Information Processing and Management 53 (2017) 780–798

Contents lists available at ScienceDirect

Information Processing and Management

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

# On early detection of high voted Q&A on Stack Overflow

Mahmood Neshati

*Faculty of Computer Science and Engineering, Shahid Beheshti University, G.C, Tehran, Iran*

ARTICLE INFO

*Article history:*
Received 4 October 2016
Revised 12 January 2017
Accepted 7 February 2017

ABSTRACT

Early detection of high quality content on community question answering platforms is an important emerging problem in which the main goal is the detection of high quality questions and answers in a short time right after their submission. Improving the process of question routing, reducing the number of questions with no answers, improving the user experience and also promoting the content quality of a CQA by rejecting low quality contents are all benefits of solving the early detection of high quality content problem in CQA. The main challenge of solving this problem is that the value of a few features is available in a short time after submission of a content in CQA. In other words, unlike previous related research, it is not possible to utilize comprehensive set of features to detect high quality content. In this paper, we view the content quality from the perspective of the voting outcome. Specifically, we consider those Q&A which will get more votes than a certain threshold as high quality posts. Analyzing large amount of data in a CQA, we observed two important patterns which help us with early detection of high quality content. We named the first pattern as *accepted answer effect* and the second pattern as *answer competition effect*. According to the first pattern, the chance of a high quality question to get an accepted answer is higher than the chance of other questions and vice versa. According to the second pattern, only few number of answers of a specific question will be high quality answers. We show that these patterns are valid in a short time after the submission of content on CQA. Utilizing these patterns, we propose a unified relational classification framework to solve the problem. In our proposed framework, the quality of a given question and its associated answers can be predicted simultaneously soon after their submission. We conduct several experiments on six data collections gathered from Stack Overflow in order to show the efficiency of the proposed models. Our experiments indicate that the performance of high quality content detection can improve up to 10.7% and 35.3% in comparison with a state-of-the-art independent classifier for questions and answers, respectively. Moreover, we found 1.2% and 11.8% F-measure gain in average versus a recent strong baseline by Yao et al. (2015) for questions and answers, respectively.

## 1. Introduction

Community Question and Answering (CQA) websites are new data sharing platforms which help users to find the best answer to their information need directly, instead of searching and reading long and detailed documents. Successful CQA

*E-mail addresses:* m_neshati@sbu.ac.ir, mahmood.neshati@gmail.com

http://dx.doi.org/10.1016/j.ipm.2017.02.005
0306-4573/© 2017 Elsevier Ltd. All rights reserved.

websites include both general websites such as *Yahoo! Answers*[1] and *Quora*[2], and also domain specific QA websites like *StackOverflow*[3] and *Mathematics Stack Exchange*[4].

Previous research confirms that altruism can be considered as the main reason to motivate users to contribute in a community like a CQA (Ostrom, 1990). However, in addition, CQA websites usually use approaches like gamification to enhance the motivation of the users and accordingly to improve their quality and quantity of contribution. Voting on questions and answers, approving the satisfying answer (i.e. the accepted answer), awarding badges for an outstanding contribution of users, uniquely representing users by avatar, demonstrating the reputation of users and etc. are all elements of such gamification mechanism to enhance the motivation of users to provide high quality contents.

Some of these gamification elements (e.g. the number of votes of a question or answer, the reputation score of a user) can be considered as the quality indicators in CQAs (Arora, Ganguly, & Jones, 2015; Hsu, Khabiri, & Caverlee, 2009; Ponzanelli, Mocci, Bacchelli, Lanza, & Fullerton, 2014b; Yao et al., 2015). Users of CQAs usually utilize these indicators to recognize high quality contents (i.e. Q&A) and high quality experts. As an example, intuitively, the probability of reading an answer is enhanced by the number of its votes.

Detecting and promoting high-quality contents and users is a primary success factor of a CQA which can lead to its overall quality and accordingly improves the site reputation, user experience, user engagement, and boost web result rankings.

Finding high quality contents and users (i.e. experts) on CQA is a well studied problem. Several aspects of this problem are investigated which can be divided in three main categories:

1. **Feature identification and selection:** The main approach in this category is to use a classifier to classify a given content (i.e. question, answer) or user into high or low quality item (Agichtein, Castillo, Donato, Gionis, & Mishne, 2008; Blooma, Goh, & Chua, 2012; Molino, Aiello, & Lops, 2016; Ponzanelli et al., 2014b; Toba, Ming, Adriani, & Chua, 2014; Yao et al., 2015). Here, the main goal is the identification and selection of important discriminative features. Several features like textual (Agichtein et al., 2008), temporal (Figueroa, Gómez-Pantoja, & Herrera, 2016), semantical (Figueroa & Neumann, 2016) and behavioral (Fu et al., 2016) have been proposed to detect the high quality content and users in CQAs. The methods in this category use separate classifiers for questions, answers and users. In other words, the quality label of each item (i.e. question, answer and user) is determined independent of other items.
2. **Quality dependency modeling:** Questions, answers and users are the main components of a CQA. The approaches in this line of research (Bian, Liu, Zhou, Agichtein, & Zha, 2009; Yao et al., 2015) utilize the quality dependencies between these components to improve the accuracy of the high quality content detection. For example, (Bian et al., 2009) uses the circular quality dependency between user, question and answer to improve the accuracy of prediction. This research assumes that an expert user usually provide a high quality answer in response to a good (i.e. high quality) question which is asked by an experienced user. Similarly, Yao et al. (2015) recognized a strong correlation between the number of votes on a question and its best answer and used it to simultaneously detect high quality questions and answers on CQA.
3. **Early detection of High quality items:** More recently, a new variation of high quality content detection problem on CQA is emerged in which the main goal is to predict of high quality posts (or users) in a short time soon after their submission (or user activity). For example, van Dijk, Tsagkias, and de Rijke (2015); Pal, Farzan, Konstan, and Kraut (2011) proposed a method to detect expert users in Stack Overflow after few weeks of their activity. Similarly, the problem of early high quality post detection has been recently introduced in Yao et al. (2015). Here, the main idea of research is to recognize and utilize time invariant patterns which can help to identify high quality items soon after their submission.

The main idea proposed in this paper can be categorized in number two and three of the above mentioned categories. Specifically, by investigating large amount of CQA data, we recognize two simple and important quality dependency patterns which can be used to enhance the high quality post detection problem. These two patterns are time invariant[5] and we show that they can be used to early detection of high quality posts on CQA.

We called our first observed pattern on CQA data as the *Accepted Answer Effect*. According to this pattern, the quality of a question (i.e. its number of votes) is strongly dependent on whether that question has (or will get) an accepted answer or not. Intuitively, questions with accepted answer attract more attention from the community in comparison with other questions, because a question with accepted answer is usually relevant, important and general enough to motivate someone to answer it.

According to the result of our experiments, we show that this pattern is also valid in a short time period after submission of a question on CQA and accordingly, we use it for early high quality post detection.

In addition, while in previous research (Yao et al., 2015), two classifiers[6] (one classifier to detect high quality questions and another one for high quality answers) have been used to detect high quality questions and answers, our contribution in this part is to utilize another classifier to predict whether the question will get an accepted answer in a short time after

[1] answers.yahoo.com

[2] www.quora.com

[3] www.stackoverflow.com

[4] math.stackexchange.com

[5] By time invariant, we mean that these patterns are valid soon after the submission of a post on CQA.

[6] These two classifiers can be separate or relational.

submission or not. Intuitively, the output of the proposed classifier is dependent on the output of the question classifier and the answer classifier. Specifically, a question will get an accepted answer if it is a high-quality question and on the other hand it will get an accepted answer, if there is any high quality answer for that question. Due to such circular dependency between output of classifiers, we use a relational classification method to solve the problem.

Our second observed pattern is named *Answer Competition Effect*. According to this observation, the number of votes on answers of a specific question follows a power law distribution. In other words, for a given question, only a few number of its answers usually get a high number of votes and most of them get a few number of votes. This effect can be explained by the competition that happens among the answers of a particular question. In general, this competition has only a few number of winners (i.e. high-voted answers). This pattern can also be explained by the question selection bias property introduced in Pal, Harper, and Konstan (2012). According to this property, an expert user (i.e. high quality users who usually provide high quality answers) often selects a question to give an answer which either has no any answer or has only low quality answers. In other words, if a question already has a high quality answer, then the probability of answering that question by an expert user substantially decreased. We show that the *Answer Competition Effect* is also a time invariant property and can be utilized to early detection of high quality post on CQA. Our contribution in this part is proposing the answer competition effect and encoding it into our proposed relational classification model to enhance the accuracy of prediction. The mentioned patterns can be easily encode in our model which is another benefit of the proposed model.

In this paper, the main research questions are: RQ1- How can we predict high quality posts in CQA in short time after their submission? Which features are more beneficial in this task? If the accepted answer effect and answer competition effect are valid in short time after Q&A submission? RQ2- How can we utilize both the Q&A features and the mentioned patterns in a unified framework to detect high quality content in CQA? RQ3- How does the proposed framework perform in comparison with baseline approaches?

To sum up, the main contributions of this paper includes:

- Investigating two important observations that indicate the quality dependencies between CQA posts.
- Utilizing the mentioned observations in a uniform prediction model to solve the early high quality post detection problem.
- Testing our algorithm on real data collections gathered and processed from the Stack Overflow.

The rest of this paper is organized as follows. In Section 2, we review some related work and a description of the background and preliminaries. In Section 3, we introduce the problem of early high quality post detection and also propose our solutions. In Section 4 and 5, we define the experimental setup and report the experimental results. Finally, we present the conclusions and future work in Section 6.

## 2. Preliminaries and related work

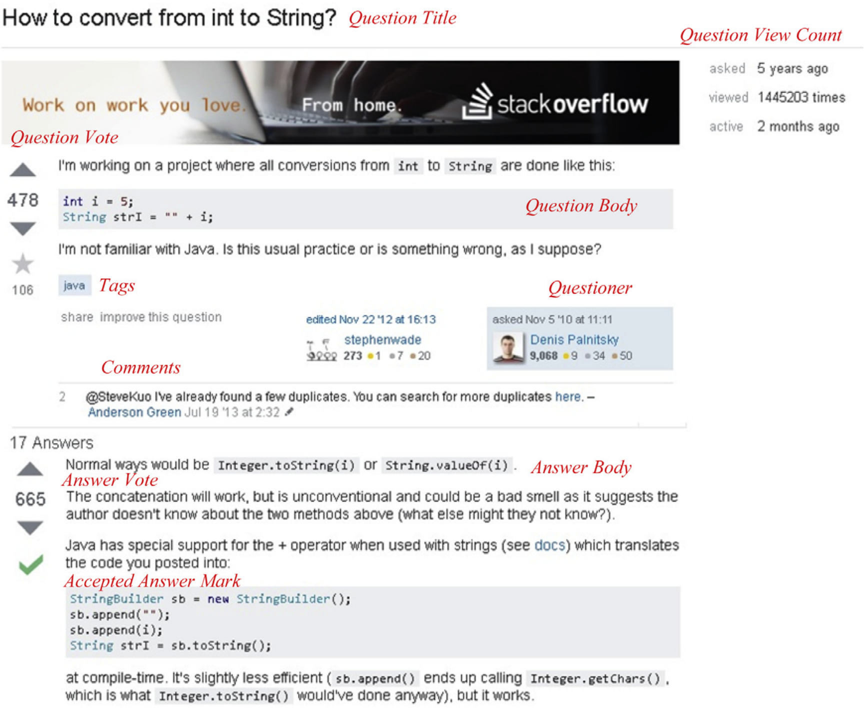
In this section, we briefly introduce the Stack Overflow CQA, its components and the major properties of questions, answers, and user interactions. Then, we investigate the related works in Section 2.1.

