Predicting the Timing and Quality of Responses in Online Discussion Forums

Patrick Hansen*, Richard Junior Bustamante*, Tsung-Yen Yang†, Elizabeth Tenorio‡,
Christopher G. Brinton§, Mung Chiang§, and Andrew S. Lan¶
*The College of New Jersey, †Princeton University, ‡Zoomi Inc.,
§Purdue University, ¶University of Massachusetts Amherst
*{hansenp2,sancher6}@tcnj.edu, †ty3@princeton.edu, ‡elizabeth.tenorio@zoomi.ai,
§{cgb,chiang}@purdue.edu, ¶andrew.lan@cs.umass.edu

***Abstract*—We consider the problem of jointly predicting the quality and timing of responses to questions asked in online discussion forums. While prior work has focused on identifying users most likely to answer and/or to provide the highest quality answers to a question, the promptness of the response is also a key factor of user satisfaction. To address this, we propose point process and neural network-based algorithms for three prediction tasks regarding a user's response to a question: whether the user will answer, the net votes that will be received on the answer, and the time that will elapse before the answer. These algorithms learn over a set of 20 features we define for each pair of user and question that quantify both topical and structural aspects of the forums, including discussion post similarities and social centrality measures. Through evaluation on a Stack Overflow dataset consisting of 20,000 question threads, we find that our method outperforms baselines on each prediction task by more than 20%. We also find that the importance of the features varies depending on the task and the amount of historical data available for inference. At the end, we design a question recommendation system that incorporates these predictions to jointly optimize response quality and timing in forums subject to user constraints.**

I. Introduction

Community Question Answering (CQA) services for knowledge dissemination and information seeking have exploded in popularity over the past decade. Platforms like Quora, Stack Overflow, and Yahoo! Answers have provided venues for Internet users to crowdsource answers to questions that they may not have otherwise found through general purpose web search. The rise of CQA has come with its share of challenges too, particularly around the timing and quality of user-generated answers; askers may have to wait up to several days until the "best" answer is determined [1], [2].

To address this issue, researchers have proposed algorithms for question routing, i.e., recommending questions newly posted on discussion forum sites to eligible answerers [2]–[4]. A major focus of such work has been identifying users most likely to answer a question [3] and/or to provide the highest quality responses [2], which in turn enables platforms to make answerer recommendations by e.g., personalizing user news feeds based on those predicted to produce desirable answers [4]. These prediction algorithms learn their parameters over data collected and stored on CQA sites, such as net votes received on posts, topics tagged in questions, and user expertise ratings [5]. A common metric these algorithms seek to optimize is response quality, typically quantified by either the net votes an answer receives or as a binary measure of whether it will be marked by the asker as the best answer [3].

In addition to response quality, there is another important dimension of the question recommendation problem that impacts user satisfaction: the time delay of answers provided [1], [6]. Ideally, these two (possibly competing) objectives would be optimized in a recommendation system concurrently, so that a user can receive an acceptable answer to their question without having to wait significantly longer for a marginally better response, e.g., several more hours for an answerer expected to accrue just one vote higher [4]. Motivated by this, we ask: *How can both the timing and quality of a user's answer to a question be predicted simultaneously in advance?* The design of accurate predictors for these attributes in turn would enable the development of more effective question-answerer allocation systems [2], potentially taking into account several factors such as each asker's objective, the urgency of the question, and the load imposed on answerers [4].

To address this research question, in this paper, we develop novel point process and neural network-based algorithms that predict response quality and timing by learning over a set of 20 features describing user-question pairs in CQA discussion forums. These features include both topical and structural aspects of user discussions, and give insight into the Social Learning Networks (SLNs) [7] that emerge on CQA sites. In evaluating our method on a real-world dataset, we also analyze the importance of each feature, and investigate whether tradeoffs exist between answerer response time and quality.

A. Related Work

Online discussion forums have received a plethora of research interest in the past several years. Many such works have focused on information retrieval tasks, including textual and semantic analysis of discussion threads [8], [9], identification of authoritative users by activity levels [10] and trends in link formation [5], inference of the social graphs connecting users based on thread discussion co-occurrences [1], [4], and analysis of the efficiency of communication among users [4], [11]. Our work is instead focused on prediction tasks for forums; in particular, predicting user response time and quality.

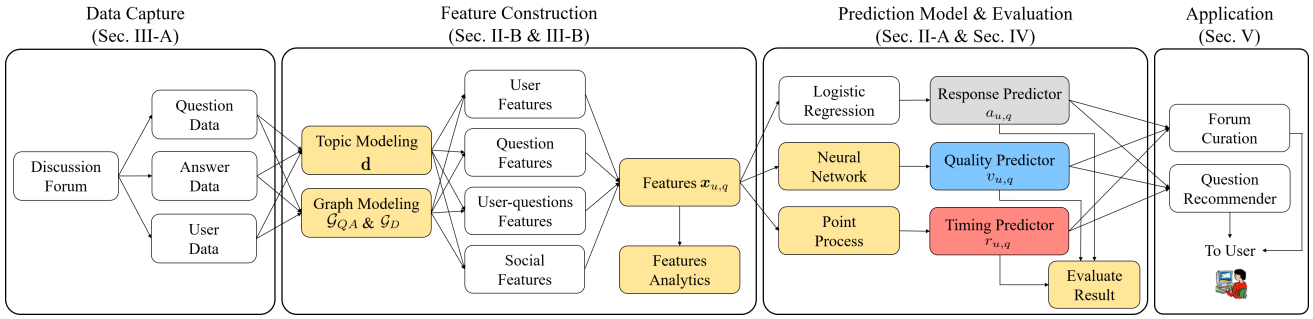


Fig. 1: Block diagram summary of the discussion forum question recommendation methodology developed in this paper. Highlighted components are those given particular emphasis. The shading pattern on the predictors is reused throughout the evaluation in Sec. IV.

In this regard, some recent works have studied prediction tasks for discussion forums. A few algorithms have been developed to predict user interactions, including whether a user will upvote/downvote an answer [12] and the formation/strength of links between between users [6], [13]. Our methodology defines some similar topic-based and structural features to those proposed in [6], [13], including user-to-user discussion similarities and resource allocation indexes, but we instead consider predictions for the purpose of question recommendation. Regarding this specific objective, recent works have built predictors focusing on two main tasks for question recommendation: determining which users will answer newly posted questions [3], [14], and estimating the quality of response that a user will provide to a question [2], [15], [16].

Similar to [2], [16], our work considers both of these tasks together, i.e., whether a user will answer and the quality of the answer. In particular, [16] proposed a set of algorithms that account for coupling between questions/answers, temporal dynamics of features, and non-linearities in predicting votes, while [2] proposed a generative tagword topic model to infer user interest and expertise on questions. While our methodology accounts for topical features and non-linear relationships with target variables (through neural networks), it additionally considers the structural aspects of the inferred social network, which we find are important features for question routing prediction tasks. Further different from these works, our method simultaneously predicts the timing of responses, which is acknowledged as an important objective in [6], [15]. For this prediction, we propose a point process model that learns from the same set of features as the net vote predictor.

B. Summary of Methodology and Contributions

Figure 1 summarizes the key components of the methodology developed in this paper. From the data collected on users, questions, and answers through the posts made in an online discussion forum (Sec. III-A), a set of prediction features is constructed (Sec. II-B). In particular, we define four groups of features for each pair of user and question: (i) *user* and (ii) *question* features, which describe the answering tendencies of the user and attributes of the question, respectively, (iii) *user-question* features, which quantify the topical match between the user and the question, and (iv) *social* features, which measure centralities and similarities between users. In doing so, we develop graph models to quantify the structure of interactions between forum users, and topic models to describe the discussions across forum posts, which are both key components of the Social Learning Network (SLN) [4].

The next component of our methodology is prediction algorithms that learn over sets of these features for user-question pairs (Sec. II-A). We consider three prediction tasks for the question recommendation problem: (i) *who* will answer a question, (ii) the *timing* of a user's response, and (iii) the *quality* of response a user will provide. A major challenge that our algorithms must overcome is modeling under sparsity, since the vast majority of users do not answer a given question [16]. For (ii) and (iii), we develop novel point process and neural network algorithms that quantify the time-varying probability of a user posting in a thread through generalized, non-linear rate functions. For (i), we resort to a logistic regression classifier to prevent overfitting on the user-question matrix.

