Predicting the Best Answers for Questions on Stack Overflow

Jun-Wei Lin
Dept. of Informatics
junwel1@uci.edu

Tzu-Chi Lin
Dept. of Computer Science
tzucl2@uci.edu

Peter Schaedler
Dept. of Computer Science
pschaedl@uci.edu

Abstract

Question-and-answering communities are becoming widely used due to the useful information they provide. However, a few issues often exist, such as the fact that many questions are left without a best answer being selected. In this paper, we proposed a set of features and machine learning models to predict the best answers for questions on Stack Overflow. Two kinds of features, shallow features and semantic features, are used and analyzed in this work. We evaluated our classifiers on a relatively new and practical metric, question-based accuracy. The experiment results show that our techniques improved the baseline by 8.52%.

1 Introduction

Stack Overflow is one of the most popular question-answering platforms for software engineers. The high-quality answers usually appear at the top when we search programming questions on Google. Nevertheless, many questions on Stack Overflow have no best answers because none of the answers are selected or marked as an accepted answer. For instance, 45% of the questions labeled with "Python" have no user-selected answers (StackExchange, 2018). Generally, the best answer for a question comes from: (1) the one marked as accepted by the author of the question; or (2) the answer with the most upvotes (i.e., the highest score). In this paper, we propose to predict the answer that will get the highest score, because the dataset we used has no information about user-selected answers. We believe this study can benefit software developers by saving their effort in searching for or writing replies, and further improve their productivity.

Question-answering is a widely-addressed topic in NLP and ML research (Yang et al., 2011; Lezina and Kuznetsov, 2013; Calefato et al., 2016; Burel et al., 2012; Blooma et al., 2010; Shah and Pomerantz, 2010; Tian et al., 2013; Jenders et al., 2016; Elalfy et al., 2015). Prior work provided not only in-depth analysis and useful criteria for high-quality answers, but also empirical study on different machine learning models for various predictions. However, specifically for Stack Overflow, one of the largest communities for software engineers, there is little research on predicting the best answer for questions on the forum. The reason could be the absence of appropriate datasets. Recently, a considerable amount of official data are released by Stack Overflow (StackOverflow, 2016), which provides us chances to evaluate the effectiveness of different ML and NLP techniques specifically on programming-related question-answering.

In this paper, we applied various ML and NLP techniques to predict the best answer for questions labeled "Python" on Stack Overflow with the official data set (StackOverflow, 2016). Given a question and the user-generated answers, well-studied feature engineering methods, including shallow (statistical) features (e.g., content length) and semantic features (e.g., question-answer similarity) were applied. Also, mature machine learning models such as random forest and logistic regression were used to predict the answer that will get the highest score (i.e., the most upvotes). The experimental results show that our best combination of features and models improves the baseline by 8.52%. We also analyzed the importance of the features used in our study to help better distinguish quality answers.

2 Related work

Community-based question-answering is widely addressed in NLP and ML research. Yang et al. (Yang et al., 2011) analyzed Yahoo! Answers forum to predict whether new questions will be answered. Jenders et al. (Jenders et al., 2016) presented machine learning models that predict ac-

cepted answers to questions in a MOOC (Massive Open Online Course) forum by using historical forum data. In terms of the criteria or features used to judge quality of answers, literature such as (Burel et al., 2012; Blooma et al., 2010; Shah and Pomerantz, 2010) pointed out that they should be multifaceted, including social and textual aspects, and the relationship between questions and answers. While these papers provide practical guidelines to our feature engineering, model selection, and evaluation, they didn't use data from Stack Overflow, the platform we are interested in this study.

Particularly for Stack Overflow, Lezina and Kuznetsov (Lezina and Kuznetsov, 2013) predicted whether questions will be closed by adopting both content-based (specific to the posts) and reputation-based (specific to the user who answers the question) features. Also, Calefato et al. (Calefato et al., 2016) found that the data on Stack Overflow can be used to train a classifier to help automatic migration of crowd-sourced knowledge from legacy forums to modern Q&A sites. Finally, Tian et al. (Tian et al., 2013) and Elalfy et al. (Elalfy et al., 2015) proposed a set of features and ML models to predict the best answer on Stack Overflow, i.e., if the answer is going to be selected as the best or not. Although these two papers are highly related to our study, ours is different from prior work in two aspects. First, we specifically care about programming questions (i.e., the questions labeled with "Python"), while prior work doesn't differentiate types of questions. Answers for programming questions may have more code snippets and thus pose more challenges than general questions. Second, the metric we used in this study is question-based accuracy, i.e., the ratio that whether the question is correctly answered, while prior work only evaluates their techniques on answer-based accuracy, i.e., if the answer is correctly classified as the best or not. Improving question-based accuracy is more difficult. For instance, consider a question with four answers and one of them is the best. A binary classifier predicting all the four answers as non-best gets accuracy $3/4 = 75\%$. On the other hand, the accuracy for our classifier is either $1/1 = 100\%$ or $0/1 = 0\%$, depending on whether it correctly selects the best answer. We believe our metric is more practical and challenging.

