Which Answer is Best?
Predicting Accepted Answers in MOOC Forums

Maximilian Jenders
Hasso Plattner Institute
Potsdam, Germany
maximilian.jenders@hpi.de

Ralf Krestel
Hasso Plattner Institute
Potsdam, Germany
ralf.krestel@hpi.de

Felix Naumann
Hasso Plattner Institute
Potsdam, Germany
felix.naumann@hpi.de

ABSTRACT

Massive Open Online Courses (MOOCs) have grown in reach and importance over the last few years, enabling a vast user-base to enroll in online courses. Besides watching videos, user participate in discussion forums to further their understanding of the course material. As in other community-based question-answering communities, in many MOOC forums a user posting a question can mark the answer they are most satisfied with. In this paper, we present a machine learning model that predicts this accepted answer to a forum question using historical forum data.

Keywords

Massive open online course, MOOC, forum, reputation system, Answer recommendation, forum posts

1. INTRODUCTION

In recent years, MOOCs have become continuously more important, with an ever-increasing range of free courses on various topics. A MOOC platform typically offers many courses , the internationally best known MOOC platforms being Coursera, Stanford Online, and Udacity, and edX, as well as openHPI and iversity for the German-speaking community. These courses are predominantly provided by universities, which produce videos for a pre-determined course length of a few weeks, usually with frequent self-tests, graded homeworks, and a final exam. These MOOCs can be taken by any user from their homes, without having to enroll at university. Each week, users can freely choose when to engage in the course and how much time they are willing to invest to study.

This encourages very different users to engage in a MOOC – a student who is interested in a topic their university is not covering, a working person wanting to expand their horizon, someone looking for work who wants to improve their chances by gaining additional skills, or a pensioner interested in new developments. Regardless of their background, MOOCs reach a vast number of users with ease through internet participation.

In order to support the teaching process, MOOCs usually provide users the opportunity to engage in a forum, in which they can discuss the current coursework, ask questions about content they are unclear about, and offer help to each other. Additionally, the lecturer and teaching assistants typically also frequent the forums to answer questions and provide feedback. On some forums, users can mark the most suitable answer, sometimes called the *accepted answer*, indicating the question has been sufficiently answered by this post. A platform that has accepted answers offers multiple advantages to its users: The user who created the thread indicates that an answer has satisfied his information need, and points to the best answer in his opinion. The user who wrote the answer that became accepted gains confidence, receives positive feedback and earns reputation in the forum, as others see that he provided the best answer. Any user browsing the thread and also struggling with the original poster's problem can easily find an answer that should completely explain and resolve the issue for him, thus providing a better learning experience. Teaching staff browsing the forum can give lower priority to threads that already have an existing answers, instead dedicating more time to writing replies to thread for which no accepted answer has been pointed out.

Allowing users to mark answers as accepted is also widely spread in community-based question answering (CQA) services, such as Yahoo Answers and Stack Overflow, where research already has been done on determining the factors making out a good answer and the prediction of accepted answers.

However, while those CQA communities share many characteristics with MOOC forums, there are also differences: Although in both cases users may have a varying degree of understanding of the subject, in MOOC forums the questions are usually concentrated to a very specific topic, namely the content discussed in the current week's coursework. Also, since all participants have to engage in the same homework, there are some common problems every user faces. For both MOOC forums and traditional CQA communities, one challenge is the fact that only few users actively mark answers

WWW'16 Companion, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4144-8/16/04.
http://dx.doi.org/10.1145/2872518.2890567 .

as accepted. This might be due to various reasons: There might be no answers satisfactory to the user, the user might experience difficulties to choose one answer among multiple good ones, or the user might simply be uninterested in or oblivious to the fact that answers can be marked as accepted.

As such, it is an interesting challenge to apply the prediction of accepted answers to MOOC forums, i.e., identifying the best candidate answer out of a set of candidate answers, while finding features inherent to MOOCs that help this process. Such predicted answers can then be suggested to the original poster of the question.

In this paper, we aim to enhance the user experience of participants by using machine learning techniques to predict the correct answer of a thread. We are using data from the Hasso Plattner Institute's own MOOC system openHPI, which has offered 20 courses and has over 100.000 registered users.

2. RELATED WORK

There has been extensive research both on CQAs and MOOC forums, but to the best of our knowledge, our work is the first that uses approaches to predict accepted answers in MOOC forums.

2.1 Community-based question answering services

Various research has been done on community-based question answering services platforms that is relevant for our work.