As one of the most popular CQAs, Stack Overflow works as a platform for collaborative information sharing. Users can ask questions, answer them, vote up or down and also edit posts. Some of these actions (e.g. vote up or down) are restricted to active members of Stack Overflow. Users can earn reputation points and *badges* for their valuable behavior. For example, users are awarded ten reputation points for receiving an up vote on their answers given to a question, and can earn badges for their valuable contributions (e.g. asking a question with score of 10 or more) which represents a kind of gamification of the traditional Q&A site or forum.

Fig. 1 indicates a sample question and answer on Stack Overflow. Each question has a title, body, a number of votes (i.e. difference between up and down votes), a set of tags (selected by the questioner), a questioner and possibly a set of comments by users. Besides, the number of views of the question, the date and the time of the question are visible for each question. For each answer, the body of the answer and its number of votes are indicated. The green checkmark sign in Fig. 1 indicates the *accepted answer* of the question. When the original author of the question receives a satisfactory answer to his or her question, he/she may accept that answer. Acceptance is indicated by a green checkmark next to the answer that has been accepted by the questioner. “Accepting an answer is not meant to be a definitive and final statement indicating that the question has now been answered perfectly. It simply means that the author received an answer that worked for him or her personally, but not every user comes back to accept an answer, and of those who do, they may not change the accepted answer if a newer, better answer comes along later.” (Stack Overflow, 2016)

Stack Overflow has some processes to refine the quality of its questions. Specifically, low-quality questions and the ones that does not fit into the Stack Overflow may be put *on hold* state by experienced community members. While questions are *on hold*, they cannot be answered but can be edited to make them eligible for reopening. Questions that are not reopened within five days will change from *on hold* to *closed* state. Some of the main categories of questions that may be closed by the community include (Stack Overflow, 2016):

- Duplicated question.
- Off topic.
- Unclear.
- Too broad.



How to convert from int to String?

asked 5 years ago
viewed 1445203 times
active 2 months ago

478

I'm working on a project where all conversions from `int` to `String` are done like this:

```
int i = 5;
String strI = "" + i;
```

I'm not familiar with Java. Is this usual practice or is something wrong, as I suppose?

106

java

share improve this question

edited Nov 22 '12 at 16:13
stephenwade
273 ●1 ●7 ●20

asked Nov 5 '10 at 11:11
Denis Palnitsky
9,068 ●9 ●34 ●50

2 @SteveKuo I've already found a few duplicates. You can search for more duplicates here. – Anderson Green Jul 19 '13 at 2:32

17 Answers

665 ✓

Normal ways would be `Integer.toString(i)` or `String.valueOf(i)`.

The concatenation will work, but is unconventional and could be a bad smell as it suggests the author doesn't know about the two methods above (what else might they not know?).

Java has special support for the + operator when used with strings (see docs) which translates the code you posted into:

```
StringBuilder sb = new StringBuilder();
sb.append("");
sb.append(i);
String strI = sb.toString();
```

at compile-time. It's slightly less efficient (`sb.append()` ends up calling `Integer.getChars()`, which is what `Integer.toString()` would've done anyway), but it works.

**Fig. 1.** An example of a question on Stack Overflow, Q: How to convert from int to string? for which one out of seventeen, answers is shown. (For interpretation of the references to color in the text, the reader is referred to the web version of this article.)

### 2.1. Related work

High-quality content detection has attracted a great deal of attention, mostly due to the launching of the user generated content platforms (Agichtein et al., 2008). While Dalip, Gonçalves, Cristo, and Calado (2016) recently proposed a multi-view and multi-dimensional framework to assess content quality in web 2.0, we specifically focus on content quality factors and algorithms related to community question answering platforms in this section. We review the principal related works to our research according to the classification depicted in Fig. 2 which summarizes related works according to their main properties and features. For more information about different aspects of community question answering platforms, refer to the recent survey by Srba and Bielikova (2016). The first property which we apply to review the related works is the *Quality Evaluation Approach*. We divide the relevant studies into two main approaches (i.e. manual and automatic) regarding this property.

In manual approaches, human experts evaluate the quality of user generated contents and assign a label to each item. Examples of this line of research include Toba et al. (2014), Figueiredo et al. (2013), Suryanto, Lim, Sun, and Chiang (2009) and Momeni, Tao, Haslhofer, and Houben (2013). In contrast, the automatic evaluation approaches assess the quality of user generated contents based on a defined measure (i.e. formula)(Blooma et al., 2012; Flekova, Ferschke, & Gurevych, 2014; Lu, Tsaparas, Ntoulas, & Polanyi, 2010; Ponzanelli, Mocci, Bacchelli, & Lanza, 2014a; Ponzanelli et al., 2014b; Ravi, Pang, Rastogi, & Kumar, 2014). As an example of automatic quality evaluation, in Ponzanelli et al. (2014b), the deleted or closed questions with negative count of votes are defined as the low quality and the rest of questions as the high quality questions.

Ravi et al. (2014) proposed a new question quality measure which is independent of the popularity of a question. In this study, the ratio of the upvotes of a question and its number of visits is used to evaluate the quality of the question (i.e. the quality indicator). Although our research can be categorized as the automatic evaluation approach, we cannot use the proposed measure in Ravi et al. (2014) to estimate the quality of both the questions and the answers on Stack Overflow, due to no view count for the answers in Stack Overflow dataset.

The number of votes on comments (Hsu et al., 2009), the user rating for wiki articles (Flekova et al., 2014), the number of votes of a question/answer on Stack Overflow (Arora et al., 2015; Blooma et al., 2012) and the number of votes of a user review (Lu et al., 2010) are other examples that can be classified as the automatic evaluation approaches.

Ponzanelli et al. (2014a) defined four levels of quality for a question based on its number of votes and its final status (i.e. closed, deleted). In this study, questions with more than seven votes are defined as the best quality class (i.e. class A) and closed and deleted questions are marked as the lowest quality questions (i.e. class D).

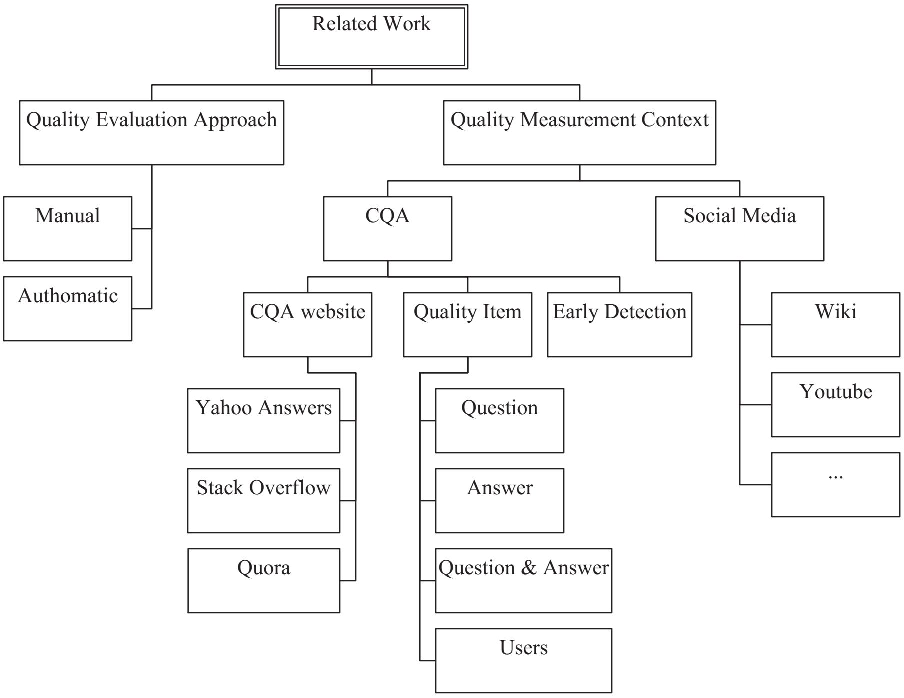


**Fig. 2.** Classification of related work.

According to Fig. 2, studies on quality of the user generated content has been performed in several contexts such as community question answering (CQA) (Agichtein et al., 2008; Arora et al., 2015; Bian et al., 2009; Blooma et al., 2012; Li, Jin, Lyu, King, & Mak, 2012; Liu, Bian, & Agichtein, 2008; Ponzanelli et al., 2014a; Ponzanelli et al., 2014b; Ravi et al., 2014; Suryanto et al., 2009; Toba et al., 2014; Yao et al., 2015), social media (Figueiredo et al., 2013; Hsu et al., 2009; Lu et al., 2010), Fliker (Momeni et al., 2013) and Wikipedia (Anderka, Stein, & Lipka, 2012; Flekova et al., 2014).

Although, initial research on estimating the quality of user generated data on CQA are performed on ”Yahoo Answers”, more recent studies focused on Stack Overflow which is a promising gamified CQA platform. Different aspects of content quality on Stack Overflow has been studied recently including content classification (Arora, Ganguly, & Jones, 2016), mining successful answers (Calefato, Lanubile, Marasciulo, & Novielli, 2015) and mining the duplicate questions (Ahasanuzzaman, Asaduzzaman, Roy, & Schneider, 2016). Apart from the Yahoo answers and Stack Overflow, some recent research studies (Patil & Lee, 2016) focuses on Quora[7] which is one of the successful general purpose CQAs.

As the most related line of research to our work, we categorize the user-generated quality estimation methods in CQA context to four main branches which are explained as follows.

### 2.2. Question quality

This line of research (Arora et al., 2015; Chua & Banerjee, 2015; Li et al., 2012; Ponzanelli et al., 2014a; Ravi et al., 2014) focuses on the prediction of the quality of questions asked by users. The primary approach in these studies is to reduce the high-quality question prediction problem into a classification problem. Li et al. (2012) utilized the relation between the expertise of a questioner and the quality of his/her questions to predict the question quality.

Agichtein et al. (2008) investigate the effect of a diverse set of features on question quality prediction. Textual features (grammatical, punctuation and typos, syntactic and semantic complexity), questioner expertise features and the number of views of questions are the main feature groups used by this research. As another related research, Ravi et al. (2014) proposed a new metric for question quality which is beyond mere popularity. They proposed techniques using latent topic models to predict the quality of questions automatically based on their content. They focused only on the quality of the questions and did not consider the quality of answers and the co-relation between them. They also proposed the content based and topic based features to predict the quality of questions. They showed that latent topical aspects shared between related questions are good predictors of question quality.

