Distribution, Correlation and Prediction of Response Times in Stack Overflow

Preeti Arunapuram
Oracle
Santa Clara, CA USA
preeti.arunapuram@oracle.com

Jacob W. Bartel, Prasun Dewan
Department of Computer Science
University of North Carolina
Chapel Hill, NC USA
{bartel, dewan}@cs.unc.edu

***Abstract*—The sending of a message raises two important questions about its response: When will the first response arrive? When will the first acceptable response arrive? These questions can be partly or completely answered by identifying distributions of response times, correlating features with response times, and/or predicting the actual response times. We address distribution, correlation and prediction of response times in Stack Overflow. We analyzed response times of over two million question-answer threads. We found no strong correlation between response times and features studied in other messaging domains: (a) use of various kinds of pronouns and punctuations, and (b) the time of day, and day of week when messages were sent. We found that title lengths show a quadratic relationship with median response time and that mean response times vary according to the tags used in a post. We explored a large design space of prediction algorithms based on the distributions of response times. These approaches predicted ranges of time that were automatically determined using a clustering algorithm. The best results were given by an approach that combines, using an index-base weighted-average algorithm introduced here, the most frequent time-ranges in the distributions for the tags in the posts.**

Keywords-online forums; response time; prediction; Stack Overflow

I. INTRODUCTION

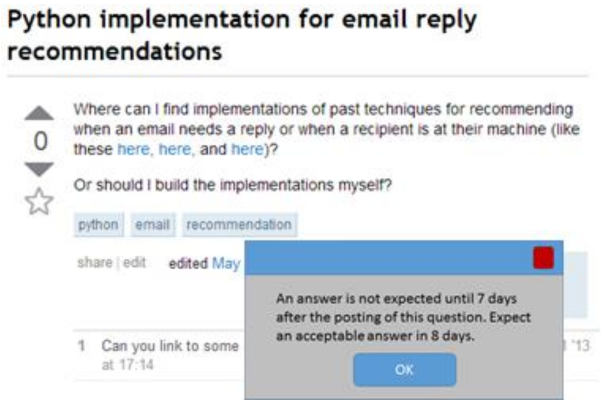
When we send an electronic or paper message to others, it is useful to know how long it will take to get a response back. When the message is sent to a group of users or is part of a thread in which further clarifications are made, the responses can be separated based on whether they were acceptable or not. It would be useful to know the time to the first response and the first acceptable response.

This information is useful because often times, the senders may urgently require a response to an important question. Being able to predict responses to messages would help them gauge whether or not they want to wait for an answer to a question before moving on to another information source. More subtle, the sender could tailor the post, by for instance shortening or expanding it or sending it a particular time of day, to improve response times.

If the messaging system is electronic, then it is attractive if the system could not only deliver messages but also make predictions regarding these times, as shown in Figure 1. Such prediction work can be divided into two categories:

- *Feature Correlation*: Identification of characteristics or features of available data that correlate with response times.
- *Time Prediction*: Prediction of actual response times.

The correlation work is often a prelude to time prediction, as the features, typically, form the input to prediction algorithms. It is useful in its own right as it can help senders tailor their message contents and times, as mentioned above.



Python implementation for email reply recommendations

Where can I find implementations of past techniques for recommending when an email needs a reply or when a recipient is at their machine (like these here, here, and here)?

Or should I build the implementations myself?

An answer is not expected until 7 days after the posting of this question. Expect an acceptable answer in 8 days.

1 Can you link to some
at 17:14

Figure 1. Mockup of Ideal Prediction System

There has been some research on response-time correlation in various messaging domains but, to the best of our knowledge, none on the much more difficult problem of prediction of response times. None of the correlation research has looked at Stack Overflow - an online forum on which people can post and respond to questions concerning computer programming. In this paper, we report on initial research on both correlation and prediction in Stack Overflow that assumes no prior knowledge about senders and recipients of messages.

This work is preliminary in that it is far from the system of Figure 1. However, it addresses some important issues worth investigating on our path to such a system:

- *Investigation of existing features*: Which message features from other message domains correlate with response times in Stack Overflow?
- *New features*: Are there any new message features that correlate with these response times?

- *Feature-less distribution-based prediction:* What is the design space of some apparently promising ways to predict response times that do not consider any features and look only at the distribution of response times?
- *Feature and distribution-based prediction:* How do these algorithms compare with those that consider both distributions and message features?

The rest of the paper is organized as follows. In the next section, we describe previous work on response times in various messaging domains, identifying some of the correlating features in these domains. Next, we discuss how well these features and, the additional feature of title lengths, correlated with the answer times in the Stack Overflow data we analyzed. We then address distribution-based prediction, with and without the features. We compare these predictions using a new metric, and end by presenting our evaluation results, conclusions, and directions for future work.

II. Related Work

A large body of work has analyzed and made predictions about how users will react to shared information. Perhaps one of the earliest reactions analyzed is the rating a given user will give to a shared piece of information. Resnick et al. [9] and Sarwar et al. [10] both attempted to make predictions about user ratings for Usenet posts using collaborative filtering. Resnick et al. developed a tool called GroupLens that predicted 5-star Likert ratings for a given user and a given item based on similar users. Sarwar et al. similarly predicted 5-star Likert ratings for a given user and item, but made predictions based on similar items rather than similar users. Similar approaches have since been used to predict user ratings or preference for a variety of types of items, such as movies [5, 8], products [6], and music [5].

Our work addresses a particular form of reaction to shared information – message response times. These have been studied to some extent in almost all domains that allow message exchange.

Arguello et al. [1] addressed Usenet groups. They studied the effects of pronoun choice, linguistic complexity, rhetoric, and context on community responsiveness in these groups as well as several other factors such as whether the poster is a newcomer and if the post has been cross listed. The focus was on determining links between these features and responsiveness, rather than the actual prediction of response times. They considered not only when answers w generated, but also how communication flowed on specified Usenet threads.

In a work more relevant to the domain of this study, Teevan, Morris, and Panovich [11] addressed responsiveness in social networks. They assessed factors that affected response of questions posted in social network statuses. Two factors led to faster responses: explicitly framing the post as a question, and using only one sentence in the post.

