Great Question! Question Quality in Community Q&A

Sujith Ravi[†] **Bo Pang**[†] **Vibhor Rastogi**[*] **Ravi Kumar**[†]
[†]Google [*]Twitter
Mountain View, CA San Francisco, CA
{ravi.sujith, bopang42, vibhor.rastogi, ravi.k53}@gmail.com

Abstract

Asking the right question in the right way is an art (and a science). In a community question-answering setting, a good question is not just one that is found to be useful by other people—a question is good if it is also presented clearly and shows prior research. Using a community question-answering site that allows voting over the questions, we show that there is a notion of question quality that goes *beyond* mere popularity. We present techniques using latent topical models to automatically predict the quality of questions based on their content. Our best system achieves a prediction accuracy of 72%, beating out strong baselines by a significant amount. We also examine the effect of question quality on the dynamics of user behavior and the longevity of questions.

Introduction

> What we observe is not nature itself, but nature exposed to our method of questioning.
>
> — Werner Heisenberg

Life is replete with questions. Questions ranging from the health perils of accidentally consuming water with a drowned cockroach to the benefits of practicing daily yoga to playing April fool pranks on colleagues. We have questions all the time and we seek answers by hook or crook. In fact, in almost every human endeavor, progress is made when we start asking the right questions faster than we can answer them. Our work itself is from a casual conversation question: what makes a question great? Thankfully, in this century, the Web has answers to *all* the questions. Almost.

Not all questions are destined to be equal. While the value of a question is a function of the collective benefits a community derives from it, the *quality* of a question is a subtler and possibly less objective notion. Intuitively, quality must reflect scholarship, discernment, and research effort—a good question simply cannot stem out of indolence. The widely-used slang RTFM arose out of our frustration in dealing with poor quality questions—those that could have been readily answered by reading the fine manual. FAQs are another by-product of efforts to glean good quality questions (with not-so-easy-to-find answers) that might confront many of us.

For online community question-answering websites (such as `quora.com`, `answers.yahoo.com`, `stackoverflow.com`), the notion of question quality is critical for many reasons. Quality begets quality: promoting or rewarding high quality questions will lead to more of the kind. This will improve the site reputation, drive traffic, provide a better user experience, encourage experts to frequent the site to seek questions that challenge their expertise, and boost web result rankings. Many question-answering sites offer badges and other incentives to encourage users to ask high quality and less frivolous questions.

While there has been a lot of research effort to address answer quality, to the best of our knowledge, there has been very little work on understanding the quality of a question, i.e., what would make a question great? As an illustration, consider the following two questions about the C programming language:

1. "Why does C have a distinction between `->` and `.`?"[1]
2. "Putting user input into char array (C Programming)."[2]

It can be argued that the former is a higher quality question than the latter since the former might cause some of us to pause and ponder why Dennis Ritchie made this syntax decision in C. In fact, the behavior of the StackOverflow community precisely reflects this: even though the latter has an order of magnitude more views (23K vs. 1.5K), the number of up votes for the former is an order of magnitude more (44 vs 1); an *up* vote for a question in StackOverflow means the question "shows research effort; it is useful and clear." This example also clearly shows that the number of views, which is a proxy for the popularity of the question in the community, may not be a faithful reflection of question quality.

Our contributions. In this work we study the notion of question quality. For this purpose, we consider the questions in StackOverflow, a community question-answering site that is publicly downloadable. Our endeavor begins with the definition of a quality of the question. This definition is based on a careful analysis of the interplay between the number

[*]The research described herein was conducted while the author was working at Google.

[1] `stackoverflow.com/questions/1813865/`
[2] `stackoverflow.com/questions/1407461/`

of views and the number of up votes a question has garnered. We develop a binary classifier for this problem using the question content. We employ models that capture the latent topical aspects of questions at three levels: (i) a global model that captures the topics for the question as a whole, (ii) a local model that captures the topics at a sentence level, and (iii) a global topic structure (Mallows) model (Fligner and Verducci 1986) that enforces structural constraints over the sentence-level topics within a question.

Our methods do not rely on signals such as the number of views since such signals would be unavailable for newly posted questions. Nonetheless, our classifier is able to achieve an accuracy of 72%, significantly beating out strong baselines such as the number of views.

