Towards Identifying the Resolvability of Threads in MOOCs

Diyi Yang, Miaomiao Wen, Carolyn Rose
Language Technologies Institute, Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, 15213
{diyiy,mwen,cprose}@cs.cmu.edu

Abstract

One important function of the discussion forums of Massive Open Online Courses (MOOCs) is for students to post problems they are unable to resolve and receive help from their peers and instructors. There are a large proportion of threads that are not resolved to the satisfaction of the students for various reasons. In this paper, we attack this problem by firstly constructing a conceptual model validated using a Structural Equation Modeling technique, which enables us to understand the factors that influence whether a problem thread is satisfactorily resolved. We then demonstrate the robustness of these findings using a predictive model that illustrates how accurately those factors can be used to predict whether a thread is resolved or unresolved. Experiments conducted on one MOOC show that thread resolveability connects closely to our proposed five dimensions and that the predictive ensemble model gives better performance over several baselines.

1 Introduction

Massive Open Online Courses (MOOCs), run by organizations such as Coursera, have been among the most news worthy social media environments in the past year. While usage of social media affordances such as discussion forums in such courses is small relative to usage of videos or assignments, participation in the discussion forums is an important predictor of commitment to the course (Yang et al., 2013). We hypothesize that supporting a positive experience in such forums has the potential to increase retention in such courses. In this paper, we specifically study the behavior of students in a MOOC course for learning Python programming. We present empirical work that elucidates an important problem in existing MOOC discussion forums, propose a practical solution, and offer promising results in a corpus based evaluation.

MOOCs for programming skills can be seen as important resources for the professional development of programmers and programmers in training. While MOOCs for learning programming are a recent phenomenon, they are not the first web accessible resources for development of such skills. In recent years, a plethora of question/answer sites for programming have become available that have grown into thriving communities of practice for programmers. In these online communities, programmers can get mentoring from those who are more expert than them and offer mentoring to programmers who are less expert than them. For example, StackOverflow[1] has become a forum not only for getting specific questions answered, but for negotiating the pros and cons of alternative ways of solving technical problems. The code proposed as part of alternative solutions remains as part of the community memory, which is then accessible for those who come later with similar concerns.

Where StackOverflow falls short is in providing an appropriate environment for the active involvement of very novice programmers. When such novices come to a forum like StackOverflow and present their naive questions, they are frequently met with sarcastic responses if they get a response at all.

MOOCs for learning programming skills fill a gap left open by such environments, in that they welcome the very novice and provide forums where naive questions are not shunned. Nevertheless, discussion forums that only include such novice programmers would be akin to *the blind leading the blind* were it not for the involvement of a few more expert students and the teaching staff. This does not fully solve the problem, however. Many threads are

[1] http://stackoverflow.com/

Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 21–31,
October 25-29, 2014, Doha, Qatar.

still left without a satisfactory resolution. Currently, it is challenging for the teaching staff and expert participants to know where in the massive amount of communication to look for opportunities where their support is most needed. This is the problem we aim to address in this paper, i.e. automatically identify whether a thread is resolved and provide potential for better allocation of instructor and student resources.

In the remainder of the paper we first survey related work. Next we describe the formulation of the problem. We then present a series of two experiments, the later one building on the successful findings and results from the former. The results conducted on one MOOC show that our proposed model of thread resolveability better captures the difference between resolved and unresolved threads and that the ensemble logistic model outperforms several baselines. We conclude the paper with a discussion of the limitations of this work and next steps.

2 Related Work

MOOCs have received more and more attention recently, with the promise of providing many of the benefits of traditional classroom learning but not limited by time, location or finances. Much prior work has focused on analysis of such platforms to motivate the design of better student learning experiences. In various ways, the issue of students needing support from instructors and students has been addressed (Lieberman, 1995).

An important component in the Coursera environment is the discussion forums, which students can use to learn new knowledge from each other and from the teaching staff when they participate. In support of the importance of the discussion forums in connection with major problems like attrition, models are proposed to predict student dropout based both on their video watching behavior and also discussion forum posting behavior, such as how many posts a student has made (Balakrishnan, 2013). Student behavior in the discussion forum is also focused by other prior works (Yang et al., 2013). Yang et al. analyze drop out along the way, demonstrating the predictive power of features extracted within time windows of student behavior within the forums. The results of their work suggest that interaction with other students is important for keeping students motivated, which is further confirmed by many works (Yang et al., 2014; Rosé et al., 2014). Besides, linguistic reflections are also crucial for students engagement (Wen et al., 2014).

Other work highlights the importance of interaction in the form of feedback during participation in MOOCs. For example, some prior work (Piech et al., 2013) has explored peer grading, especially in helping grading of open ended assignments, in courses with thousands or tens of thousands of students. Other work takes a more holistic approach to assessment of student behavior. For example, in one such example (Kizilcec et al., 2013), instead of looking at students' assignments, students were classified based on their patterns of interaction with video lectures and assessment activities. This behavior trace was processed using a simple and scalable classification method that could identify a small number of longitudinal engagement trajectories that potentially provide the impetus for tailored feedback or mentoring.