Similarly, Dabbish et al. [3] considered the domain of email. As messages are directed at specific set of recipients, it is possible to investigate responsiveness of specific recipients. They looked not a response times but on *liveness* - whether a message will result in a response. They studied the effects of message content, the importance of a message, and where a message would be filed on liveness.

Avrahami and Hudson [2] addressed the more real-time domain of instant messaging. They studied both timing and liveness in this domain. They focused on factors such as the time of day, computer activity (e.g. mouse movements), importance of tasks being performed by the receiver, the relationship between the sender and the recipient, and demographic characteristics of the receiver, such as age and gender.

As we see from this discussion, previous research has not considered the domain of Stack Overflow. Moreover, previous work has not attempted prediction of response times in any domain. Addressing both limitations raises special challenges. Stack Overflow posts contain code. Thus, features from other domains are difficult to apply directly to this domain. Prediction of response times in any domain creates further challenges in comparison to prediction of other human activities:

- *Multi-human dependency*: In other cases, a prediction is made about a single person – such as the rating the person will give to a bought item or a Usenet post. In community domains such as Stack Overflow, a response may come from one or more members of a large group. Thus, apparently, the response times depend on a larger number of factors.
- *Human vagaries:* In other cases, the correct answer is known by the human involved. For example, given a Usenet message, the users rating it know what rating they want to give. In the case of message response times, recipients themselves may not know when they will respond until they have responded. It is possible, however, that these vagaries are neutralized by having a large number of potential responders.
- *Large number of choices*: Typically, the range of choices among which a prediction system picks is relatively small. Often the choice is binary – for example, will a user like an item or not. Even in the message rating example, the number of choices is fairly limited. In the case of messaging systems in general and Stack Overflow in particular, the range of response times is very large, varying from a few seconds to months.

Correlation and prediction are only two aspects of responsiveness that have been studied in the literature. A third class of research, related to this work, has to do with identifying distributions of response times. A study of email by Kalman and Rafaeli of over 16000 sent emails [4] shows 97% of users responded to at least 30% (70%) of emails within a day (5 days). A study of Stack Overflow by Mamykina et al. [7] found that questions received the first answer with a median time of 11 minutes, and the accepted answer with a median of 21:10 minutes. They did, however, find a long tail lasting to several years. We contribute to this work by presenting a more recent report on distribution response times and identifying mean response times for certain popular tags.

Previous distributions-based research suggests an apparently simple initial step to predicting response times: Use the distributions of response times to predict the response time for a particular post. This approach, as stated above, does not take into account any features of the message. Thus, an alternate approach is to somehow combine distributions with features. These are the two directions we pursued in the prediction part of the project. The second approach requires identification of appropriate relevant features in Stack Overflow. Both approaches require us to gather data to determine the features and determine distributions. Let us consider the data gathering task next.

III. Data Set

For this study, we used the Stack Overflow public data dump from September 2012. . We then filtered the data set to only include questions that had at some point received an accepted response, which left us with over 2 million question-answer threads.

For each of question, the data set included the question ID, question title, the creation date of the post, the creation dates of all answers given in the post, the owner of the question, and any subject tags associated with the question.

Data about the answerers of the thread was also accessible, but we wished to predict answer times without prior knowledge about the responders. Thus, this information was not helpful in this study. We were concerned with predicting both the time until the first answer to a question, and the time until the accepted answer.

IV. Investigating Features Linked to Response Times

As mentioned previously, one of our goals was to identify features that are linked to response times in Stack Overflow. However, rather than start from scratch, we chose to use the wealth of knowledge from past work to focus on some of features that have shown to have links to response time in other systems. Naturally, investigating all of these features is an arduous task beyond the scope of one paper. The features we investigated are as follows.

A. *Title length*

Teevan, Morris, and Panovich [11] investigated the effects of limiting the number of sentences in a post. Their results indicated that posts that were only one sentence long achieved faster responses than posts that were longer. Analysis of the number of sentences and words in Stack Overflow does not work because Stack Overflow questions often contain snippets of code which cannot be translated into words and sentences. They could be translated into program units such as lines, variables and functions. Instead, we used a simpler feature - the title length in words - to see whether longer post titles resulted in faster or slower answers.

We found no direct correlation between title lengths and elapsed times of either type. However, the distributions of elapsed times tended to change systematically over different title lengths. Upon closer inspection, we found that the median elapsed time of each of these distributions shares a strong quadratic relationship with the title length in words. This relationship amongst the medians is shown in Figure 3. As evidenced by the R^2 values in the figure, we found that the association between median elapsed times by title length and title lengths themselves was stronger with accepted answer response times.

B. *Keywords*

Arguello et al. [1] found that certain words affected the responsiveness and response quality of a given Usenet thread. More specifically, the use of 3rd person pronouns increased the likelihood that a response would be generated. In the same vein, we wanted to investigate the effect of such variables on the response time.

The three categories of particular interest were 1st person, 2nd person, and 3rd person pronouns. Each data entry was given a binary feature for each of the categories: 1 if the pronoun was found in the post title and 0 if it was not. Then we compared the distributions of answer times for questions that contained the feature and those that did not. With the available data, we saw no difference in the time taken to generate the first answer or an accepted answer if 1st, 2nd or 3rd person pronouns are present in the question title. Additionally, we performed principal component analysis (PCA) for any signs of redundancy among different pronoun groups. However, PCA showed no such indications, and thus the data matrix could not be reduced in dimension. If such clustering had been discovered, it would be an indication that certain features correlated with each other with the presence or lack thereof of the pronoun group in question.