En route, we study the web search queries that lead to clicks on StackOverflow questions (and answers). Our analysis of the web queries lends further credence to the notion of question quality: higher quality questions continue to be queried at a rate higher than lower quality questions.

Related work

Predicting question quality. Unlike predicting answer quality, little attention has been devoted for analyzing and predicting question quality. A recent line of work (Li et al. 2012; Bian et al. 2009; Agichtein et al. 2008) has emerged to address this problem in the context of Yahoo! Answers. Since questions in Yahoo! Answers do not have a well-defined notion of quality, attempts have been made to define such a (subjective) notion. Bian et al. use manual labels for defining quality for a set of 250 questions. Li et al. define question quality as some combination of the number of tags of interest, the number of answers, and the reciprocal of time to obtain the first answer; they use domain experts along with authors' judgments for getting the ground truth. In both these works, the ground truth is limited to a relatively small number of questions. On the other hand, our prediction task uses StackOverflow questions, where there is a well-defined notion of up votes and down votes for questions; this enables us to define question quality in a less subjective, principled, and automatic fashion for a large number of questions.

Another important difference between our work and existing work on predicting question quality is the prediction model. Existing work (Li et al. 2012; Bian et al. 2009; Agichtein et al. 2008; Anderson et al. 2012) model question quality as a function of the reputation of the question asker, the question category, and simple statistics about question content, such as length, number of typos, number of words per sentence, etc. The models do not use the actual question content (such as its ngrams or its latent topics), but instead pay particular attention on propagating the asker's reputation to question quality. However, such approaches would suffer for questions asked by new users. For this purpose, we use the actual question content and topic models, and show them to be powerful predictors of question quality.

Other related work in CQA. There has been extensive prior work on Community-based question answering (CQA) that focused on analyzing and predicting answer quality (Jeon et al. 2006; Shah and Pomerantz 2010; Tian, Zhang, and Li 2013) or asker satisfaction (Liu, Bian, and Agichtein 2008). Given the focus on *answer* quality, they are only tangentially related to our work. There has also been work on discovering expert users in CQA sites, which has largely focused on modeling expert *answerers* (Sung, Lee, and Lee 2013; Riahi et al. 2012). We leave it as interesting future work to further extend our models, which focus on the content of questions, to also incorporate asker information in order to capture the intuition that certain people tend to ask good questions. Work on discovering expert users was often positioned in the context of routing questions to appropriate answerers (Li and King 2010; Li, King, and Lyu 2011; Zhou, Lyu, and King 2012). This, in turn, is related to work on question recommendation (Wu, Wang, and Cheng 2008; Qu et al. 2009; Li and Manandhar 2011; Szpektor, Maarek, and Pelleg 2013), which seeks to identify questions of interest for answerers by optimizing for topical relevance as well as other considerations such as diversity. Not surprisingly, some of these studies have explored global topics (the topics of the question as a whole). We differ from such work in two aspects. First, we are addressing a different task: rather than identifying questions that are of interest to certain answerers, we wanted to identify questions that are appreciated by people (not just those who are qualified to answer) who are interested in this subject. Second, in addition to modeling the global topics of questions (what is the question about), we also attempted to capture different aspects of question formulation (how is the question stated).

Building topic models. Recently, Bayesian models have become increasingly popular tools for solving a variety of structured prediction problems in NLP (Chiang et al. 2010) and other areas. A prominent use of Bayesian inference is in topic modeling, which has found applications in information retrieval and NLP for a broad variety of tasks such as summarization (Daumé and Marcu 2006), inferring concept-attribute attachments (Reisinger and Paşca 2009), selectional preferences (Ritter, Mausam, and Etzioni 2010), name ambiguity resolution (Kozareva and Ravi 2011), and cross-document co-reference resolution (Haghighi and Klein 2010). Topic models such as Latent Dirichlet Allocation (LDA) (Blei, Ng, and Jordan 2003) are generative models for documents and represent hidden topics (where a topic is a probability distribution over words) underlying the semantic structure of documents. An important use for methods such as LDA is to infer the set of topics associated with a given document or a document collection.