Outside of MOOC discussion forums, there has also been work investigating the conditions under which questions receive appropriate feedback in more general Question Answering (QA) sites. In particular, this work has been framed as research on thread resolveability in QA sites. It can be conceived as the human counterpart to fully automated question answering systems (Prager et al., 2000; Perera, 2012; Jeon et al., 2006; Agichtein et al., 2008). Much of this work has emphasized the importance of having effective features to model question and answer processes.

In some of this prior work, the focus has been on identifying whether a thread is answered given a question and a set of potential answers (Sung et al., 2013; Tian et al., 2013). The prior work (Anderson et al., 2012) has focused on understanding the dynamics of the surrounding community activity, like the process through which answers and voters arrive over time. Based on understanding of such factors, a prediction can be made about the long term value for the community of a question being answered. Similarly, Agichtein and colleagues (Agichtein et al., 2009) presented a general prediction model of information seeker satisfaction in community question answering, and developed content, structure and community focused features for the question answering task. A collection of other related work (Liu and Agichtein, 2008) has developed personalized models of asker satisfaction to predict whether a particular question starter will be satisfied with the answers given

by others. This is solved by exploring content, structure and interaction features using standard prediction models.

Work on automated question answering systems can also be seen as relevant since questions that can be answered automatically do not need a human response, and therefore might reduce the load on available human effort. Instead of predicting whether a problem is answered, strategies for predicting are explored when a question answering system is likely to give an incorrect answer (Brill et al., 2002). To further understand how a question is answered, researchers (Yih et al., 2013) have studied the answer sentence selection problem for question answering and improves the model performance by using lexical semantic resources. That is, they construct semantic matches between question and answers. In terms of the extent to which the question is answered, Shah and colleagues (Shah and Pomerantz, 2010) evaluated answer quality by manually rating the quality of each answer. Then they extracted various features to train classifiers to select the best answer for that question. Liu et al. (Liu et al., 2011) proposed to use a mutual reinforcement based propagation algorithm to predict question quality based. The model makes its prediction based on the connection between askers and topics, and how those connections predict differences in quality.

The above question answering work is all about general discussion forums (Qu et al., 2009; Kabutoya et al., 2010), such as Yahoo! Answers[2]. In our work, in addition to taking advantage of existing QA work, we also adopt a linguistic perspective (Jansen et al., 2014) and take semantic matching into account using a latent semantic approach. To the best of our knowledge, this is the first work on thread resolvability analysis in a MOOC context.

3 Research Problem Introduction

In this section, we focus on how to identify the resolveability of threads in the MOOC forums. We firstly introduce the research context and dataset, then we formulate our resolveability problem.

3.1 Research Context and Dataset

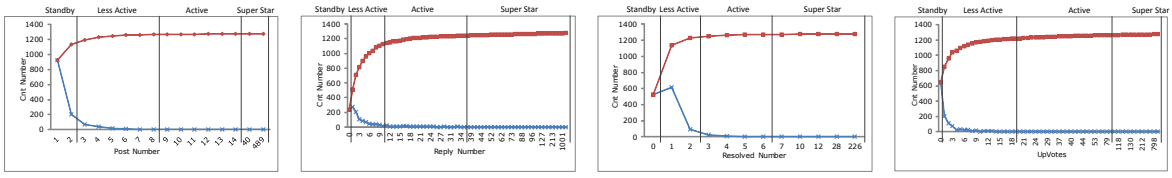
In programming MOOCs, when students encounter problems working on the programming assignments, or when something is not clear from the readings or lectures, students have the opportunity to initiate a thread in the course forum, in order to engage other students in the class as well as the teaching staff. For example, if a student were confused about the distinction between an argument and a parameter in Python, he/she would post the question to the variables subforum, marking it unresolved at the same time. In the ideal case, another participant would reply to this question with some detailed explanation and example, which would solve that problem. When the student who initiated the thread receives the response, assuming it is adequate, that student may mark it as resolved. Others may join in as well, and individual posts may be rated through upvotes and downvotes. In contrast to existing QA sites, no *best answer* option is available. Thus, the resolved/unresolved button provides the closest equivalent groundtruth.

The data for this paper was crawled from a Python language course. Our focus was specifically to investigate the inner workings of threads related to getting answers to questions or help with programming difficulty. In order to avoid including threads in our dataset that are off-topic or otherwise irrelevant, we limited the set of forums to the subforums that focus strongly on course content, including those indicated to focus on lectures, exercises and assignments as well as the final exam. That is, we discarded posts in the study groups, social discussion, and other discussion areas that do not have unresolved buttons. In the final dataset, there were 2508 threads (1244 resolved threads) in total, and 2896 users (12 instructors and staffs) who had at least one post. Each question is associated with a label indicating whether it is resolved or not.