Yang et al. analyzed the Yahoo! Answers platform and discovered that many questions are left without any answer at all [20]. They used a set of old questions to predict whether new question will be answered. As answers might be closed by a moderator, e.g., for being off topic or not constructive, a classifier was built in [10] using reputation-based used features in addition to post-specific features to estimate whether a question will be closed.

Yahoo! Answers was further used by Adamic et al. to research knowledge sharing activities, as interactions can resemble expertise sharing forums or discussion or advice [1]. The focus on activity led to the discovery that some users narrowly focus on specific topics, whereas others participate over multiple categories, and that individual categories exhibit different participation levels, such as average post length and number of answers. An answer's quality can be judged not by its content alone, but also by incorporating question types, as they have an impact on how an answer's quality is perceived [15].

The task of answer predictions has also received focus from the research community. For example, Shah et al. constructed quality criteria for answers and created features from questions, answers, and users [13]. These features were then used to predict the best answer. Blooma et al. identified social, textual, and manually determined content-appraisal features and researched their influence on the selection of the best answer [2]. Other work has also included the relationship between an answer and the question. Prediction models to identify an accepted answer were built using information about the answer, the question, and the relationship between both [11, 14]. We have expanded on the features by incorporating information about the user and MOOC-specific data. Burel et al. extracted user, content, and thread features to predict best answers across three question answering communities [3]. The authors reported that features like the total number of answers a user has written or the number of times an author's answer has been accepted showed only a small impact when measured with information gain. In our experiments, these features contributed the most using information gain.

Accepted answer were not the only ground truth used in answer ranking: Yao et al. focused on predicting the long-term impact of a forum post, measured as the number of votes it receives within a timespan of six months [21]. The number of votes an answer has received was used to train Random Forests for ranking in [7]. An extensive feature set is used, and only threads with at least four answers are considered. The authors of [22] leveraged public user profile information to identify the level of engagement, authority, and recognition of users and used this information for answer ranking on Yahoo! Answers. In contrast to typical CQA platforms, openHPI does not give users badges or assigns levels to signify their social reputation.

While the issue of predicting accepted answers or ranking answers has been addressed by the research community for CQA platforms, we are applying using MOOC forums, which are used only during short time spans in which a huge number of users discus very limited topics. We are further incorporating features unique to MOOCs to improve the prediction of accepted answers.

2.2 Research using MOOC forum data

Forum threads in MOOC courses have also been used for various kinds of research: MOOC forum data has been used in the task of predicting learning instructor intervention, i.e., if a teaching staff member will reply in a thread. Chaturvedi et al. reflected thread structure using chain based models [5], while others used data from 61 Coursera courses in their predictor[4], finding that forum type plays an important role in the prediction task and that sometimes, simple baselines outperform machine learning models as staff member may hold different views on when and how often to interact in the forum.

Linguistic models were also used by Wen et al. to quantify user engagement in the forums [18]. In a survival analysis, students with different engagement types displayed deviating dropout rates. In other research, user activity in forums and linguistic features were used to distinguish passive learners from active ones, which can be used to predict student performance [12].

MOOC forums were evaluated by Coetzee et al., finding that active forum use correlates with a higher retention rate [6]. Additionally, forums using a reputation system (e.g., badges, best answer selection, and votes on forum posts) produce faster response times and a larger volume of replies, but show no significant impact on grades and student retention. Dropout rates were also discussed by Wen et al., with a focus on the sentiment of MOOC forums [17]. Sentiment in forum posts was correlated to student dropout rates.

Thread and discourse structure has been analyzed to judge whether threads have been resolved using data from technical Linux communications [16] and student undergrad forums [9]. Finding useful and informative threads in MOOC forums was done using feature-based matrix factorization [19]. Features used included social peer connections, forum activities, and the content of posts.

3. MOOC DATA

The data used for the prediction model in this paper was collected from the openHPI platform. Section 3.1 gives information about the specific workings of openHPI, while Section 3.2 discusses the platform and its forum activity.