Allamanis and Sutton (2013) propose a technique to analyze StackOverflow questions using LDA, considering each question as a document. They propose three LDA models, each differing in the construction of document for a given question. The first one performs LDA over the entire question body, the second over code snippets in the question, and the third on the part of question obtained by removing noun phrases. The first two models cluster questions based on noun concepts like 'applets', 'games', 'java'; the third model clusters questions based on activity (i.e., verbs) rather

than nouns: such as questions focusing on 'build issues' independent of the language. However, they do not try to predict quality of the questions using their model. On the other hand, we show that latent topics inferred from the question content can be powerful predictors of the question quality. We also construct a more intricate model to capture structural similarities such as topic ordering among related questions.

Other work. Techniques have also been proposed to automatically infer an interesting set of features from the question-answer pairs. Heilman and Smith (2010) propose an approach for automatically generating questions by overgenerating and ranking the questions based on some quality metric; both the methodology and goal of this work is different from ours.

Data

The primary dataset used in this paper is a publicly available dataset from StackOverflow. We obtained a copy of the data dump of StackOverflow[3] made available by the site. Along with the textual content of questions, this dataset also includes a rich set of meta-information. We extracted two years worth of questions from this data dump, which includes all questions, answers, and comments, timestamped between 2008 and 2009. In total, we extracted 410,049 questions. For each question, we extracted basic information like its title and body, author and timestamp. We also extracted the total number of views (*ViewCount*), as well as the number of up votes and down votes on this question. These will form the basis of our definition of question quality and will be discussed later in detail.

In addition, we also extracted 10,284,555 answers and 22,666,469 comments posted during the same time period and aligned them to the questions in our dataset. Answers are supposed to provide an answer meant for the general public, whereas comments could be used to request clarifications and can be considered as a message to the person who posted the question, though there can be misuses or confusions as to the etiquette[4]; comments also require a higher reputation[5]. Overall, 99.4% of the questions in the dataset received at least one answer and 37% of questions received at least one comment.

A notion of question quality

What makes a question *good*? As we mentioned earlier, each question on StackOverflow can be voted up or down by users of the site. According to prompts on the site, a question should be voted up if the "question shows research effort; it is useful and clear." Suppose we take this as the definition of a *good* question. Let the number of up (resp. down) votes received by question q_i be denoted by n_i^+ (resp. n_i^-). StackOverflow defines a $s_i = n_i^+ - n_i^-$, which is the *Score* prominently displayed right next to the question. Can we use this as the quantitative measure for question quality?

Intuitively, one might think *Score* is a good measure of question quality: questions with a positive *Score* would be good and ones with a negative *Score* would be bad (a zero *Score* can either mean a equal number of non-zero up and down votes or no votes at all). But we notice that the *Score* distribution is highly skewed (Table 1).

Score	< 0	$= 0$	> 0
% of questions	1.0	26.8	72.2

Table 1: Percentage of questions for different *Score* ranges.

Does this mean predominantly many questions on StackOverflow are considered good by its community? To better understand this, we need to study the relationship between voting and user reputation. On StackOverflow, each new user starts with a reputation of one point and they can gain or lose points based on their behavior on the site[6]: among other things, users gain (or lose) points when their questions or answers are voted up (or down). In turn, voting on StackOverflow is modulated by this user reputation system: up votes can be cast by users with 15 or more reputation, and down votes can be cast by users with 125 or more reputation, subject to the constraint that each user can cast no more than 30 question votes per day[7]. As a result, the number of users eligible for down votes is significantly smaller than the number of users eligible for up votes. Furthermore, during the timespan of this dataset, voting a question[8] or an answer down cost one point for the voter. Overall, we observe that the number of down votes cast for questions (29,585) is an order of magnitude smaller than the number of up votes (302,834). But this could be due to the above factors, rather than a reflection of the intrinsic question quality distribution.

What we can infer from the above observation is the following: if a question were to get a vote at all, it is much more likely to be an up vote. As an approximation, we can ignore the down votes and simply consider an up vote as an endorsement and its absence (when there are enough views) as a silent disapproval. But if we simply use positive *Score* vs. non-positive *Score* to label questions as good vs. bad, we may be conflating quality with popularity—a question that is viewed many times has more chances of getting a vote. This leads us to study *ViewCount* in conjunction with *Score*.