To evaluate the performance of our predictors and assess the impact of our feature set, we perform several experiments on a real world-dataset of 20,000 question threads from Stack Overflow (Sec. IV). Our key findings are as follows:

- We show that our predictors obtain substantial improvements of 22-23% over baselines on each prediction task.
- We observe that user and question features vary in importance significantly between prediction tasks, while user-question and social features are more consistent.
- We find that the user, question, and user-question feature groups can each be the most important depending on the prediction task and amount of historical data available.
- We observe, rather surprisingly, that the timing and quality of user responses are uncorrelated quantities.

The final component of our methodology is the question routing algorithm that recommends newly posted questions to eligible answerers (Sec. V). To do this, we formulate a joint optimization of predicted response quality and timing subject to constraints on user load over a recent time window. We also discuss considerations for future work regarding the integration of this methodology into online forum platforms.

II. Forum Prediction Methodology

In this section, we formalize our prediction models. We first present our point process and neural network learning algo-

rithms for response timing and quality (Sec. II-A), followed by the learning features (Sec. II-B) used in the predictors.

A. Response Prediction Algorithms

An online (CQA) discussion forum is generally comprised of a series of threads, with each thread corresponding to one user-generated question as well as answers to that question [1]. In this paper, we let $u \in U$ denote user u in the set of users U, and $q \in Q$ denote question q in the set of questions Q comprising the dataset under consideration. p_{qn} will refer to the nth post made in the thread for question q, with p_{q0} corresponding to the question itself and $p_{q1}, \ldots$ being the answers, collectively forming thread q. We say that each post p contains text written by a creator $u(p)$ at timestamp $t(p)$, and received $v(p)$ net votes (up-votes minus down-votes).

As discussed in Sec. I, for each question q, we are interested in predicting three attributes of each user u: (i) whether u will answer q, (ii) the net votes that u's answer to q will receive, and (iii) the time that will elapse before u's answer of q. We denote these quantities as (i) $a_{u,q} \in \{0, 1\}$, a binary indicator with 1 corresponding to the user answering, (ii) $v_{u,q} \in \mathbb{Z}$, a positive or negative integer value, and (iii) $r_{u,q} \in \mathbb{R}^+$, a positive real number, for each user-question pair (u, q).[1] If $a_{u,q} = 0$, then $v_{u,q}$ and $r_{u,q}$ do not exist, though they may still be predicted for question recommendation. The predicted versions of these variables will be denoted $\hat{a}_{u,q}$, $\hat{v}_{u,q}$, and $\hat{r}_{u,q}$.

Our prediction algorithms are as follows:

1) Predicting $a_{u,q}$: We model the probability of a user u posting an answer to question q according to

$$P(a_{u,q} = 1 \mid \boldsymbol{x}_{u,q}) = \frac{1}{1 + e^{-\boldsymbol{x}_{u,q}^T \boldsymbol{\beta}}},$$

i.e., a logistic regression classifier. Here, $\boldsymbol{x}_{u,q} \in \mathbb{R}^d$ is our vector of engineered features for the user-question pair (u, q), which we will detail in Sec. II-B, and $\boldsymbol{\beta} \in \mathbb{R}^d$ is the vector of regression coefficients.

We choose a linear model on our features for $a_{u,q}$ for a few reasons. First, it will allow us to establish the general predictive capability of the features $\boldsymbol{x}_{u,q}$ themselves in Sec. IV, i.e., without more complex input-output mappings as is done for $v_{u,q}$ and $r_{u,q}$ below. Second, the sparsity of $a_{u,q}$ in discussion forums in general – with most users answering few questions [1], [4], [6] – renders nonlinear techniques prone to overfitting for this prediction task [16]. We will explore the sparsity of our own dataset in Sec. III.

2) Predicting $v_{u,q}$: We propose a fully-connected (and possibly deep) neural network for net vote prediction. Specifically, we model $v_{u,q}$ according to

$$\begin{aligned} \boldsymbol{h}_1 &= \sigma(\boldsymbol{W}_1^T \boldsymbol{x}_{u,q} + \boldsymbol{b}_1) \\ \boldsymbol{h}_2 &= \sigma(\boldsymbol{W}_2^T \boldsymbol{h}_1 + \boldsymbol{b}_2) \\ \vdots &= \vdots \\ v_{u,q} &= \sigma(\boldsymbol{w}_L^T \boldsymbol{h}_L + b_L), \end{aligned} \tag{1}$$

where $\boldsymbol{x}_{u,q}$ is the vector of input features, and the parameters are weight matrices $\boldsymbol{W}_1, ..., \boldsymbol{W}_{L-1}$, the weight vector $\boldsymbol{w}_L$, the bias vectors $\boldsymbol{b}_1, ..., \boldsymbol{b}_{L-1}$, and the bias scalar b_L. L controls the number of hidden layers $\boldsymbol{h}_1, ..., \boldsymbol{h}_L$ in the model, while we allow the number of hidden units (i.e., dimension of each $\boldsymbol{h}_1, ..., \boldsymbol{h}_L$) to vary across layers. σ denotes a nonlinearity function, e.g., $\tanh$ or rectified linear units (ReLU).

3) Predicting $r_{u,q}$: We develop a point process model [17] to model a user's response time in a question thread. The (latent) rate function of this process for each (u, q) dictates the time-varying probability that u will post an answer to q at a particular point in time. We model this rate, $\lambda_{u,q}(t)$, as an initial excitation that decays exponentially over time, i.e.,

$$\lambda_{u,q}(t) = \mu_{u,q} e^{-\omega_{u,q}(t - t(p_{q0}))},$$

where t denotes the current time and $t(p_{q0})$ is the timestamp when question q was posted, i.e., if u responds at t then the observed response time is $r_{u,q} = t - t(p_{q0})$. $\mu_{u,q}$ denotes the initial excitation of q on u, which characterizes the strength of influence the question has on the user, while $\omega_{u,q} > 0$ denotes the decay rate on the influence of the question post. We further model the initial excitation and decay rate as

$$\mu_{u,q} = f_{\boldsymbol{\Theta}}(\boldsymbol{x}_{u,q}), \qquad \omega_{u,q} = g_{\boldsymbol{\Theta}}(\boldsymbol{x}_{u,q}),$$

where $f_{\boldsymbol{\Theta}}(\cdot)$ denotes a function with parameter set $\boldsymbol{\Theta}$, and $\boldsymbol{x}_{u,q}$ is the vector of input features for this (u, q) pair.

As a generalization over prior methods that restrict $f_{\boldsymbol{\Theta}}(\cdot)$ to be a linear function [18], we use two separate (non-linear) fully connected neural networks for $f_{\boldsymbol{\Theta}}(\cdot)$ and $g_{\boldsymbol{\Theta}}(\cdot)$. $\boldsymbol{\Theta}$ contains all the weights and biases from the two neural networks f and g, as detailed in (1). Additionally, our choice of setting the decay rate ω to be a function of $\boldsymbol{x}_{u,q}$ – and thus varying across user-question pairs – is significantly different from the setting in [18] where ω is set to a constant value.

