Supporting Synchronous Social Q&A Throughout the Question Lifecycle

Matthew Richardson
Microsoft Research
One Microsoft Way
Redmond, WA 98052 USA
mattri@microsoft.com

Ryen W. White
Microsoft Research
One Microsoft Way
Redmond, WA 98052 USA
ryenw@microsoft.com

ABSTRACT

Synchronous social Q&A systems exist on the Web and in the enterprise to connect people with questions to people with answers in real-time. In such systems, askers' desire for quick answers is in tension with costs associated with interrupting numerous candidate answerers per question. Supporting users of synchronous social Q&A systems at various points in the question lifecycle (from conception to answer) helps askers make informed decisions about the likelihood of question success and helps answerers face fewer interruptions. For example, predicting that a question will not be well answered may lead the asker to rephrase or retract the question. Similarly, predicting that an answer is not forthcoming during the dialog can prompt system behaviors such as finding other answerers to join the conversation. As another example, predictions of asker satisfaction can be assigned to completed conversations and used for later retrieval.

In this paper, we use data from an instant-messaging-based synchronous social Q&A service deployed to an online community of over two thousand users to study the prediction of: (i) whether a question will be answered, (ii) the number of candidate answerers that the question will be sent to, and (iii) whether the asker will be satisfied by the answer received. Predictions are made at many points of the question lifecycle (e.g., when the question is entered, when the answerer is located, halfway through the asker-answerer dialog, etc.). The findings from our study show that we can learn capable models for these tasks using a broad range of features derived from user profiles, system interactions, question setting, and the dialog between asker and answerer. Our research can lead to more sophisticated and more useful real-time Q&A support.

Categories and Subject Descriptors

H.5.3 [**Information Interfaces and Presentation**] – Group and Organization Interfaces

General Terms

Algorithms, Measurement, Experimentation, Human Factors

Keywords

Synchronous social Q&A, question answering, predictive models

1. INTRODUCTION

Question-and-answer (Q&A) services provide searchers with an additional mechanism, beyond general-purpose search engines, through which to get questions answered. Web Q&A sites, such as Yahoo! Answers, allows askers to post questions and have others answer them. These sites are increasingly popular and can provide high quality answers [8]. However, they can also have high latency even when the number of registered users is large [11]. Research on predicting answer quality and asker satisfaction (e.g., [8][18][21]) has typically been used *a posteriori* (e.g., to rate unrated answers) rather than predict outcomes as questions progress.

Real-time Q&A services such as *Aardvark* [10], *Twitter Answers* (ask.mosio.com/twitter), *Zephyr* [1], or *IBM Community Tools* [23] address latency concerns and leverage synchronous communication channels such as instant messaging (IM) for question asking and answering. The synchronicity of these systems allows questions and answers to reach their users directly, leading to faster answers and faster updates on answers. These systems often provide a way for askers to rate answer quality, which can be used to improve the effectiveness of question routing, and understand and train models to automatically predict asker satisfaction.

Q&A services are typically not designed to balance the needs of the asker with the availability of the answerer; askers may want answers urgently, but broadcasting questions can result in many costly interruptions for potential answerers [11][23]. To manage asker expectations of question success and avoid interrupting answerers unnecessarily, it may be useful to accurately predict important outcomes such as the likelihood that a question will be answered, the rating that the answer will receive from the asker, and the number of potential answerers that will be interrupted with the question until an answer is received. Being able to make predictions throughout the lifecycle of the question, e.g., before it has been distributed to candidate answerers, or even before it has been asked (based on time, answerer availability, etc.), may help systems intervene early, saving wasted effort from the asker and answerer if a predicted outcome is negative. In addition, if question and answer dialogs are archived, predicting answer quality ratings may help construct higher quality knowledge repositories.

In this paper we present an investigation of predicting a variety of outcomes during the question lifecycle, using data from a live synchronous social Q&A system deployed to a community of over two thousand users. The system, called *IM-an-Expert* and developed by the authors, receives questions via IM, automatically identifies candidate answerers by ranking all users by representations of their interests and expertise, routes questions only to those available and most able to answer, and mediates the dialog between the asker and answerer. IM-an-Expert operates under the principle that all users are experts and ask or answer questions; to ask, users must be willing to answer. In our previous work [24], we conducted a user study analyzing the effect of IM-an-Expert parameter settings on s of community size and contact rate on IM-an-Expert performance. We have since deployed the system more broadly and gathered interaction data from real users. In this work we characterize the live system and leverage relevant aspects of user behavior in these data for the prediction of (i) whether a question will be answered, (ii) the number of candidate answerers

WWW 2011, March 28–April 1, 2011, Hyderabad, India.
ACM 978-1-4503-0632-4/11/03.

that will be interrupted by the question, and (iii) the answer rating obtained. As we demonstrate through experimentation, we obtain good performance in all tasks.

The remainder of the paper is structured as follows. In Section 2 we describe related work on the use of IM for Q&A, the prediction of answer quality and answer ratings in social Q&A settings, and quality predictions in other settings such as Web search (e.g., query performance) and in education (e.g., essay scores). In Section 3 we describe IM-an-Expert, and in Section 4 we describe the lifecycle of a question submitted to the system. Section 5 provides statistics on the usage of the service, such as availability by time of day. Section 6 describes our experimental setup, including the prediction tasks, data set, features used to train our predictive models, and the experimental findings. Section 7 discusses the findings and their implications, and we conclude in Section 8.

2. RELATED WORK

The most relevant research areas are (i) social Q&A, (ii) the use of IM for synchronous social Q&A, (iii) predicting answer quality and asker satisfaction in social Q&A, and (iv) predicting answer quality in other settings like query performance prediction.

Social question and answer (Q&A) services are designed to facilitate the transfer of information and expertise. Recently, such services have manifested themselves in the form of online community Q&A sites, where community members can post and respond to one another's questions. In recent years, community-based Q&A sites have become increasing popular. Yahoo! Answers, one of the most popular sites like this, has millions of users [15]. These sites offer an alternative to traditional search engines and act as communities where users can contribute and share expertise. Users can post a question to the site and other users in the community can respond and answer the posted questions. The generated knowledge can be archived and is valuable as a knowledge repository, especially if answer quality ratings are available.

IM is popular in virtual communities, and, despite its drawbacks for archiving and navigation, IM is informal and provides instant support for negotiation of meaning and the characteristics needed for free flow and sharing of tacit knowledge. IM has been used for some time as a way to support collaboration between individuals and within small groups [19]. There are also systems that use IM directly as a question-answering mechanism. *Zephyr* [1] allows users to send messages to a chat channel called an "instance" and questions are distributed to everyone who is subscribed to that instance at that time. *Mimir* [11], a market-based Q&A service, employs a strategy that is similar to an email distribution list (i.e., broadcasting all questions to all users), since they do not filter question recipients based on personalization. *Aardvark* [10] is an IM-based synchronous social Q&A system that removes the need for users to select the target of the question prior to asking. Instead, Aardvark automatically routes incoming questions to the asker's social network. Avrahami and Hudson [3] showed that they could build predictive models of answerer responsiveness to an incoming IM using status and incoming event features. However, in those settings, little work has been done on predicting answer *likelihood*, asker *satisfaction*, or *interruption costs* across candidate IM respondents who receive the question.