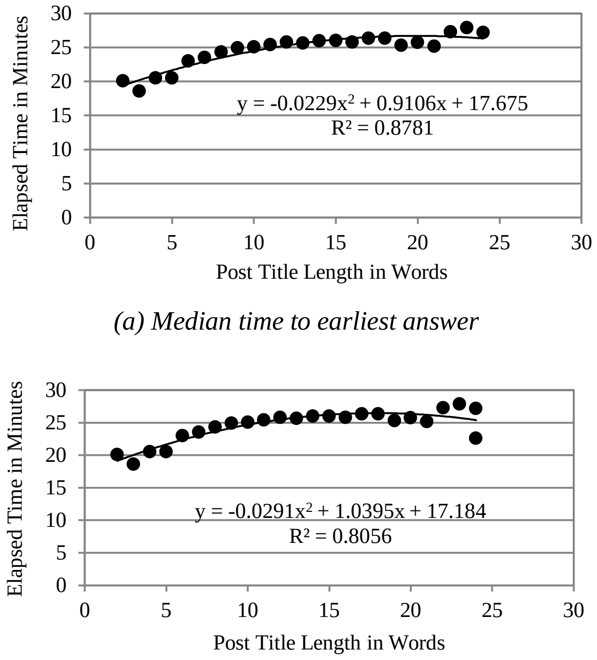
C. *Punctuation*

Teevan et al. [11] found that explicitly framing social networking posts as questions rather than as statements generated faster responses. We wanted to investigate the impact of punctuation on response times. Post titles were checked for occurrences of punctuation such as question marks, periods, exclamation points, semicolons, and others that signify the finishing of a thought. Contrary to the findings in Teevan et al. [11], the presence of punctuation did not lead to significantly faster response times in Stack Overflow posts. The difference between the median answer times for those posts that contained punctuation and those that did not were 1.49 minutes and 1.61 minutes for earliest answers and accepted answers, respectively. Moreover, the specific presence of question marks also did not lead to significantly faster response times, with the median differences between questions that contained question marks and questions that did not amounting to 3.25 minutes and 5.13 minutes for earliest answers and accepted answers, respectively.

There may be several reasons for this result. For one, post titles are oftentimes not framed as complete thoughts. Question marks may have had no impact because of a possible underlying assumption that those who post on Stack Overflow seek to have a question answered. This differs from the general social networking atmosphere in which it is unknown whether poster are asking questions in their status or simply making statements.

D. *Time of day*

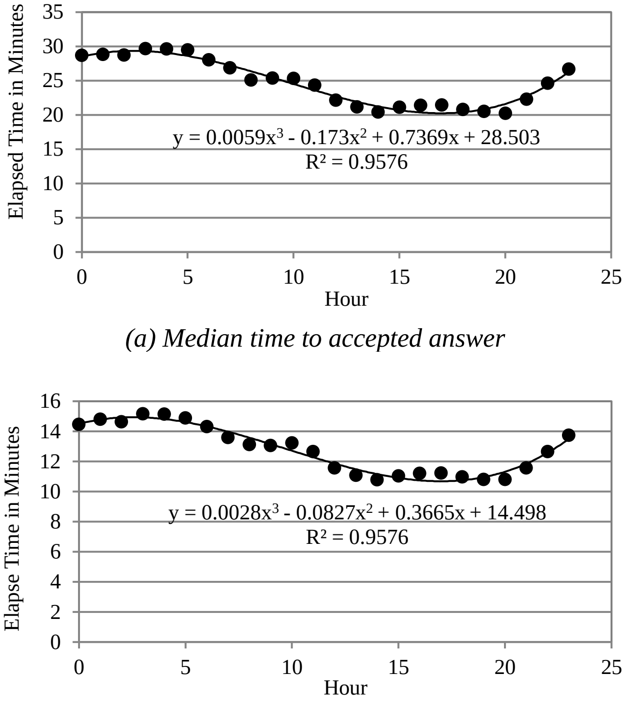
Avrahami and Hudson [2] found that both the day of week and the time of day influences response speed in instant



(a) Median time to earliest answer

(b) Median time to accepted answer

(c) Figure 3. Median elapsed times and title lengths



(a) Median time to accepted answer

(b) Median time to earliest answer

Figure 2. Elapsed time by hour of posting

messaging conversations. Specifically, they found that responsiveness improved during the morning hours and at night compared to the afternoon.

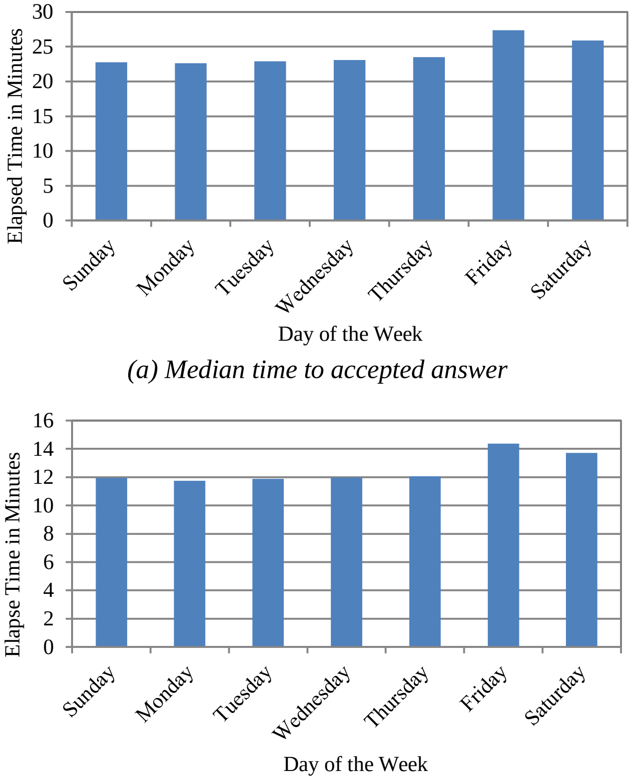
Table 1. Median elapsed times by tag for some popular tags

Median Elapsed Times (Minutes)		
Tag Name	**Accepted**	**Earliest**
algorithm	13	3
android	4	4
apache	29784	29784
api	3476	3451
arrays	5	2
asp.net-mvc-3	4904	4904
database	12	12
debugging	316681	162559
facebook	22	22
forms	61821	62
function	5	5
git	3741	404
google-app-engine	189	189
html	292558	292558
osx	30883	796
qt	10269	737
query	1	1
ruby-on-rails-3	18	18
ruby-on-rails	181	17
svn	51976	32
swing	150	146
tsql	16	3
unit-testing	14	14
validation	47	47
visual-studio-2008	1	1
visual-studio-2010	8	8
visual-studio	190060	21018
wcf	25	25
web-services	7	7
winapi	20	6
windows-phone-7	1477	1477
windows	1696	81
wordpress	1053	1053

To test this feature in Stack Overflow, we extracted the time of day and the day of week according to the UTC time zone, and compared against response times. In the case of time of day, we took the medians of accepted and earliest elapsed answer times and plotted them against the hour of posting, which is shown in Figure 2. The x-axis shows the hour number, where hour 0 represents 12:00 AM. For the day of week analysis, we plotted a bar chart of the median elapsed times by the day of week, which is shown in Figure 5.