As shown in Figure 1, *ViewCount* for 87.9% of questions falls in the range $(100, 10k)$: 42% $\in (100, 1k)$ and 45.9% $\in (1k, 10k)$. Figure 2 shows how different the distribution of *Score* can be for different ranges of *ViewCount*. For instance, 41.6% of questions with *ViewCount* in the range

[3] http://blog.stackoverflow.com/2009/06/stack-overflow-creative-commons-data-dump/

[4] http://meta.stackoverflow.com/questions/17447/answer-or-comment-whats-the-etiquette

[5] http://meta.stackoverflow.com/questions/7237/how-does-reputation-work

[6] http://meta.stackoverflow.com/questions/7237/how-does-reputation-work

[7] http://meta.stackoverflow.com/questions/5212/are-there-any-voting-limits

[8] Note that since May 2011, down-voting on questions no longer cost a point, in an effort to incentivize more balanced voting (meta.stackoverflow.com/questions/90324/should-downvotes-on-questions-be-free).

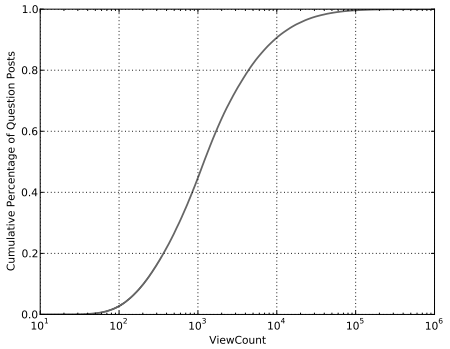

Figure 1: Cumulative distribution of *ViewCount*.

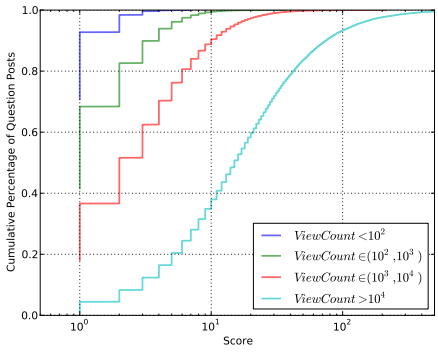

Figure 2: Cumulative distribution of *Score* for different ranges of *ViewCount*.

$(100, 1k)$ have non-positive *Score* and the percentage drops to less than 1% for questions with *ViewCount* more than $10k$. Clearly, questions with a higher *ViewCount* are more likely to get a higher *Score*. If we were to define a classification (or regression) task based on *Score* alone, it is unclear if this notion can capture the quality of a question as much as it captures its popularity.

To address this issue, we consider the value $p_i = s_i/v_i$, where v_i is the *ViewCount* for question q_i. The average value of p_i is about 0.002 (0.003 when conditioned on questions with positive p_i). We focus on questions with v_i at least 1000: this way, if we observe a zero p_i, we are more confident that this reflects the poor quality of the question rather than the question not being viewed by a qualifying user who can vote. And, if we observe p_i more than 0.001, we are more confident that this reflects the good quality of the question, rather than an incidental click on the up vote. More specifically, we label questions with $p_i = 0$ as *bad* and those with $p_i > 0.001$ as *good*. We then downsample the good questions to create an equal distribution of good/bad samples. This labeled dataset is then split into training data Q_{train} and test data Q_{test} with 33,199 questions each. The good/bad label distribution is uniform for both Q_{train} and Q_{test}.

Predicting question quality

Given the importance of question quality and its observed effects on the interest generated around the question, it is natural to ask the following:

> Can we predict the quality of a question (good vs. bad) posted on a community question-answering site using content of the question alone?

In this section we describe our approach to modeling question quality, motivated by the previously-mentioned aspects that define a good question: is the question useful and clear; does it show research effort? A bag-of-words model can capture each of these aspects partially: a clearly stated question may be worded differently and a question that shows research effort may include phrases like "I have searched" or "I have tried." But some of these aspects may be better captured if we go beyond a bag-of-words representation. For instance, a good question may show *structural clarity*, where the body of the question follows specific patterns of a discourse structure. Before we explore each of these considerations in more detail, we start with a description of the learning framework we adopt.