One setting in which answer quality has been studied in detail is in Web forums, where askers pose their question and others post answers. There has been significant interest from researchers on information seeking behaviors, selection of resources, and in particular on predicting asker satisfaction or answer quality in these online communities of askers and answerers.

Liu et al. [18] present a general prediction model of asker satisfaction in community Q&A, and develop a variety of content, structure, and community-focused features for this task. Through large-scale evaluation on data from Yahoo! Answers, they demonstrate the feasibility of modeling and predicting asker satisfaction given the question and the answer. They complement their findings with an investigation of the interactions and information-seeking patterns in Q&A communities that correlate with information seeker satisfaction. Others have moved beyond asker satisfaction to make predictions regarding the answer quality and factors that contribute to this quality. Shah and Pomerantz [21] evaluate and predict the quality of online answers again using Yahoo! Answers. They selected well-answered questions, and asked Amazon Mechanical Turk workers to rate answer quality for each question based on a number of criteria (e.g., completeness, readability, relevance) and matched their quality assessments with the asker rating. They showed that their quality criteria matched with asker's perception of a quality answer, and trained classifiers to select a question's best answer given features of the question, the asker, the answer, and the answerer. Harper et al. [8] investigate predictors of answer quality through a comparative, controlled field study of responses provided across several online Q&A sites. Along with several quantitative results concerning the effects of factors such as question topic and rhetorical strategy, they show that (i) answer quality was higher on a fee-based site than in the free sites, and paying more money for an answer led to better outcomes, and (ii) the community of users contributes to its success; sites where anybody can answer questions outperformed those sites that depend on specific individuals to answer questions, such as library reference services. Jeon et al. [12] use non-textual features, such as click counts, to predict the answer quality in Q&A sites. They show that their quality measure can be successfully incorporated into language modeling-based retrieval model.

Moving beyond general predictions, Liu and Agichtein [17] hypothesize that satisfaction with the contributed answers is largely determined by the asker's prior experience, expectations, and personal preferences. They develop personalized models of asker satisfaction to predict whether a particular question author will be satisfied with answers contributed by community participants. They explore a variety of content, structure, and interaction features for this task using standard machine learning techniques. They show that it is beneficial to personalize satisfaction predictions when sufficient prior user history exists, significantly improving accuracy over a "one-size-fits-all" prediction model. Kim et al. [13] attempt to better understand how people seek, share, and evaluate information in a social Q&A environment. They identify the selection criteria people employ when they select the best answers in the context of relevance research. Using content analysis, they analyzed the comments people left upon selecting the best answers to their own questions and grouped comments into value categories (e.g., cognitive, socio-emotional, utility). Such findings demonstrate the importance of considering features of question context as well as the question itself in assigning ratings. Indeed, in our predictions, we include features of the asker, the answerer, and the question setting, such as time of day and the match between question and available answerer profiles.

Information retrieval (IR) and natural-language processing (NLP) researchers have studied the prediction of answer quality. In the question-answering community, there is growing interest in the automatic evaluation of answers to complex questions. Lin and Demner-Fushman [16] propose a technique for evaluating based on n-gram co-occurrences between machine output and a human-generated answer key. IR researchers have investigated the devel-

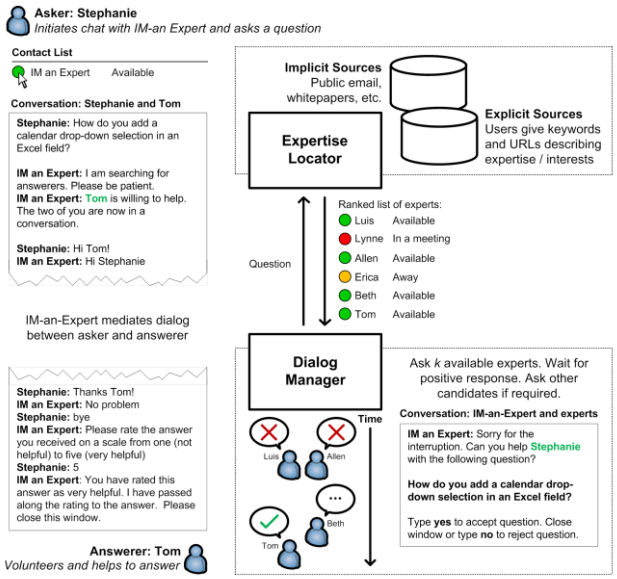

Figure 1. Components and interaction flow in IM-an-Expert.

opment of automated methods to estimate search engine performance and facilitate more efficient evaluation of new retrieval models, ranking algorithms, or presentation techniques. Prior work by Cronen-Townsend and Croft [6] and others has developed automatic methods for predicting query difficulty (i.e., how good the search results are for a query). Other prior work has modeled user interaction to predict search result preferences [2], search goal success [9], or query performance [7]. There is also literature on automatic essay grading (e.g., [14]) that may be relevant in the context of predicting answer quality.

Despite the large amount of research on using IM for collaboration (more recently for social Q&A), and the prediction of answer quality in Q&A websites, there has been no work on predicting asker satisfaction, answer likelihood, or interruption costs in real-time for synchronous social Q&A. Given the rise in popularity of real-time social Q&A services on the Web, it is critical to investigate ways to reduce the tension between asker and answerer demands in such settings. Though we conduct our study in a large online community within Microsoft Corporation, we believe that our findings are not limited to the enterprise. Because the instant messaging medium is similar between enterprise- and Web-based settings, we believe that our findings will extend to Web-based synchronous social Q&A, and would be useful to anyone interested in deploying social Q&A solutions at scale.

We now describe IM-an-Expert, the system used in our study.

3. IM-AN-EXPERT

IM-an-Expert is an automated service that receives questions via IM, locates and contacts potential answerers with expertise or interest in the question topic, and mediates the dialog between asker and answerer. Figure 1 illustrates the interaction flow in the IM-an-Expert service. The asker initiates an IM conversation with IM-an-Expert and poses the question. The question is used to retrieve a candidate set of experts based on profile information (described later). A small group of those experts who are currently available (not busy, away, in a meeting, in a call, etc. available via presence information from the IM client) are contacted via IM *three at a time*, in descending order of their expertise, to determine whether they are willing to help answer the question. If and when an answerer accepts, other requests are canceled. If a candidate answerer does not respond in time or rejects the question, the service asks others. Once an answerer accepts, IM-an-Expert mediates the conversation between the asker and answerer. Once the conversation ends, the asker is asked to optionally rate the quality of the answer they received on a scale from one (not helpful) to five (very helpful). To support the functionality described in this paragraph, the system has two components: (i) *expertise locator*: selects users who are most likely able to answer a question, and (ii) *dialog manager*: handles question processing and communication management throughout the question lifecycle.

3.1 Expertise Location

To locate subject matter experts, IM-an-Expert searches user profiles created from explicit and implicit sources:

Self-reported knowledge (explicit): We provide a Web interface where users can create and update a profile. The interface elicits keywords about which a user is knowledgeable. Emphasis can be placed on keywords by purposely repeating them. For example, a math expert can enter "math math math" to emphasize that area. Although this is not the ideal design (e.g., we could have provided a way to weight each keyword on a scale), the repetition concept is easily understood and keywords can be entered rapidly, which is important given the profile creation overhead. The interface also allows users to provide URLs of pages about them (e.g., their homepage). Periodically, and prior to profile indexing, the service downloads text from these URLs and adds it to profiles.