Similar to studies in this part, Ponzanelli et al. (2014a) present an approach to automate the classification of questions according to their quality. They used several features including the contents features and the community-related aspects.

[7] www.quora.com

In a recent research (Arora et al., 2015), the question quality have been evaluated in Stack Overflow. This study proposes a method for enhancing the question quality prediction performance by using the content extracted from previously asked similar questions in the forum. They investigated various document (questions already asked in the forum) and query (current question to be classified as good or bad) representation alternatives and different retrieval models in order to retrieve the set of similar questions. Their result showed that the performance of the question classification tasks depends on how effectively they can retrieve this set of similar questions.

These works simply ignore the effect of quality dependencies between questions and answers and just focus on finding robust features to detect high-quality questions in CQA.

### 2.3. Answer quality

The second group of research on prediction of the quality of CQA user generated data focuses on answer quality. Toba et al. (2014) propose a hybrid hierarchy-of-classifiers framework to detect high-quality answers. First, they analyze the question type (e.g. Factoid, Yes/No question, etc.) to guide the selection of the right answer quality model. They manually label each question-answer pair with a quality label in good and bad judgment. In their research, they only predict the high-quality answers and ignore the relationship between the quality of answers and questions.

As another research in this scope, Suryanto et al. (2009) indicates that answer quality can improve the QA performance significantly. They showed that quality, apart from the relevance, is an important criterion for selecting right answers for a given question. However, in this research, the authors did not use content based features and only focused on user-related features to derive answer quality. They proposed several graph-based algorithms to predict the quality of answers and then use it to improve the answer ranking for a given question. Blooma et al. (2012) study the predictors of high-quality answers in Yahoo Answers CQA. They used the social features of the person who responded to the question as well as the content features of answers to predict its quality. More recently, Wei et al. (2016) proposed a summarization approach for high quality and novel answer retrieval in CQA. In addition, Liu, Feng, Liu, Hu, and Wang (2015) proposed a method to predict the quality of answers in CQA by utilizing non-textual features.

### 2.4. User quality

Apart from the question and the answer, the third main component in each CQA is the user. Several research studies has been performed to detect high quality users on CQA (Bouguessa & Romdhane, 2015), routing questions to appropriate users (Yan & Zhou, 2015), analysing of the abusing behavior of users on CQA (Kayes, Kourtellis, Quercia, Iamnitchi, & Bonchi, 2015) and evaluating the reliability and expertise of the users on CQA (Pelechrinis et al., 2015). However, in this paper, we focus on question and answer and our main goal is to find high quality content in a short time after their submission.

### 2.5. Joint quality model

This line of research is the most relevant one to our work. In these studies, the mutual quality relation between questions and answers have been used to enhance the quality prediction of the user generated data.

Yao et al. (2015) recently proposed a method to detect high-quality posts in CQA. The key intuition is that the voting score of an answer is strongly positively correlated with that of its question, and they verified such correlation in two real CQA data sets. According to this idea, they proposed a family of algorithms to jointly detect the high-quality questions and answers soon after they are posted in the CQA sites. Although our first observation in detection of high-quality Q&A is similar to their idea, they did not consider the answer competition effect which can help substantially to improve the high-quality answer detection.

As another related research in this category, we can mention the research proposed in Bian et al. (2009). Bian et al. proposed to propagate the labels through a user-question-answer graph, so as to tackle the sparsity problem where only a small number of questions/answers are labeled. In contrast, we formulate an optimization problem to penalize the differences between question and answer labels. As the last related work in this part, Li et al. (2012) adopt the co-training approach to employ both question features and answer features. However, at the methodology level, these two methods still treat question and answer quality prediction as separate problems.

As indicated in Fig. 2, another related research topic to this paper is the early high quality content detection. Here, the main goal is to predict of high quality posts (or users) in a short time soon after their submission (or user activity). For example, van Dijk et al. (2015) and Pal et al. (2011) proposed a method to detect expert users in Stack Overflow after few weeks of their activity. Similarly, the problem of early high quality post detection has been recently introduced in Yao et al. (2015) which we used as one of the baselines in our experiments.

It is worth mentioning that the result of high quality content detection algorithms can be used to enhance the result of several related IR tasks on CQA such as expert finding (Neshati, Beigy, & Hiemstra, 2014a; Neshati, Hashemi, & Beigy, 2014b), expert matching (Neshati, Hiemstra, Asgari, & Beigy, 2014c) and also expert profiling.

### 2.6. Research position

According to the categorization in Fig. 2, Our proposed models can be classified as the automatic evaluation approach. Similar to Yao et al. (2015), we used the number of votes on questions and answers as the quality indicator. Specifically, we defined questions and answers with more than ten votes as the high-quality posts. According to this definition, 2.7% and 3.1% of questions and answers are marked as high-quality posts in our data set, respectively.

In terms of *quality measurement context*, our research focused on finding high quality contents on Stackoverflow and on finding both high quality questions and answers simultaneously but ignores the effects of users. In addition, our main focus in this research is finding high quality contents in a short time interval after the submission of the content.

## 3. Early high quality post detection in CQA

The problem of early high quality post detection is defined as follows Yao et al. (2015):

**Given**: a set of posts(i.e. questions and answers); their features; and the final quality label of each post.

**Find**: the quality label of a new question and its associated answers.

In this problem, an important constraint is that the posts features are observed only in a short time period after the post submission. Similar to Yao et al. (2015), we divide the posts features into two categories. The first category includes features which are observable at the submission time and the second category includes the ones observed in one hour after the post submission.

A standard approach to predict high quality post in CQA is to reduce the problem into a classification problem (Agichtein et al., 2008; Blooma et al., 2012; Li et al., 2012; Ponzanelli et al., 2014a; Ponzanelli et al., 2014b; Toba et al., 2014; Yao et al., 2015). These works use discriminative textual, behaviorial and temporal features to detect high quality posts in CQA. In Section 3.1, we introduce the logistic regression as a typical classification algorithms for solving the high quality post detection problem. In Section 3.2, we describe our observations on post's quality dependencies and then in Section 3.3, we propose our solution to exploit these dependencies to solve the early detection of high quality detection.

### 3.1. Independent classifier (logistic regression)

Several discriminative features associated with each question/answer can be used to train two logistic regression classifiers. The first classifier can be used for detection of high quality questions and the second classifier will be used for high quality answer detection. The important point here is that each classifier is trained independent of other classifier and more importantly, in this approach the label of each question/answer is predicted independent of other posts.

Two set of training instances can be used to train two separate classifiers for questions and answers:

$TrainSet_Q = \{(q_1; l_1)\dots(q_n; l_n)\}$ and $TrainSet_A = \{(a_1; l'_1)\dots(a_m; l'_m)\}$ where $q_i$ and $a_j$ are the feature vector associated with the $i$th question and $j$th answer; $l_i \in \{true, false\}$ and $l'_j \in \{true, false\}$ are their corresponding label and $n$ and $m$ are the number of training instances for questions and answers, respectively.

In this method, for question $q_i$, the probability $p(l_i|q_i)$ is used the predict its label:

$$p(l_i = 1|q_i) = \frac{1}{1 + \exp(\theta_q q_i)} \tag{1}$$

$$p(l_i = 0|q_i) = \frac{\exp(\theta_q q_i)}{1 + \exp(\theta_q q_i)} \tag{2}$$

where vector $q_i$ is the feature vector corresponding to the question $q_i$ and $\theta_q$ is the training parameter. The log-likelihood of the training data is as follows and should e minimized during training:

$$\log \mathcal{L}(\theta_q \mid Q, T) = \sum_{i=1}^{n} \log P(l_i|q_i; \theta_q) \tag{3}$$

In this equation, $L = \{l_1, l_2, \dots, l_n\}$ and $Q = \{q_1, q_2, \dots, q_n\}$ represent the set of labels and the set of feature vectors for each question in the training set respectively.

We use the set of features indicated in Table 3 for question's and answer's label prediction. Fig. 9a shows the logistic regression model which predicts the label of each post (i.e. black circles = target variables) independent of the label of other posts. In this figure, black circles indicate the target variables (i.e. high quality/low quality) and the rectangles indicate the feature vector of question and answers. In this figure, $A_{ij}$ is the $j$th answer of the $i$th question.

### 3.2. High quality post dependency observations

As mentioned before, the logistic regression model predicts the label of each post independent of other posts. In this section, we explain our observations on dependencies between label of posts in a CQA. These observation can be embedded in a relational classification framework to improve the quality of high quality post detection problem.

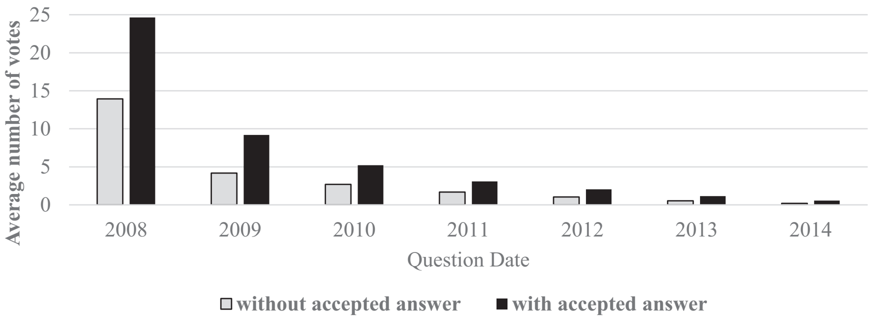


**Fig. 3.** Accepted answer effect on average number of votes in different years.

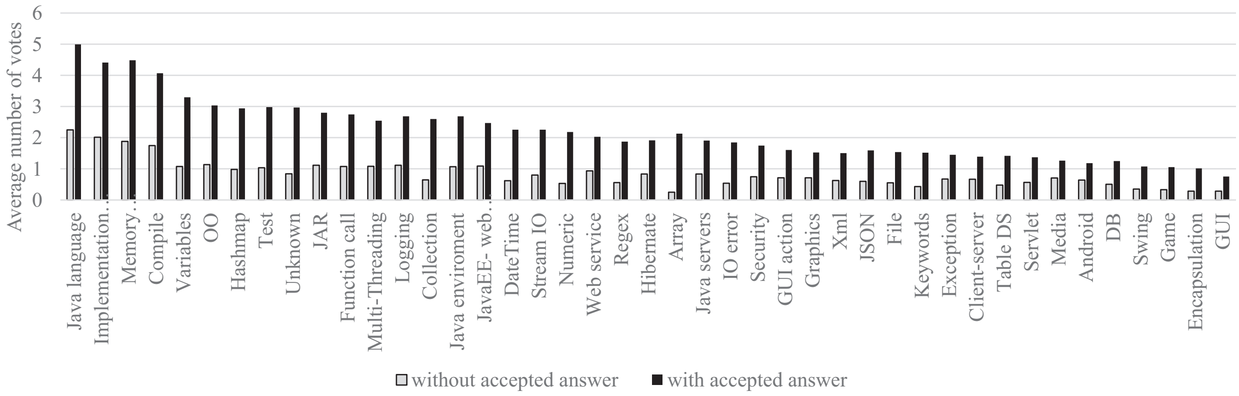


**Fig. 4.** Accepted answer effect on average number of votes for different topics.

- **Accepted answer effect**

  Our first observation is about the *Accepted Answer Effect*. Intuitively, when a user finds a question without accepted answer[8] on CQA the probability of reading the question and consequently upvoting it, will be decreased dramatically. To be more specific, the number of votes on a question (i.e. the quality of a question) is dependent on the quality of its answers and whether it has (or will get) an accepted answer. The dependency between the number of votes on a question and its chance to get an accepted answer is a bidirectional and reinforcing relation.