3 Proposed Approach

We first extract features from the questions and corresponding answers, and represent the answers as real-value vectors. Next, we feed the answer vectors into different machine learning models, and train classifiers to predict the probability that an answer is the best for a question. Here we define the best answer for a question as the answer with the highest score. In terms of features, we consider both shallow features (e.g., length of content and code snippets, number of links, and reputation of the answerer) and semantic features (e.g., question-answer similarity and answer-answer similarity.)

3.1 Shallow Features

For the answers, we considered the shallow features of the content. By shallow, we mean the numerical or statistical values derived from the content. We introduced five shallow features as follows:

- **TotalLength**: The length (word counting) of the content, including the text and code snippets. Longer and more detailed answers are more likely to be chosen as best answers. More detailed answers are likely to include links to outside resources (e.g., tutorial or API documents), and may have longer code snippets. However, the most useful answers may also be those with shorter, more concise code snippets, so our models may find that to be the case instead.

- **LinkCount**: The number of hyperlinks in the answer.

- **CodeLength**: Word counting of the code snippets.

- **PostOrder**: The chronological order of the answer. First-posted answer is set as 1, second answer is 2, and so on. Intuitively, we expected the answers submitted most quickly to have a higher chance of being selected as the best answer, since there is more time for users to look at them and upvote. Answers submitted later are therefore shown to a smaller audience and may not have as high of scores.

- **Reputation**: The reputation score of the person writing the answer. The score of a user is computed by aggregating all upvotes she got from all the answers (in the training dataset) she delivered. The intuition behind this feature is that prestigious people usually produce good answers.

3.2 Semantic Features

Semantic similarities among questions and answers may influence the quality of the answers. In

Questions

	Id	OwnerUserId	CreationDate	Score	Title	Body
0	469	147.0	2008-08-02T15:11:16Z	21	How can I find the full path to a font from it...	<p>I am using the Photoshop's javascript API t...
1	502	147.0	2008-08-02T17:01:58Z	27	Get a preview JPEG of a PDF on Windows?	<p>I have a cross-platform (Python) applicatio...
2	535	154.0	2008-08-02T18:43:54Z	40	Continuous Integration System for a Python Cod...	<p>I'm starting work on a hobby project with a...

Answers

	Id	OwnerUserId	CreationDate	ParentId	Score	Body
0	497	50.0	2008-08-02T16:56:53Z	469	4	<p>open up a terminal (Applications->Utilit...
1	518	153.0	2008-08-02T17:42:28Z	469	2	<p>I haven't been able to find anything that d...
2	536	161.0	2008-08-02T18:49:07Z	502	9	<p>You can use ImageMagick's convert utility f...

Figure 1: A few sample entries from the dataset

this paper, we computed similarity between two documents by a series of transformations: bag-of-words, tf-idf (term frequency-inverse document frequency), and LSA (Latent Semantic Analysis). The cosine similarity between two documents in a latent vector space is used. We considered the following semantic features:

- **SimToQ**: The similarity between an answer and the question. We want to know if a good answer is similar to the original question.
- **MaxSimToA**: The maximal similarity between an answer and other answers of the original question. We want to know if a good answer is similar to other candidate answers.
- **MinSimToA**: The minimal similarity between an answer and other answers of the original question.

3.3 Model

We used logistic regression, random forest and XGBoost (gradient boosted decision trees) as our models, and evaluated them in Section 4.

4 Experiments

4.1 Dataset

We used the official dataset provided by Stack Overflow (StackOverflow, 2016). The dataset primarily comes with two parts: (1) questions, consisting of a title, body, creation date, score, and owner ID; and (2) answers, consisting of a body, creation date, score, question ID, and owner ID, as illustrated in Figure 1. An issue with the dataset is that not all the questions have corresponding answers, and the corresponding answers may be indistinguishable (i.e. having the same score). Another issue is that the body of the questions and answers is HTML formatted, While we had to perform data cleaning and preprocessing to extract the propsed features, the formatted body was also quite convenient because it allowed us to easily parse and recover links, code snippets, and other semantic elements of the text, such as emphasis by bolding or italicizing words.