Let each labeled question q_i be represented as $\langle \ell_i, \mathbf{f_i} \rangle$, where ℓ_i is the binary (good or bad) label for the question and $\mathbf{f_i}$ is the covariate vector comprising a set of question features. Let $Q = Q_{\text{train}} \cup Q_{\text{test}}$ denote the set of all questions, where ℓ_i is unknown for questions in Q_{test}. We treat the task of predicting ℓ_i from $\mathbf{f_i}$ as a binary classification problem. Given a function $\phi(\cdot)$ on the features, the learning objective is to find a weight vector $\mathbf{w}$ that minimizes the mis-classification error on the training corpus Q_{train}:

$$\underset{\mathbf{w}}{\text{minimize}} \sum_{q_i \in Q_{\text{train}}} \|\ell_i - \langle \mathbf{w}, \phi(\mathbf{f_i}) \rangle\|_2^2 + \lambda \|\mathbf{w}\|_2^2, \qquad (1)$$

where λ is a regularization parameter to prevent over-fitting.

We use two types of models for prediction based on the feature function $\phi(\mathbf{f})$: (i) *linear models* that use the identity mapping $\phi(\mathbf{f}) = \mathbf{f}$ and (ii) *nonlinear models*, where $\phi(\mathbf{f})$ represents a nonlinear mapping using Nystrom approximation for Gaussian RBF kernels. More specifically, let $k(\mathbf{f}, \mathbf{f}') = \exp(-\gamma \|\mathbf{f} - \mathbf{f}'\|^2)$ denote a Gaussian RBF kernel as commonly used in kernel methods (Smola and Schölkopf 2000; Navalpakkam et al. 2013), where γ is a parameter. This kernel can be approximated by

$$\tilde{k}(\mathbf{f}, \mathbf{f}') = \langle \tilde{\phi}(\mathbf{f}), \tilde{\phi}(\mathbf{f}') \rangle, \text{ where}$$
$$\tilde{\phi}(\mathbf{f}) = K^{-\frac{1}{2}} \cdot (k(\mathbf{f}_1, \mathbf{f}), \ldots, k(\mathbf{f}_n, \mathbf{f}))$$

Here $\mathbf{f}_1, \ldots, \mathbf{f}_n$ are feature vectors corresponding to questions from Q_{train}[9] and K is an $n \times n$ matrix obtained by forming the inner products with $K_{ij} = k(\mathbf{f}_i, \mathbf{f}_j)$. The advantage of the mapping $\tilde{\phi}(\mathbf{f})$ is that it can be used to learn a linear function in the transformed feature space that is equivalent to learning a nonlinear function in the input space.

We use the *Vowpal Wabbit* package (Langford, Li, and Strehl 2007) to solve the optimization problem (1). We extract several information signals pertaining to the question

[9] In our experiments, we set n to be 10% of the training data size.

and use them as features to build prediction models. Next, we describe the details of the prediction models.

Question content

We first parse the content from a given question by applying a sequence of preprocessing steps—HTML parsing, sentence splitting, tokenization, stopword removal, and stemming. We treat the question title and body as separate signals for building the prediction model. We then extract the following two types of features:

(i) Length: the word/token count and sentence count in the title and body.

(ii) Text: the ngram features extracted from the title and body after tokenization. We use unigram text features in our experiments. We experimented with higher-order ngrams on a held-out dataset but did not observe any noticeable gain in performance for the prediction task. So, we only add unigram features which keeps the models more compact and leads to faster training.

Global topic models

Questions posted on StackOverflow can be categorized into different general topics (e.g., "help with general programming concepts" vs. "debugging errors"). The voting pattern for a certain type of question can differ considerably from others. The community interested in a certain topic might be more (or less) parsimonious about handing out up votes. Or a question about an arcane topic may not generate too much interest. We explore the topic information as an additional signal in the prediction task. Since this information reflects what the entire question is about, we refer to it as the global topic (in order to differentiate with other topic models described later in the paper). The topic can be learned in an unsupervised manner using latent topic models.