Now, for each question thread q comprised of posts $p_{q0}, p_{q1}, ...$, the log likelihood of q is given by

$$\begin{aligned} L_q &= \sum_{n:n>0} \log \lambda_{u(p_{qn}),q}(t(p_{qn})) - \sum_{u \in U} \int_{t(p_{q0})}^{T} \lambda_{u,q}(\tau) \mathrm{d}\tau \\ &= \sum_{n:n>0} \log f_{\boldsymbol{\Theta}}(\boldsymbol{x}_{u(p_{qn}),q}) - \sum_{n:n>0} g_{\boldsymbol{\Theta}}(\boldsymbol{x}_{u,q})(t(p_{qn}) - t(p_{q0})) \\ &\quad - \sum_{u \in U} f_{\boldsymbol{\Theta}}(\boldsymbol{x}_{u,q}) \frac{1 - e^{-g_{\boldsymbol{\Theta}}(\boldsymbol{x}_{u,q})(T - t(p_{q0}))}}{g_{\boldsymbol{\Theta}}(\boldsymbol{x}_{u,q})}, \end{aligned}$$

where $T = \max_{q,n} t(p_{qn})$ denotes the timestamp of the last answer in the dataset (assuming that the first question is posted at $t = 0$). Using this expression, the total log-likelihood of a particular set of questions $\Omega \subseteq Q$ can then be computed as $\sum_{q \in \Omega} L_q$. Since this total log-likelihood is a smooth function of the neural network parameters $\boldsymbol{\Theta}$, we can estimate the parameters using gradient descent algorithms.[2]

[1] It is possible (though rare) for a user to submit multiple answers to the same question. We will address this in our data processing in Sec. III.

[2] We use the standard Adam optimizer in TensorFlow https://www.tensorflow.org/.

Once we have obtained the estimates of the parameters $\boldsymbol{\Theta}$, denoted $\widehat{\boldsymbol{\Theta}}$, we can calculate the expected time at which user u will respond to question q as

$$\begin{aligned}\mathbb{E}[t_{u,q}] &= \int_{t(p_{q0})}^{T} \tau P(\text{response between } \tau \text{ and } \tau + \mathrm{d}\tau) \\ &= \int_{t(p_{q0})}^{T} \tau \lambda_{u,q}(\tau)\mathrm{d}\tau \\ &= \widehat{\mu}_{u,q} \int_{0}^{T-t(p_{q0})} \tau e^{-\widehat{\omega}_{u,q}\tau}\mathrm{d}\tau \\ &= \frac{\widehat{\mu}_{u,q}}{\widehat{\omega}_{u,q}^2}\left(1 - e^{-\widehat{\omega}_{u,q}(T-t(p_{q0}))}(1+\widehat{\omega}_{u,q}(T-t(p_{q0})))\right),\end{aligned}$$

where $\widehat{\mu}_{u,q} = f_{\widehat{\boldsymbol{\Theta}}}(\boldsymbol{x}_{u,q})$ and $\widehat{\omega}_{u,q} = g_{\widehat{\boldsymbol{\Theta}}}(\boldsymbol{x}_{u,q})$. This expectation constitutes our prediction of when the user will respond to the question, from which we can subtract the time $t(p_{q0})$ when the question was created to obtain our prediction $\hat{r}_{u,q}$ of the response time $r_{u,q}$:

$$\hat{r}_{u,q} = \mathbb{E}[t_{u,q}] - t(p_{q0})$$

B. Feature Engineering for $\boldsymbol{x}_{u,q}$

In this section, we will develop four groups of features that constitute $\boldsymbol{x}_{u,q}$. In order to do so, we first detail our methods for inferring post topics and constructing the Social Learning Network (SLN) graph structure of the forums.

Topic models. We divide the text comprising each post p into two groups: words $x(p)$ and code $c(p)$ (using the fact that code on forums is delimited by specific HTML tags). A topic distribution $\mathbf{d}(p) = (d_1(p), ..., d_K(p))$ is associated with each p based on analysis of $x(p)$, where K is the number of topics, $d_i(p) \in (0,1)$ is the proportion of p constituted by topic i, and $\sum_i d_i(p) = 1$. Similar to [4], [6], we infer $\mathbf{d}(p)$ through Latent Dirichlet Allocation (LDA), which extracts post-topic and topic-word distributions across a set of forum questions when each post p comprising the set of questions is treated as a separate document.[3] Moving forward, we will let $\Omega \subseteq Q$ denote a general partition of the questions in the dataset for feature computation and model training; the methods used for cross validation will be described in Sec. IV.

Graph models. We consider two graphs of users for the SLN. First is the question-answer graph $\mathcal{G}_{QA}$, where a link is created between users u and v if u creates a question and v posts an answer, or vice versa. Formally, let $w_{u,v} = \mathbb{1}\{\exists q \in \Omega, i > 0 : u(p_{q0}) = u, u(p_{q,i}) = v \;||\; u(p_{q0}) = v, u(p_{q,i}) = u\}$, where $\mathbb{1}$ is the indicator function; then $[w_{u,v}]$ is the binary adjacency matrix of $\mathcal{G}_{QA}$. Second is a denser graph $\mathcal{G}_D$ where answerers in the same thread are also connected to each other, with "density" reflecting the proportion of node pairs that are connected [6]. In this case, $w_{u,v} = \mathbb{1}\{\exists q \in \Omega, i \geq 0, j \geq 0 : u(p_{q,i}) = u, u(p_{q,j}) = v\}$ is the adjacency matrix. Note that since links are bidirectional, both $\mathcal{G}_{QA}$ and $\mathcal{G}_D$ are symmetric.

[3]We use the Latent Dirichlet Allocation function in Python's Gensim package.

We now define 20 user, question, user-question, and social features for each (u, q) pair to form the feature vector $\boldsymbol{x}_{u,q}$:

User features. These aim to quantify user u's observed behavior in answering questions, including overall activity, quality and speed of responses, and topics of interest. In particular, based on u, the following features are computed:

(i) Answers provided a_u: The number of answers given by u, i.e., $a_u = \sum_{q\in\Omega, i>0} \mathbb{1}\{u(p_{q,i}) = u\}$.

(ii) Answer ratio o_u: The smoothed ratio of answers generated to questions asked by u, i.e., $o_u = (\sum_{q\in\Omega,i>0} \mathbb{1}\{u(p_{q,i}) = u\})/(1 + \sum_{q\in\Omega,i=0} \mathbb{1}\{u(p_{q,i}) = u\})$.

(iii) Net answer votes v_u: The net votes on answers given by u, i.e., $v_u = \sum_{q\in\Omega,i>0} (\mathbb{1}\{u(p_{q,i}) = u\} \cdot v(p_{q,i}))$.

(iv) Median response time r_u: A measure of the average time before u responds to a question, i.e., $r_u = \text{median}\{t(p_{q,i}) - t(p_{q0}) : q \in \Omega, i > 0, u(p_{q,i}) = u\}$.[4]

(v) Topics answered $\mathbf{d}_u$: The average distribution of topics answered by u, i.e., $\mathbf{d}_u = \text{mean}\{\mathbf{d}(p_{q,i}) : q \in \Omega, i > 0, u(p_{q,i}) = u\}$ where the average is taken element-wise.

Question features. This group of features aims to quantify attributes of the question q that may attract responses from particular users, including popularity, length, and constituent topics. In particular, the following are computed for q:

(vi) Net question votes v_q: The net votes on q, i.e., $v(p_{q0})$.

(vii) Question word length x_q: The length of words written in q in characters, i.e., $|x(p_{q0})|$.

(viii) Question code length c_q: The length of code written in q in characters, i.e., $|c(p_{q0})|$.

(ix) Topics asked $\mathbf{d}_q$: The topic distribution $\mathbf{d}(p_{q0})$ of q.

User-question features. These quantify potential relationships between user u and question q, such as similarities in topics discussed and the quality of answers u provided to related questions. In particular, the following are computed:

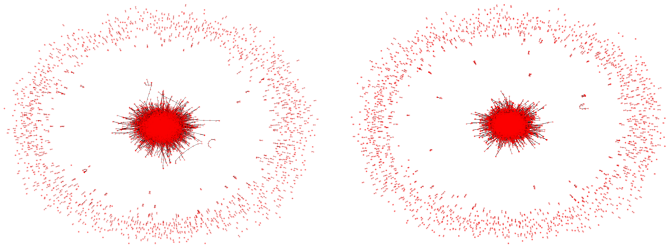
(x) User-question topic similarity $s_{u,q}$: The total variation distance between the user and question topic distributions expressed as a similarity, i.e., $s_{u,q} = 1 - \frac{1}{2}\|\mathbf{d}_u - \mathbf{d}_q\|_1$.