Email sent to mailing lists (implicit): For privacy reasons we cannot access all user email. Instead, we use email sent to internal mailing lists, available on a range of topics. We crawl and index these archives, collecting over 300,000 emails for around 30,000 people. Email is preprocessed to exclude headers and quoted text so that each profile contains only the user's authored text.

History (implicit): We associate with each user the questions that user answered, which allows the system to improve over time.

Text (excluding common stopwords) from all three sources is combined to form a textual representation of each user's interests and expertise. We now describe how we rank users for a question.

3.1.1 Expert Ranking

Selecting those who may *best* provide needed information is the *expert finding* problem, well investigated by the IR community e.g., [4]. Since the purpose in this investigation is not to develop the optimal ranking function, but rather to study social Q&A, we use BM25 [22], an established ranking function used extensively in IR research and practice. BM25 ranks profiles by considering the frequency of question words within each profile and the inverse of the frequency of those same words across all profiles.

In their profiles, users can configure the minimum time between questions, β, (default is 20 minutes) and limit the number of questions received per day (default is 15). Additionally, the BM25 score for each employee in relation to a question is multiplied by a decay function which has low value if this user was recently contacted. This helps balance question load across all users, an important factor in online communities [5]. The function is:

$$Decay = \begin{cases} 0 & \Delta t \leq \beta \\ 1 - e^{-\Delta t/\alpha} & \Delta t > \beta \end{cases}$$

where Δt is the number of minutes that have passed since the last question, and α is set to 120 / max questions per day. To prevent users knowledgeable about popular topics from being overloaded, each question is routed to at most 45 users. This set comprises the top users (up to 20) whose score exceeds a threshold, followed by 25 randomly-selected users that have not been contacted recently.

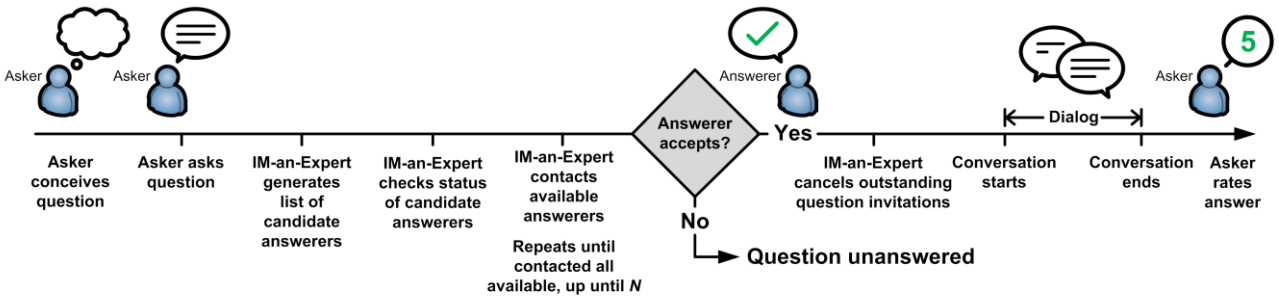

Figure 2. Question lifecycle in IM-an-Expert.

3.2 Dialog Management

A dialog management component coordinates the flow of messages between the asker and answerers. A session is initiated by a user asking a question via IM. The system determines the other users most likely to be able to answer the question and sends an IM to the top-*k* of them (in our study $k = 3$), asking if they are willing to help answer the question. Targeting only a small number of participants per question (rather than broadcasting IMs, as has been described previously [1][11]) helps ameliorate the effects of IM interruptions. The system only asks users who are currently available to answer. If a potential answerer does not respond within 45 seconds or indicates that she cannot help at the moment (by either closing the IM window or sending "no" in response to the invitation), the system contacts the next user on the list, until up to 45 users have been asked. The system informs the user whether they have been selected based on expertise or randomly. Expired candidate answerers who kept their dialog window open still have the option to accept the question until another user agrees to help.

Once a user indicates that she is willing to help, other potential answerers are thanked for their time and informed that they are no longer needed. The system then acts as a bridge between the asker and answerer, relaying messages from one to the other. In this way, the system has full control over the conversation; for instance, it can mask asker or answerer identity, record the conversation, and watch for termination. This also lets us record the dialog for later analysis, archival, and sharing (with permission). Upon completion, the asker is asked to rate the answer.

4. QUESTION LIFECYCLE

The question-answering process in IM-an-Expert can be represented in the form of a *question lifecycle* beginning with the conception of the question by the asker and running through to the conclusion of the answer dialog and the asker providing an answer quality rating. Figure 2 illustrates the question lifecycle in IM-an-Expert, comprising the following steps and opportunities for the system to use predictions to support the asker and answerers:

1. Asker conceives question: The asker decides on the question that they want to ask and formulates the text of the question. At this stage in the lifecycle, the system may provide general indications of availability based on time or day that could help the asker make an informed decision about whether to type the question.

2. Asker asks question: The asker types a question into an IM client and sends that question to IM-an-Expert. At this stage, the system knows more about the information need of the asker, and is able to provide indications to the asker on the likelihood of the question being answered or the likely quality of the answer based on past usage data. Providing such indications prior to the asker sending the question may prevent inappropriate or highly-difficult questions being sent to the system (a challenge in real-time Q&A settings, as highlighted in [23]), saving askers the wait time for a response and saving answerers from interruption.

3. IM-an-Expert generates list of candidate answerers: IM-an-Expert compares the question text to the profiles of other users and generates a ranked list of candidate answerers.

4. IM-an-Expert checks status of candidate answerers: The system obtains status information on all of the candidate answerers and prunes the set to those who are available at question time, while preserving their relative rank order in the list from Stage 3.

After steps 3 and 4, the current availability of users with expertise to answer is known and the system could communicate predictions to the user regarding answer likelihood, answer quality, and the estimated interruption cost of sending the question. If needed, the system could suggest that the asker reformulate their question or ask the question at a different time.

5. IM-an-Expert contacts available answerers: The system contacts the available answerers three at a time (in the order of the ranked list), waiting for each answerer to respond (or timeout) before contacting the next. If no one offers to help within a certain time then the question is unanswered. If an invited answerer agrees to answer then:

6. IM-an-Expert cancels outstanding question invitations: IM notifications are sent to those that have been contacted but have not yet replied, informing them that another user is answering and thanking them for their time.

7. Dialog between the asker and answer begins: The asker and the answerer engage in a dialog to resolve the question. This is another point in the lifecycle where predictions regarding the likely outcome or effort may be useful for the asker (to support a graceful exit if a good answer is unlikely) or for the answerer (to allow the answerer to redirect the question to another user or switch to a higher bandwidth communication channel, such as telephone, if the question seems likely to be challenging).

8. Asker rates the quality of the answer: The asker provides a rating on answer quality. IM-an-Expert can learn based on these ratings to improve the effectiveness of the question routing.