Contrary to Avrahami and Hudson’s findings, the day of week did not have a strong relationship with response time. The time of day seems to exhibit a pattern when plotted against both accepted and earliest answer elapsed times, and when fitted to a three degree polynomial curve, had a moderately strong correlation. However, the data seems to systematically rise and fall around this fitted curve, indicating that response times may be driven by multiple distributions rather than a single one. This is consistent with the fact that unlike IM-messages, Stack Overflow posts (a) are directed at multiple people living in different work cultures and time-zones, and (b) do not contain “frivolous” conversations to be relegated to non-work morning and night hours. Therefore, the distribution of response times by time day may differ base on the time zone of the answerer.

It may be possible to find a better fit for these time-of-day data points with a higher order polynomial. However, this also increases the risk of over fitting the data and ignores the possible mixed-model nature of the distributions. Consequently, we did not use the time of day as a factor in our predictions.



(a) Median time to accepted answer

(b) Median time to earliest answer

Figure 5. Median elapse times by day of week

E. *Subject tags*

The idea behind using subject tags on Stack Overflow is to gear a question toward a more specific audience. For example, a person may post a question relating to C++ and include the C++ subject tag so that people who generally answer question relating to C++ may view it more quickly. As such, it could be that certain community groups on Stack Overflow simply respond to questions faster than other community groups do.

This feature has not been heavily researched in past works on factors influencing response speed. It is suggested by the research of Arguello et al. [1], which showed that cross posting messages decreased response times. Table 1 shows that this is a promising direction as median response times for different tags show a large variance. It also shows that use of more specific tags such as visual-studio-2010/2008, ruby-on rails-3, instead of visual-studio or ruby-on-rails can dramatically reduce response times.

V. Prediction of Actual Times

A. *Predicting Scale using Distributions*

We wanted to go a step beyond previous work and try and actually predict response times, despite the inherent challenges of doing so mentioned earlier. However, because of these challenges we were aiming for conservative approaches that were sufficient to give the user a good sense of the *scale* of the time of the expected response (minutes, days, weeks/months). As mentioned earlier, the study of response times in Stack Overflow by Mamykina et al. [7] suggests an apparently simple initial step to predicting scales of response times: Use the distributions of response times to predict the response time for a particular post

Table 2. Response times in the data set

	Accepted Answer Elapsed Time (Hours)	Earliest Answer Elapsed Time (Hours)
Mean	176.02	75.74
Median	8.02	1.41

B. *Distribution-based Answer-Time Prediction*

Two of our baseline approaches in this space were to predict that all questions would be answered by mean and median, respectively, of all answer times These values are substantially higher than those found by Mamykina et al. [7] and shown in Table 2.

C. *Baseline Approach*

The mean and median values show that the distribution of answer times for both earliest answers and accepted answers is skewed in one direction. As a result, the average elapsed time is not a good representative of the entire distribution of times. While the median is also not the best metric to use for such a distribution, it is less susceptible to influence from extreme answer times (some of which are above two years). In both cases, this baseline approaches are not very intelligent; we sought other approaches.

D. *Dynamic Timerange Partitioning*

As the response time is a continuous numerical value, it is natural to use regression, but when we used this approach our predictions were off by weeks and months in many cases. Our intuition was that more success may be afforded in predicting ranges of time rather than pinpointing time values. This is consistent with the fact if asked when users would respond to a message, they are likely to give a range rather than a precise time. More important, it is consistent with our goal of giving users an idea of not the exact response times but the scales of the times.

Therefore, we partitioned the entire time range into distinct time bins. There were a few different ways we could have partitioned the entire time range. A constant time range method

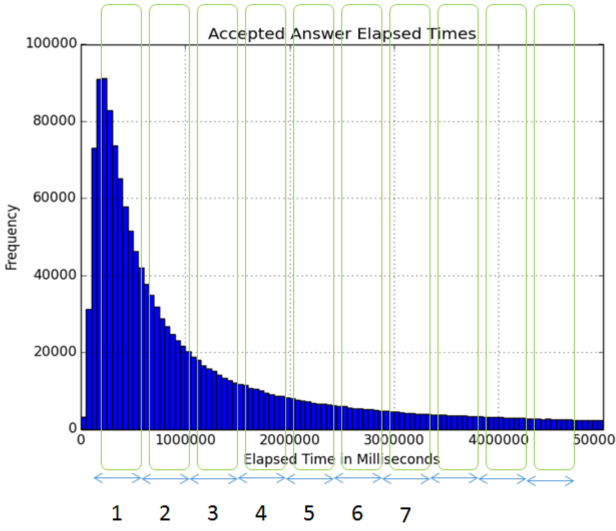


Figure 4. Equal sized bins

would partition the answer times into predetermined, equal-sized ranges, such as those shown Figure 4 for accepted-answer response times. We could then predict that any given question would be answered within the most probable time range.

There are various problems when using this approach. It is not possible to determine the distribution of response times beforehand, and thus it is difficult to know exactly what time bin size should be used. If the size of the bin is too small, prediction methods may fail to adequately predict the correct time range though the difference in the actual elapsed time between post and answer may not be very significant. On the other hand, too large a bin size would yield the opposite effect.