3.2 Problem Formulation

Work on the related problem of analysis of QA websites has grown in popularity in recent years. However, due to differences in how MOOCs work as temporary online communities, it is necessary to consider how findings from prior work in these areas may or may not generalize to this new context as we formulate our research problem. In particular, MOOCs are different from existing QA websites, such as Yahoo! Q&A, Stack Overflow. The purpose of QA sites is primarily for people to get answers. While people may learn from their interactions on such sites, those sites are not designed in particular to support learning. Thus,

[2]http://answers.yahoo.com/



(a) Question Post (b) Reply Devotee (c) Resolved Favor (d) UpVotes

Figure 1: Starter Influence Statistics. Each Figure has two curves; the below one indicates how many users have made the associated number of posts/reply. The above one is the cumulative version of the same.

different characteristics are needed in the MOOCs discussion threads. One implication is that the discussions in MOOCs may need to be more interactive than those found in environments such as StackOverflow. Students who post problems can be expected to be less capable of fully comprehending an answer even if it is a good one. This demands more effort from those with the ability to offer helpful responses. In order for the discussions to be effective, the threads must include a balance of naive participants and participants with more knowledge. A related issue is that it is not yet ubiquitous for participants in MOOCs to have the opportunity to earn a reputation score for offering useful answers and other instructional support. In other QA sites, this is both a valuable motivator as well as an important predictor of resolved versus unresolved question threads (Anderson et al., 2012). Thus, students who post questions may need to sell their problem in order to attract those who can offer help. Taking these interrelated issues into account, an important aspect of our modeling work is in recognizing the different roles that users play in the community. Related to this, we will describe below how we develop models that include latent variables related to the propensity of users to initiate problem threads that attract useful responses, and the propensity of others to contribute useful responses in such contexts. We refer to these complementary variables as *starter influence* and *expert participation* respectively.

Secondly, all are welcome to learn in a MOOC and participate actively even if they have no prior knowledge. In an educational context, it would not be appropriate to meet a naive question with a sarcastic response. In contrast, in Stack Overflow, it would be treated as unremarkable for a naive question to get a sarcastic response. While naive participants may not enjoy such responses, they learn to expect them. Since approaching posted problems with patience and friendliness is important for avoiding discouraging new learners, we include a variable called *friendliness* that represents friendly and polite discussion behavior. None of these would ultimately result in thread resolution if the answers that are offered were not targeted to the problems raised by the students who initiated the threads. This is one place where our work is very aligned with earlier work on QA sites. And thus we adopt a similar practice where we include in our model an estimate of answer appropriateness in a latent variable we refer to as *content matching*.

Now we define important terms used in our discussion. First, we define roles within discussion threads that are relevant for our work. For a given thread, the user who initialized the thread is called the **Starter**; the teaching staff including both official course instructors and TAs are referred to as **Instructors**; and any other users who have replied or commented in the thread are referred to as **Participants**. We count a thread in our dataset as **resolved** only if the thread starter personally changes the Unresolved button to Resolved. Otherwise, we count the as **unresolved**.

We are interested in the conditions under which a thread is marked as resolved or unresolved:

Thread Resolveability: Given a thread with its associated question and set of replies, which may not have been explicitly marked as resolved, identify whether it should have been marked as resolved or not.

4 Latent Variable Modeling

We laid the foundation for a conceptual model above to understand the factors associated with resolved versus unresolved threads and introduced five latent factors we referred to as Starter Influence, Expert Participation, Thread Popularity, Friendliness, and Content Matching. In this section, we

further formalize these latent factors by specifying associated sets of observed variables that will ultimately enable us to evaluate our conceptual model. All latent and observed variables are enumerated in Table 1.

4.1 Starter Influence

The person who serves as the Thread starter is responsible for formulating the question that is addressed, and therefore the focus of that discussion. Some participants post many questions and are very adept at formulating their questions in ways that engage the attention of people who have the ability to provide answers. If the starter posts a lot and his/her questions often get resolved, this can be taken as an indication that this person is popular. Questions contributed by him/her may be more likely to attract attention and receive replies. This simple indication of popularity, which can be easily computed, may in some way compensate for the lack of an established badge system where they are not in use. We propose to measure this form of user influence by using the following four indicators. (1) Question Devotee x_{Pst}, indicates how many threads this question starter has proposed in this forum. Based on Figure 1(a), we divide users in this discussion forum into four types to indicate the propensity to post, i.e. post number ranges from 1-2 as standbys, 3-5 as less active, 6-14 as active, 40-489 as superstars. Similar partition method is adopted for all the following indicators. (2) Reply Devotee x_{Rep}, means how many times a person acts as a *Participant* in a thread posted by other students as shown in Figure 1(b). If he/she usually replies to others, then it is possible that his/her question will be paid more attention in return. (3) Resolved Favor x_{Res}, means in how many threads the person acts as the *Starter* in threads that get resolved. (4) Praised Responder x_{Uvt}, indicates the proportion of all the posts this starter makes in the forum that received upvotes, as displayed in Figure 1(d). This connects to how others recognize this starter and to what degree.