(xi) Topic-weighted questions answered $g_{u,q}$: Total topic similarity between the question and questions previously answered by u, i.e., $g_{u,q} = \sum_{r\in\Omega,r\neq q} (\mathbb{1}\{\exists i > 0 : u(p_{r,i}) = u\} \cdot s_{r,q})$ where $s_{r,q} = 1 - \frac{1}{2}\|\mathbf{d}_q - \mathbf{d}_r\|_1$.

(xii) Topic-weighted answer votes $e_{u,q}$: Net votes on answers given by u weighted by the question-question similarity, i.e., $v_{u,q} = \sum_{r\in\Omega,r\neq q,i>0} (\mathbb{1}\{u(p_{r,i}) = u\} \cdot v(p_{r,i}) \cdot s_{r,q})$.

Social features. We also consider features of the inferred SLN topologies that may give insight into user u's (a) overall question answering tendency, such as centrality measures, and (b) potential for answering the particular question q, such as thread co-participation and discussion topic similarity between u and the creator $v = u(p_{q0})$ of question q. In particular, we compute the following:

[4]The median is taken here to prevent the effect of outliers in timing data.



(a) $\mathcal{G}_{QA}$ (b) $\mathcal{G}_D$

Fig. 2: Visualization of the (a) question-answer and (b) denser graph models of the SLNs in our dataset. Each has roughly 14K user nodes. Higher degree users are plotted closer to the center.

(xiii) User-user topic similarity $s_{u,v}$*:* The similarity between topics discussed by the user and the user who asked the question, i.e., $s_{u,v} = 1 - \frac{1}{2}\|\mathbf{d}_u - \mathbf{d}_v\|_1$.

(xiv) Thread co-occurrence $h_{u,v}$*:* The number of threads that both u and v contribute to as either questions or answers, i.e., $h_{u,v} = \sum_{q\in\Omega} \mathbb{1}\{\exists m : u(p_{q,m}) = u, \exists n : u(p_{q,n}) = v\}$.

(xv) QA closeness centrality l_u^{QA}*:* The closeness of u measured over the social graph $\mathcal{G}_{QA}$, i.e., $l_u = (|U| - 1)/\sum_{v\neq u} z_{u,v}(\mathcal{G}_{QA})$ where $z_{u,v}$ is the shortest path distance between u and v.

(xvi) QA betweenness centrality b_u^{QA}*:* The betweenness of u measured on $\mathcal{G}_{QA}$, i.e., $b_u = \sum_{s\neq t\neq u} \sigma_{s,t,u}(\mathcal{G}_{QA})/\sigma_{s,t}(\mathcal{G}_{QA})$ where $\sigma_{s,t}$ is the number of shortest paths between s and t and $\sigma_{s,t,u}$ is the number of these paths that u lies upon.

(xvii) QA resource allocation index $Re_{u,v}^{QA}$*:* The resource allocation index of u and v in the social graph $\mathcal{G}_{QA}$, i.e., $Re_{u,v}^{QA} = \sum_{n\in\Gamma_u\cap\Gamma_v} 1/|\Gamma_n|$ where $\Gamma_u = \{t : w_{u,t} = 1\}$ is the set of u's neighbors. Among the topology features proposed for link prediction in [6], this was found most predictive.[5]

(xviii) Denser closeness centrality l_u^D*:* The closeness centrality of u measured over the social graph $\mathcal{G}_D$ instead.

(xix) Denser betweenness centrality b_u^{QA}*:* The betweenness centrality of u measured on $\mathcal{G}_D$ instead.

(xx) Denser resource allocation index $Re_{u,v}^D$*:* The resource allocation index of u and v measured on $\mathcal{G}_D$ instead.

These 20 features constitute vector $\boldsymbol{x}_{u,q}$. Since two of the 20 defined features are topic distributions of length K, the resulting dimension of $\boldsymbol{x}_{u,q}$ is $18 + 2K$. In Sec. IV, we will analyze the importance of each feature to each prediction task.

III. Dataset and Descriptive Analytics

To evaluate our methodology, we consider a dataset from Stack Overflow, a popular CQA site for software developers.[6] In this section, we detail our data collection (Sec. III-A) and analyze the dataset in terms of the model features (Sec. III-B).

[5]If u and v have no common neighbors, $Re_{u,v} = 0$. Similarly, when there are no paths between u and v, these terms are removed from l_u and b_u.

[6]www.stackoverflow.com

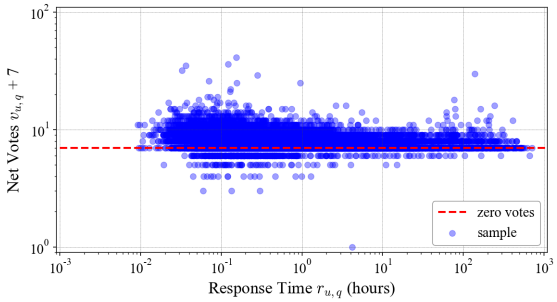


Fig. 3: Plot of net votes against response time for each user-question pair in the dataset. The smallest number of observed votes (-6) is calibrated to 1. Surprisingly, there is no apparent tradeoff relationship between response quality ($v_{u,q}$) and timing ($r_{u,q}$), indicating that these question routing objectives may not always be competing.

A. Data Collection and Processing

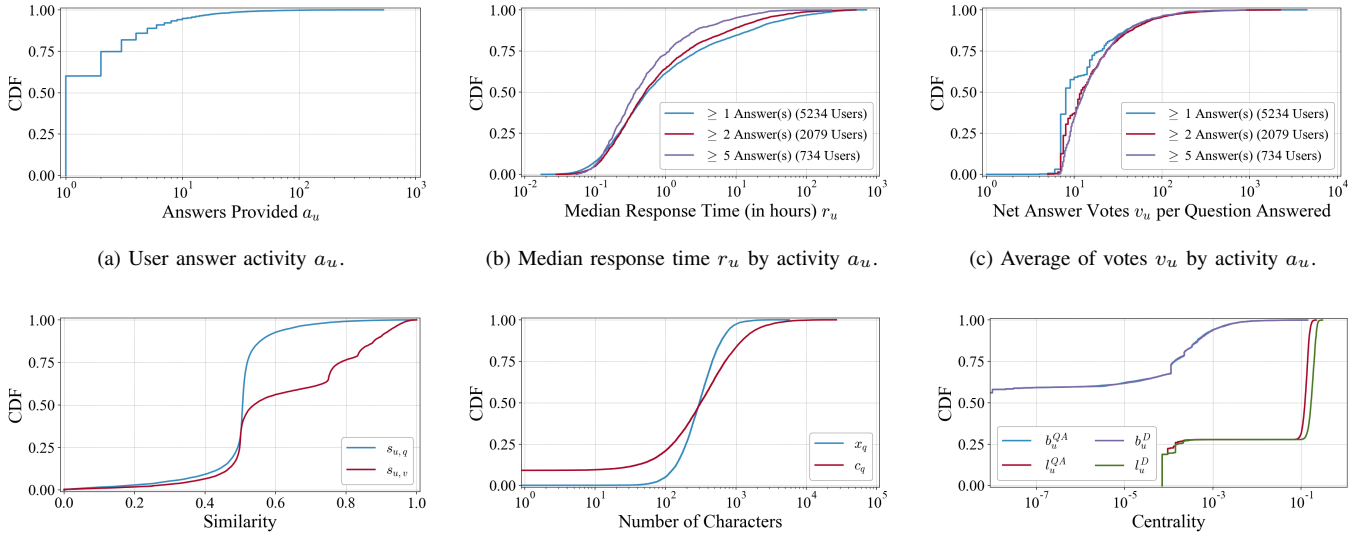
We queried the Stack Exchange API[7] for all questions created on Stack Overflow with the generic tag "Python" in the 30-day span from June 3 to July 3, 2018. This process yielded 20,923 questions and 19,934 total answers generated by 9,947 askers and 6,451 answerers, with 14,643 distinct users.

