected neural networks for time series forecasting. *Journal of Computational Science*, 2019, 36: 101020. DOI: 10.1016/j.jocs.2019.07.007.

[30] Cheng Y P. Backpropagation for fully connected cascade networks. *Neural Processing Letters*, 2017, 46(1): 293–311. DOI: 10.1007/s11063-017-9588-4.

[31] Deng W, Liu H L, Xu J J, Zhao H M, Song Y J. An improved quantum-inspired differential evolution algorithm for deep belief network. *IEEE Trans. Instrumentation and Measurement*, 2020, 69(10): 7319–7327. DOI: 10.1109/TIM.2020.2983233.

[32] Hao L Y, Li J, Guo G. A multi-target corner pooling-based neural network for vehicle detection. *Neural Computing and Applications*, 2020, 32(18): 14497–14506. DOI: 10.1007/s00521-019-04486-1.

[33] Hao L Y, Zhang H, Guo G, Li H. Quantized sliding mode control of unmanned marine vehicles: Various thruster faults tolerated with a unified model. *IEEE Trans. Systems, Man, and Cybernetics: Systems*, 2021, 51(3): 2012–2026. DOI: 10.1109/TSMC.2019.2912812.

[34] Priya S S, Gupta L. Predicting the future in time series using auto regressive linear regression modeling. In *Proc. the 12th International Conference on Wireless and Optical Communications Networks*, Sept. 2015. DOI: 10.1109/WOCN.2015.8064521.

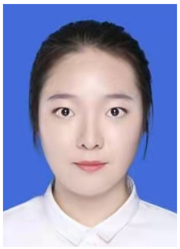
[35] Nguyen B, Morell C, De Baets B. Large-scale distance metric learning for *k*-nearest neighbors regression. *Neurocomputing*, 2016, 214: 805–814. DOI: 10.1016/j.neucom.2016.07.005.

[36] Li Z J, Li Y X, Yu F, Ge D H. Adaptively weighted support vector regression for financial time series prediction. In *Proc. the 2014 International Joint Conference on Neural Networks*, Jul. 2014, pp.3062–3065. DOI: 10.1109/IJCNN.2014.6889426.

[37] Park J G, Jo S. Approximate Bayesian MLP regularization for regression in the presence of noise. *Neural Networks*, 2016, 83: 75–85. DOI: 10.1016/j.neunet.2016.07.010.

[38] Hu Q, Wu W B, Friedl M A. Mapping sub-pixel corn distribution using MODIS time-series data and a random forest regression model. In *Proc. the 6th International Conference on Agro-Geoinformatics*, Aug. 2017, pp.108–112. DOI: 10.1109/Agro-Geoinformatics.2017.8047051.

Shi-Kai Guo received his B.S. degree in computer science, in 2012, from the Information Science and Technology College, Dalian Maritime University, Dalian, where he is currently working toward his Ph.D. degree in computer science and technology. His research interests include mining software repositories, search-based software engineering, fuzzy measures and integrals, and imbalance learning from big data.

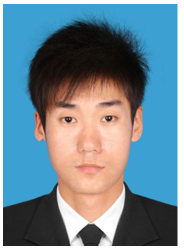


Si-Wen Wang received her Bachelor's degree in information and computing science from Shenyang University of Technology, Shenyang, in 2018, and her Master's degree in computer technology from Dalian Maritime University, Dalian, in 2021. She is currently a Ph.D. candidate in the Navigation College from Dalian Maritime University, Dalian. Her current research interests include bug reports assignment, text summarization and trajectory prediction.

Hui Li received his Bachelor's, Master's and Doctoral degrees from Northeastern University, Shenyang, in 2006, 2008 and 2013, respectively. He is an associate professor of Information Science and Technology College, Dalian Maritime University, Dalian. His research interests include software test, artificial intelligence and complex networks.

Yu-Long Fan received his Bachelor's degree in information and computing science from Dalian Maritime University, Dalian, in 2017. He graduated with his Master's degree in software engineering from Dalian Maritime University, Dalian, in 2020. His current research interests include image classification, image color correction and noise reduction processing.

Ya-Qing Liu is an associate professor of software engineering in Dalian Maritime University, Dalian. He received his Ph.D. degree in computer application technology from Dalian Maritime University, Dalian, in 2011. His research interests are in software testing and quality assurance, model-driven development, and software maintenance.



Bin Zhang received his B.S. degree in information and computational science from Qilu University of Technology, Jinan, in 2008, and his M.S. degree in computer application from Dalian Maritime University, Dalian, in 2010. He is currently a lecturer in Computer Science and Technology College from Shandong Technology and Business University, Yantai. His current research interests include combinatorial optimization and scheduling theory.