Table 3. Response time cluster ranges

(a) Time to earliest answer clusters

Range Number	Lower Time Limit	Upper Time Limit
1	0:00:00	0:02:51
2	0:02:51	0:04:35
3	0:04:35	0:06:46
4	0:06:46	0:09:44
5	0:09:44	0:13:59
6	0:13:59	0:20:11
7	0:20:11	0:29:26
8	0:29:26	0:43:30
9	0:43:30	1:04:54
10	1:04:54	1:37:20
11	1:37:20	2:26:10
12	2:26:10	3:39:20
13	3:39:20	5:28:19
14	5:28:19	8:09:13
15	8:09:13	12:00:33
16	12:00:33	17:10:11
17	17:10:11	23:44:38
18	23:44:38	1 day 9:48:38
19	1 day 9:48:38	1 day 22:57:00
20	1 day 22:57:00	2 days 15:58:06
21	2 days 15:58:06	3 days 17:17:02
22	3 days 17:17:02	5 days 2:22:07
23	5 days 2:22:07	6 days 17:14:26
24	6 days 17:14:26	8 days 17:49:45
25	8 days 17:49:45	1081 days 20:24:33

(b) Time to accepted answer clusters

Range Number	Lower Time Limit	Upper Time Limit
1	0:00:00	0:19:04
2	0:19:04	0:49:24
3	0:49:24	1:35:42
4	1:35:42	2:41:49
5	2:41:49	4:13:29
6	4:13:29	6:19:19
7	6:19:19	9:08:10
8	9:08:10	12:48:54
9	12:48:54	17:18:25
10	17:18:25	22:37:58
11	22:37:58	1 day 6:16:47
12	1 day 6:16:47	1 day 16:51:30
13	1 day 16:51:30	2 day 5:37:11
14	2 days 5:37:11	2 days 19:47:10
15	2 days 19:47:10	3 days 15:12:05
16	3 days 15:12:05	4 days 19:53:23
17	4 days 19:53:23	6 days 7:22:52
18	6 days 7:22:52	7 days 23:08:02
19	7 days 23:08:02	9 days 23:41:48
20	9 days 23:41:48	14 days 11:02:15
21	14 days 11:02:15	28 days 15:40:43
22	28 days 15:40:43	34 days 17:54:27
23	34 days 17:54:27	43 days 16:19:38
24	43 days 16:19:38	414 days 7:56:00
25	414 days 7:56:00	1450 days 16:44:13

A prediction algorithm could manage to predict the correct time range, but that specific time range could be meaningless. This relates also to the issue of how large the full time range itself should be. A longer time range may require that more time ranges are necessary to partition it. Arguably, constant time ranges also do not allow us to take into account the fact that the size of a time range should be proportional to its limits - the difference between 10 minutes and 20 minutes is more significant than the difference between 1 month and 10 minutes and a month and 20 minutes!

Partitioning the time range into unequal time ranges allows one to use the entire time range and account for relative differences in time. However, we have to decide how these time ranges are defined. Statically defining time ranges allows one to use familiar time measurements (minutes, hours, days, etc.). However, these divisions are arbitrary, and a user study would have to be done in order to discern the appropriate equivalence classes. Moreover, there may be an alternate structure in the dataset that statically-defined time ranges may not capture

Instead, we used a dynamic partitioning approach to determine the time ranges. In order to automatically divide up the full time range, we used simple K-means clustering of the two kinds of times with k being given a somewhat arbitrary value of 25. Table 3 shows the resulting partitions for accepted answer elapsed times and earliest answer elapsed times. As we see here, the time-range sizes increase with time. As mentioned earlier, this is a property we want in prediction – if we are optimizing relative error, then the size of the range should be proportional to the time. Interestingly, all earliest answer responses more than a day are put in the last earliest-response range, and all acceptable answer responses more than 414 days are put in the last accepted-response range. The upper limit of the first time cluster in Table 3(b), interestingly, is the same as the median time for accepted answer reported previously [7].

Each question in the training data set and the test data set was placed into one of 25 time ranges according to the lower and upper limits of each time range. From here, we used several different approaches to predict answer times within one of these ranges.

E. Average and median time range

Our baseline approaches used the average/median elapsed times and predicted them constantly for all the test questions. Therefore we developed two corresponding additional baseline approaches that predicted the average/median time range of all the questions.

F. Most frequently occurring time range

This approach took the most frequently occurring time range for both accepted answers and earliest answers and predicted that every question would have its accepted and earliest answer within these two ranges. This approach is particularly promising given the skewed nature of the distribution of elapsed times.