We retrieved 44184 questions with at least four and at most ten answers from the dataset, and made sure that the best answer can be determined from the candidates. We split the dataset as a training set and a test set. Classifiers were trained on the training set using 3-fold cross validation, and tested on the test set.

4.2 Evaluation

The models will be evaluated by the accuracy of choosing the best answer, i.e. the percentage of questions for which our models correctly predicted the best answer. Because our dataset does not have information about which answer was actually chosen on the Stack Overflow website, we are assuming that the answer with the highest score is the best answer and using that when computing accuracy. This question-based accuracy is different from prior work (as stated in Section 2), but we believe this metric is more practical and challenging.

4.3 Baseline

As a baseline measurement, we look at the following three candidates: the first-submitted answer, the last-submitted answer, and the answer with the longest content. As mentioned, we believe that the first answer and the longest content features may contribute to which answers are chosen as highest due to time users have to upvote them, as well as the amount of detail put into longer answers. We also consider the last-submitted answer to further prove this thought. The results are 39.70% for first-submitted, 7.65% for last-submitted, and 29.89% for longest content. These results suggest that first-submitted answers may be a large contributing factor to what may be chosen as best answer, but as discussed, length of an answer may only be partially related to its success. Users may prefer more succinct answers that give a simple solution to the problem, even if that is just a single line of code.

4.4 Results

Question-based accuracy. We used logistic regression, random forest, and XGBoost as our models and compare with the baselines. Table 1 shows the predicted accuracy for our baselines and proposed models. We can see that first answer is a simple but competitive baseline as we expect. In addition, we outperformed the baseline by 8.52 percent by using XGBoost, and 7.52 percent by using random forest.

Table 1: Result of all the methods

Method	Accuracy
First answer	39.70%
Last answer	7.65%
Longest answer	29.89%
Logistic Regression	34.97%
Random Forest	47.22%
XGBoost	48.22%

Feature Importance. Among the eight proposed features illustrated in Table 2, we analyzed their importance determined by our random forest model. From Figure 2, we can see that the most important feature is Reputation, i.e., whether the person answering the question has delivered high quality answers. In other words, a user with high reputation is likely to give a higher quality answer. Also, as speculated, PostOrder of an answer influences the score it eventually received. Another interesting thing is that the semantic features (SimToQ, MaxSimToA, MinSimToA) are also important to the score of the answer. Finally, LinkCount seems to be the least important.

Table 2: Set of Features

Index	Symbol	Feature Description
0	TotalLength	Length of the content
1	LinkCount	Number of hyperlinks
2	CodeLength	Length of the code snippet
3	PostOrder	Chronological order of the answer
4	Reputation	Reputation of the answerer
5	SimToQ	Similarity to the question
6	MaxSimToA	Max similarity to other candidate answers
7	MinSimToA	Min similarity to other candidate answers

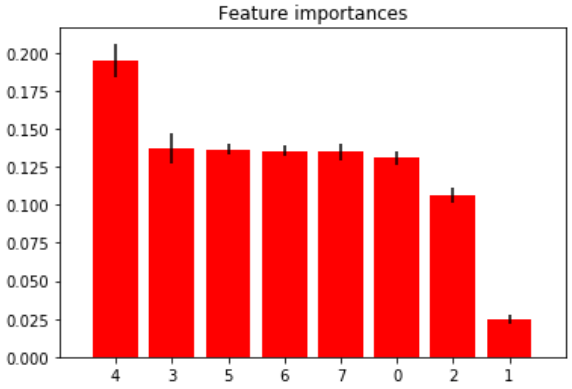

Figure 2: Compare the importance of each feature

The features can be further explained by using LIME (Ribeiro et al., 2016), a Local Interpretable Model-Agnostic Explanation. Take an answer with ID 21218427 as an example, Figure 3 shows that it has a higher chance to be selected as the best answer. The reason is that it is one of the first few answers (PostOrder ≤ 2) and the reputation of the answerer is high enough (Reputation > 403). Moreover, the values of all other features, including TotalLength, MinSimToA, LinkCount, SimToQ, and CodeLength, indicate that it is likely to be the best answer. However, its value of MaxSimToA decreases the possibility to be the best answer.