To incorporate our earlier intuition, we first train a *global LDA topic model* over all questions Q. For any $q \in Q$ we add a feature for topic t with weight θ_{qt} to the prediction model, where $\theta_{qt} = \Pr[t \mid q]$. For the rest of the paper, let K be the number of topics.

Learning global topic models. For training the LDA model, we use the online variational Bayes (VB) algorithm (Hoffman, Blei, and Bach 2010). Unlike typical LDA inference approaches, this method is more efficient in dealing with massive document collections as well as streaming data, which is applicable to our setting if we want to incorporate new questions posted to the site into our model incrementally. The algorithm is based on online stochastic optimization and fits a topic model to the entire data in multiple passes as described below for completeness.

ONLINE LDA INFERENCE ALGORITHM.
Until converged:
(i) Choose a mini-batch of questions randomly.
(ii) For each question in that mini-batch:

Estimate approximate posterior over what topics each word in each question came from.

(iii) (Partially) update approximate posterior over topic distributions based on what words are believed to have come from what topics.

We train an online LDA model with $K = 10$ topics using a mini-batch size of 256 and ran the inference algorithm for 100 passes over the questions Q.[10] We use sparse Dirichlet priors in the model by setting the hyperparameters to a low value.[11] This encourages the model to learn fewer topics per question and sparse topic word distributions. Finally, we extract the topic distribution θ_{q_i} inferred for a given question $q_i \in Q$ and incorporate this as K features $(\theta_{q_i 1}, ..., \theta_{q_i K})$ for the prediction task.

Local topic models

The global topic model follows the same assumption as traditional topic models that topics are randomly spread throughout a document. The model learns latent topics that correspond to global properties of questions (e.g., "android app", "server client programming", etc.) rather than internal structure within a particular question. For example, a good, useful question is one that contains a clear *problem statement* and in addition demonstrate sufficient details indicating *background research* that the user may have done for finding a solution. This would convey useful information to other users who attempt to answer the question and could affect whether it gets up-voted.

In order to model local properties within individual questions, we propose a different approach. The hypothesis is that internal structural aspects of questions could be better captured using *local topic models*. We introduce two different types of models for this purpose, a sentence-level model and a global topic structure model.

Sentence topic model (STM). Previously, latent variable models have been used to capture local aspects within documents and online reviews (Titov and McDonald 2008). It has been noted that topics extracted by running LDA over sentences rather than documents can better capture local aspects (Brody and Elhadad 2010). We follow a similar approach for our task. First, each question q_i is split into sentences $\{q_i^j\}$ and then we aggregate sentences across all questions to create a corpus $\cup_{ij} q_i^j$. Next, we train a topic model on the sentence corpus using the online LDA inference algorithm. The goal is to learn a model that captures latent aspects internal to a question rather than at the question level. As in the global model, we use sparse Dirichlet priors for learning sentence-topic and topic-word distributions for this model.

Once inference is completed, we compute the topic distribution for a given question q_i by aggregating the topic weights from its constituent sentences:

$$\theta_{q_i t} = \sum_j \theta_{q_i^j t},$$

where $t \in [K]$ is a topic and $\theta_{q_i^j t}$ is the inferred topic weight for t corresponding to sentence j. Finally, we add the resulting topic weights as features to our prediction system.

[10] We use the online LDA implementation in the Vowpal Wabbit package (Langford, Li, and Strehl 2007) for our experiments.

[11] We set the hyperparameter values $\alpha = 0.01, \rho = 0.01$ for fitting the topic-word and document-topic distributions.

An alternative to the sentence model is to learn joint topic models at multiple granularity levels (Titov and McDonald 2008). Note that the STM model completely ignores the ordering of topics within the question. Next, we propose a model that captures such long-range dependencies within a question.

Global topic structure models (GTSM)

The models presented so far do not consider the structural properties (such as topic ordering) of a question while determining topic assignments. An ideal question model should be able to capture certain discourse-level properties: (i) adjacent sentences (e.g., occurring in the same section) should coherently share the same topics and (ii) questions that are related tend to present similar topics in similar orders.