In processing the data to create user-question pairs, we filtered out any question that did not receive at least one answer. Then, where a user posted more than one answer to a question (only about 50 cases total), we took the one with the highest score. Additionally, some answers were found to be posted at the same time as the question was asked; we also removed these user-question pairs from the dataset. After these preprocessing steps, we were left with 12,488 questions asked by 9,318 users, and 18,414 answers posted by 5,234 users, for a total of 14,064 unique users. If we define $\mathbf{A} = [a_{u,q}]$ as the user-question answering matrix over all users u who answered at least one question, then, only 0.03% of the elements in $\mathbf{A}$ for our dataset are 1. This underscores an extreme sparsity of user-question pairs for response prediction, and further justifies our choice of classifier for $a_{u,q}$ in Sec. II to prevent overfitting.

Social graphs. Figure 2 visualizes the two social graphs $\mathcal{G}_{QA}$ and $\mathcal{G}_D$ defined in Sec. II-B across the entire dataset, i.e., taking $\Omega = Q$ over all 12K questions. The nodes here are the 14K users, with links between them according to the corresponding adjacency matrices. In these visualizations, users with higher degree are drawn closer to the center. With $\sum_v w_{u,v}$ as the degree of node u in an undirected graph, the average user degree is 2.6 in the question-answer graph, and rises to 3.7 in the denser graph that connects all users posting in the same thread. Despite this difference, we see from the outer rings in Figure 2 that both social graphs are disconnected, i.e., many user pairs do not have paths connecting them. This implies that there is high variance in the degree distributions, which further motivates the inclusion of structural features like centrality measures in Sec. II-B.

Net votes vs. response time. Recall from Sec. I our discussion on the possibility of response quality $v_{u,q}$ and timing $r_{u,q}$

[7]https://api.stackexchange.com



(a) User answer activity a_u.

(b) Median response time r_u by activity a_u.

(c) Average of votes v_u by activity a_u.

(d) User-question $s_{u,q}$ and user-user $s_{u,v}$ topic similarities.

(e) Word text x_q and code c_q lengths.

(f) Betweenness b_u^{QA}, b_u^D and closeness l_u^{QA}, l_u^D centralities, each normalized to a maximum of 1.

Fig. 4: Cumulative distribution functions (CDFs) of select quantities comprising the $\boldsymbol{x}_{u,q}$ feature vectors. They show that (a) users are relatively active, while those who are more active tend to (b) have shorter response times but (c) not necessarily more average votes. Users also (d) tend to be more similar to question askers than the questions themselves and (f) have substantial variation in centrality measures.

being two competing objectives for question routing. To investigate this, after computing the dependent prediction variables $v_{u,q}$, $r_{u,q}$, and $a_{u,q}$ for each user-question pair in the dataset, we plot $v_{u,q}$ against $r_{u,q}$ in Figure 3 for all user-question pairs with $a_{u,q} = 1$. Surprisingly, there is no correlation between these quantities, implying that *a shorter response time does not necessarily come at the expense of a lower quality answer or vice versa*. The objectives may not always be competing after all. This further underscores the importance of including both quality and timing as prediction tasks in our methodology, since one cannot be inferred from the other yet both are important components of user satisfaction.

B. Statistical Analysis of Features

After computing the feature vectors $\boldsymbol{x}_{u,q}$ over the full dataset, we plot the distributions of selected features in Figure 4. These are the subject of the following observations:

(i) Answers provided a_u (Fig. 4a): The number of answers posted by a user is an indication of their activity level. We see that roughly 40% of users posted two or more answers, indicating that many users were active on Stack Overflow during this period. This contrasts other types of discussion forums, e.g., those for Massive Open Online Courses (MOOCs) [4] where activity is centered around a small number of users.

(ii) Response time r_u (Fig. 4b): A user's median response time r_u over the answers they provided is lessened noticeably as their activity level a_u increases. For example, roughly 80% of users with $a_u \geq 5$ have $r_u \leq 1$ hr, while this percentage drops to 60% for $a_u \geq 1$. Users who spend more time on the forums may be more aware of newly posted questions, foreshadowing an observation we will make in Sec. IV that a_u is among the most predictive features for response timing $r_{u,q}$.

(iii) Average net votes v_u (Fig. 4c): The average votes received v_u across questions answered reflects the quality of u's responses. We see that while all users with at least one answer ($a_u \geq 1$) tended to have lower average votes than users who provided multiple responses, as long as users answer more than once ($a_u \geq 2$) there is no significant variation between the distributions of different total answers. Answering more than one question therefore may be a threshold beyond which a user tends to be perceived by others as authoritative.

(iv) Topic similarities $s_{u,q}, s_{u,v}$ (Fig. 4d): Recall from Sec. II-B that the user-user $s_{u,v}$ and user-question $s_{u,q}$ topic similarities are calculated as differences in inferred topic distributions $\mathbf{d}$. These features show an interesting trend: *answerers tend to have more similarity to the user who asked the question than to the question itself.* For example, 90% of the user-question pairs have a similarity of $s_{u,q} \leq 0.6$, compared to only 60% for the same threshold on $s_{u,v}$. This is consistent with an observation we will make in Sec. IV that social similarity between users is more predictive of posting activity than user-question topic similarity.

(v) Question lengths x_q, c_q (Fig. 4e): The median lengths of word text x_q and code c_q appearing in questions are both roughly 300 characters. The variation of c_q across questions is significantly higher than x_q, however, with an apparent limit on the words users will write; this is consistent with the length of code likely needing to vary by the type of question.

(vi) User centralities b_u^{QA}, b_u^D, l_u^{QA}, l_u^D (Fig. 4f): Four of the social features from Sec. II-B are betweenness b_u^{QA}, b_u^D and closeness l_u^{QA}, l_u^D centralities measured on the two graphs.

Task	Metric	Baseline	Our model	Improvement
$a_{u,q}$	AUC	0.699 ± 0.005	$\mathbf{0.860 \pm 0.004}$	23.0%
$v_{u,q}$	RMSE	1.554 ± 0.057	$\mathbf{1.213 \pm 0.118}$	21.9%
$r_{u,q}$	RMSE	34.247 ± 4.641	$\mathbf{26.353 \pm 3.566}$	22.8%

TABLE I: Performance on all three prediction tasks over the full dataset. Our models significantly outperform the baselines in each case.

We see that each measure exhibits significant variation across users, consistent with the observations from Figure 2. Closeness and betweenness are also markedly different from one another, with 60% of users having zero b_u while l_u has clusters around 10^{-4} and 10^{-1}. The fact that l_u changes between graphs while b_u does not implies that while the dense graph lowers path distances, it does not create many new paths between users unconnected in the question-answer graph.

IV. Prediction Evaluation

We now evaluate the methodology proposed in Sec. II. After describing the evaluation procedure and baseline algorithms (Sec. IV-A), we investigate overall performance (Sec. IV-B), the importance of specific features (Sec. IV-C), and the impact of historical data (Sec. IV-D) on the prediction tasks.

A. Evaluation Setup

For each experiment, each algorithm is evaluated over several iterations. In each iteration k, a training set $\mathcal{S}_k^T$ and a testing set $\mathcal{S}_k^e$ of user-question pairs (u, q) are sampled over the partition of questions $\Omega \subseteq Q$ under consideration such that $\mathcal{S}_k^T \cap \mathcal{S}_k^e = \emptyset$. The feature vector $\boldsymbol{x}_{u,q}$ for each sample (u, q) in $\mathcal{S}_k^T$ and $\mathcal{S}_k^e$ is computed over a set of questions $\mathcal{F}(q)$. The choice of Ω and $\mathcal{F}(q)$ will vary by experiment, particularly to analyze the effect of historical data.

Baselines. We establish one baseline for each prediction task.

(i) SPARFA for $a_{u,q}$: The sparse factor analysis (SPARFA) algorithm [19] was developed to predict the correctness of a user's response to a question. We use this as the baseline for the task of predicting whether a user will answer a question since it has consistently outperformed other binary matrix completion methods.

(ii) MF for $v_{u,q}$: Collaborative filtering techniques have had demonstrable success in recommender system prediction tasks involving user-item matrices [2], [7], [20]. As a result, we employ (non-binary) matrix factorization (MF) [21] as the baseline for net vote prediction. The fact that SPARFA and MF learn over user u and question q indices allows us to evaluate the quality of our features $\boldsymbol{x}_{u,q}$ by comparing their prediction performance against our models.