4.2 Expert Participation

Who participates a discussion is as important as who initiates the discussion. Students with some expertise in the related content can often provide quality replies (Anderson et al., 2012). Since user reputation score information is not available in this MOOC, it is necessary to for us to identify observable indicators. We define a person as Expert x_{Exp} in our forum as follows. A person is an Expert if and only if he/she is one of the instructors or his/her reputation score as we compute it is ranked in the top 1% among all students. The reputation score of student u is computed based on his/her question devotee u_{Pst}, reply devotee u_{Rep}, resolved favor u_{Res}, and praised recognition u_{Uvt} as we defined in the previous section. The contribution of each factor to reputation score is controlled using parameters α, β, γ.

$$\begin{aligned} score(u) =& \alpha u_{Pst} + \beta u_{Rep} + \gamma u_{Res} \\ &+ (1 - \alpha - \beta - \gamma) u_{Uvt} \end{aligned} \quad (1)$$

4.3 Thread Popularity

How much attention is paid to a question may be linked to the attractiveness of the thread based on how it is presented to the community. Thus modeling thread popularity may be valuable for accounting for variation in level of participation across threads. In particular, a reply is given upvotes when others think it is informative or good. Thus upvotes could indicate how others evaluate the replies in connection with the question. We design three observable factors here that may contribute to a model of thread popularity. The *Total UpVotes* x_{Tvt} and *Max UpVotes* x_{Mvt} are used to represent the credit this thread has received and how others recognize the current discussion. Based on our analysis, people rarely give a downvote to others' posts. The *Question Votes* x_{Svt} indicates whether the starter formulates a problem that wins recognition from others. For Total Upvotes, we find that in resolved threads, it is 6.10 compared to 3.15 in unresolved thread. Thus, intuitively, thread popularity has the potential to give a useful prediction of thread resolveability.

4.4 Friendliness

Friendliness (Danescu-Niculescu-Mizil et al., 2013; Burke and Kraut, 2008) concerns whether the current conversation is conducive for others to discuss ideas. This has not been considered in existing question answering work, and we thus discuss our operationalization of politeness here. We hypothesize that resolved threads posses more polite words, such as 'thank'. For example, a resolved thread might end with gratitude to thank others for providing help, and indeed we see this. Thus, we specify a set of observed indicators that may be useful in a latent variable model of politeness. (1) *Start with Thanks*: x_{Stx},

Var	T	Description	Var	T	Description	Var	T	Description
Pae	N	Please Count	Qa1	N	1st Match Score	Svt	N	Question Votes
Thx	N	Thanks Count	Qa2	N	2nd Match Score	Mvt	N	Max Votes
Dfe	N	Deference	Qa3	N	3rd Match Score	Uvt	N	User Votes
Etx	B	End with Thx	Len	N	Max Length	Rep	N	Reply Number
Stx	B	Start with Thx	Sim	N	Similarity	Res	N	Resolved Count
Exp	B	Expert Join	Tvt	N	Total Votes	Pst	N	Post Number
Sin	-	Starter Influence	Epr	-	Expert Participation	Con	-	Content Matching
Pop	-	Thread Popularity	Fen	-	Friendliness	Label	B	Resolved or not

Table 1: Variables used in the Structural Equation Model (SEM). Var is the factor variable that is used, which also corresponds to Figure 2. T indicates what type of values a variable can take. B is short for Binary. N is short for Numeric. '-' means it is a latent unobserved variable.

indicates whether this starter shows politeness when he/she posted the question. (2) *End with Thanks*: x_{Eth}, stands for whether the starter says thanks after receiving others' help. (3) *Thanks Count*: x_{Thx}, measures overall friendliness in the current discussion. We evaluate this by counting the thanking related words. (4) *Deference*: x_{Dfe}, is a count of positive polite words occurring in the discussion, such as using the words 'Nice','Great', or 'Awesome', as in prior work (Danescu-Niculescu-Mizil et al., 2013). Such words are used as markers to conduct counting. (5) *Please*: x_{Pae}, captures whether friendly question asking words were used, i.e. how many times words such as 'Please', 'Will', occur in current conversation.

4.5 Content Matching

Matches between the content of a thread and its replies indicate whether replies are relevant to answering the question instead of some off-topic discussion. In order to estimate this, we build an Eigenword bipartite graph to capture semantic similarities. Each node in the bipartite graph is the corresponding Eigenword[3] of a given word, with the left side representing the words that occurred in the thread starter, and the right side representing the words in a given reply. The edge is a similarity score computed by using the cosine similarity metric. In order to better identify whether a reply is discussing the content of the question, a semantic match between the thread question and its replies is needed. The top 3 matching scores are denoted as $x_{Qa1}, x_{Qa2}, x_{Qa3}$. Additionally, TF-IDF similarity x_{Sim} is computed (the correlation between x_{Sim} and $Qa1, Qa2, Qa3$ are $0.3280, 0.3572, 0.3569$ separately) and the maximum answer length x_{Len} is used to assist in computing the matching score.