Following the termination of the clarification dialog the asker may fail to provide a rating. If we want to store the questions and their associated dialog in a knowledge repository for later retrieval, then we may want to automatically assign answer ratings or determine which unrated questions were well-answered.

The question lifecycle has a number of stages, and intervention at certain points throughout this process can help the askers make more informed decisions and reduce unnecessary interruptions.

5. USAGE OF IM-AN-EXPERT

In this section we present some general statistics on the usage of the IM-an-Expert service within our enterprise. We also present the findings of some additional analysis on availability, and investigate the effect of availability on system performance.

5.1 Summary Statistics

We deployed IM-an-Expert within a large enterprise (Microsoft Corporation) and solicited employees to use it to find answers to their questions. This initial publicity resulted in approximately 200 users. Throughout the following year (and to date), the system grew through word-of-mouth to over 2000 users. The data we use for this paper come from this latter period of time, after the initial publicity, when users were using the system and coming or leaving at will. IM-an-Expert was occasionally used by ourselves or a product team for demonstration purposes; we removed these questions from the set. The resulting data set has 1725 questions, 1009 (58%) of which were answered, of which 794 (79%) were rated.

Around 65% of our users asked at least one question on the service and a quarter of users asked and answered half of the questions. The average time-to-answer is 2 minutes 48 seconds (median = 1 minute 58 seconds), and the average answer rating is 4.1 (median=5). On average, dialogs between askers and answers lasted six minutes and comprised 10 dialog turns, evenly distributed between askers and answerers. Figure 3 shows histograms of the answer ratings obtained, the number of users interrupted per question, and the time to obtain an answer.

Figure 3a shows that our answer ratings are positively skewed, with most answers receiving a rating of five. In some cases the answer ratings could be attributable to properties of the question itself, since this is an important determinant in who was selected to receive the question and on whether invited answerers would accept the invitation.

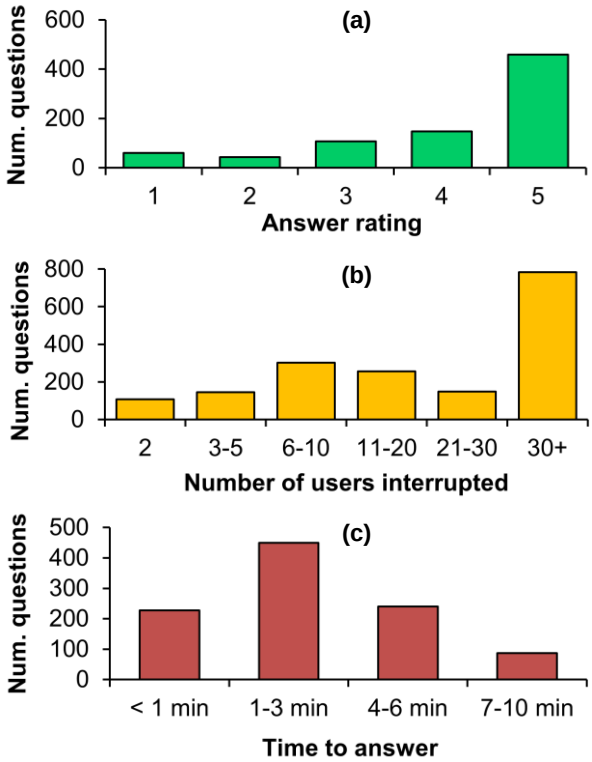

Figure 3. Histograms of (a) answer rating, (b) number of people interrupted, and (c) time to answer, in IM-an-Expert.

Table 1. Example questions and answer ratings they received.

Question	Rating
1. What is Unified Communications?	5
2. Is Veterans Day a Microsoft holiday?	5
3. Excel: how do I set default pivot table to "Classic"?	5
4. Can DPM backup based on VSS writer services?	1
5. Excel related: is there a way to have the "Classic PivotTable" as default in Excel 2007? When I create new pivots I have to go to "PivotTable Options" and then "Display" to change it to classic... I just don't like the "new" pivot format, so I use the classic all the time. Thanks.	1
6. I have a problem	1
7. Does Hitachi have a JBOD SATA option that a customer can use with Ex2010?	no ans.

In Table 1 we present examples of questions. The questions with high ratings in the table are simple (1), require a yes/no answer (2), or are clearly stated (3). Those questions in the table with a low rating (or unanswered) are confusing (4 and 7 have multiple acronyms), contain extraneous information and are potentially overwhelming (5), and ambiguous (6). Note that these are not representative of all questions received by IM-an-Expert; for many it is challenging to predict the rating based on the question. For example, the apparently simple "What is the capital city of Afghanistan?" received a rating of one, whereas the seemingly difficult "OCS TCP UDP question" received a rating of five.

Figure 3b shows that most answered questions only need to interrupt 6-10 answerers. When a question goes unanswered (which happened for 42% of questions), 45 people will be interrupted with an IM popup. This illustrates why it is important for synchronous social Q&A systems to be able to predict outcomes throughout the question lifecycle, even if only to dissuade askers from sending questions such as "I have a problem."

Figure 3c shows that over 20% of our answered questions receive that answer in less than one minute, and around two-thirds are answered in less than three minutes. This demonstrates the potential power of synchronous social Q&A, but it also illustrates the low latency that askers may come to expect from such systems, re-emphasizing the importance of balancing all users' needs.

5.2 Availability Effects

We suspected that answerer availability would have an effect on answer ratings, answer likelihood, and number of users interrupted, as well as the time to obtain an answer. To measure whether there were any effects, we compared the factors of interest with the availability of users across time. To measure availability, we ran a separate process that automatically polled the IM client status of all of users every five minutes over a three-week period. We recorded the fraction of users whose IM client status was *available* (not *busy*, *away*, *in a meeting*, *in a call*, etc.) suggesting that they could be asked a question at that time. Figure 4 shows the distribution of availability throughout the day.

The figure shows that time-of-day has a large effect on availability, with 3-4 times as many users being available between 9am and 5pm (the traditional work day) as are available at the start or end of the day (20% at peak time versus 5-7% in morning or evenings). There is a slight dip in availability around noon (presumably for lunch) and there are more people available in the evenings than in the mornings. The distribution is consistent over days and weeks, and slightly compressed on weekends.

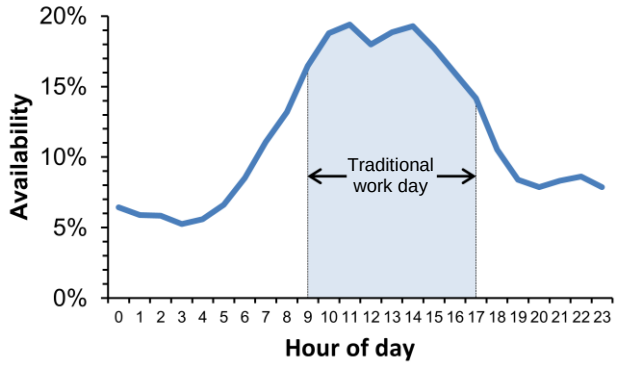

Figure 4. Availability throughout the day. Traditional work day (9am – 5pm) is highlighted in light blue for reference.