(iii) PR for $r_{u,q}$: Since response time prediction has not been a focus of prior research on question recommendation, we resort to Poisson regression (PR) as a baseline, which has been used to model e.g., web traffic inter-arrival times [22]. In our context, we use the features $\boldsymbol{x}_{u,q}$ as regressors, and the target $\bar{r}_{u,q} = \lceil r_{u,q} \rceil$ is a discretized (ceiling) version of $r_{u,q}$.

Metrics. We employ two metrics to evaluate the performance of the trained predictors on the test set $\mathcal{S}_k^e$ in each iteration k:

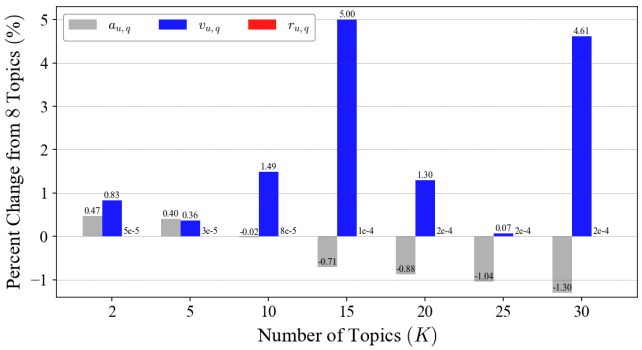


Fig. 5: Performance of models on prediction tasks from varying the number of topics K from the default of 8. There is virtually no effect on $r_{u,q}$, only small impact on $a_{u,q}$, and relatively larger change on $v_{u,q}$. Thus, while $K = 8$ obtains significant improvements over baselines in Table I, better results may be possible for $v_{u,q}$.

(i) AUC: The area under the ROC curve (AUC) assesses the tradeoff between the true and false positive rates of a classifier. We apply this metric to the binary prediction task for $a_{u,q}$, i.e., comparing $\hat{a}_{u,q}$ and $a_{u,q}$ over $\mathcal{S}_k^e$. We employ AUC rather than e.g., accuracy due to dataset imbalance [6].

(ii) RMSE: For the non-binary tasks $v_{u,q}$ and $r_{u,q}$, we calculate the root mean squared error (RMSE) between the predictions $\hat{y}_{u,q}$ and targets $y_{u,q}$ on $\mathcal{S}_k^e$ [7]. Formally, this is calculated as

$$\sqrt{\frac{1}{|\mathcal{S}_k^e|} \sum_{(u,q) \in \mathcal{S}_k^e} (\hat{y}_{u,q} - y_{u,q})^2}$$

where y is either response time r or net votes v.

Training and testing. 5-fold stratified cross validation is used to train and evaluate each predictor. More specifically, in each iteration k, 20% of user-question pairs over Ω with $a_{u,q} = 1$ are allocated randomly to each fold, with four then used as $\mathcal{S}_k^T$ and one used as $\mathcal{S}_k^e$. Due to variation in user activity (Fig. 4a), each user's answers are allocated uniformly (stratified) across folds. This procedure is followed for both the $v_{u,q}$ and $r_{u,q}$ prediction tasks. For $a_{u,q}$, negative samples ($a_{u,q} = 0$) are also needed; as a result, we follow a procedure similar to [6] and sample $|\mathcal{S}_k^T \cup \mathcal{S}_k^e|$ user-question pairs with $a_{u,q} = 0$ equally across questions Q and randomly allocate them to $\mathcal{S}_k^T$ and $\mathcal{S}_k^e$. In this way, each fold has a balanced number of samples. Cross validation is repeated 5 times, for a total of 25 iterations. In each iteration, the features $\boldsymbol{x}_{u,q}$ are computed over the set of questions $\mathcal{F}(q)$ for each pair.

Two different network configurations are used for response quality $v_{u,q}$ and initial excitation $\mu_{u,q}$. The network configuration for response quality is $L = 4$ with 20 hidden units in each layer and nonlinearity $\sigma = \text{ReLU}$. For the initial excitation, a shallower network configuration was used with $L = 2$ having 100 and 50 hidden units in each layer, respectively, and nonlinearities $\sigma = \tanh$ for the hidden layers and $\sigma = \text{ReLU}$ for the output layer. On the other hand, we found that neural networks for the decay rate $\omega_{u,q}$ did not yield benefit over a constant value $\omega_{u,q} = 10,000$ on this dataset, though

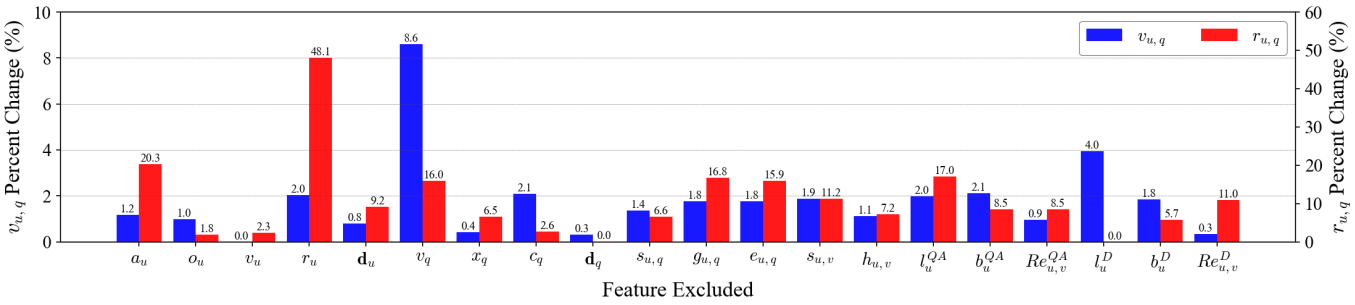


Fig. 6: Feature importance analysis for response quality $v_{u,q}$ and timing $r_{u,q}$ predictions. The percent change in RMSE from the full feature set $\boldsymbol{x}_{u,q}$ is shown when each is removed one-by-one (left axis for v, right axis for r). The importance of user and question features tend to vary widely by task (with r_u and v_q being most important, respectively), while the user-question and social features are more consistent.

we believe increased performance can be achieved in other applications by modeling nonlinearities in both $\mu_{u,q}$ and $\omega_{u,q}$. For MF and SPARFA we set the latent dimension to 5 and 3, respectively [19], and for LDA we set $K = 8$ topics [4]. Larger parameter values did not alter our results substantially.

B. Performance Comparison with Baselines

To establish the overall quality of our methodology, our first experiment evaluates each predictor on the full set of questions (i.e., $\Omega = Q$), with each feature vector computed on all prior question data (i.e., $\mathcal{F}(q) = \{q' : q' \leq q\}$ with questions ordered chronologically). Table I shows the means and standard deviations of the metrics obtained on each prediction task for both the baselines and our model. Overall, we find that *our algorithms outperform the baselines for each task*, with improvements of 22-23% in each case. The improvement that our algorithm obtains on $a_{u,q}$ validates our defined set of features, and that of $v_{u,q}$ and $r_{u,q}$ validates our point process and neural network model design. These results also shows that *it is possible to predict both response timing and quality simultaneously*, despite these quantities having been seen to be entirely uncorrelated in Figure 3.

For completeness, we also run an experiment varying the number of topics K used for each prediction task. The results are shown in Figure 5, where we measure the percent change in each evaluation metric from the default $K = 8$ for several choices of K. The number of topics has virtually no effect on the $r_{u,q}$ task, while it has a small affect on $a_{u,q}$ and a more noticeable impact on $v_{u,q}$. While $K = 8$ obtains close to the best results for $a_{u,q}$, up to 5% increase in performance can be obtained for $v_{u,q}$ by changing $K = 15$. This implies that even further improvement could be obtained over the baseline for $v_{u,q}$ in Table I by treating K as a tunable parameter.