  In our dataset, the average number of votes for questions with accepted and questions without accepted answer are **2.29** and **0.76** votes, respectively. In other words, questions with accepted answer get about 200% more votes in comparison with questions without accepted answer which clearly support the *accepted answer effect*.

  Several features can affect the number of votes on a question. For example, the duration (i.e. the time elapsed from the post date) of a question has a substantial effect on the number of its votes. Intuitively, we expect that old questions get more votes in comparison with newly asked questions. Fig. 3 indicates the average number of votes on questions which are classified by the year of the question. This figure shows that independent of the duration of a question, questions with accepted answer get more votes in comparison with questions without accepted answer.

  The popularity of a question topic is also a major factor that can affect the number of its votes. Intuitively, questions with general topics usually get more votes in comparison with very specific topics. Fig. 4 indicates that independent of the topic[9] of questions, the accepted answer effect is observable.

  In addition, according to the Fig. 5, the lag time between a question and its first answer also can affect the number of its votes. However, also, in this case, the accepted answer effect is valid.

  In our last experiment in this section, we indicate the accepted answer effect over time in Fig. 6. Here, our specific question is that whether or not this observation is valid in short time period after submission of the question. In this Figure, the average number of votes of questions which eventually get an accepted answer is compared with the average number of votes of questions which does not get an accepted answer in their whole life cycle. The mentioned averages are computed in several snapshots in consecutive days after the submission of the question. For example, the average number of votes of questions with accepted answer is 2.31 after 500 days of their submission while this average equals to 1.44 for other question.

[8] Refer to Section 2 for more detail on the meaning and the mechanism of accepting an answer on CQA.

[9] We use LDA (Blei, Ng, & Jordan, 2003; McCallum, 2002) to detect the question's topic and then manually label the name of each topic.

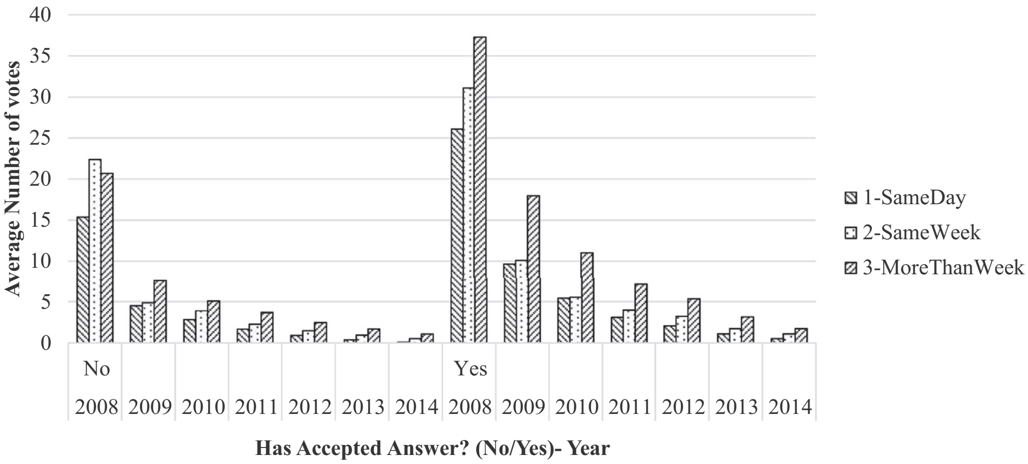


**Fig. 5.** Accepted answer effect on average number of votes for different lag time between question and its answers.

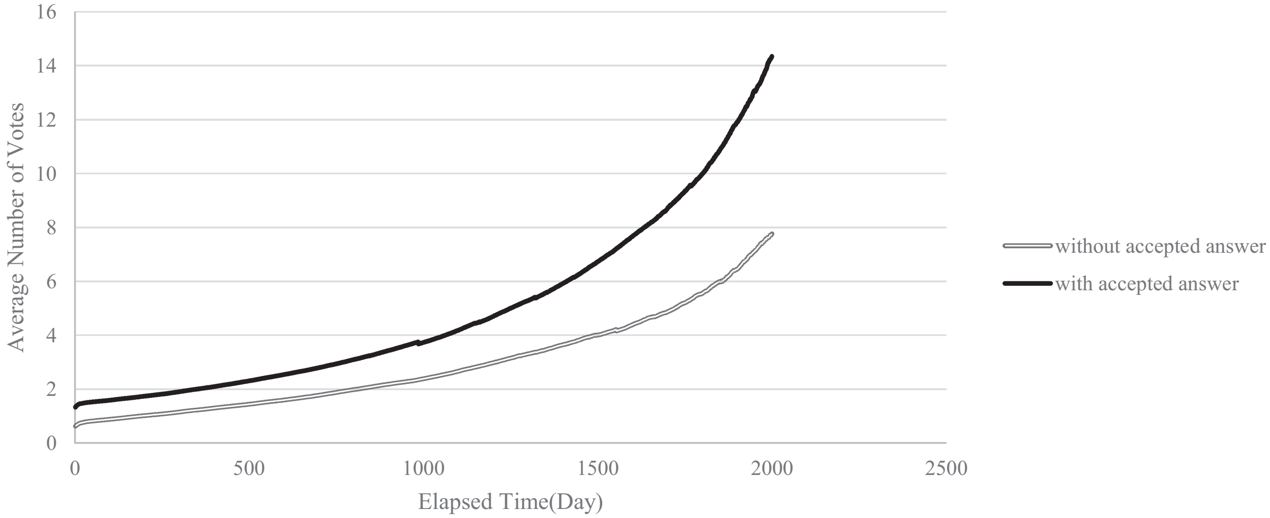


**Fig. 6.** Accepted answer effect observed on consecutive days after the submission of the question.

According to Fig. 6, the accepted answer effect is valid in all snapshots after the question submission. Specifically, even in the first day of question submission the average number of votes of question with accepted answer is significantly higher than the other questions. This observation indicates that we can exploit the accepted answer effect in a short time period soon after the question submission.

- **Answer competition effect**

According to this observation, the distribution of votes answers of a specific question follows the power law distribution. In other words, for a given question, only a few number of answers usually get a high number of votes while most of them get a very few number of votes.

Fig. 7 shows the answer competition effect. For the $k$th rank (horizontal axis), we plot a point which indicates the average ratio[10] of the number of votes on the $k$th best answer[11] of a question to the number of votes on the best answer to that question ($\frac{\#\text{ of votes on the } k\text{th best answer}}{\#\text{ of votes on the best answer}}$). For example according to this figure, the average number of votes on the second best answer is only 0.37 of the number of votes on the best answer.

Fig. 7 indicates that there is a competition among answers of each specific question to get votes from the users. In this competition, in average, there are only a few number of winners (i.e. high-voted answers) and a large number of losers (i.e. low-voted answers).

In Fig. 8 the answer competition effect is demonstrated over time. In this figure, for each question, the number of votes of the $i$th best answer is divided by the number of votes of the best answer and the average over all questions is

[10] The average ratio is computed on all questions.

[11] $k$th answer with most number of votes.

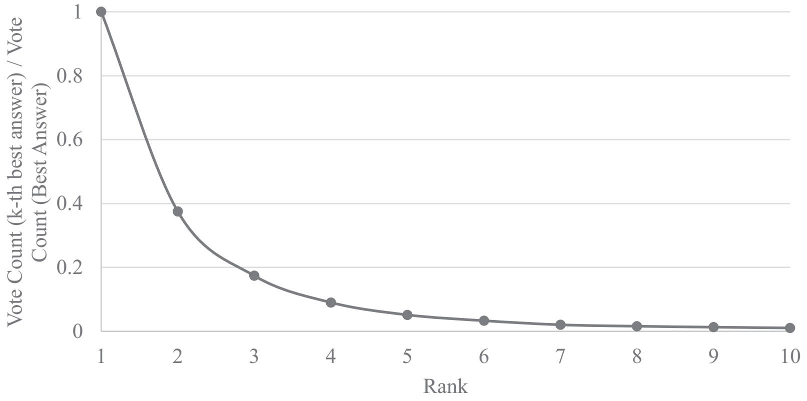


**Fig. 7.** Answer competition effect.

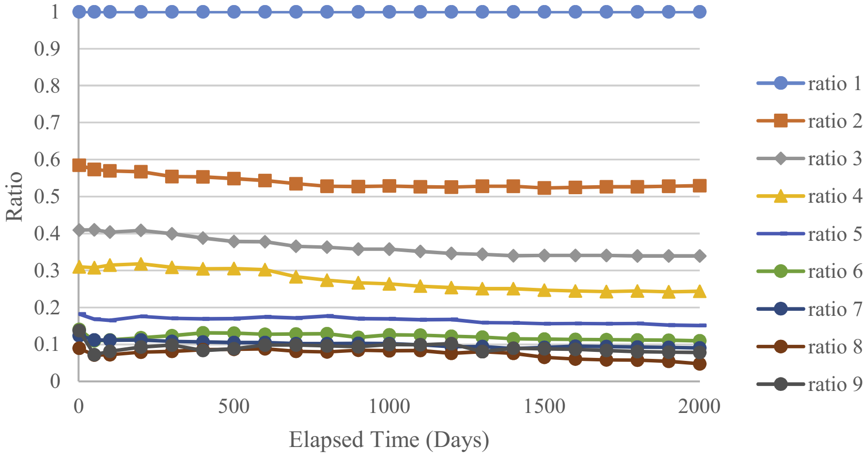


**Fig. 8.** Answer competition effect over time.

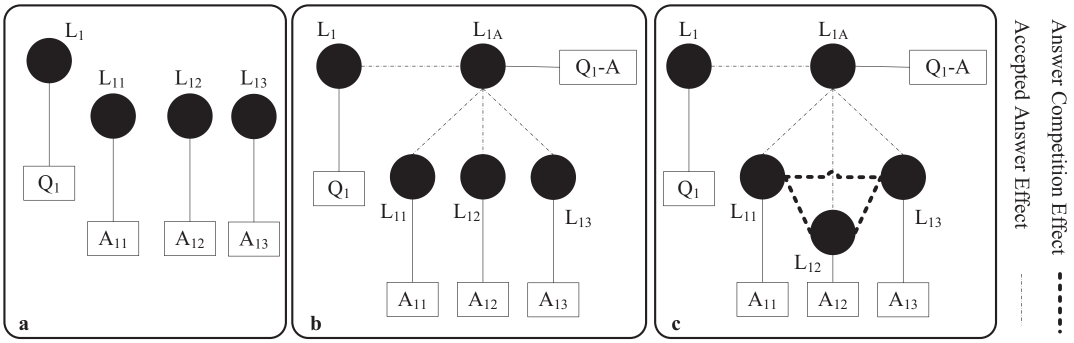


**Fig. 9.** Proposed relational classification models. Black circles indicate dependent/target variables.

reported as the *ratio i* in Fig. 8. According to this figure, even in the first day after the question submission, the answer competition effect is valid and accordingly we can exploit it in early detection of high quality answers.