[3]http://www.cis.upenn.edu/ ungar/eigenwords/

5 Experimental Investigation

In the above section, we described five latent factors we hypothesize are important in distinguishing resolved and unresolved threads along with sets of associated observed variables. In this section, we conduct two studies on thread resolveability, including validating the influence of each latent factor on thread resolution using a Structural Equation Model (SEM), and evaluating the generality of the identification of the resolveability using a predictive model. Experiments are conducted on the Python dataset with performance measurement under different evaluation metrics.

5.1 Conceptual SEM Validation

Our conceptual model is implemented as a Structural Equation Model (SEM) and is introduced as an evaluations of the effect of each latent factor on thread resolveability, as shown in Figure 2.

5.1.1 Conceptual SEM Model

A Structural Equation Model (Bollen, 1987), is a statistical technique for testing and estimating correlational (and sometimes causal) relations in cross sectional datasets. To explore the influence of our five latent factors, we take advantage of SEM to formalize the conceptual structure in order to measure what contributes to thread resolveability. The designed latent factors are specified as latent variables within the model, with the associated observed variables discussed above. We define the conceptual structure of how a thread gets resolved as well as a mathematical expression of each latent variable in Equation 2.

Related variables are explained above and

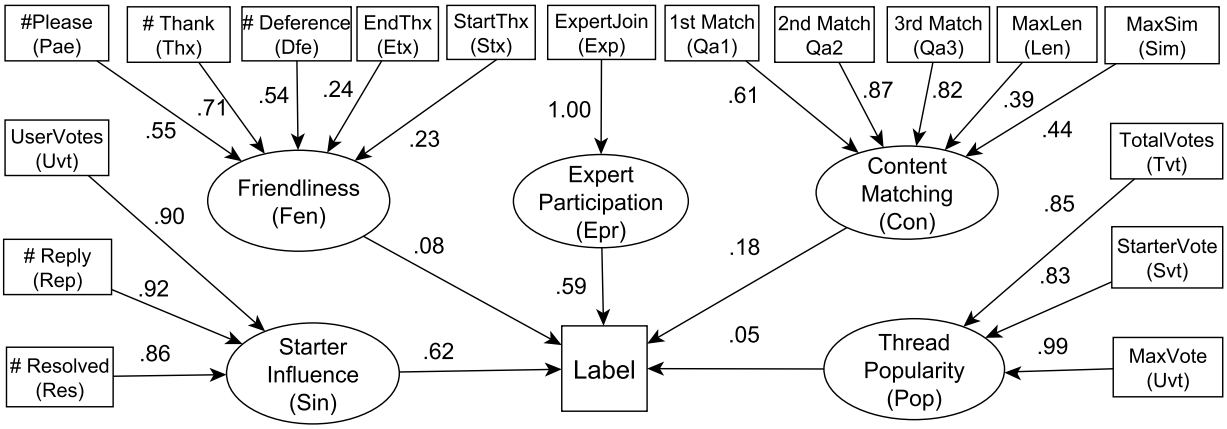


Figure 2: SEM Model Factor Analysis Result. Each directed edge indicates the predictive relationship. Weight on each directed edge is the estimated influence strength of one node to another. Table 1 illustrates the denotation. Only significant node influences whose p-value ($p < 0.05$) are presented. Circles stand for latent variables while rectangles signify observed variable.

summarized in Table 1. $Label$ refers to the label of a unknown thread, taking the value of Resolved or Unresolved. $Label$ (L) is a linear combination of each latent factor set. For each variable in a latent factor set, it is associated with a weight parameter γ in the SEM. Specifically, this conceptual structure of how a thread gets resolved relates to five aspects, i.e. (1) whether the thread starter has enough influence on others, (2) whether the relevant experts participated at least once in the discussion, (3) whether the thread polite and conducive to encouraging others to be willing to provide help, (4) whether the thread is popular, and (5) whether replies aim at answering questions instead of off topic discussion.

$$
\begin{aligned}
Con &= \gamma_{ci} \sum_{i=1}^{3} x_{Qai} + \gamma_{c4} x_{Sim} + \gamma_{c5} x_{Len} \\
Fen &= \gamma_{p1} x_{Stx} + \gamma_{p2} x_{Etx} + \gamma_{p3} x_{Thx} \\
&+ \gamma_{p4} x_{Dfe} + \gamma_{p5} x_{Pae} \\
Sin &= \gamma_{u1} x_{Rep} + \gamma_{u2} x_{Pst} + \gamma_{u3} x_{Res} + \gamma_{u4} x_{Uvt} \\
Pop &= \gamma_{t1} x_{Cmt} + \gamma_{t2} x_{Tvt} + \gamma_{t3} x_{Mvt} + \gamma_{t4} x_{Svt} \\
Epr &= \gamma_{a0} x_{Exp} \\
Label &= \zeta_1 Con + \zeta_2 Fen + \zeta_3 Sin \\
&+ \zeta_4 Pop + \zeta_5 Epr
\end{aligned}
\tag{2}
$$