5.2.1 Time-to-Answer

To measure the time-to-answer we computed the time span between the time that the question was posed and the time that an invited answerer agreed to answer. Figure 5 shows the time-to-answer throughout the day, including the standard error (in pink).

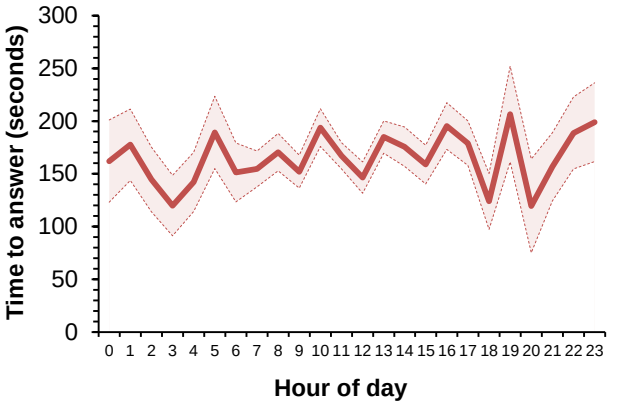

Figure 5. Time-to-answer throughout the day (± SEM).

Figure 5 shows that time-to-answer varies on an hourly basis throughout the day, it is typically in the 100-200 second range. There is more variance in the time-to-answer later in the evening and throughout the night (from 6pm to 6am), because the system receives fewer questions during that time. Most importantly, from these findings, it appears that time-to-answer is largely unaffected by the fraction of users that are available (Pearson's correlation (R) = 0.04). It may be that 5-7% of our user population is still sufficient to answer questions, although we wanted to probe more deeply regarding answer ratings, likelihoods, and interruptions.

5.2.2 Answer Ratings, Likelihoods, and Interruptions

In a similar way to time-to-answer, we investigated answer ratings, likelihoods, and interruptions and found that over days and weeks, answer rating held constant at around 4.1 (out of 5), answer likelihood held constant at around 60%, and the number of candidate answerers who were interrupted per question held constant at around 6-10 users. All correlations between these factors and answerer availability were low (all $R < 0.07$), suggesting little or no relationship between availability and question outcomes.

It may be that even though the fraction of people available varies dramatically depending on the time of day, there is still sufficient expertise in our user pool to answer an incoming question. Indeed, many of the questions submitted to IM-an-Expert system could be answered by a reasonable fraction of users; 49% of questions were answered by users selected at random (i.e., users who do not have the question topic listed in their profile). We would therefore expect that the number of people interrupted prior to getting an answer, the likelihood of obtaining an answer, and the answer rating would be somewhat constant over time. That said, users appeared to notice a difference in the *quality* of the answers received by those returned by the search algorithm (average answer rating = 4.29) and those selected at random (average rating = 3.89) (differences significant at $p < .001$ with independent measures t-test, $t(792)=3.35$). This suggests that availability of those with expertise (determined at stage 4 of the question lifecycle) may be important in predicting at least the quality of the answer, and perhaps for our other prediction tasks too.

We now provide details on our prediction experiments.

6. PREDICTING QUESTION OUTCOMES

There are three primary tasks that we wish to explore: predicting whether a question will find an answerer (*answered*), predicting how many users will be interrupted during the search for an answerer (*interruptions*), and predicting whether the asker will be satisfied with the answer received (*satisfied*). Each of these tasks corresponds to a number of prediction problems, based on where in the question lifecycle it is considered. For example, we can try to predict how satisfied the asker will be based on the question, or once we have found an answerer but before the dialog has begun, or throughout the dialog. To perform these predictions, we use logistic regression on features computable at runtime. We begin by describing the prediction setting and the features. Following that, we present results along with implications.

6.1 Prediction Setting

For all three tasks, we use machine learning to predict the target—answered, interruptions, or satisfied—given a set of features of the asker, the answerer, the question, and the question setting.

For the *satisfied* task, we use the 794 questions which received a rating, and consider the asker to be satisfied if the rating was at least three (out of five). This concurs with previous work by Liu et al., who tackled a similar problem on Web Q&A forums [18].

For the other tasks, we use the full set of 1725 questions. In *answered*, a question is considered to be answered if there was at least one user who was willing to try to help the asker (note, this task does not predict whether the question was answered well, or whether the asker was satisfied with the answer, simply whether an answerer at least tried to help). For *interruptions*, the target was the number of users (including the eventual answerer) who received an IM asking for help. This number ranges from 1 to 45 (the maximum allowed before cancelling a question).

Because our training set is small, and we have a large number of features (see below), we opted to use logistic regression for the two classification tasks (*satisfied* and *answered*). To prevent overfitting the training set, we employed both L1 and L2 regularization. Both are common forms of regularization; L1 encourages sparsity (reduces the number of features with non-zero weight), and L2 encourages weights to be close to zero. Attempts to use more complex learners (boosted decision trees and averaged perceptron) led to no better results, likely due to the small amount of training data. For the regression task (*interruptions*), we used stochastic gradient descent to find a linear combination of feature values that optimizes squared loss with respect to the target value.

Each result is obtained through ten-fold cross validation. Statistical testing is performed using a paired t-test (pairing each fold across the 10 folds) at $p < .05$ unless otherwise stated.

6.2 Features

The features are broken in to four sets, associated with the different stages of the question lifecycle: *conception* (before the question is asked), *question asked* (once the question and the list of potential available answerers are known), *answerer found* (once a user has indicated willingness to help), and *dialog* (while the asker and answerer are in discussion over IM). The features are accumulated; for example, predicting *satisfaction* once the answerer is found involves all three feature sets: *conception*, question *asked*, and *answerer found*. We now describe each set's features.

6.2.1 Question Conception

At this stage in the process (stage 1 in the question lifecycle), the system is only aware that an asker is about to ask a question. At this stage, what is known is simply the asker's identity, and statistics about the current time. The features thus include the day (7 binary features), hour, whether it's a weekday, and the average availability of users at this time of day (i.e., what fraction of users are typically online at this time, shown in Figure 4). Availability statistics were computed by polling availability over the course of three weeks, collected after the time period in the data set. Such polling is resource intensive. Using the average (rather than actual) availability allows us to conduct this polling less frequently.

We also compute features based on the asker's profile: the number of expertise keywords provided, whether the asker elected to be anonymous, whether he is willing to accept random questions, and the maximum number of questions the asker is willing to receive per day. In total, this phase contains 49 features.

6.2.2 Question Asked

At this stage, the system knows the question and has also retrieved the candidate answerers (end of stage 4 in the question lifecycle). We therefore compute features about the question, including its length (characters and words) and whether it contains a newline or tab (these are uncommon in IM conversations and may indicate the question text was pasted from elsewhere). We also include binary features indicating whether the question contained the words "help," "please," "thanks," or ends with a question mark, and six more features indicating individually whether the question starts with "who," "what," "why," "where," "when," or "how".

We also include statistics about the list of potential answerers that the expertise locator retrieved. We compute the average and the maximum retrieval score among the set of users, and again for the set of users who are actually available at question time. The first indicates general knowledge of the question among the system users, which may indicate the generality of the question, whereas the second indicates the quality of the potential-answerer pool. In all, this phase adds 22 features to our overall feature set.