Consider Table 1 which indicates five sample questions from Stack Overflow[12]. Questions $Q_1$ and $Q_2$ indicate the *Answer Competition Effect*. Specifically, these two questions have several answers, but it should be noticed that only a few number of them are high voted answers[13]. Questions $Q_3$ and $Q_4$ indicate the *Accepted Answer Effect*. While question $Q_3$ has only

[12] This data reflect the status of questions and their answers in our snapshot of Stack Overflow dataset and it may be different from the live version.

[13] In this paper, we consider questions and answers with more than ten votes as high voted posts. Please refer to Section 4.

**Table 1**
Five sample questions on Stack Overflow and their statistics.

| Question | # of Votes of question | # of Votes of accepted answer | Answer vote distribution |
|---|---|---|---|
| $Q_1$ | 48 | 35 | 35, 30, 16, 9, 6, 5, 5, 4, 4, 4, 1, 1, 1, 1, 0, 0, 0 |
| $Q_2$ | 33 | 37 | 37, 30, 15, 7, 5, 5, 4, 3, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2 |
| $Q_3$ | 28 | 31 | 31 |
| $Q_4$ | 6 | No accepted answer | 6, 1, 1, 0, 0 |
| $Q_5$ | 19 | 4 | 16, 10, 5, 5, 5, 4 |

$Q_1$ "Java lib or app to convert CSV to XML file?" $Q_2$ "How would you access Object properties from within an object method?" $Q_3$ "What is the meaning of the type safety warning in certain Java generics casts?" $Q_4$ "How to get started writing a code coverage tool?" $Q_5$ "What are the different methods to parse strings in Java?"

one answer, it is an accepted answer and because of the *Accepted Answer Effect* the number of votes on question $Q_3$ and its answer is higher than the question $Q_4$ which does not have an accepted answer.

Question $Q_5$ is a special case which the number of votes on its accepted answer is lower than other answers. However in this case, both the *Accepted Answer Effect* and the *Answer Competition Effect* are also visible. Specifically, $Q_5$ is a high-voted question because of *Accepted Answer Effect* and also compliant with *Answer Competition Effect*, only two answers are among high voted posts.

### 3.3. Relational classifier model using conditional random field

#### 3.3.1. Modeling accepted answer effect

Fig. 9b represents the model which utilizes the accepted answer effect to predict the high-quality posts. In this model, we introduce a new *dependent variable* (i.e. black circle connected to $Q_1 - A$) which indicates whether question $Q_1$ will get an accepted answer or not. As mentioned before, each black circle in Fig. 9 indicates a dependent variable associated with a binary label (i.e. true/false) which should be determined using the relational classifier.

According to Fig. 9b, the high quality post detection problem can be converted to simultaneously predict the quality of a question (i.e. $L_1$ in Fig. 9b), its question-answer node (i.e. $L_{1A}$ in Fig. 9b) and all of its related answers (i.e. $L_{11}$, $L_{12}$ and $L_{13}$ in Fig. 9b). To be more precise, the label $L_1$ indicates whether question $q_1$ is a high quality question or not, the label $L_{1A}$ indicates whether question $q_1$ will get an accepted answer or not, and finally, $L_{11}$, $L_{12}$ and $L_{13}$ indicate whether $A_{11}$, $A_{12}$ and $A_{13}$ are high quality answers or not.

As described in Section 3.1, the label $L_1$ and $A_{11}$ are dependent on the feature vector of the question $q_1$ and its answer $A_{11}$, respectively. However, according to the accepted answer effect mentioned in Section 3.2, these labels are dependent on the $L_{1A}$. Therefore, we expect better label prediction by considering both effects of post features and their label dependencies according to the accepted answer effect, simultaneously.

Two types of potential function (node potential function and edge potential function) are defined in our model to capture the mentioned dependencies.

Node potential function is responsible for capturing the dependency of the label of a post (e.g. $L_1$ and $L_{11}$) on its observed features (i.e. features of question $Q_1$ and $A_{11}$, respectively) and edge potential is responsible for modeling the label dependency among neighbor nodes in Fig. 9b.

Using log-linear potential functions, the conditional probability of the label set $L$ given the observed feature variable $X$ can be re-written as follows:

$$p(L|X) = \frac{1}{Z'}\exp(\sum_{Q_i \in Q}^{n} \psi_1(l_i, Q_i) + \sum_{A_{ij} \in A} \psi_2(l'_{ij}, A_{ij}) + \sum_{Q_iA \in Q-A} \psi_3(l''_i, Q_iA) + \sum_{e_{st} \in E_1} \psi_4(l_s, l_t) \tag{4}$$

In above equation, each potential function $\psi_1$, $\psi_2$, $\psi_3$, $\psi_4$ is represented by weighted combinations of feature vectors in the following form:

$$\psi_1(l_i, Q_i) = \sum_{m=1}^{M_1} \theta_m f_m(l_i, Q_i)$$

$$\psi_2(l'_{ij}, A_{ij}) = \sum_{m=1}^{M_2} \alpha_m g_m(l'_{ij}, A_{ij})$$

$$\psi_3(l''_i, Q_iA) = \sum_{m=1}^{M_3} \gamma_m h_m(l''_i, Q_iA)$$

$$\psi_4(l_s, l_t) = \sum_{m=1}^{M_4} \beta_m i_m(l_s, l_t)$$

**Table 2**
General statistics of topical datasets. The number of questions, answers, high-quality (HQ) questions and answers and also the ratio of high-quality posts are illustrated.

| Tag | # Question | # Answer | # HQ question | # HQ answers | HQ Q prop. | HQ A prop. |
|---|---|---|---|---|---|---|
| Java | 810,071 | 1,510,813 | 21,571 | 46,209 | 2.66% | 3.06% |
| Android | 106,861 | 165,252 | 1639 | 3400 | 1.53% | 2.06% |
| Swing | 48,723 | 81,271 | 576 | 1356 | 1.18% | 1.67% |
| Eclipse | 34,584 | 58,783 | 1186 | 2099 | 3.43% | 3.57% |
| Spring | 33,534 | 49,563 | 799 | 1463 | 2.38% | 2.95% |
| Hibernate | 24,008 | 35,994 | 561 | 973 | 2.34% | 2.70% |

Where $\theta$, $\alpha$, $\gamma$, and $\beta$ should be determined during training phase, *f*, *g*, *i*, and *h* represent features vectors and $M_1$, $M_2$, and $M_3$ represent the number of features for each potential function.

For node functions, we used the features introduced in Table 3 and for edge functions (i.e. $\psi_4$), three binary features have been introduced to capture the consistency of labels assigned to the corresponding nodes. The binary features associated with $\psi_4$ are defined as follows:

$$h_1(l_s, l_t) = \neg l_s \wedge \neg l_t$$

$$h_2(l_s, l_t) = \neg l_s \wedge l_t \vee \neg l_t \wedge l_s$$

$$h_3(l_s, l_t) = l_s \wedge l_t$$

In above equations, the feature $h_2$ indicates conflicting label assignment and $h_1$ and $h_3$ indicate homogeneous label assignment for two neighbor nodes. Specifically, if the classifier assigns true label (i.e. high-quality label) to the question $Q_1$ and assigns the false label (i.e. the question will not get an accepted answer) to node $Q_1 - A$ then only feature $h_2$ will be true and $h_1$ and $h_3$ will be false.

### 3.3.2. Modeling answer competition effect

Fig. 9c indicates the relational classification model which utilizes both the first and the second observation to predict high-quality posts. In this classification model, each two answers related to a specific question are connected to each other to simulate the answer competition effect.

According to the answer competition effect, we expect that only a few number (e.g. one or two) of answers to a particular question will be a high-quality answer. In other words, we expect that the least likely label assignment for a pair of connected answers ($A_{ij}$, $A_{ik}$) would be (high quality, high quality). We can complete the learning model introduced in Eq. (5), to include a new edge potential function which models the answer competition effect as indicated bellow:

$$p(L|X) = \frac{1}{Z'} \exp(\sum_{Q_i \in Q}^{n} \psi_1(l_i, Q_i) + \sum_{A_{ij} \in A} \psi_2(l'_{ij}, A_{ij}) + \sum_{Q_iA \in Q-A} \psi_3(l''_i, Q_iA) \sum_{e_{st} \in E_1} \psi_4(l_s, l_t) + \sum_{e_{uv} \in E_2} \psi_5(l_u, l_v) \tag{5}$$

We can determine the unknown parameters of the proposed model by minimizing the log-likelihood of the training data. After training the models, for a given instance of problem, approximate belief propagation can used to determine the most likely quality labels for all posts simultaneously.

## 4. Experiments

### 4.1. Data

The experiments in this section are designed to determine how accurate the proposed approaches are to identify the high-score posts.

We test our proposed models on data sets collected from a dump of Stack Overflow data[14]. For each dataset, we restrict the original data set to questions which have a particular tag (illustrated in Table 2). The selected tags are the most frequent Java related concepts and technologies in Stack Overflow. Table 2 indicates the number of questions, their answers and also the ratio of high-quality posts.

For each question, we have only 1.9 answers in average which makes our relational classification model entirely feasible. In our experiments, for each dataset, we randomly choose 80% of questions and their associated answers as the training set and use the rest as the test set. Precision, Recall, and the F-measure are used to evaluate the proposed models.

[14] Data Dump can be downloaded from http://archive.org/details/stackexchange

**Table 3**
list of features for $\psi_1$, $\psi_2$ and $\psi_3$ potential functions and their description.

| Function name | Feature |
|---|---|
| $\psi_1$ | $f_1$: The length of the question body |
| | $f_2$: The length of the question title |
| | $f_3$: Does the question contains source code? (yes/no) |
| | $f_4$: The reputation of questioner |
| | $f_5$: Number of previous questions of questioner |
| $\psi_2$ | $g_1$: The length of the answer |
| | $g_2$: Does the answer contains source code? (yes/no) |
| | $g_3$: The Jaccard textual overlap score of the question and answer |
| | $g_4$: The reputation of answerer |
| | $g_5$: Number of previous answers of answerer |
| $\psi_3$ | $h_1$: Number of previous questions of questioner with accepted answer |
| | $h_2$: Number of comments of question one hour after question creation |
| | $h_3$: Number of comments of all submitted answers one hour after question posted |
| | $h_4$: Number of submitted answers one hour after question posted |

### 4.2. Experiments setup

In our experiments, we compared the predication performance of following algorithms:

1. Model A (i.e. the logistic regression classifier).
2. Model B (i.e. the relational classifier which consider *Accepted Answer Effect*).
3. Model C (i.e. the relational classifier which consider *Answer Competition Effect*).
4. Model B&C (i.e. the relational classifier model which consider both *Accepted Answer Effect* and *Answer Competition Effect*).
5. The proposed algorithm in Yao et al. (2015) for high-quality post detection on CQA. We use the best-proposed algorithm in Yao et al. (2015) (as one of the most recent and relevant works to our research) as our baseline model. The main idea of this study is that the voting score of an answer is strongly correlated with that of its question. According to this intuition, they proposed a joint voting prediction algorithm to predict the high-quality question and answers in community question answering websites. In their learning model, they used a voting consistency factor which enforces the co-relation between the number of votes on questions and answers.