C. Feature Importance Analysis

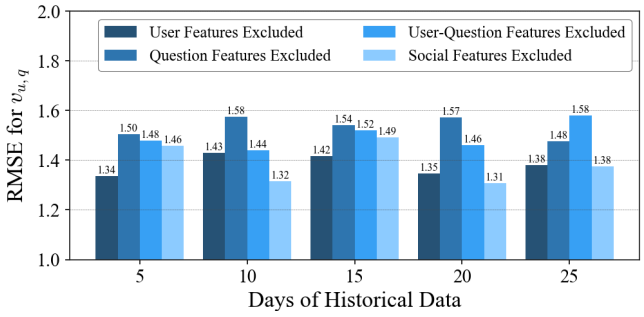
We now assess the impact of each feature comprising $\boldsymbol{x}_{u,q}$ (specified in Sec. II-B) on the response time $r_{u,q}$ and quality $v_{u,q}$ prediction tasks. To do this, we run 20 experiments with each feature excluded one-by-one, taking $\Omega = Q$ and $\mathcal{F}(q) = \{q' : q' \leq q\}$, and measure the average percent increase in RMSE from the full feature set case. The results are plotted in Figure 6; in what follows, we discuss key observations.

At a high level, we see that individual features tend to be more important for $r_{u,q}$ (right axis) than $v_{u,q}$ (left axis). This implies that *response time is a more significant interplay between the features than response quality*. In particular, the greatest percent change in RMSE for $r_{u,q}$ was 48% from excluding r_u, the average response time of the user, as opposed to 8.6% for $v_{u,q}$ due to v_q, the votes received on the question.

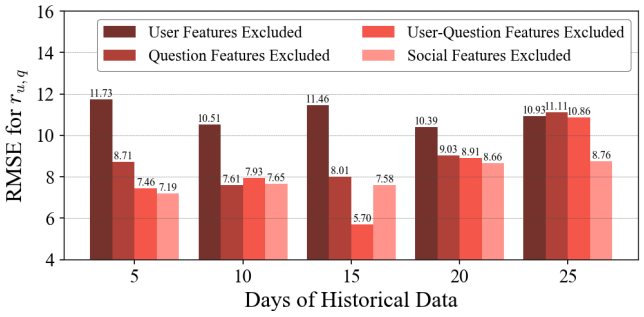
It is intuitive that v_q is important to the response quality task. On the other hand, v_u, the prior user votes, was not seen to affect the RMSE of this task at all. In fact, *none of the user features are particularly important to predicting net votes with the exception of* r_u, which is surprising due to the observed lack of correlation between these quantities in Figure 3. The importance of v_q and r_u to $v_{u,q}$ implies that the desire by users for a question to be answered can lead to a better response, or at least that having more users interested in the thread generates more reactions when a response is posted. In contrast, *the user features are as a whole rather important to predicting response time*, with r_u and a_u – the number of prior answers – being the most predictive. This suggests that active answerers are more likely to respond quickly, a tendency which is captured by the point process model, and is consistent with Figure 4b.

The user-question features tend to have higher importance than the user features for $v_{u,q}$, and than the question features for $r_{u,q}$. $g_{u,q}$, the topic-weighted questions answered, and $e_{u,q}$, the topic-weighted answer votes, are both rather important to $r_{u,q}$, suggesting that while an answerer's history of net votes is not indicative of their response time, *the votes they received towards the topic of the question motivate them to respond quicker*. Also, $s_{u,q}$, the topic similarity between answerer and question, is less important than $s_{u,v}$, that between the asker and answerer. This suggests that *user discussion similarities are more predictive of timing and quality than the questions themselves*, reinforcing the observation in Figure 4d. The fact that the user and question topic distributions ($\mathbf{d}_u$ and $\mathbf{d}_q$) are not as important as $s_{u,v}$ also suggests that users respond according to similarity rather than universally popular topics.

Some of the social features are even more important than user-question features: the exclusion of l_u^{QA}, the closeness centrality of the question-answer graph, causes 2.0% and 17% improvements in RMSE for $v_{u,q}$ and $r_{u,q}$, while l_u^D, the



(a) Net votes $v_{u,q}$ task.



(b) Response timing $r_{u,q}$ task.

Fig. 7: Results for the impact of the length of historical data on the predictive capability of each group of features. In each experiment, a group of features is excluded from $\boldsymbol{x}_{u,q}$, the included features are computed over a window of historical data, and the RMSE of the resulting model is evaluated on the last five days of threads. The user, question, and user-question features are each most important in at least one case, underscoring the importance of including a diverse set of features as the level of historical data can vary in practice.

closeness in the denser graph, changes the RMSE of $v_{u,q}$ by 4.0%. Overall, this suggests that *topics aside, the inferred social network structure has features that are rather predictive of both tasks*. The betweenness centralities (b_u^{QA} and b_u^D) of both graphs are also important (though more so for the question-answer graph), which can be explained by users who are connected to multiple sub-communities being more active and/or possibly able to collect information on topics across complementary threads. The importance of social features on both graphs is consistent with their variation observed in Figure 4f, suggesting in general that *care must be taken in how the network structure is defined* for prediction. Also, the lesser importance of the resource allocation indices ($Re_{u,v}^{QA}$ and $Re_{u,v}^{D}$) to the tasks compared with other features is in contrast to [6], which found this feature most predictive of user interactions in the forums of online courses: topic similarity seems to play a more noticeable role on CQA sites.

D. Impact of Historical Data

Finally, we study how the importance of each group of features varies based on the timeframe of historical data available for inference. To do this, letting $D_i \subset Q$ be the set of questions created in day $i = 1, \ldots, 30$ of the dataset, we fix $\Omega = \{q \in D_{25} \cup \cdots \cup D_{30}\}$ as the last days for evaluation, and run 20 experiments for the $v_{u,q}$ and $r_{u,q}$ prediction tasks varying both (i) which of the four feature groups is excluded from $\boldsymbol{x}_{u,q}$ and (ii) the inference set $\mathcal{F}(q) = \{q' \in D_{25-i} \cup \cdots \cup D_{25}\}$ for $i = 5, 10, \ldots, 25$. Higher values of i give more days of historical data on which each feature is computed. The results are given in Figure 7, showing the average RMSE obtained in each case; a taller bar implies higher importance of the excluded feature for the given experimental setting.

Overall, we see that each of the user, question, and user-question features have at least one instance in which they are most important. This underscores *the necessity of including diverse feature groups in the model*, as the level of historical data available can vary. More specifically, over the first 20 days of historical data, the question and user feature groups are most important to $v_{u,q}$ and $r_{u,q}$, respectively, consistent with the finding in Figure 6 of v_q and r_u having the highest impact. With 25 days of data, however, the user-question features become most critical to $v_{u,q}$, and are among the most important to $r_{u,q}$. This implies that *associations between answerers and questions are particularly sensitive to the amount of historical data*, which may be due to topic similarities ($s_{u,v}$) and topic-weighted votes ($e_{u,q}$) becoming more stable over time.

The social features show opposing trends between the two prediction tasks: for $r_{u,q}$, they monotonically increase in importance over time, while for $v_{u,q}$ they are more variable and actually decrease with more historical data. This indicates that *when viewed over a long timescale, the social network structure contains information more predictive of response times, while a recent window of interactions may be more indicative of response quality*. This may be explained by answerers tending to respond quicker to askers they have a long history of interaction with, while the newer connections arising in a more recent time window tending to be formed by answerers seeking out topics they have expertise in.

V. Discussion and Question Recommendation

The results in Sec. IV show overall that our methodology can effectively predict who ($a_{u,q}$), when ($r_{u,q}$), and with what quality ($v_{u,q}$) a question posted on an online discussion forum will be answered, with large performance improvements over baselines (Table I). They also indicate that user, question, user-question, and social features are each important to the prediction tasks in their own right (Figure 6), while the most predictive features may vary depending on the specific task and length of historical data available for training (Figure 7).

Referring back to Figure 1, the final step of our methodology is to build a question recommendation system. We will now formulate a question routing algorithm that uses our predictors to jointly optimize response time and quality.