6.2.3 Answerer Found

At this stage, the system is aware of who is going to attempt to answer the question. We generate features based on the answerer's profile as was done with the asker in 6.2.1 (profile size, anonymity settings, etc.). Additionally, we include information about the degree of match between the answerer and the question: the retrieval score, whether they were selected randomly or by retrieval, and their rank in the list of potential answerers. We also include the time that elapsed between the question and the answerer offering to help. In all, this phase adds 29 features.

6.2.4 Dialog

Once the answerer has accepted the question, the asker and answerer engage in IM dialog. At any point in the dialog (e.g., partway through, or when the dialog is complete), we may find it helpful to predict whether the question will be well answered. We thus compute a set of features based on the dialog to-date: the time elapsed, total number of messages by the asker (answerer), characters typed by the asker (answerer), fraction of turns taken by the asker, and whether the last turn is by the asker or answerer. We also provide four binary features of whether the answerer says "I don't," "I can't," "sorry," or "don't know." Similarly, whether the last thing typed by the asker is "thank you" or does it end with an exclamation mark. Also, whether the asker (answerer) typed a smiley or frown emoticon or URL. We also include whether the last thing said is by the asker has a question mark or is "no problem" or "you're welcome" (or variants thereof). We will refer to the 26 features computed when the dialog is halfway or fully complete as the *50% dialog* and *post-dialog* phase, respectively.

6.3 Results

In this section, we present results for the *satisfaction*, *answered*, and *interruptions* tasks, at various points in the question lifecycle (*conception*, *question asked*, *answerer found*, and *dialog*).

6.3.1 Satisfied Task

We first present the results for the *satisfied* task. Recall that the task is to predict whether the asker will be satisfied by the answer to the question, as indicated by giving it a rating of three or more (hereafter referred to as the *satisfied* task, since we also explore a very satisfied variant, *satisfied4*, later in this section). In evaluating model performance we use precision (P) (the fraction of instances predicted to be true that were actually true) and recall (R) (the fraction of all true instances that were predicted to be true).

6.3.1.1 Basic Results

In Figure 6 we show the precision-recall (PR) graph of our results for *satisfied*. The prior precision (always guessing that the asker will assign a rating of three or more) is 87.2%. As shown in the figure, the precision at reasonable levels of recall is significantly higher (e.g., 96.4% precision at a recall of 50%, marked).

For simplicity of exposition, the remaining results are presented at recall levels of 25%, 50%, and 75%. The "correct" recall level is dependent on the task and the relative cost of making a mistake. On our tasks, we generally feel precision is more important than recall, since the remedy is usually to interrupt the user and, e.g., ask them to rephrase the question or if they want to be connected with a new expert. We thus tend to focus on recall levels of 25% in our discussion. In Table 2, we present results for the *satisfied* task at various stages of the question lifecycle.

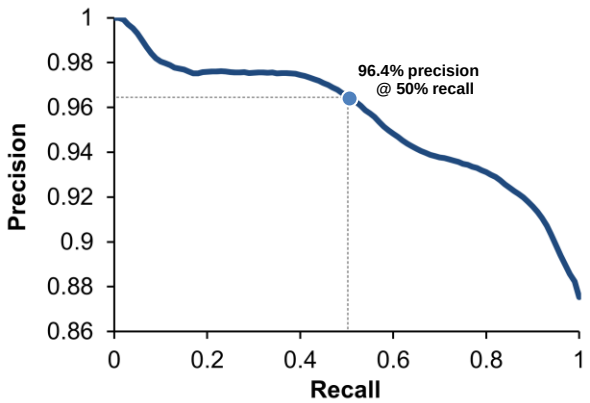

Figure 6. PR curve for *satisfied* task using entire feature set. Precision (96.4%) at 50% recall is marked for reference.

Table 2. Results for the *satisfied* task at various recall levels. Bold indicates statistical significance ($p<0.05$) vs. prior. Plus symbol indicates significance vs. the previous stage.

Question Stage	P@25%	P@50%	P@75%
Prior	87.2	87.2	87.2
Conception	89.2	**89.8**$^{+}$	88.3
Question asked	88.4	**89.7**	**88.6**
Answerer found	**90.9**	**89.1**	**88.3**
50% dialog	**96.7**$^{+}$	**93.2**$^{+}$	**91.5**$^{+}$
Post-dialog	**97.6**	**96.4**$^{+}$	**93.6**

Overall, we are able to improve precision significantly. Unsurprisingly given limited information, we achieve only relatively small improvements in precision during the early stages of the question lifecycle (*conception*, *question asked*). However, once the answerer is known, and particularly once the dialog is progressing, precision increases significantly. There is a curious (though not statistically significant) drop in precision after the question is asked, possibly indicating that the learner is over-fitting the training data (without over-fitting, the learner should have at least the same performance as during conception, since the features available are a strict superset of those available during conception).

6.3.1.2 During the Dialog

The findings in Table 2 show that the most significant gains in precision occur once the dialog is underway. To better understand the changes in precision throughout the course of the dialog, we divided up the dialogs into five portions (e.g., 20% of dialog, 40% of dialog, etc.) and made predictions regarding whether the asker will be satisfied after each portion. In Figure 7 we plot the precision over time as a function of the percent of dialog completion.

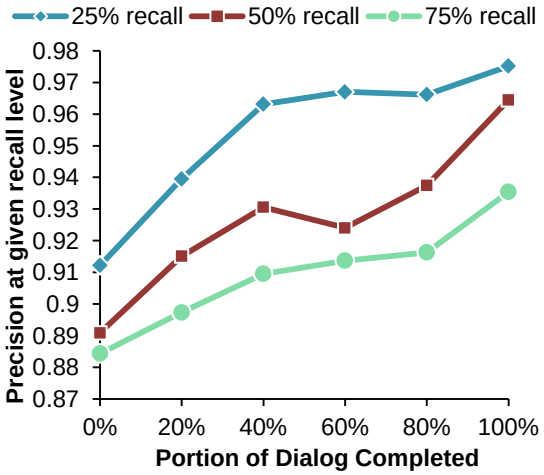

Figure 7. Performance on *satisfied* task during the dialog.

Figure 7 illustrates that precision increases steadily throughout the dialog (as opposed to increasing only near the end of the conversation), providing evidence that system support for askers could provide benefit at all points in the dialog.

In addition to the basic satisfied task, we also studied two important variants: the "very satisfied" task and the "reverse" task.

6.3.1.3 Very-Satisfied Variant

From reviewing the questions and answers, it appeared that ratings of four and five were assigned for good answers, whereas a rating of three was often offered for a fair attempt at answering. We ran a similar analysis to that reported in Section 6.3.1.1 for the *satisfied4* (or *very satisfied*) task, considering an asker to be satisfied only if he rates the question with a four or more (see Table 3).

Table 3. Results for the *satisfied4* task at various levels of recall, considering a rating of four or more as satisfaction.