Table 3 indicates three potential functions and their associated features. Function $\psi_1$ is the potential function which predicts the quality of the question. Specifically, $f_1$ and $f_2$ indicate whether the question is explained enough or not. $f_3$ indicates whether the question contains source code or not. According to some recent research (Bhat, Gokhale, Jadhav, Pudipeddi, & Akoglu, 2014), this feature is a strong signal for the question quality on Stack Overflow. Besides, the expertise level of the questioner is also an outstanding signal of the quality of questions (Asaduzzaman, Mashiyat, Roy, & Schneider, 2013) which is modeled by the features $f_4$ and $f_5$.

Function $\psi_2$ is the potential function which predicts the quality of the answer. Specifically, $g_1$ indicates whether the length of an answer is suitable or not. The feature $g_2$ indicates whether the answer contains source code or not. The feature $g_3$ measures the similarity of question and answer. The features $g_4$ and $g_5$ measure the expertise and maturity of the answerer. It is worth mentioning that instead of using complicated features like the vocabulary gap between questions and answers (Momtazi & Klakow, 2015), we only used the simple vocabulary overlap in this paper.

Function $\psi_3$ is the potential function which predicts whether the associated question will receive an accepted answer or not. Specifically, $h_1$ indicates the number of previous questions accepted by the same questioner. Features $h_2$, $h_3$ and $h_4$ show the interest of community in submitted question which is intuitively proportional to the chance of the question to get an accepted answer.

## 5. Results

An extensive set of experiments were conducted on the Stack Overflow test collections (i.e. Table 2) to address the following questions:

- How efficient is the Model A in detection of high quality posts?
- Can the *accepted answer effect* improve the quality of high quality question/answer detection?
- Can the *answer competition effect* improve the quality of high quality question/answer detection?
- How efficient is the proposed models in comparison with the method proposed in Yao et al. (2015)?

Figs. 10 and 11 depict the performance of *Model A* to predict high-quality questions and answers, respectively. The performance of high-quality question prediction is higher than high-quality answer prediction. Additionally, for some datasets with general topics (e.g. “eclipse”) the overall performance is higher than the datasets with more specific topics (e.g. “Android” and “swing”).

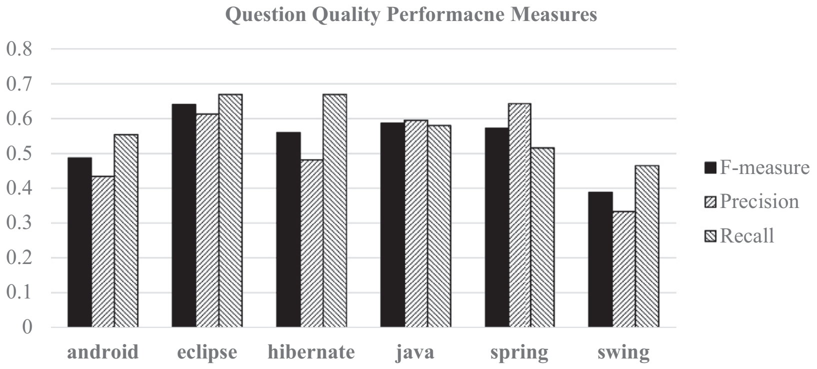


**Fig. 10.** *Model A* High-Quality Question Prediction. The precision, recall and f-measure on Stack Overflow datasets are illustrated.

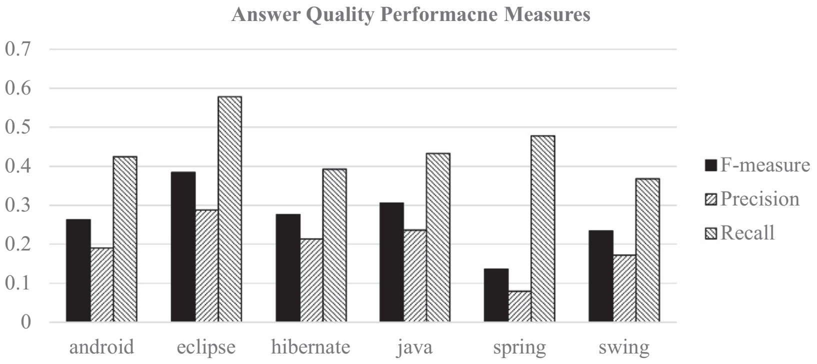


**Fig. 11.** *Model A* High-Quality Answer Prediction. The precision, recall and f-measure on Stack Overflow datasets are illustrated.

**Table 4**
High-quality post detection, independent versus joint prediction models. Comparisons are based on F-measure. The *and **symbols indicate statistical significance at 0.9 confidence interval against Model A and Model Yao et al. (2015), respectively.

| Post Type | Dataset | Model A | Model B | Model C | Model B&C | Model Yao et al. (2015) |
|---|---|---|---|---|---|---|
| Question | Android | 0.487 | 0.523 | 0.487 | **0.524*** | 0.519 |
| | Eclipse | 0.640 | 0.658 | 0.640 | **0.667*** | 0.660 |
| | Hibernate | 0.560 | 0.566 | 0.560 | **0.572*, **** | 0.559 |
| | Java | 0.587 | 0.639 | 0.587 | **0.647*** | 0.641 |
| | Spring | 0.572 | 0.629 | 0.572 | **0.634*, **** | 0.621 |
| | Swing | 0.388 | **0.404*** | 0.388 | 0.399* | 0.399 |
| Answer | Android | 0.263 | 0.283 | **0.320*, **** | 0.304 | 0.281 |
| | Eclipse | 0.384 | 0.391 | 0.423 | **0.424*, **** | 0.394 |
| | Hibernate | 0.276 | 0.265 | **0.309*, **** | 0.288 | 0.270 |
| | Java | 0.306 | 0.323 | **0.447*, **** | 0.414*, ** | 0.312 |
| | Spring | 0.136 | 0.133 | 0.144 | **0.148*, **** | 0.139 |
| | Swing | 0.234 | 0.219 | 0.254 | **0.260*, **** | 0.238 |

In next experiment, we compare the prediction performance of the *Model A* with the joint label prediction models (i.e. *Model B*, *Model C* and *Model B&C*). In *Model B* and *Model C*, we utilize the accepted answer effect and the answer competition effect, respectively. We exploit both these effects in *Model B&C*.

Table 4 indicates the F-score result of these models on different test collections for both high quality question and answer detection problems. According to Table 4 *Model B* substantially improves the F-score of high quality question prediction in all test collections. Besides, the *Model B* improves the F-score of high-quality answer prediction in some test collections.

The improvement of high-quality question prediction using the *Model B* can be explained by the fact that this model selects a question as a high-quality candidate if it finds a high-quality answer candidate simultaneously. As a result, in overall, this property can improve the quality prediction performance for both questions and answers.

In order to better analyze the behavior of *Model B*, Figs. 12 and 13 comparing *Model A* and *Model B* in terms of precision and recall measures can help.

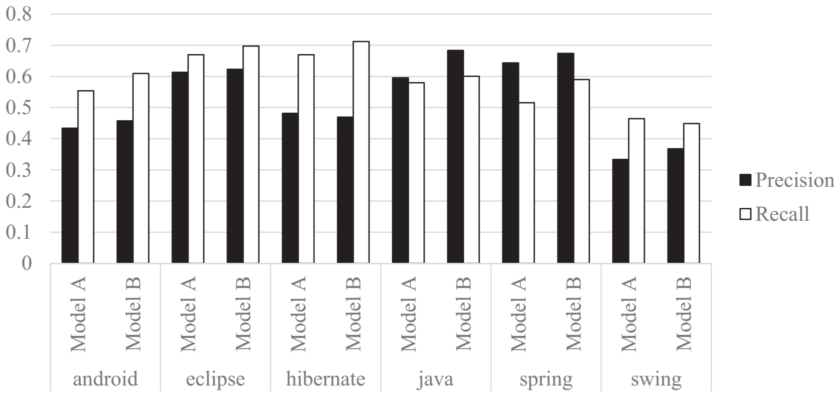


**Fig. 12.** *Model A* versus *Model B*. High-Quality Question Prediction. The precision and recall on Stack Overflow datasets are illustrated.

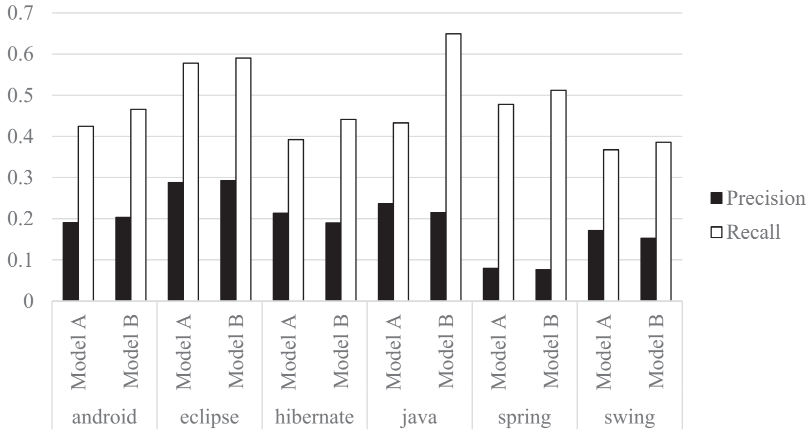


**Fig. 13.** *Model A* versus *Model B*. High-Quality Answer Prediction. The precision and recall on Stack Overflow datasets are illustrated.

Fig. 12 indicates that *Model B* improves the recall score but retains the precision at the same level in comparison with *Model A* for high-quality question prediction. In average the *Model B* improves the precision and recall of high quality question prediction, **5.6%** and **5.9%**, respectively.

Fig. 13 indicates that although *Model B* decreases the precision score of high-quality answer prediction, it can significantly improve the recall. In average, the *Model B* reduces the precision of high-quality answer detection, **4.2%**, but improve the recall of high-quality answer detection **13.9%**.

As mentioned before, *Model C* considers the answer competition effect. As a result, based on Table 4, the *Model C* does not have any impact on the performance of high-quality question detection. However, this model can significantly improve the F-measure (in average 18.6% improvement) of high-quality answer detection. Fig. 14 shows the precision and recall of *Model A* versus *Model C* for high quality answer detection. According to that, although *Model C* decreases the recall measure, it can significantly improve the precision. The improvement of precision can be explained by the fact that *Model C* selects only a few number of high-quality answers for a given question. Although, this behavior can cause the reduction of the recall but the significant improvement of precision, improves the F-score of *Model C* in comparison with *Model A*.