G. Weighted random choice

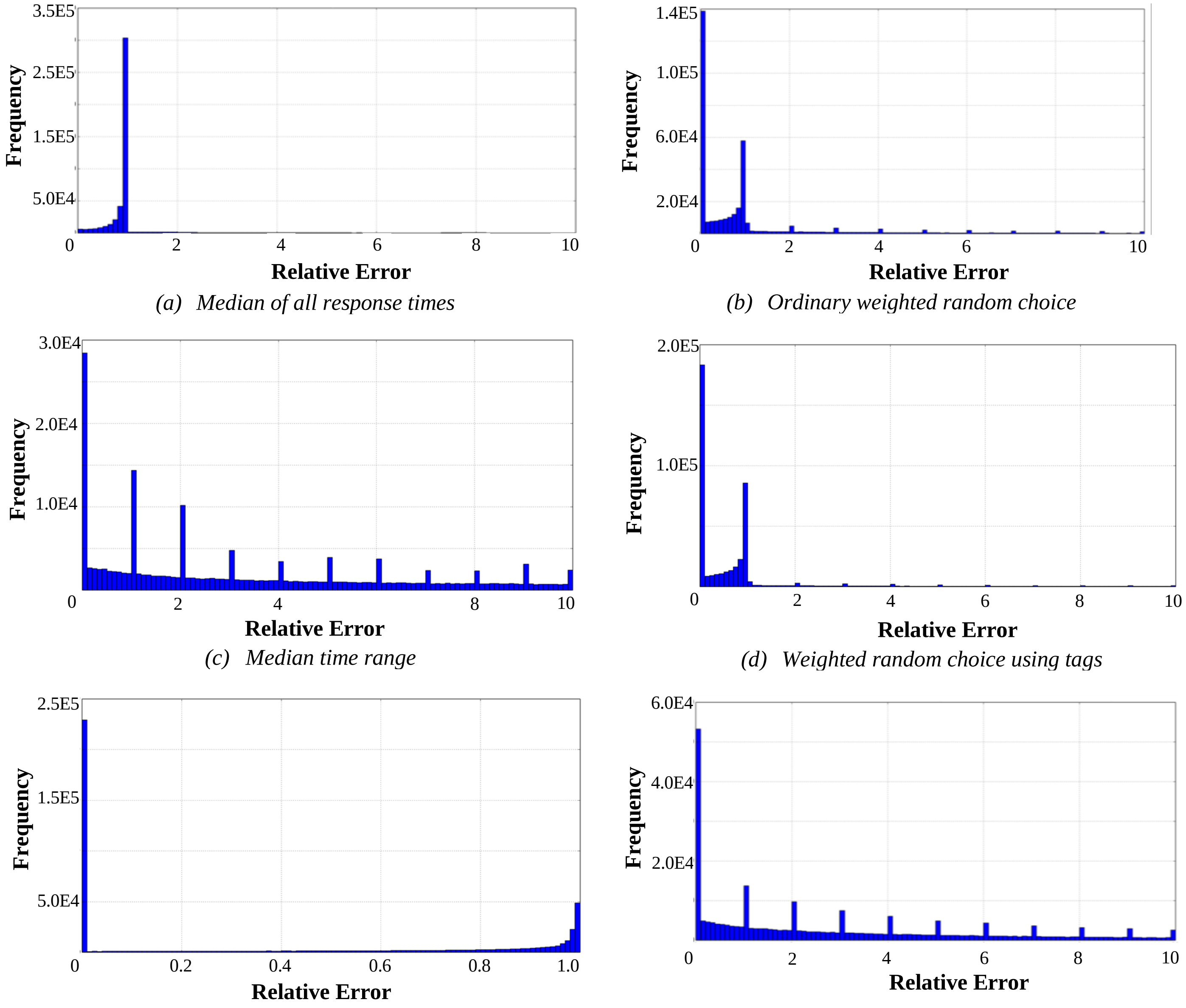
The issue with the previous methods is that they allow for no variability in prediction. Questions that are not answered within the time range used for constant prediction will always be attributed an incorrect time range. In order to increase the probability that other time ranges will be predicted, we used a weighted random choice algorithm. The algorithm works similarly to a roulette wheel where there are different probabilities for the wheel to stop in a certain section. For each question in the test set, a time range was drawn probabilistically from the existing distributions of time ranges for both earliest answers and accepted answers. If, for example, there are 3 choices of time ranges with probabilities 0.1, 0.3, and 0.6 respectively of being chosen, a value between 0 and 1 would be randomly selected. The list of possible outcomes would be as follows: If the random value is between 0 and 0.1, choose the first time range. If the random value is between 0.1 and 0.4, choose the second time range. For all other generated random values, choose the third time range.

This algorithm does not take into account features. We describe below variants that uses both tags and title lengths – the two promising features we found.

H. Feature-based Prediction

The basic idea behind using features in a distribution-based approach is to compute not an overall distribution but multiple distributions for different discrete values taken by the features. Given our two features, this means computing different distributions for different tags and title lengths. In our evaluation, we used 25 different title lengths and the top 100 tags (though in reality, thousands exist on Stack Overflow).

Creating a feature-based weighted random choice or most frequent time-range approach for title lengths is simple, as there is no possibility of a post containing more than one title length. Thus, we use the time range distribution corresponding to the title length of a post when applying the weighted random choice algorithm or choosing the most frequent time range. If a



(a) Median of all response times

(b) Ordinary weighted random choice

(c) Median time range

(d) Weighted random choice using tags

(e) Most frequent time range

(f) Weighted random choice using title length

Figure 6. Relative error plots for some of the accepted answer time predictions

post contains a single tag, the tag-based approach works similarly, using the distribution for the post tag rather than the post title length.

In many cases, a post contained multiple tags, which means we have to somehow combine the results from multiple distributions. Our basic idea was to apply the notion of weights, to not only the time ranges of each distribution as in the weighted random choice algorithm but also the time ranges returned using different distributions. Each distribution returns a time range with a certain probability of occurrence. We use these probabilities as weights in our choice.

This approach raises two issues based on the fact that each predicted time range has a width (upper limit – lower limit). Which properties of a time range should be used in the weighted average – lower limit, upper limit, average of the two limits, or some other property? And what should be the width of the predicted range – the maximum of the widths of the combined ranges, the minimum, or some other value?

We developed an elegant solution to this problem that has the characteristic that it does not predict an "artificial time range" – a time range not found by our clustering algorithm. As weights, it uses, not the absolute values of the limits of the combined time ranges, but the relative indices in Table 3, which have the property that increased indices are associated with higher limits. It then chooses a time range whose index is closest to the weighted index average. Suppose two tags produced time ranges 2 and 4 (using either the random weighted choice algorithm or the most frequent time range selection) with probabilities (frequencies) 30% and 60% respectively. The weights here are 30/(30+60) = 1/3 and 60/(30+60) = 2/3. The weighted average of the time ranges is 2(1/3) + 4(2/3) = 3.333. This value rounds to 3, so that is the time range used.

VI. Evaluation

A. Evaluation Metric

One approach to measuring the goodness of response-time predictions is to measure the absolute difference between predicted and actual times. However, as we were concerned with predicting the scale of response times, we wanted a metric that captured, for instance, that if the actual response time was 2 minutes/hours/days, a predicted response time of 6 minutes/hours/days is acceptable. This meant we had to measure relative errors.

The following metric is one possible way to measure relative error:

$$\frac{|actual\ time - predicted\ time|}{actual\ time}$$

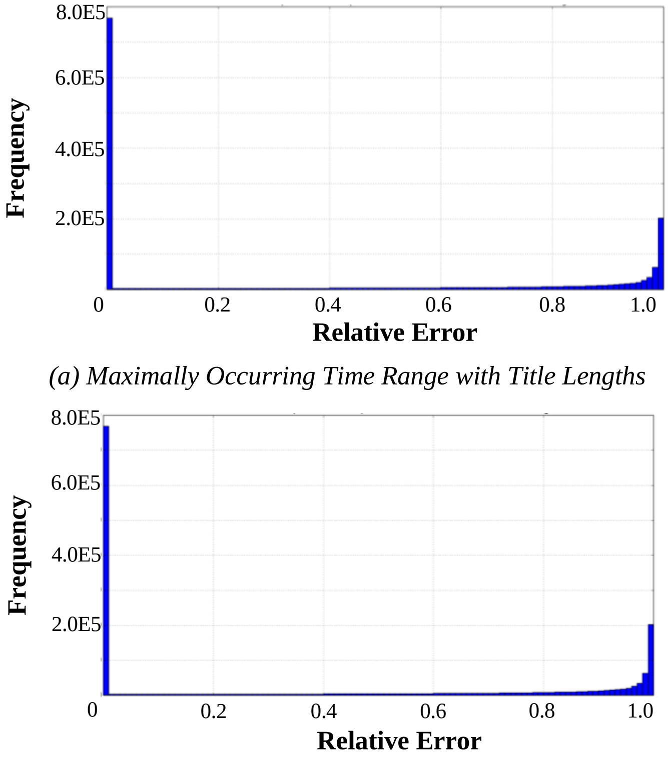
To illustrate the nature of this metric, let us assume that we decided that a relative error of 600% is considered acceptable. Under this metric and threshold, if the actual response time is 10 minutes, then a prediction of 1 hour would be considered acceptable, and it does gives the user the sense that a response will occur in the next hour rather than the next minute. However, this example also illustrates a problem with this metric - a prediction of near zero time would also be considered acceptable. In fact, a prediction of near zero time would always result in a relative error of less than 100%. Thus, this is a good metric when the response time is smaller than the predicted time but not when it is much larger. The dual of this metric is to use the predicted time in the denominator. However, in that case a predicted time of infinity would always be considered acceptable – this would be a good metric when the response time is larger than the predicted time.

To consider situations in which the predicted times can be both smaller and larger than the actual times, we used the following metric:

$$\frac{|actual\ time - predicted\ time|}{\min(actual\ time, predicted\ time)}$$

By dividing by the minimum of the actual and predicted elapsed times, we ensured that we are using the maximum possible relative error for any response time prediction. To illustrate, when the predicted response time is 8 hours, if the next response actually occurs in (a) 1 minute, the relative error is 480, (b) 1 day, the relative error is 3.

This metric, as defined above, does not address what value to use for predicted time when a range is predicted. Relative error becomes most important, naturally, when the predicted time range is incorrect. If so, the predicted time is the time range limit that is closest to the actual elapsed time until an answer. For example, if the actual time until an answer is 230 days and the time limits of the predicted time range are 1 year and 2 years, the predicted time in the relative error formula would be 1 year. Taking the midpoint of the time ranges as the predicted value would have allowed us to compare our results



(a) Maximally Occurring Time Range with Title Lengths

(b) Maximally Occurring Time Range with Tags

Figure 7. Relative error plots for accepted answer time for the two best performing approaches

Table 4. Relative error of baselines and prediction approaches

Average Relative Errors	Accepted Answers	Earliest Answers
Mean Time Range Baseline	3.3897	2.6052
Median Time Range Baseline	3.2801	2.6052
Maximally Occurring Time Range	0.3996	1.8306
Ordinary Weighted Random Choice	1.2246	1.9701
Maximally Occurring Time Range with Tags	0.4044	1.3613
Maximally Occurring Time Range with Title Lengths	0.4044	1.9398
Weighted Random Choice with Title Lengths	0.8393	1.5170
Weighted Random Choice with Tags	2.8551	1.9549

to other algorithms that predict specific times or different ranges. However, the actual difference in observed relative errors between our approach and a midpoint one would likely be small as the range sizes are the same magnitude as their lower limits, and our goal was to give an indication of the scale of the response times.

The results of the average relative errors for first and accepted answers are shown in Table 4.

B. Accepted Answers

Figure 6 and Figure 7 show the relative error plots for the approaches used in prediction of accepted answer elapsed times. In this figures, Figure 6 (a)-(c) show baseline results, and Figure 6 (d)-(f) and Figure 7(a-b) and show results from our predictions. In terms of baselines, the mean and median of all response times showed little to no variation in relative error from each other. Therefore, we only showed the result of median response time. The median range, shown in Figure 6(b), was the worst performing of the baselines, since it had a larger distribution of results with higher relative error than any of the other baselines. For the same reason, we judge median and mean response times to be better than the median time range, because the median time range has a larger number of results with relative errors at the larger end of the spectrum.

When comparing predictions to baselines, all the approaches outperformed the baselines for predicting the elapsed time till the accepted answer. Taking the maximally occurring time range worked well in general, regardless of whether or not tags or title lengths were used. Of the three such methods, performing the tag variant yielded the highest percentage of predictions with 0 relative error (Figure 7). As weighted random choice and its variants were used, the relative error increased, indicating poorer predictions.

Why did using the most frequently occurring time range work so well? One possible explanation is the concentration of elapsed times in this time range. About 45 percent of all questions were answered within the most frequently occurring time range. This did not hold with earliest answers, where the distribution of time ranges was far less concentrated. It is also partially the reason why predicting the elapsed time till the earliest answer was more difficult. The fact that it did not work as well for earliest answers can be explained by the conjecture that correct answers are provided by fewer people, and the time to reply is more uniform for them.

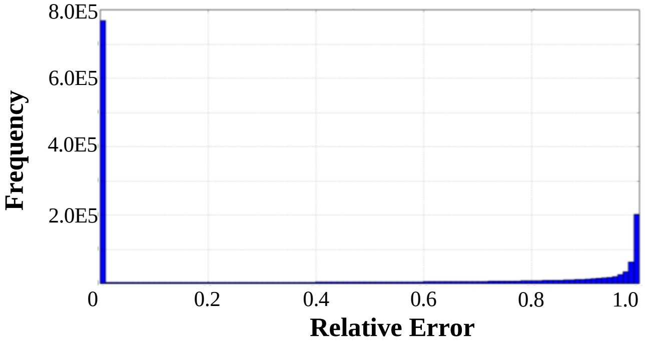


Figure 8. Relative error plots for earliest answer time for Maximally Occurring Time Range with Tags

C. Earliest Answer

Because of lack of space, we do not give the plots for earliest answers of all of the approaches; we give only the error plot for our best performing approach (Figure 8). All the proposed predictions approaches outperformed the baselines. Again, the tag variant of the maximally occurring time range prediction method yielded both the lowest average relative error and the highest number of correct predictions. Unlike in the case for accepted answers, using the title length variant of the same method did not lead to as good predictions. This is an interesting result considering that, as mentioned earlier, the correlation between median elapsed times and title lengths is stronger when considering accepted answers. It would appear that the variation of time ranges for questions of various title lengths is much larger when considering the elapsed time till the earliest answer. This can be explained by the fact that in the case of accepted answers, there was a much better approach, which did not work for earliest answers. Nonetheless, these results provide more evidence that mean times are correlated with title lengths and tags.