We present a generative model that encodes these long-range dependencies within a question. The model posits a process by which a corpus Q of questions, given as a sequence of sentences q_i^j, can be generated from a set of hidden topic variables. The model's final output consists of a topic assignment z_i^j to each sentence in the question. Topically similar sentences are grouped together and there is a single underlying distribution over a question's entire topic structure. This enforces that the internal topic ordering is shared globally across related questions, thereby allowing us to capture discourse-level properties that are difficult to represent using local dependencies such as those induced by hidden Markov models.

The goal is to learn similar topic structures (sequences) for related questions by modeling a distribution over the space of topic permutations. In order to learn this ordering efficiently, we employ the *generalized Mallows model* (GMM) (Fligner and Verducci 1986), which is an exponential model over permutations. It provides us with a distribution over the space of topic permutations but concentrates the probability mass on a small set of similar permutations, thereby allowing us to group together questions that are structurally similar. Previously, GMMs have been used for ranking (Lebanon and Lafferty 2002; Meila et al. 2007) and other applications such as cross-document comparison and document segmentation (Chen et al. 2009). We follow the same approach of Chen et al. (2009) and use GMM to build a *global topic structure model* over questions posted by users. Unlike previous work, we study its extrinsic effect on predicting question quality.

To recap, our input is a corpus Q of questions, where each question q_i is an ordered sequence of sentences q_i^j, and a number K of topics. Each sentence is represented as a bag of words. We learn a probabilistic model that explains how words in the corpus were generated. The final output is a distribution over topic assignment variables for each sentence. The topic ordering variable π_q for each question q is a permutation over $[K]$ that defines the order in which topics appear in the question (namely, those assigned to sentences from the question); here, π_q is drawn from GMM.

Generalized Mallows model (GMM). GMM represents a permutation as a vector $(v_1, ..., v_{K-1})$ of inversion counts with respect to the identity permutation on $[K]$, where v_j is the number of times a value greater than j appeared before j in the permutation. Every inversion count vector uniquely identifies a single permutation. GMM assigns probability mass to a given permutation according to its distance from the identity using $K-1$ real-valued parameters $(\rho_1, ..., \rho_{K-1})$. The GMM probability mass function is

$$\text{GMM}(\mathbf{v}|\rho) = \frac{e^{-\sum_j \rho_j v_j}}{\psi(\rho)} = \prod_{j=1}^{K-1} \frac{e^{-\rho_j v_j}}{\psi_j(\rho_j)},$$

where $\mathbf{v}$ represents a vector of inversion counts, the distribution over topic orderings is parameterized by ρ and ψ_j is the normalization term

$$\psi_j(\rho_j) = \frac{1 - e^{-(K-j+1)\rho_j}}{1 - e^{-\rho_j}}.$$

A higher value for ρ_j assigns more probability mass to v_j being close to zero, thereby encouraging topic j to have fewer inversions (or reorderings). Next, we describe the details of the generative model:[12]

1. For each topic k, draw a topic-specific word distribution $\theta_k \sim \text{Dirichlet}(\theta_0)$.
2. Draw a topic distribution $\tau \sim \text{Dirichlet}(\tau_0)$, which encodes how likely each topic is to appear overall.
3. Draw the topic ordering distribution parameters $\rho_j \sim \text{GMM}_0(\rho_0, \nu_0)$ for $j = 1$ to $K-1$; this controls how rapidly probability mass decays when having more inversions for each topic.
4. For each question q with N_q sentences:
   (a) Draw a bag of topics $\mathbf{t}_q$ by drawing N_q samples from Multinomial(τ).
   (b) Draw a topic ordering π_q by sampling a vector of inversion counts $\mathbf{v}_q \sim \text{GMM}(\rho)$.
   (c) Compute the vector of topic assignments $\mathbf{z}_q$ for question q's sentences by sorting $\mathbf{t}_q$ according to π_q.
   (d) For each sentence s in question q, sample each word $w_{q,s,j} \sim \text{Multinomial}(\theta_{z_{q,s}})$.

In order to perform inference under this model, we follow the approach of Chen et al. (2009) using a Gibbs sampling scheme.[13] We use $K = 10$ topics and set the Dirichlet hyperparameters to a small value (0.01) to encourage sparse distributions. For the GMM, we set the decay parameter ρ_0 to 1, and sample size prior $\nu_0 = 0.1|Q|$.