5.1.2 SEM Result Analysis

In this section, we discuss what we learn from the SEM about the influence of each factor within the model. We adopt the Structural Equation Model in R (Rosseel, 2012) to conduct the validation, and evaluate it by looking at the Comparative Fit Index (CFI), Root Mean Square Error of Approximation (RMSEA) and Standardized Root Mean Square Residual (SRMR) (Barrett, 2007). Figure 2 shows the influence of each observed variable on its corresponding latent variable, and in turn the latent variable on the resolved label. The weights on each directed edge represent the standard estimated parameter for measuring the influence. For the model fitting, we get a RMSEA of 0.09 and SRMR of 0.06, with a CFI of 0.89. The fit is not extremely high, but it is moderate, and it is within the range one would expect from a good fitting model when a large set of variables is considered.

Based on Figure 2, firstly, starter influence and expert participation contribute a lot to thread resolveability, with a standard estimated parameter of 0.619 and 0.587. This makes sense that who posts the question and who gives replies matter a lot in identifying whether a thread is resolved. Next, content matching contributes 0.178 to the resolving of a thread, which means matching between question and replies does differentiate between resolved and unresolved threads, but less so than who participates, perhaps because the observed variables are very shallow indicators of relevance. Friendliness is not very strongly predictive of resolvability. Similarly, Thread popularity contributes only 0.051 to the prediction, without significant influence compared to the other four latent variables, which are all significant. Thus we conclude that starter influence, expert participation, and content matching are strong factors while friendliness and thread popularity could help us separate resolved and unresolved

threads, but less so than the other two.

5.2 Resolveability Prediction

The influences of five latent factors on thread resolveability are demonstrated as above. In this part, we build an ensemble logistic regression model to leverage those findings to predict whether a given thread is resolved or not.

5.2.1 Ensemble Regression Model

An ensemble logistic regression model is proposed to deal with the prediction of whether a thread is resolved or not. That is, given the question and a set of potential replies, as well as the five latent variables and associated observed variables, we want to predict whether a question has been answered. Our ensemble logistic model works in the following way. Firstly we train a separate logistic model for each of the five aspects defined above, i.e. five sub logistic model of how each aspect predicts the resolved property. Then those sub-models are included together in an ensemble in order to contribute to a final logistic model, which takes those results as the input features. Similar to generalized boosting (Friedman et al., 1998), this regression model integrates five weak predictors that capture five different aspects of thread resolveability, and construct a two layer logistic ensemble, which is distinct from a linear voting strategy. Our ensemble model relaxes the assumption of linearity and thus offers more flexibility in finding an effective predictive model. This process is formalized below.

$$\ddot{R}_j = \frac{1}{1 + e^{-\sum_{i=1}^{k} \alpha_i \cdot \dot{R}_{ij}}} \tag{3}$$

Here, k refers to the number of latent aspects. $\ddot{R}_j$ is the predicted resolved score for thread j; if it is larger than a threshold, the prediction of that thread question is resolved, otherwise it remains unresolved. $\dot{R}_{ij}$ is the predicted resolved score of latent factor set i on thread j, trained on the corresponding latent factor set.

5.2.2 Prediction Results

To demonstrate the predictive abilities of the five latent factors, we use our ensemble regression model to predict thread resolution. 10-fold cross validation is used, and the prediction results will be evaluated using the metrics Recall, Precision, and AUC (Area under Curve). For baselines, we begin with the simplest model **EndThx**, which simply bases the prediction on whether the current thread ends up with a gratitude sentence. This makes sense because it is natural that students will express their gratitude after receiving others' help. One simple baseline is the **Majority**, which predicts the testing thread as the majority status (unresolved in our dataset), leading to a accuracy of 0.503; **Si+Ep** is a combination of the latent aspect of starter influence and expert participation; and **Si+Ep+Ct** adds the content matching latent set on **Si+Ep**; **Si+Ep+Ct+Fr** is defined similarly. **ALL-Linear** is adding all five latent factor sets and predicts the resolved or not using a linear logistic regression. Comparably, **ALL-Ensemble** is trained using the nonlinear ensemble logistic regression model. The combination results as well as a comparison are summarized in Table 3. For the influence of each single latent aspect on the same prediction task, we present them correspondingly in Table 2, where Si, Ep, Ct, Tp, and Fr represent Student Influence, Expert Participation, Content Matching, Thread Populratiy, and Friendliness respectively.