VII. Conclusions and Future Work

The main contribution of this paper is to present issues, techniques, and results related to making predictions about response times in messaging systems in general and Stack Overflow in particular. The specific contributions include:

- *Survey of related work:* We surveyed relevant research in the related domains of email, Usenet forums, and social networks, identifying some of the features in these domains that correlate with response times and reporting on some of the data gathering results.
- *Distribution of response times:* We have provided a finer-grained report of mean response times than previous work that identify mean response times for different tags.
- *Previous features investigation*: We showed that we could not apply several of the previous features to Stack Overflow. Some of these features such as the number of sentences in a message did not make sense as Stack Overflow message bodies contain code. Other features such as time of day and use of pronouns and punctuations did not correlate based on PCA analysis and graphs

plotting response times as functions of different values of these features.

- *Title length correlation*: One new feature we investigated was the length of the title of the post, which is a variation of the previously identified feature of number of sentences in a post body. For each title length we plotted the distribution of response times. We found a quadratic relationship between the median of these distributions and the title lengths. The relationship was stronger with accepted answers than earliest answers.

- *Design space of prediction granularity:* As a step towards prediction of actual times, we defined a large design space based on the granularity of prediction. Specifically, we separate the approaches based on whether (a) specific times or time-ranges were predicted, (b) time ranges were of constant or variable size, and (c) variable-sized time ranges were identified through user studies or clustering the time ranges.

- *Identification of 25 clusters:* Using the K-Means clustering algorithm, we identified 25 clusters of times for both earliest and accepted answers. These clusters have the intuitively desirable property that the span of a time range goes up with its limits.

- *Design space of distribution-based time-range prediction:* We developed several time-range prediction algorithms based on response-time distributions. Two baseline approaches simply predicted the median/average time range. One variation predicted the most frequently occurring time-range. A more complicated algorithm, weighted random choice, predicted multiple values based on the frequencies of time ranges. An even more complicated algorithm created different distributions based on features of messages, applied weighted random choice to each distribution, and probabilistically combined the results from each application.

- *Evaluation of prediction design space:* Using dynamic time ranges in response time prediction outperformed the baseline approaches of taking the mean and median of all the response times. Predicting the most frequent time range, regardless of whether or not title lengths or tags were used, gave the best result for response times of accepted answers, because 45% of the answers fell in that range. The earliest answers were most spread out, and using tags led to the best relative error.

This is very preliminary work in response time prediction for Stack Overflow in particular and messaging systems in general; and there are several further avenues possible for future research.

We did not thoroughly investigate the effect of tags on response times. This is partially due to the fact that, though we only considered the top 100 tags used on Stack Overflow, there are many more that we did not look into. Accounting for all these tags would create many more variables than may be desired in a prediction algorithm. Future work may involve clustering tags to reduce the total number of variables used in response time prediction and to create larger communities of tags with similar response rates.

Of the features that we investigated, only title length and the subject tags were promising in predicting response time in Stack Overflow. It would be useful to apply the set of correlating features in other domains to Stack Overflow more thoroughly using perhaps additional data mining techniques. If further validation our results is provided by follow-up research, it would indicate that the predictability of response time is not constant across multiple systems and requires different features, if not entirely different prediction models

It would be useful to investigate the application of title lengths and tags to other domains. While it makes intuitive sense to make the widths of time ranges increase with their limits, it would be useful to carry out user studies to (a) validate this intuition, and (b) identify specific widths and limits for various domains. It is unfair to use our metric to compare approaches pinpointing times with those giving time ranges, and a more sophisticated metric is needed. Using distributions is a very simplistic approach, and work is needed to explore more sophisticated techniques that perhaps use sender and recipient information to provide smaller relative errors. This paper provides a basis for pursuing these future directions.

Acknowledgment

This research was supported in part by the NSF award IIS 0810861

References

[1] Arguello, J., Butler, B.S., Joyce, E., Kraut, R., Ling, K.S., Rosé, C. and Wang, X. 2006. Talk to Me: Foundations for Successful Individual-group Interactions in Online Communities. *ACM CHI* 2006.

[2] Avrahami, D., Fussell, S.R. and Hudson, S.E. 2008. IM Waiting: Timing and Responsiveness in Semi-synchronous Communication. *ACM CSCW 2008*.

[3] Dabbish, L.A., Kraut, R.E., Faussell, S. and Kiesler, S. 2005. Understanding Email Use: Predicting Action on a Message.. *ACM CHI* 2005.

[4] Kalman, Y.M. and Rafaeli, S. 2005. Email Chronemics: Unobtrusive Profiling of Response Times. *HICSS* 2005.

[5] Koren, Y., Bell, R. and Volinsky, C. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer*. 42, 8 (2009), 30–37.

[6] Linden, G., Smith, B. and York, J. 2003. Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing*. 7, 1 (Jan. 2003), 76–80.

[7] Mamykina, L., Manoim, B., Mittal, M., Hripcsak, G. and Hartmann, B. 2011. Design lessons from the fastest q&a site in the west.. *ACM CHI* 2011.

[8] Miller, B.N., Albert, I., Lam, S.K., Konstan, J.A. and Riedl, J. 2003. MovieLens Unplugged: Experiences with an Occasionally Connected Recommender System. *ACM IUI* 2003.

[9] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J. 1994. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. *ACM CSCW* 1994.

[10] Sarwar, B., Karypis, G., Konstan, J. and Riedl, J. 2001. Item-based Collaborative Filtering Recommendation Algorithms. *Proceedings of WWW* 2001.

[11] Teevan, J., Morris, M.R. and Panovich, K. 2011. Factors Affecting Response Quantity, Quality, and Speed for Questions Asked Via Social Network Status Messages. *AAAI ICWSM 2011*.