Question recommendation. At time indices $n = 1, 2, \ldots$ separated by a fixed interval (e.g., once an hour), we are interested in recommending a new question q' that arrives between n and $n+1$ to the set of users that are predicted to post high-quality answers in a short period of time. Using the

available sets of questions $Q(n)$ and users $U(n)$, the feature vectors $\boldsymbol{x}_{u,q'}(n)$ are computed for each $u \in U(n)$, along with the predictions $\hat{a}_{u,q'}(n) = F_a(\boldsymbol{x}_{u,q'}(n))$, $\hat{v}_{u,q'}(n) = F_v(\boldsymbol{x}_{u,q'}(n))$, and $\hat{r}_{u,q'}(n) = F_r(\boldsymbol{x}_{u,q'}(n))$. With this, the set $U_{q'} = \{u : \hat{a}_{u,q'}(n) \geq \epsilon\}$ of eligible answerers to q' is obtained (where ϵ is a tunable parameter), and the following optimization problem is solved for q' over $U_{q'}$:

$$\begin{aligned} \underset{\boldsymbol{p}^{q'}(n)}{\text{maximize}} \quad & \sum_{u \in U_{q'}} (\hat{v}_{u,q'}(n) - \lambda_{q'} \hat{r}_{u,q'}(n)) \cdot p_u^{q'}(n) \\ \text{subject to} \quad & 0 \leq p_u^{q'}(n) \leq c_u - \sum_q \sum_{i=0}^{I} z_{u,q}(n-i), \ \forall u \in U_{q'} \\ & \sum_{u \in U_{q'}} p_u^{q'}(n) = 1. \end{aligned} \tag{2}$$

Here, $\boldsymbol{p}^{q'}(n) = (p_1^{q'}(n), p_2^{q'}(n), \ldots)$ is a probability distribution over the eligible set of users $U_{q'}$, and $p_u^{q'}(n)$ can be interpreted as the probability that u will be recommended to answer q'. $\epsilon \in (0, 1)$ controls the tradeoff between conforming to answerer behavior (i.e., recommending questions they would likely answer anyway) and the number of choices $|U_{q'}|$ available to the recommendation system. $\lambda_{q'}$ is another parameter, controlling the importance of response quality ($\hat{v}_{u,q'}$) versus timing ($\hat{r}_{u,q'}$) for the particular question q', and might be set by the question asker. c_u is an upper bound on the amount of questions u can answer in a time period I (due to external factors, e.g., time commitments), from which the number of observed answers is subtracted, with $z_{u,q}(n) = a_{u,q}(n) - a_{u,q}(n-1)$ denoting whether u answered q between $n-1$ and n. Like $\lambda_{q'}$, c_u may also be user specified, or could be inferred from user behavior collected over time.

The choice of $p_u^{q'}(n)$ as a probability across users rather than a binary assignment to a single user has several advantages. First, it makes (2) a linear program, which can be solved substantially faster than integer programs at this scale [4]. Second, it generates a ranking of potential responders that can be drawn from several times until an answer is recorded.

VI. Conclusion

In this paper, we developed novel methodology for the joint prediction of response quality and timing in online discussion forums. Our neural network and point process-based algorithms learn over a set of 20 features for each sample that we formulated and divided into four groups: user, question, user-question, and social features. Through evaluation on a dataset from Stack Overflow consisting of 20,000 question threads, we found that our models were able to obtain substantial improvements of more than 20% over baselines for each prediction task, and that the features most important to the predictions vary based on the task and amount of historical data available. Using our models, we finally proposed a question recommendation system that utilizes our three predictors to recommend questions to be answered by jointly optimizing net votes and response time. The main next step for future work is incorporating our recommendation system into an online forum platform to observe its impact; the quality of the approach could be evaluated through A/B testing, comparing the net votes and response times observed in a group with the system in use to one with it not. The learnt features can provide analytics to forum administrators too.

References

[1] G. Wang, K. Gill, M. Mohanlal, H. Zheng, and B. Y. Zhao, "Wisdom in the Social Crowd: An Analysis of Quora," in *WWW*. ACM, 2013, pp. 1341–1352.

[2] X. Cheng, S. Zhu, S. Su, and G. Chen, "A Multi-Objective Optimization Approach for Question Routing in Community Question Answering Services," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 9, pp. 1779–1792, 2017.

[3] T. C. Zhou, M. R. Lyu, and I. King, "A Classification-based Approach to Question Routing in Community Question Answering," in *WWW*. ACM, 2012, pp. 783–790.

[4] C. G. Brinton, S. Buccapatnam, F. M. F. Wong, M. Chiang, and H. V. Poor, "Social Learning Networks: Efficiency Optimization for MOOC Forums," in *IEEE INFOCOM*. IEEE, 2016.

[5] M. Bouguessa, B. Dumoulin, and S. Wang, "Identifying Authoritative Actors in Question-answering Forums: The Case of Yahoo! Answers," in *ACM SIGKDD*. ACM, 2008, pp. 866–874.

[6] T.-Y. Yang, C. G. Brinton, and C. Joe-Wong, "Predicting Learner Interactions in Social Learning Networks," in *IEEE INFOCOM*. IEEE, 2018.

[7] C. G. Brinton and M. Chiang, "MOOC Performance Prediction via Clickstream Data and Social Learning Networks," in *IEEE INFOCOM*. IEEE, 2015, pp. 2299–2307.

[8] M. Qu, G. Qiu, X. He, C. Zhang, H. Wu, J. Bu, and C. Chen, "Probabilistic Question Recommendation for Question Answering Communities," in *WWW*. ACM, 2009, pp. 1229–1230.

[9] L. Wang, B. Wu, J. Yang, and S. Peng, "Personalized Recommendation for New Questions in Community Question Answering," in *IEEE ASONAM*. IEEE, 2016, pp. 901–908.

[10] J. Yang, S. Peng, L. Wang, and B. Wu, "Finding Experts in Community Question Answering Based on Topic-Sensitive Link Analysis," in *IEEE DSC*. IEEE, 2016, pp. 54–60.

[11] F. M. F. Wong, Z. Liu, M. Chiang, F. Ming Fai Wong, Z. Liu, and M. Chiang, "On the efficiency of social recommender networks," *IEEE/ACM Transactions on Networking*, vol. 24, no. 4, pp. 2512–2524, 2016.

[12] M. Glenski and T. Weninger, "Predicting User-Interactions on Reddit," in *IEEE/ACM ASONAM*. ACM, 2017, pp. 609–612.

[13] R. Xiang, J. Neville, and M. Rogati, "Modeling Relationship Strength in Online Social Networks," in *WWW*. ACM, 2010, pp. 981–990.

[14] A. S. Lan, J. C. Spencer, Z. Chen, C. G. Brinton, and M. Chiang, "Personalized Thread Recommendation for MOOC Discussion Forums," in *ECML-PKDD*, 2018.

[15] Y. Yao, H. Tong, T. Xie, L. Akoglu, F. Xu, and J. Lu, "Joint Voting Prediction for Questions and Answers in CQA," in *IEEE/ACM ASONAM*, 2014, pp. 340–343.

[16] Y. Yao, H. Tong, F. Xu, and J. Lu, "Scalable Algorithms for CQA Post Voting Prediction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 8, pp. 1723–1736, 2017.

[17] D. J. Daley and D. Vere-Jones, *An Introduction to the Theory of Point Processes*. Springer, 2003.

[18] M. Farajtabar, S. Yousefi, L. Tran, L. Song, and H. Zha, "A Continuous-Time Mutually-Exciting Point Process Framework for Prioritizing Events in Social Media," *arXiv preprint arXiv:1511.04145*, Nov 2015.

[19] A. S. Lan, A. E. Waters, C. Studer, and R. G. Baraniuk, "Sparse Factor Analysis for Learning and Content Analytics," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1959–2008, 2014.

[20] D. Yang, D. Adamson, and C. P. Rosé, "Question Recommendation with Constraints for Massive Open Online Courses," in *ACM RecSys*. ACM, 2014, pp. 49–56.

[21] Y. Koren, "Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model," in *ACM SIGKDD*. ACM, 2008, pp. 426–434.

[22] T. Karagiannis, M. Molle, M. Faloutsos, and A. Broido, "A Nonstationary Poisson View of Internet Traffic," in *IEEE INFOCOM*, vol. 3. IEEE, 2004, pp. 1558–1569.