In our next experiment, we compare the performance of *Model B&C* (i.e. the model which consider both effects) with the other joint prediction models (i.e. *Model B* and *Model C*). According to Table 4, for the high-quality question prediction, the *Model B&C* dominates all other methods in terms of F-score. Besides, for the high-quality answer prediction, the performance is comparable with *Model C* and is the best performing method in comparison with other methods in three test collections. Figs. 15 and 16 compares the precision and recall of joint prediction model for all test collections.

According to Fig. 15, the *Model B&C* slightly improves the precision (i.e. 0.4%) of high-quality question detection and also improves the recall (i.e. 1.4%) of high-quality question detection. The accepted answer effect can explain this improvement. The *Model B&C* considers this effect and accordingly it can detect high-quality answers more precisely. As a result, this precision growth can cause an improvement of high-quality questions via accepted answer effect.

According to Fig. 16, in average, the *Model C* is the best performing method regarding precision for high-quality answer detection, while the *Model B* is the best performing method regarding recall. Actually, the *Model B&C* makes a balance between the precision and the recall. Specifically, The *Model B&C* significantly improves the F-measure in comparison with

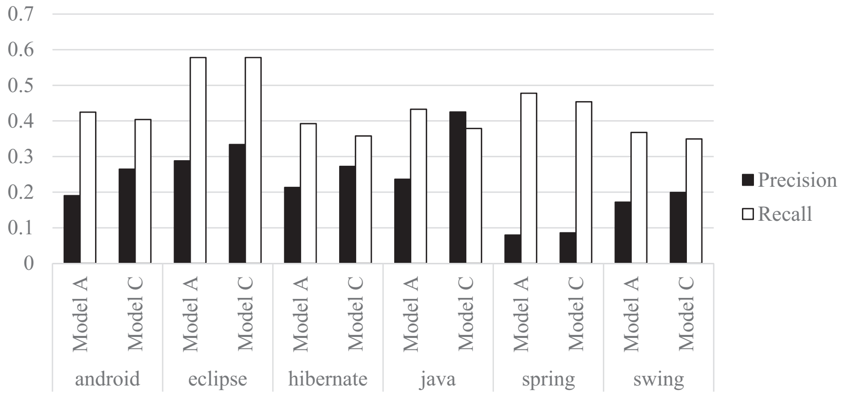


**Fig. 14.** *Model A* versus *Model C*. High-Quality Answer Prediction. The precision and recall on Stack Overflow datasets are illustrated.

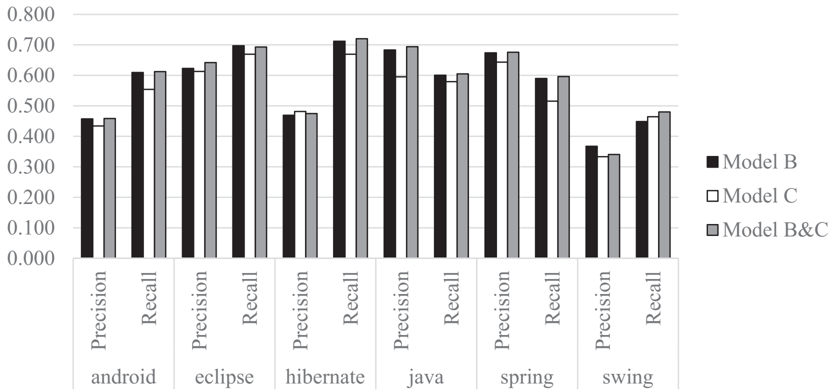


**Fig. 15.** Comparison of joint prediction models. High-Quality Question Prediction. The precision and recall on Stack Overflow datasets are illustrated.

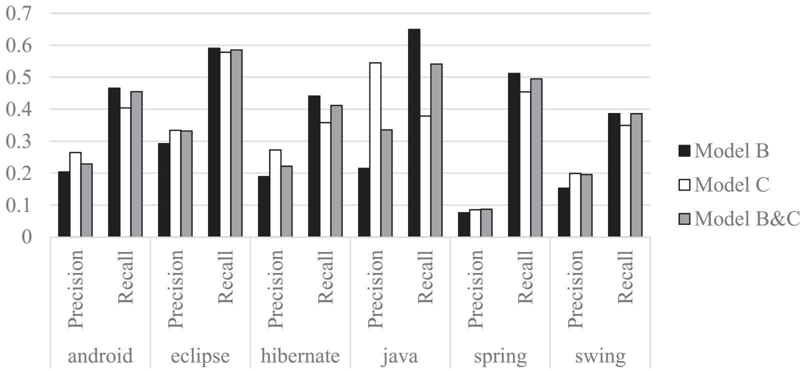


**Fig. 16.** Comparison of joint prediction models. High-Quality Answer Prediction. The precision and recall on Stack Overflow datasets are illustrated.

*Model B* while has a comparable F-measure compared with *Model C*. The results of this experiments indicate that there is a trade-off between the accepted answer and the answer competition effects. The accepted answer works for the recall while the answer competition effect improves the precision. In some of our test collections, the combinations of these methods work better than each one separately.

In our last experiment, we compare the prediction performance of our proposed models with the proposed method in Yao et al. (2015). As mentioned before, Yao et al. (2015) utilize the question-answer vote correlation to adjust the vote prediction and accordingly improve the high-quality post detection. They proposed an optimization formulation which considers voting consistency between a question and the average number of votes of its answers.

According to the discussion in Section 3.2, the accepted answer effect has a similar behavior with the voting consistency because, if a question has (or will get) an accepted answer then its number of votes and the number of votes of the accepted answer increase while for questions without accepted answer the number of votes on the question and its answers is

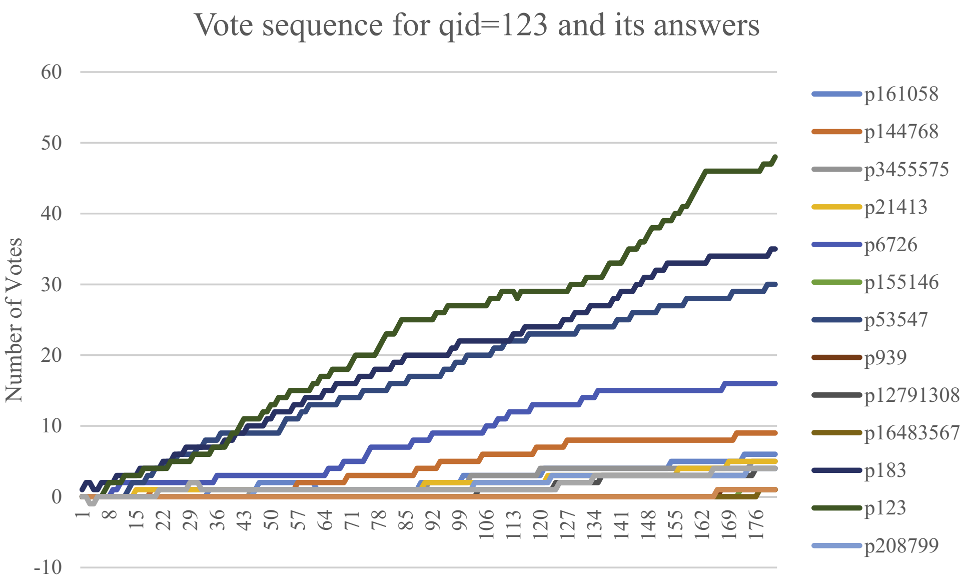


**Fig. 17.** Vote sequence for question id = 123 and its associated answers.

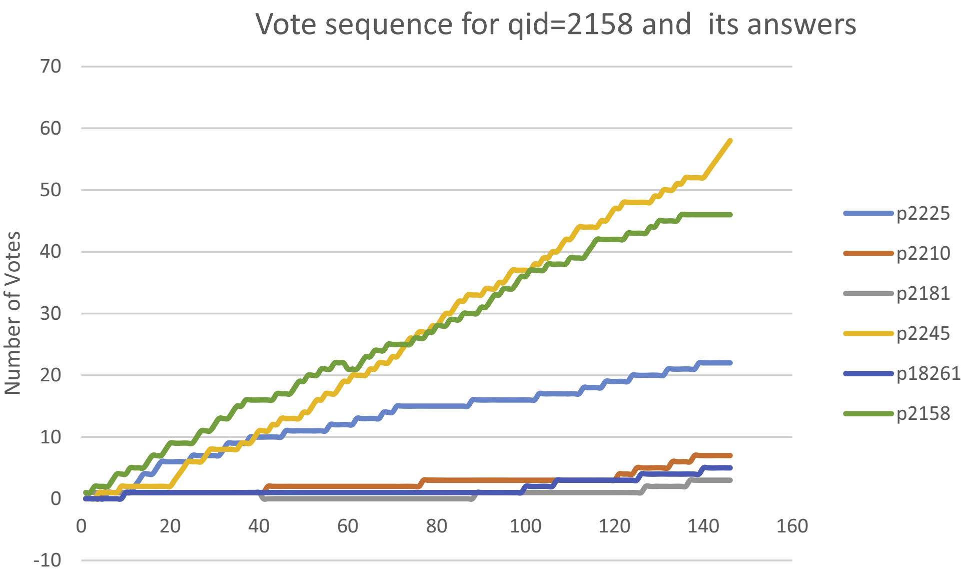


**Fig. 18.** Vote sequence for question id = 2158 and its associated answers.

getting lower. Similar to Yao et al. (2015), *Model B* considers the vote consistency between a question and its answer. However, the *Model B &C* considers both the accepted answer effect (similar to voting consistency) and the answer competition effect simultaneously. Table 4 indicates that *Model B* and vote consistency method (Yao et al., 2015) have almost the same performance on all data collections, but *Model B &C* surpass the vote consistency approach by a large margin. Our proposed relational learning model is flexible enough to consider both the accepted answer effect (similar to voting consistency) and the answer competition effect simultaneously.

To be more specific, the methods proposed in Yao et al. (2015) use the correlation between the number of votes of a question and the *average* number of votes of its associated answers to adjust the number of votes of both the question and its associated answers.

However, according to the answer competition effect this correlation is only valid for few number of answers. As a result, the limitation of the model proposed in Yao et al. (2015) causes lower precision and recall in comparison to our models. To better describe the behavior of our model and the model proposed in Yao et al. (2015), please consider Figs. 17 and 18. These figures indicate the vote sequence provided by users over time for two sample questions and their related answers. In other words, these figures indicate the trend of votes for question 123 and 2158 of Stack Overflow as well as their associated answers over time (each new vote over time is mapped to a point on horizontal axis). According to these samples, there is a strong correlation between number of votes of a question and only few number of its associated answers over time. While the constraints in models proposed in Yao et al. (2015) force to use the correlation between the number of votes of a question and the *average* number of votes of its answers, our model uses the quality label of the most probable best answer to adjust the label of the question. The pattern indicated in Figs. 17 and 18 is observable in most cases and we just select two examples to describe it.