Question Stage	P@25%	P@50%	P@75%
Prior	73.9	73.9	73.9
Conception	72.9	73.9	73.8
Question asked	75.5	74.0	73.9
Answerer found	73.8	74.9	74.3
50% dialog	**81.7**$^{+}$	**79.5**$^{+}$	**77.3**
Post-dialog	**90.2**$^{+}$	**86.9**$^{+}$	**81.6**$^{+}$

We found similar results to our earlier analysis in the *satisfied* task: precision climbs dramatically from a prior of 73.9% to 90.2% precision at 25% recall. Once again, we can see that dialog features deliver most of the performance gains; the learner outperforms the prior mid-way through the dialog, and obtains further significant gains from the remainder of the conversation.

6.3.1.4 Reverse Variant

We also consider the reverse task, predicting whether the asker will be *dis*satisfied with the answer provided. The reverse task is important if the system is to act on negative results, such as informing users they are unlikely to receive a good answer. As above, this can be done by considering satisfaction to mean *satisfied* or *satisfied4*. We only report the *dissatisfied* cases in Table 4, but summarize *dissatisfied4* performance in the subsequent text.

Table 4. Results for *dissatisfied* task at various recall levels.

Question Stage	P@25%	P@50%	P@75%
Prior	12.8	12.8	12.8
Conception	15.0	**16.9**$^{+}$	**13.8**$^{+}$
Question asked	17.2	**16.1**	**13.4**
Answerer found	16.1	**15.1**	**14.1**
50% dialog	**32.7**	**24.6**$^{+}$	**18.2**$^{+}$
Post-dialog	**45.0**$^{+}$	**35.7**	**23.1**

Because the prior is so low (12.8% precision), it is challenging to achieve high precision rates. With all of the features, the learner is able to raise the P@25% from 12.8% to 45%. As before, we see modest improvements in precision for the early stages of the question lifecycle. However, there is a significant increase in precision beginning even just halfway through the dialog with the answerer. Performance trends were similar for *dissatisfied4*, with P@25% increasing from a prior of 26.1% to 52.6% after the dialog was complete. These findings are promising, since they suggest that we may be able to predict in some cases, even if only at low recall, whether the asker is likely to be unhappy with the outcome of the dialog and can alert them of this prediction, invite additional answerers to the conversation, or provide a mechanism to re-ask the question to the community.

6.3.1.5 Feature Importance

The most useful features can be estimated by performing greedy feature selection; a set of "selected" features is initialized to be empty and each feature is evaluated independently to be added to the set. The best-performing feature is added and the process repeats until no further benefit is obtained. In Table 5 we list the features that were selected, in the order they were added. We give features for the *satisfied* task, for two phases, *post-dialog* and *question asked*. Note that the best-performing features may vary with each feature set as new features can subsume the benefits that other features were previously providing.

The features useful at *post-dialog* can be grouped as indicators of an inadequate answer (answerer says "sorry," last turn is question), availability of expertise (time to answer, retrieval score of

Table 5. Features selected by greedy feature selection (given in the order in which they were selected) for predicting *satisfied* in different phases of the question lifecycle.

(a) Features selected for the *post-dialog* phase
Characters typed by answerer during the dialog
Answerer says sorry during the dialog
Last thing said during the dialog is a question
Time to answer
Asker wants to be anonymous in transcript of dialog
The last thing said by the asker is thank you
The question begins with "How"
The retrieval score of the top available answerer candidate
The number of dialog turns taken by the asker
The number of characters typed by the asker
Number of characters in the question
The question contains the word "help"
(b) Features selected for the *question asked* phase
Asker wants to be anonymous in transcript of dialog
The retrieval score of the top answerer candidate
The question begins with "How"
The question begins with "Why"

top available answerer), question characteristics (number of characters, question contains "help" or "how"), asker dialog (number of asker turns, characters typed), and profile settings (anonymity in transcript – so that their name is not visible in archived questions). Similar features are also important for *question asked*.

6.3.2 Answered Task

In addition to predicting whether a question will be satisfied, we also predict whether a question will be answered in the *answered* task. Note that these predictions can only be made early in the question lifecycle, before a user has volunteered to answer. Table 6 shows precision values for the prior and the model at each stage.

Table 6. Results for the *answered* task at various recall levels.

Question Stage	P@25%	P@50%	P@75%
Prior	58.5	58.5	58.5
Conception	61.4	**60.1**$^{+}$	59.4
Question asked	**71.2**$^{+}$	**66.1**$^{+}$	**62.6**$^{+}$

The results shown in Table 6 are qualitatively similar to that of the early stages of the satisfied task; P@25% rises from a baseline of 58.5% to 71.2%, and gains at all reported precision levels are significant over the prior once we reach *question asked*. With a lower prior than *satisfied*, *answered* may be a more challenging task, but we are still able to obtain significant gains, even so early in the question lifecycle. Results of the reverse task (i.e., predicting if a question will not be answered) showed similar gains, with P@25% rising from a baseline of 41.5% to 53.6%.

6.3.3 Interruptions Task

Predicting the number of users that are interrupted by a question is a regression problem and is our third prediction task. Here, our metric is how closely the predicted number of users who received IM requests matches the actual number. We report two metrics: the mean absolute error (MAE) and the mean squared error (MSE). The learner was optimized for MSE. The results are presented in Table 7. As with the *answered* task, interruption predictions only make sense before an answerer has agreed to help.

Table 7. Results for the *interruptions* task for MAE and MSE.

Question Stage	MAE	MSE
Prior	10.77	152.5
Conception	**10.69**$^{+}$	**151.2**$^{+}$
Question asked	**10.13**$^{+}$	**141.7**$^{+}$

The findings in Table 7 show that we achieve a 6.0% and 7.1% reduction in MAE and MSE at the *question asked* phase. In real terms, this means that we are inaccurate with our predictions by around 10 answerers. The gains over the prior as we progress through the question lifecycle are encouraging, but the modest results demonstrate the challenging nature of the task and suggest that further development is necessary before we can provide user benefit.

6.4 Summary

We have presented experiments on predicting whether a user will be satisfied, whether a question will be answered, and the number of answerers who will be interrupted with a question. We showed good gains in accuracy for predicting satisfaction throughout the question lifecycle, and especially within dialog, with a smooth performance increase throughout. Results for answered and interruptions also show gains from the prediction models, but those tasks may be more challenging and the models lacked dialog information that was shown to be useful in *answered*.

7. DISCUSSION AND IMPLICATIONS

Synchronous social Q&A is an important method on the Web and within enterprises for resolving questions that urgently need a response and/or may require rapid clarification dialog. Questions submitted to synchronous social Q&A systems have a lifecycle in which they are asked, dispatched to candidate answerers, and answered via dialog. Using data from a Q&A system deployed to a large online community within our enterprise, we explored the task of making important predictions at key junctures in the question lifecycle. We showed a clear trend in availability depending on time of day, but also no relationship between any of the variables under study and time. We selected a large number of features derived from user profiles, system interactions, the question setting, and IM dialog for each prediction task that varied depending on where in the lifecycle the prediction was being made. We showed that as we move through the stages of the lifecycle, prediction accuracy in predicting answer ratings improves from the mid-80% to mid-90%. Interestingly, we also showed large gains from the dialog between askers and answerers, which were observed throughout the entire dialog.