Single Model	Precision	Recall	AUC
Si	0.697	0.696	0.791
Ep	0.602	0.590	0.572
Ct	0.626	0.616	0.647
Tp	0.594	0.579	0.626
Fr	0.639	0.633	0.685

Table 2: Prediction Result of Single Latent Factor

Model	Precision	Recall	AUC
EndThx	0.629	0.612	0.593
Si + Ep	0.803	0.802	0.857
Si+Ep+Ct	0.819	0.815	0.884
Si+Ep+Ct+Fr	0.823	0.823	0.893
ALL-Linear	0.826	0.826	0.894
ALL-Ensemble	**0.831**	**0.831**	**0.896**

Table 3: Prediction Result

Looking at the five latent aspects, (1) we conclude that, starter influence has the most powerful influence on thread resolution. It improves a lot on the Precision metric, and 50.25% on AUC compared to the **EndThx**. It makes sense that, if a user posts a lot, and often helps answer others' questions, it is more likely that his/her question will get a lot attention; (2) Thread Popularity, by itself works better than the baseline under the metric of AUC. The features in this set are not so directly

connected to thread resolution from a conceptual standpoint compared to whether a thread ends with thanks. However, it unexpectedly achieves an AUC of 0.626, which is higher than the baseline. (3) For content matching, the precision is similar to that of **EndThx**, but in contrast, this model achieves a good improvement on AUC. Content matching describes the similarities between a question and a reply, which is a direct indication of whether the reply is trying to answer the question. (4) Friendliness has a significant predictive ability in connection with thread resolution. For the AUC, it offers about a 13% improvement over the baseline. It is reasonable that a resolved thread tends to be more polite, which means people use 'please', 'thanks' more than in other unresolved threads.

To build the ensemble models, we combine the latent factor sets in the order of their strength of estimated influence on resolveability. We firstly integrate the starter influence and expert participation, as we can see, it achieves significant improvement over the simpler baselines, with 28% higher on Precision, 31% on Recall and 45% on AUC. It even performs better on the three metrics than any of the single models in Table2. **Si+Ep+Ct** also gives a substantial increase on the metrics and when adding semantic content matching, **Si+Ep+Ct+Fr** is about 3% better than **Si+Ep** on precision and recall. This indicates that friendliness and content matching are capturing different aspects of the thread resolveability from starter influence and expert participation. Besides, the **ALL-Linear** performs best among all one layer regression models. This shows that even though thread popularity contributes least to resolved or not based on the SEM result, it gives a different perspective of the thread resolveability and is not to be ignored. When we applied our proposed ensemble regression model **ALL-Ensemble** using the five latent factor sets, we find that it outperforms all one layer logistic regressors, especially in Recall and Precision. This demonstrates that the two-layer ensemble logistic regression model's added representational power is needed for this problem.

6 Conclusions and Future Research

In this paper, we have focused on improving the thread resolveability in MOOC discussion forums. Our investigation is divided into two separate studies that leverage a common conceptual model involving five latent factors that are associated with thread resolution. Our first study validates the five latent variable structures using a SEM model, which helps us to validate our assumptions and hone in on those factors that are most promising to leverage in subsequent work. It enables us to assess the relative strength of each factor's influence on thread resolveability, and provides a foundation for the other study. The second study's focus is predicting thread resolution based on the first phase's findings. In addition to serving as a test of generality from trained data to unseen data, the predictive model may also have a practical benefit. In particular, thread resoveability identification could provide the potential to achieve a better allocation of valuable human resources to work on unresolved threads, which increases the potential for students to get their support needs met in Massive Open Online Courses. Our work is contenxtualized in the specifics of MOOCs as an online context including the particulars of interaction practices within those contexts. Thus, in addition to building on existing QA work in our feature engineering, we also introduce new directions, such as the linguistic modeling of speaker politeness, and conduct forms of latent semantic matching that have proven effective in dialogue systems.

However, we believe there is a need for further modeling in order to fully understand thread resolveability. A limitation of the current work is that it was conducted in only one course. Thus, we will be in a stronger position for moving forward if we explicitly address the question of generalizability across courses with further corpus based investigation. Besides, how to transfer the prediction models from forums with resolved buttons to ones that have no such affordances, which may be challenging because of differences in the distribution of behaviors.

Acknowledgement

This research was funded in part by NSF grants IIS-1320064 and OMA-0836012 and funding from Google.

References

Eugene Agichtein, Carlos Castillo, Debora Donato, Aristides Gionis, and Gilad Mishne. 2008. Finding high-quality content in social media. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, WSDM '08, pages 183–194, New York, NY, USA. ACM.

Eugene Agichtein, Yandong Liu, and Jiang Bian. 2009. Modeling information-seeker satisfaction in community question answering. *ACM Trans. Knowl. Discov. Data*, 3(2):10:1–10:27, April.

Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec. 2012. Discovering value from community activity on focused question answering sites: A case study of stack overflow. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 850–858, New York, NY, USA. ACM.

Girish Balakrishnan. 2013. Predicting student retention in massive open online courses using hidden markov models. Master's thesis, EECS Department, University of California, Berkeley, May.

Paul Barrett. 2007. Structural equation modelling: Adjudging model fit. *Personality and Individual Differences*, 42(5):815–824.

Kenneth A Bollen. 1987. Total, direct, and indirect effects in structural equation models. *Sociological methodology*, 17(1):37–69.