3.1 openHPI platform

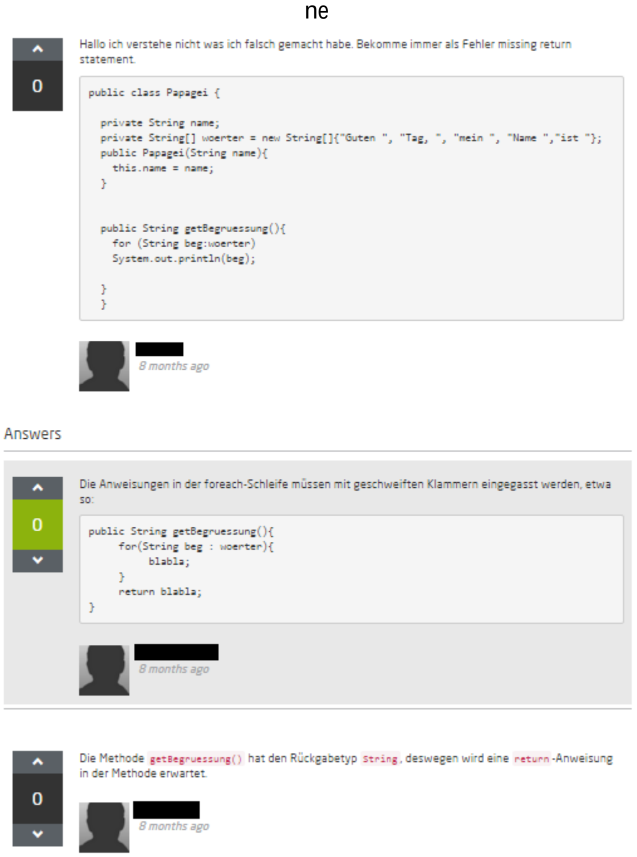
openHPI specializes on computer science subjects, offering courses in English or German, which are typically spread over six weeks and are open to everyone. Every week, participants are presented with approximately two hours of video material split into short sections to allow users to set their own pace, with optional and repeatable self-tests being provided after each video. Users also have to demonstrate their learning progress each week in a graded multiple choice homework and a final exam.

For the duration of a course, the openHPI forum can be used by participants to pose questions, comment on the current material, and to post answers to each others' questions. As such, the openHPI system distinguishes between the three types *question*, *comment*, and *answer*. While users posting a question can accept any answer given in reply, indicating they think this answer satisfies their information need, comments can not be marked as "accepted".

Figure 1 shows a typical openHPI forum thread with two answers. The first answer was accepted, illustrated by the green color of the vote button and the gray background of the answer. Users can add answers to the question or comment on the question or individual answers.

While not everyone who posts a question accepts an answer, the historical data of already completed courses gives us enough data to train a model. Using only the forum threads for which an answer was marked as accepted, we extracted numerous features, which are further described in Section 4. This dataset is then used to create a prediction model that estimates the probability of any given answer to become accepted.

It should be noted that we face the same problem that all community-based question answering services and learning platforms in general exhibit: Out of many users that are familiar with a subject, only a fraction visits the forums. Out of these users, only some actively engage in the forums to write questions and respond with answers. Even fewer users take the time to mark their favorite answer as accepted. Thus, any prediction system might exhibit a systematic bias, as users who mark accepted answers might favor different answers than those that do not mark accepted answers. The research community has noted this in their research of CQA platforms; for exaple, [20] reported that one out of eight questions on Yahoo! does not receive an answer, while [3] found that only every second thread will see an answer accepted.



Hallo ich verstehe nicht was ich falsch gemacht habe. Bekomme immer als Fehler missing return statement.

```
public class Papagei {

  private String name;
  private String[] woerter = new String[]{"Guten ", "Tag, ", "mein ", "Name ","ist "};
  public Papagei(String name){
    this.name = name;
  }


  public String getBegruessung(){
    for (String beg:woerter)
    System.out.println(beg);

  }
  }
```

[redacted]
8 months ago

Answers

Die Anweisungen in der foreach-Schleife müssen mit geschweiften Klammern eingegasst werden, etwa so:

```
public String getBegruessung(){
     for(String beg : woerter){
          blabla;
     }
     return blabla;
}
```

[redacted]
8 months ago

Die Methode `getBegruessung()` hat den Rückgabetyp `String`, deswegen wird eine `return`-Anweisung in der Methode erwartet.

[redacted]
8 months ago

Figure 1: openHPI forum thread with two answers. As the different coloring indicates, the first answer has been accepted.

3.2 Statistics

The openHPI platform started in 2012 and has offered 20 completed courses already, with approximately half of them being taught in German, the remainder in English. Depending on the language and the topic of the course, the number of enrolled users per course varies, as an introductory course into java programming might appeal to more users than an in-depth course about the semantic web. Further information can be found in Table 1.