## 6. Conclusion

Early Detection of high-quality content in CQA is a crucial step in improving the quality of service, site reputation, user experience and user engagement. In this paper, we presented two straightforward and intuitive observations on CQAs data which can help significantly improve the performance of high-quality content detection problem. The first observation considers the accepted answer effect and the second observation considers the answer competition effect. We explained how we utilized these two observations to improve the high-quality content prediction. A relational classification method was proposed in this paper which modeled observations. An extensive set of experiments on real datasets extracted from Stack Overflow proves the efficiency of the proposed model in comparison with the state-of-the-art classification model and a strong baseline method on high-quality content detection on Stack Overflow. Our experiments indicate that considering the dependency between quality of questions and answers can significantly improve the performance of prediction task. In addition, both the accepted answer effect and the answer competition effect are beneficial to improve the prediction task in a short time after submission of the content on Stack Overflow. In our next steps, we tend to extend our model to take into consideration the users behavior in high-quality content detection problem. In addition, we are planning to investigate the effect of high-quality content prediction on the performance of expert finding and expert profiling tasks on Stack Overflow.

## References

Agichtein, E., Castillo, C., Donato, D., Gionis, A., & Mishne, G. (2008). Finding high-quality content in social media. In *Proceedings of the 2008 international conference on web search and data mining*. In *WSDM '08* (pp. 183–194).

Ahasanuzzaman, M., Asaduzzaman, M., Roy, C. K., & Schneider, K. A. (2016). Mining duplicate questions in stack overflow. In *Proceedings of the 13th international conference on mining software repositories*. In *MSR '16* (pp. 402–412).

Anderka, M., Stein, B., & Lipka, N. (2012). Predicting quality flaws in user-generated content: The case of wikipedia. In *Proceedings of the 35th international ACM SIGIR conference on research and development in information retrieval*. In *SIGIR '12* (pp. 981–990).

Arora, P., Ganguly, D., & Jones, G. J. (2016). Nearest neighbour based transformation functions for text classification: A case study with stackoverflow. In *Proceedings of the 2016 ACM international conference on the theory of information retrieval*. In *ICTIR '16* (pp. 299–302).

Arora, P., Ganguly, D., & Jones, G. J. F. (2015). The good, the bad and their kins: Identifying questions with negative scores in stackoverflow. In *2015 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM)* (pp. 1232–1239).

Asaduzzaman, M., Mashiyat, A. S., Roy, C. K., & Schneider, K. A. (2013). Answering questions about unanswered questions of stack overflow. In *Proceedings of the 10th working conference on mining software repositories*. In *MSR '13* (pp. 97–100).

Bhat, V., Gokhale, A., Jadhav, R., Pudipeddi, J. S., & Akoglu, L. (2014). Min(e)d your tags: Analysis of question response time in stackoverflow. In *2014 IEEE/ACM international conference on advances in social networks analysis and mining, ASONAM 2014, Beijing, China, August 17–20, 2014* (pp. 328–335).

Bian, J., Liu, Y., Zhou, D., Agichtein, E., & Zha, H. (2009). Learning to recognize reliable users and content in social media with coupled mutual reinforcement. In *Proceedings of the 18th international conference on world wide web* (pp. 51–60).

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research, 3*, 993–1022.

Blooma, M. J., Goh, D. H., & Chua, A. Y. (2012). Predictors of high quality answers. *Online Information Review, 36*(3), 383–400.

Bouguessa, M., & Romdhane, L. B. (2015). Identifying authorities in online communities. *ACM Transactions on Intelligent Systems and Technology, 6*(3), 30:1–30:23.

Calefato, F., Lanubile, F., Marasciulo, M. C., & Novielli, N. (2015). Mining successful answers in stack overflow. In *Proceedings of the 12th working conference on mining software repositories*. In *MSR '15* (pp. 430–433).

Chua, A. Y., & Banerjee, S. (2015). Answers or no answers. *Journal of Information Science, 41*(5), 720–731.

Dalip, D. H., Gonçalves, M. A., Cristo, M., & Calado, P. (2016). A general multiview framework for assessing the quality of collaboratively created content on web 2.0. *Journal of the Association for Information Science and Technology*.

van Dijk, D., Tsagkias, M., & de Rijke, M. (2015). Early detection of topical expertise in community question answering. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. In *SIGIR '15* (pp. 995–998).

Figueiredo, F., Pinto, H., BeléM, F., Almeida, J., GonçAlves, M., Fernandes, D., & Moura, E. (2013). Assessing the quality of textual features in social media. *Information Processing & Management, 49*(1), 222–247.

Figueroa, A., Gómez-Pantoja, C., & Herrera, I. (2016). Search clicks analysis for discovering temporally anchored questions in community question answering. *Expert Systems with Applications, 50*, 89–99.

Figueroa, A., & Neumann, G. (2016). Context-aware semantic classification of search queries for browsing community question answering archives. *Knowledge-Based Systems, 96*, 1–13.

Flekova, L., Ferschke, O., & Gurevych, I. (2014). What makes a good biography? Multidimensional quality analysis based on wikipedia article feedback data. In *Proceedings of the 23rd international conference on world wide web*. In *WWW '14* (pp. 855–866).

Fu, H., Niu, Z., Zhang, C., Yu, H., Ma, J., Chen, J., … Liu, J. (2016). Aselm: Adaptive semi-supervised {ELM} with application in question subjectivity identification. *Neurocomputing, 207*, 599–609.

Hsu, C.-F., Khabiri, E., & Caverlee, J. (2009). Ranking comments on the social web. In *Proceedings of the 2009 international conference on computational science and engineering - volume 04* (pp. 90–97).

Kayes, I., Kourtellis, N., Quercia, D., Iamnitchi, A., & Bonchi, F. (2015). The social world of content abusers in community question answering. In *Proceedings of the 24th international conference on world wide web*. In *WWW '15* (pp. 570–580).

Li, B., Jin, T., Lyu, M. R., King, I., & Mak, B. (2012). Analyzing and predicting question quality in community question answering services. In *Proceedings of the 21st international conference on world wide web* (pp. 775–782).

Liu, B., Feng, J., Liu, M., Hu, H., & Wang, X. (2015). Predicting the quality of user-generated answers using co-training in community-based question answering portals. *Pattern Recognition Letters, 58*(C), 29–34.

Liu, Y., Bian, J., & Agichtein, E. (2008). Predicting information seeker satisfaction in community question answering. In *Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval* (pp. 483–490).

Lu, Y., Tsaparas, P., Ntoulas, A., & Polanyi, L. (2010). Exploiting social context for review quality prediction. In *Proceedings of the 19th international conference on world wide web*. In *WWW '10* (pp. 691–700).

McCallum, A. K. (2002). *Mallet: A machine learning for language toolkit*. Http://mallet.cs.umass.edu

Molino, P., Aiello, L. M., & Lops, P. (2016). Social question answering: Textual, user, and network features for best answer prediction. *ACM Transactions on Information Systems, 35*(1), 4:1–4:40.

Momeni, E., Tao, K., Haslhofer, B., & Houben, G.-J. (2013). Identification of useful user comments in social media: A case study on flickr commons. In *Proceedings of the 13th ACM/IEEE-CS joint conference on digital libraries* (pp. 1–10).

Momtazi, S., & Klakow, D. (2015). Bridging the vocabulary gap between questions and answer sentences. *Information Processing & Management, 51*(5), 595–615.

Neshati, M., Beigy, H., & Hiemstra, D. (2014). Expert group formation using facility location analysis. *Information Processing & Management, 50*(2), 361–383.

Neshati, M., Hashemi, S. H., & Beigy, H. (2014). Expertise finding in bibliographic network: Topic dominance learning approach. *IEEE Transactions on Cybernetics, 44*(12), 2646–2657.

Neshati, M., Hiemstra, D., Asgari, E., & Beigy, H. (2014). Integration of scientific and social networks. *World Wide Web, 17*(5).

Ostrom, E. (1990). *Governing the commons : The evolution of institutions for collective action*. Cambridge University Press.

Stack Overflow (2016). *Stack overflow help center*. http://stackoverflow.com/help. [Online; accessed 3-July-2016]

Pal, A., Farzan, R., Konstan, J. A., & Kraut, R. E. (2011). Early detection of potential experts in question answering communities. In *Proceedings of the 19th international conference on user modeling, adaption, and personalization*. In *UMAP'11* (pp. 231–242).

Pal, A., Harper, F. M., & Konstan, J. A. (2012). Exploring question selection bias to identify experts and potential experts in community question answering. *ACM Transactions on Information Systems, 30*(2), 10:1–10:28.

Patil, S., & Lee, K. (2016). Detecting experts on quora: By their activity, quality of answers, linguistic characteristics and temporal behaviors. *Social Network Analysis and Mining, 6*(1), 1–11.

Pelechrinis, K., Zadorozhny, V., Kounev, V., Oleshchuk, V., Anwar, M., & Lin, Y. (2015). Automatic evaluation of information provider reliability and expertise. *World Wide Web, 18*(1), 33–72.

Ponzanelli, L., Mocci, A., Bacchelli, A., & Lanza, M. (2014). Understanding and classifying the quality of technical forum questions. In *Proceedings of the 2014 14th international conference on quality software* (pp. 343–352).

Ponzanelli, L., Mocci, A., Bacchelli, A., Lanza, M., & Fullerton, D. (2014). Improving low quality stack overflow post detection. In *2014 IEEE international conference on software maintenance and evolution (ICSME)* (pp. 541–544). IEEE.

Ravi, S., Pang, B., Rastogi, V., & Kumar, R. (2014). Great question! question quality in community q&a. In *Proceedings of the eighth international conference on weblogs and social media, ICWSM 2014, Ann Arbor, Michigan, USA, June 1–4, 2014.*.

Srba, I., & Bielikova, M. (2016). A comprehensive survey and classification of approaches for community question answering. *ACM Transactions on the Web, 10*(3), 18:1–18:63.

Suryanto, M. A., Lim, E. P., Sun, A., & Chiang, R. H. L. (2009). Quality-aware collaborative question answering: Methods and evaluation. In *Proceedings of the second ACM international conference on web search and data mining*. In *WSDM '09* (pp. 142–151).

Toba, H., Ming, Z.-Y., Adriani, M., & Chua, T.-S. (2014). Discovering high quality answers in community question answering archives using a hierarchy of classifiers. *Information Sciences, 261*, 101–115.

Wei, W., Ming, Z., Nie, L., Li, G., Li, J., Zhu, F., … Luo, C. (2016). Exploring heterogeneous features for query-focused summarization of categorized community answers. *Information Sciences, 330*, 403–423.

Yan, Z., & Zhou, J. (2015). Optimal answerer ranking for new questions in community question answering. *Information Processing & Management, 51*(1), 163–178.

Yao, Y., Tong, H., Xie, T., Akoglu, L., Xu, F., & Lu, J. (2015). Detecting high-quality posts in community question answering sites. *Information Sciences, 302*(C), 70–82.