Eric Brill, Susan Dumais, and Michele Banko. 2002. An analysis of the askmsr question-answering system. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, pages 257–264, Stroudsburg, PA, USA. Association for Computational Linguistics.

Moira Burke and Robert Kraut. 2008. Mind your ps and qs: The impact of politeness and rudeness in online communities. In *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work*, CSCW '08, pages 281–284, New York, NY, USA. ACM.

Cristian Danescu-Niculescu-Mizil, Moritz Sudhof, Dan Jurafsky, Jure Leskovec, and Christopher Potts. 2013. A computational approach to politeness with application to social factors. In *ACL (1)*, pages 250–259.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. 1998. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28:2000.

Peter Jansen, Mihai Surdeanu, and Peter Clark. 2014. Discourse complements lexical semantics for non-factoid answer reranking. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, ACL '13=4. Association for Computational Linguistics.

Jiwoon Jeon, W. Bruce Croft, Joon Ho Lee, and Soyeon Park. 2006. A framework to predict the quality of answers with non-textual features. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 228–235, New York, NY, USA. ACM.

Yutaka Kabutoya, Tomoharu Iwata, Hisako Shiohara, and Ko Fujimura. 2010. Effective question recommendation based on multiple features for question answering communities. In *ICWSM*.

René F Kizilcec, Chris Piech, and Emily Schneider. 2013. Deconstructing disengagement: analyzing learner subpopulations in massive open online courses. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, pages 170–179. ACM.

Ann Lieberman. 1995. Practices that support teacher development: Transforming conceptions of professional learning. *Innovating and Evaluating Science Education: NSF Evaluation Forums, 1992-94*, page 67.

Yandong Liu and Eugene Agichtein. 2008. You've got answers: Towards personalized models for predicting success in community question answering. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, HLT-Short '08, pages 97–100, Stroudsburg, PA, USA. Association for Computational Linguistics.

Qiaoling Liu, Eugene Agichtein, Gideon Dror, Evgeniy Gabrilovich, Yoelle Maarek, Dan Pelleg, and Idan Szpektor. 2011. Predicting web searcher satisfaction with existing community-based answers. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 415–424, New York, NY, USA. ACM.

Rivindu Perera. 2012. Ipedagogy: Question answering system based on web information clustering. In *Technology for Education (T4E), 2012 IEEE Fourth International Conference on*, pages 245–246. IEEE.

Chris Piech, Jonathan Huang, Zhenghao Chen, Chuong Do, Andrew Ng, and Daphne Koller. 2013. Tuned models of peer assessment in MOOCs. In *Proceedings of The 6th International Conference on Educational Data Mining (EDM 2013)*.

John Prager, Eric Brown, Anni Coden, and Dragomir Radev. 2000. Question-answering by predictive annotation. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, pages 184–191, New York, NY, USA. ACM.

Mingcheng Qu, Guang Qiu, Xiaofei He, Cheng Zhang, Hao Wu, Jiajun Bu, and Chun Chen. 2009. Probabilistic question recommendation for question answering communities. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 1229–1230, New York, NY, USA. ACM.

Carolyn Penstein Rosé, Ryan Carlson, Diyi Yang, Miaomiao Wen, Lauren Resnick, Pam Goldman, and Jennifer Sherer. 2014. Social factors that

contribute to attrition in moocs. In *Proceedings of the first ACM conference on Learning@ scale conference*, pages 197–198. ACM.

Yves Rosseel. 2012. lavaan: An r package for structural equation modeling. *Journal of Statistical Software*, 48(2):1–36, 5.

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*, SIGIR '10, pages 411–418, New York, NY, USA. ACM.

Juyup Sung, Jae-Gil Lee, and Uichin Lee. 2013. Booming up the long tails: Discovering potentially contributive users in community-based question answering services. In *ICWSM*.

Qiongjie Tian, Peng Zhang, and Baoxin Li. 2013. Towards predicting the best answers in community-based question-answering services. In Emre Kiciman, Nicole B. Ellison, Bernie Hogan, Paul Resnick, and Ian Soboroff, editors, *ICWSM*. The AAAI Press.

Miaomiao Wen, Diyi Yang, and Carolyn Penstein Rosé. 2014. Linguistic reflections of student engagement in massive open online courses. In *Proceedings of the International Conference on Weblogs and Social Media*.

Diyi Yang, Tanmay Sinha, David Adamson, and Carolyn Penstein Rose. 2013. turn on, tune in, drop out: Anticipating student dropouts in massive open online courses. In *Workshop on Data Driven Education, Advances in Neural Information Processing Systems 2013*.

Diyi Yang, Miaomiao Wen, and Carolyn Rose. 2014. Peer influence on attrition in massive open online courses. In *Proceedings of Educational Data Mining*.

Wen-tau Yih, Ming-Wei Chang, Christopher Meek, and Andrzej Pastusiak. 2013. Question answering using enhanced lexical semantic models. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1744–1753, Sofia, Bulgaria, August. Association for Computational Linguistics.