As the accepted answer feature was introduced in openHPI in 2014, not all courses have questions with accepted answers. Also, as previously discussed, a majority of users who ask questions do not make the effort to mark answers as accepted, and the number of questions and number of questions with accepted answers fluctuates between courses. In total, there are 835 questions with an accepted answer. This emphasizes the value of an automated prediction system that can be used to re-rank answers in a thread for the benefit of other users. Answers with a higher prediction

Completed courses	20
Average participants per course	8351
Average participants active in forum	555
Average questions per course	585
Average answers per course	843
Average comments per course	1686

Table 1: Statistics about openHPI, as of December 2015.

score could be placed on top of a thread, helping users that are reading the question find the best answers.

4. PREDICTION MODEL

The data provided by the openHPI system is used to extract various features. We distinguish *user features*, *thread features*, and *content features*.

- **User features** describe the user who has written the question or has written one of the *candidate answers*. Candidate answers are all answers given in a thread that ultimately had an answer accepted and which hence can be used for model training. These features give information about previous forum activity of that user, such as the number of posted questions and answers, and the amount of accepted answers the user has already written. Also, MOOC-specific features, such as the number of visited courses and average homework scores, are used.
- **Thread features** hold information inherent to a thread post. Typical post features give information about answers and comments that have been already posted and some timing metadata, e.g., the time it took the answerer to reply or the time of day of the answer.
- **Content features** are specific to the textual content of the questions and answers. These features encompass the number of words a post holds (absolute and in relation to the question) as well as word-based similarity measures between question and answer, such as Jaccard similarity and KL Divergence.

While we are unable to provide a table of all features used due to space constraints, the most impactful features regarding their information gain are the number of previous answers in a thread, number of accepted answers posted by the answerer, sum of votes received by an answerer over all threads, and number of posted comments by answerer. In general, some user and thread features exhibit a relatively high information gain, whereas the content features seem to be less impactful.

Each extracted feature is then discretized into a set of distinct values to facilitate better machine learning. For our prediction, we use the Weka framework to create out-of-box methods without elaborate parameter tuning to show that our features are sufficient to create meaningful predictions [8].

5. EVALUATION

The 835 questions with an accepted answer (mentioned in Section 3.2) were responded to with at total of 1841 answers. Approximately half of those questions had only a single answer, the remaining 416 questions were responded to with at least two answers. Since we used historical data for the training of the model, we made sure to use only time-stamped features and thus only the data that was available when each answer was posted.

5.1 Evaluation measures

A machine learning model typically assigns each candidate a class probability — here, for each answer in a thread, a probability is calculated that estimates the likelihood of the answer becoming accepted. In a typical machine learning evaluation, all candidates above a certain threshold (e.g., 50%) are predicted to belong to a class, and the predicted class is then compared to the actual class to establish measures like accuracy, precision, and recall.

However, there can be only one accepted answer per thread. As such, we want to evaluate whether the accepted answer's prediction score is higher than the other answers' scores in the thread. Therefore, all evaluation must be carried out on a thread level to see if the answer that was in fact accepted also received the highest prediction score out of all answers in the thread (we call this case a *hit*), if another answer has received the highest prediction score (a *miss*), or if both the accepted answer and another answer share the highest score (a *tie*).

As training and evaluation was performed on historical data, we employed ten-fold cross-validation to mitigate the impacts of overfitting. We inserted all candidate answers of a thread into the same fold, because each thread is evaluated individually. We then made sure each fold was assigned approximately the same overall number of answers.

As baseline methods we used the features that give the highest information gain scores:

- **First answer**: The first answer to be posted in a thread is recommended.
- **Last answer**: The last answer to be posted in a thread is recommended.
- **Most accepted answers**: The answer of the user that has had the most answers accepted for previous questions is recommended.
- **Most votes received**: The answer of the user that has received the most up-votes across all forum posts is recommended.
- **Most comments**: The answer of the user that has posted the most comments is recommended.

In addition, we trained machine learning models on all features using a random forest classifier (an ensemble learning method on tree learning models), multilayer perceptron (a feedforward artificial neural network), bagging (also called bootstrap aggregating, this is a meta-learning algorithm designed to reduce variance), and Naive Bayes (a simple probabilistic classifier that assumes feature independence).

Method	Hits	Misses	Ties
First answer	192	224	0
Last answer	133	283	0
Most accepted answers	155	129	132
Most received votes	195	61	60
Most comments	193	182	41
Random forest	396	20	0
Multilayer perceptron	390	26	0
Bagging	387	29	0
Naive Bayes	237	179	0

Table 2: Evaluation on accepted answer prediction for threads with at least two answers. Baselines (top) are compared with machine learning methods (bottom). Ties indicate that multiple candidate answers share the highest score, so the algorithm would have to choose one based on other features or a random draw.