5 Conclusion and Future Work

In this paper, we proposed a set of features and machine learning models to predict the best answer for questions on Stack Overflow. We adopted NLP techniques such as tf-idf and LSA in our feature engineering, and appied ML models such as random forest and XGBoost to train classifiers. Shallow features and semantic features are proposed in this work, and the importance of the features are analyzed. We evaluated our techniques on a relatively new and practical metric, question-based accuracy. The experiment results show that our techniques outperformed baseline by up to 8.52%. In the future, we plan to introduce more features and further fine-tune our models. More NLP tech-

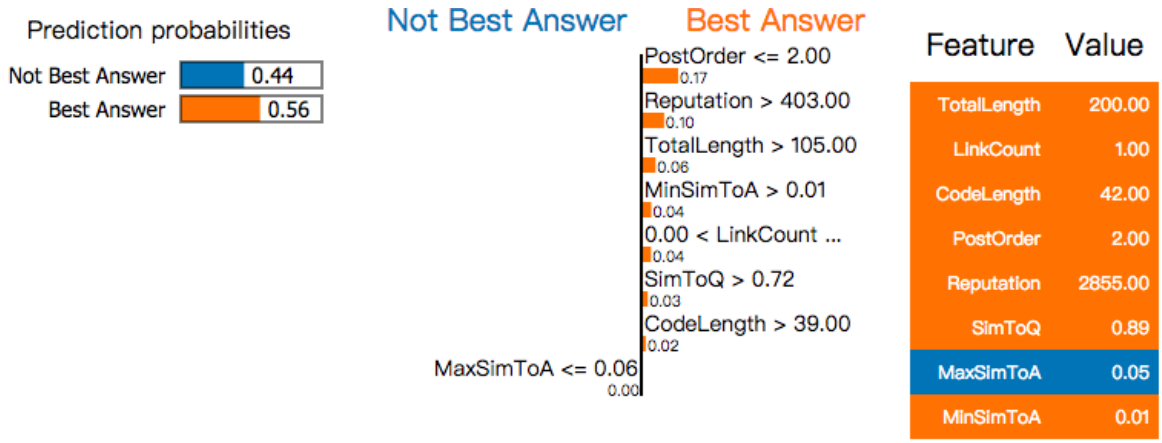

Figure 3: Compare the importance of each feature by LIME

niques such as word lemmatizing and other ML models such as recurrent neural network will also be investigated.

References

Mohan John Blooma, Alton Yeow-Kuan Chua, and Dion Hoe-Lian Goh. 2010. Selection of the best answer in cqa services. In *Information Technology: New Generations (ITNG), 2010 Seventh International Conference on*, pages 534–539. IEEE.

Grégoire Burel, Yulan He, and Harith Alani. 2012. Automatic identification of best answers in online enquiry communities. In *The Semantic Web: Research and Applications*, pages 514–529, Berlin, Heidelberg. Springer Berlin Heidelberg.

Fabio Calefato, Filippo Lanubile, and Nicole Novielli. 2016. Moving to stack overflow: Best-answer prediction in legacy developer forums. In *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ESEM '16, pages 13:1–13:10, New York, NY, USA. ACM.

D. Elalfy, W. Gad, and R. Ismail. 2015. Predicting best answer in community questions based on content and sentiment analysis. In *2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS)*, pages 585–590.

Maximilian Jenders, Ralf Krestel, and Felix Naumann. 2016. Which answer is best?: Predicting accepted answers in mooc forums. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 679–684. International World Wide Web Conferences Steering Committee.

Galina Lezina and Artem Kuznetsov. 2013. Predict closed questions on stackoverflow. In *SYRCoDIS*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1135–1144, New York, NY, USA. ACM.

Chirag Shah and Jefferey Pomerantz. 2010. Evaluating and predicting answer quality in community qa. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 411–418. ACM.

StackExchange. 2018. Stackexchange data explorer.

StackOverflow. 2016. Python questions from stack overflow.

Qiongjie Tian, Peng Zhang, and Baoxin Li. 2013. Towards predicting the best answers in community-based question-answering services. In *7th International AAAI Conference on Weblogs and Social Media, ICWSM 2013*. AAAI press.

Lichun Yang, Shenghua Bao, Qingliang Lin, Xian Wu, Dingyi Han, Zhong Su, and Yong Yu. 2011. Analyzing and predicting not-answered questions in community-based question answering services.