We saw good gains over the baselines in prediction performance for the task of predicting whether a question will be answered and predicting the number of users that will be interrupted by a question. However, the accuracy numbers for these tasks are significantly lower: P@50% of 66% when predicting answer likelihood and a MAE of 10 for predicting the number of people that will be interrupted with a question. Although the lower performance may in part be related to the smaller number of features available for each of these tasks (both only include *conception* and *question asked* features), it is also likely that the tasks are simply challenging, and a rigorous failure analysis may be needed to better understand instances of good and bad performance.

Although this study was conducted within a large multinational enterprise rather than on the Web, there is nothing about the system or our experimental setting that is specific to an enterprise. Our findings have a number of implications for improving the design of synchronous social Q&A systems and the experience for their users. Accurate predictions early in the question lifecycle (i.e., after question conception or following question entry) may benefit askers by providing them with knowledge of whether their question is likely to be answered, the expected rating an answer will receive, or a quantification of the interruption cost that their question may incur (in terms of number of users interrupted). This may make them more efficient and less likely to abuse the com-

munity. Answerers may benefit indirectly from the voluntary throttling of questions that may occur from the increased asker awareness, although care will need to be taken when messaging optimistic predictions to askers, so as to not result in large increases in the number of questions posed. Once an answerer has volunteered, knowledge of the likely outcome can benefit both parties in terms of wasted effort and impact on reputation scores. At this point, and during the dialog, a predictor of answer ratings can monitor the conversation and handle closure and re-routing of the conversation to another answerer if appropriate, overcoming some of the social awkwardness associated with terminating an unsuccessful Q&A dialog prematurely. If questions and their associated dialogs are archived in knowledge repositories for later retrieval, an ability to accurately predict the ratings for unrated answers may improve the quality of the archive and be valuable for future retrieval, where those searching the archive may wish to filter results to those recorded dialogs with successful outcomes.

There are a number of other features that could be useful that take into account information that is known up to the point at which the question is asked. For example, in a similar way to [17], we can use personalized historic information such as the average rating that the answerer has received to date since it is likely to be a useful indicator of performance on the current question. In fact, we tested this hypothesis in our setting. Using historic features required a slight change to our test procedure, since each run of cross-validation must be ordered so that the test questions come after all of the training questions. We did this by randomly shuffling the order of questions, performing a 90-10 train/test split temporally, and repeating ten times. Including historic features appeared to hurt prediction performance. Although the reason for this is unknown, it may be because we included all historic information, some of which may be irrelevant for the current question.

Another useful feature might be unigrams and bigrams from the question/dialog. In the results above, we already include features for some words that we thought would be informative ("thanks", "please," "help," emoticons, etc.), but allowing the machine learning algorithm to select other words or phrases could lead to improved performance. With only 700 training examples, it will be crucial to add unigrams and bigrams in a manner that does not cause the model to over-fit the training data. In future work, we would like to explore such features, including variants of historic features or learners that depend only on recent history.

8. CONCLUSIONS

In this paper we have investigated support for users of synchronous social Q&A systems throughout the question lifecycle, from conception to answer. We conducted this study using data gathered from IM-an-Expert, a live IM-based system deployed to a large community of users within our enterprise. We showed that we can train models to predict answer ratings and likelihood with reasonable accuracy, particularly at lower levels of recall or later stages of the question lifecycle. Such models could be deployed in an operational setting to benefit all users of a synchronous social Q&A system. In future work we will add these models to IM-an-Expert, study user experience issues around incorporating predictions, and evaluate them via user studies and large-scale (Web-based) deployments.

9. REFERENCES

[1] Ackerman, M. S. and Palen, L. (1996). The Zephyr help instance: promoting ongoing activity in a CSCW system. *SIGCHI*, 268–275.

[2] Agichtein, E., Brill, E., Dumais, S., and Ragno, R. (2006). Learning user interaction models for predicting web search result preferences. *SIGIR*, 3–10.

[3] Avrahami, A. and Hudson, S. (2006). Responsiveness and instant messaging: predictive models supporting interpersonal communication. *SIGCHI*, 731–740.

[4] Balog, K., Azzopardi, L., and De Rijke, M. (2006). Formal models for expert finding in enterprise corpora. *SIGIR*, 43–50.

[5] Beenen, G., Ling, K., Chang K., Wang, X., Resnick, P. and Kraut, R. (2004). Using social psychology to motivate contributions to online communities. *CSCW*, 212–221.

[6] Cronen-Townsend, S., Zhou, Y., and Croft, W.B. (2002). Predicting query performance. *SIGIR*, 299–306.

[7] Guo, Q., White, R.W., Dumais, S.T., Wang, J., and Anderson, B. (2010). Predicting query performance using query, result, and user interaction features. *RIAO*.

[8] Harper, F.M., Raban, D., Rafaeli, S., and Konstan, J.A. (2008). Predictors of answer quality in online Q&A sites. *SIGCHI*, 865–874.

[9] Hassan, A., Jones, R., and Klinkner, K. (2010). Beyond DCG: user behavior as a predictor of a successful search. *WSDM*, 221–230.

[10] Horowitz, D. and Kamvar, S.D. (2010). The anatomy of a large social search engine. *WWW*, 431–440.

[11] Hsieh, G. and Counts, S. (2009). Mimir: a market-based real-time question and answer service. *SIGCHI*, 769–778.

[12] Jeon, J., Croft, W.B., and Lee., J.H. (2006). A framework to predict the quality of answers with non-textual features. *SIGIR*, 228–235.

[13] Kim, S., Oh, J-S., and Oh, S. (2007). Best-answer selection criteria in a social Q&A site from the user-oriented relevance perspective. *ASIST*, 44–54.

[14] Larkey, L.S. (1998). Automatic essay grading using text categorization techniques. *SIGIR*, 90–95.

[15] Leibenluft, J. (2007). A librarian's worst nightmare: Yahoo! answers, where 120 million users can be wrong. *Slate Mag*.

[16] Lin, J. and Demner-Fushman, D. (2006). Methods for automatically evaluating answers to complex questions. *Information Retrieval*, 9(5): 565–587.

[17] Liu, Y. and Agichtein, E. (2008). You've got answers: towards personalized models for predicting success in community question answering. *HLT*, 97–100.

[18] Liu, Y., Bian, J., and Agichtein, E. (2008). Predicting information seeker satisfaction in community question answering. *SIGIR*, 483–490.

[19] Nardi, B., Whittaker, S., and Bradner, E. (2000). Interaction and outeraction: Instant messaging in action. *CSCW*, 79–88.

[20] Ribak, A., Jacovi, M. and Soroka, V. (2002). "Ask before you search" peer support and community building with ReachOut. *CSCW*, 126–135.

[21] Shah, C. and Pomerantz, J. (2010). Evaluating and predicting answer quality in community QA. *SIGIR*, 411–418.

[22] Spärck Jones, K., Walker, S., and Robertson, S.E. (2000). A probabilistic model of information retrieval: development and comparative experiments (parts 1 and 2). *Information Processing and Management*, 36(6):779-840.

[23] Weisz, J.D., Erickson, T., Kellogg, W.A. (2006). Synchronous broadcast messaging: the use of ICT. *SIGCHI*, 1293–1302.

[24] White, R.W., Richardson, M., and Liu, Y. (2011). Effects of Community Size and Contact Rate in Synchronous Social Q&A. *SIGCHI*, to appear.