5.2 Predicting the best answer

While we were most interested in predicting accepted answers in threads that had multiple answers, we could still use the 419 threads whose only answers were accepted. However, when compared to a model trained on just the 416 threads with at least two answers, the first approach provided no significant increase in prediction power, with some machine learning methods even showing a slightly decreased accuracy. This might indicate that threads in which the only answer gets accepted exhibit slightly different characteristics than those in which an answer was picked out of various competitors. Users might also have found the single answer to not be entirely satisfactory, but accepted it since there was no better response.

Hence, Table 2 shows the results of the ten-fold cross-evaluation of the baselines and machine learning methods on the 416 (multiple-answer) threads. As can be seen, the baselines recommended the accepted answer for fewer than half of all threads, while the ensemble and deep learning methods give the best predictions. We have also evaluated further simple machine learning methods, which were all outperformed by the ensemble learners. Using Random forests yielded the best result.

Inspecting the combined 75 classification errors the best three machine learning models made, 10 accepted answers were misclassified by all three models and 12 errors were shared between two models. This suggest that the prediction accuracy could be further improved by learning a further ensemble classifier on the output of a variety of machine learning models.

Furthermore, six out of the ten threads that were misclassified talked about issues in programming courses, where users had problems getting their code to run and posted snippets of the code in their questions. The question and answers usually contained great amounts of code compared to normal, textual content. This suggests that content features, while individually providing a small information gain, still provide valuable information in the prediction process. Figure 1 in Section 3 shows such an example of a misclassified programming thread.

Method	Hits	Misses
Random forest	408	11
Multilayer perceptron	411	8
Bagging	415	4
Naive Bayes	387	32

Table 3: Evaluation on accepted answer prediction for threads that only have a single answer. A decision boundary of 50% was used. Hits denotes answer that were correctly classified as accepted, misses indicate misclassifications.

5.3 Judging individual answers

So far, or focus has been to pick the best answer out of a set of answers. While including the 419 threads with only a single answer did not add value for this task, we can still aim to predict whether this answer will become accepted using the prediction score of machine learning methods. If the score is above a certain threshold, e.g., 50%, we can assume the answer will become accepted. Table 3 shows the result of this analysis on a ten-fold cross-evaluation of models trained on all 835 threads. As can be seen, in most cases the machine learning algorithms would have correctly predicted the answer to become accepted with up to 99% accuracy.

However, the significance of these predictions is hard to judge, because there is no negative gold standard, i.e., a set of threads that do not have any accepted answer, because the asker or a human judge decided that no answer is satisfactory. Such annotations could be used during training and to evaluate how often a predictor would estimate an answer to become accepted when no answer was accepted. Additionally, with such a standard, the threshold could be further optimized by finding the best setting that performs reasonably well for both accepted as well as non-accepted answers.

6. CONCLUSION AND FUTURE WORK

We have presented a model to predict accepted answers for forum questions on the openHPI MOOC platform. The model was trained on historical course data, containing features about both the forum thread and the users participating in it, and evaluated using ten-fold cross validation. The results show that automatically predicting which answer will get accepted is viable in a MOOC environment.

Possessing the means to automatically evaluate answer quality can be very useful as the data showed that only few users actively mark the best answers. Automatically re-ranking or indicating the most useful answers could help other users reading a forum thread obtain the most important information faster and thus improve the learning experience.

As users generally do not take the time to mark their favorite answers, recommending good answers might encourage them to do so, as well as a more prominent placement of the accepted answer button, user badges representing social status to mark users that had answers accepted) and other gamification approaches. Getting more users to accept answers has two benefits: It adds value to the forum, giving users reading a thread a better idea on which answers to read first, and it provides more data to do training and evaluation on. As mentioned in the previous section, a feature that allows users to specify that no answer so far has been good enough to be accepted would also greatly benefit the gold standard and thus the prediction task.

As such, we are expanding on our research that used only

tt thrn tloas tal tnecessary-386(tata)-3866and thrn-3866andwer
tro pf te tccepted pwithout-302(tdela28(y)-6(a)]TJ8.967 -11.402 Td[(AFinlly)-6(a)-3130thrn-382(expistng)-320(b)-29(e)-29(d)y-320(bf)-31
cm28(usnit29(ty]423(theink)-2752puat)-323(tredictiog)-275(accepted1(id-275(acsw)28(er))-3752pis-275(ampl-29(e)ran)29(t)-TJ0 -11.402 T