Finally, we pick the topic sequence assigned to each question q (i.e., topic order assignment for sentences in q). We then extract ngrams from the topic sequence and add them as features to the prediction system.

[12] $\nu_0, \theta_0, \tau_0, \rho_0$ are fixed prior hyperparameters for the sample size and distributions over topic-word, overall topic and topic ordering, respectively.

[13] http://groups.csail.mit.edu/rbg/code/mallows/

Model	Classification accuracy (%)
Baseline	
Random	50.0
Popularity (*ViewCount*)	61.1
Content	
Length	55.5
Text (unigrams from title, body)	65.8
Text + Length	69.9
Topic	
Global topics	64.2
Local topics	61.4
Global topic structure	55.6
Combined	
Text + Length + Global topics	70.5
Text + Length + Local topics	71.8
Text + Length + Local topics + Global topic structure	71.7
Non-linear	
Text + Length	71.0
Text + Length + Local topics	71.9
Text + Length + Local topics + Global topic structure	72.1

Table 2: Results on predicting question quality. In both linear and non-linear settings, Content+LocalTopics outperforms all other systems significantly ($p < 0.0001$), but is statistically indistinguishable from Content+LocalTopics+GlobalTopicStructure.

Experimental results

Experimental setup

As mentioned earlier, we create an equal-sized train/test split (with ∼33k questions each) for the experiments. For the binary (good/bad) prediction task, we train different models on Q_{train} and use it to predict the label for each question in Q_{test}. We compare different prediction systems in terms of their classification performance (% accuracy) on Q_{test}:

- Baseline models: For comparative purposes, we include a random baseline and a strong baseline based on the popularity of the question (i.e., using *ViewCount* as a single feature). Note that popularity is vacuous for a new question.
- Content models: Using content features (length, words) within the question directly.
- Topic models: Output from the latent variable models: (i) global model, (ii) local model, (iii) global topic structure model (topic sequences produced by the Mallows model);
- Combined models: Combining content and topic models.

For some systems (including our best model), we also compare the performance of linear vs. nonlinear models.

Prediction results

The classification performance of the baselines and the linear models are summarized in Table 2. First, we observe that popularity (*ViewCount*) by itself outperforms the random guess baseline, demonstrating that there is some degree of correlation between *ViewCount* and our notion of question quality. However, our best system outperforms this baseline by more than 10%, showing that our notion of question quality is more intricate than popularity. Since *ViewCount* is not necessarily available for newly posted questions—a case where the prediction task is especially useful—we did not use it as a feature it in our models. Nonetheless, it is an interesting baseline to compare against.

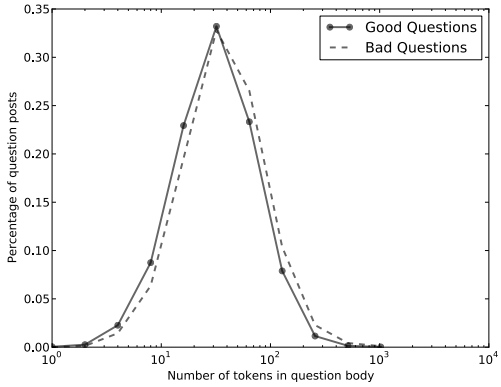

Figure 3: Distribution of question length (number of tokens in the body of the question) for good vs. bad questions. Good questions tend to be slightly shorter.

Topic	Top words				
T1	use	file	like	would	work
T2	string	public	class	int	new
T3	self	context	touch	uiview	composite
T4	table	select	queries	row	column
T5	server	error	service	connect	message
T6	id	page	value	function	name
T7	image	x	y	iphone	animation
T8	boost	detail	photo	rate	amp
T9	http	url	path	request	directories
T10	xml	qt	ds	conn	logger

Table 3: Top words for sample topics learned by the global topic model.

Using content features (length and text) produced significant improvement in performance (+9% over popularity). This validated our assumption that words used in the content of the question captures several aspects of quality. It is interesting to note that while the length of the question alone slightly outperform the random guess baseline, we observed something counter-intuitive: one might expect that a clearly presented question will tend to be longer than a question with poor quality, but in our dataset, good questions actually tended to be slightly shorter (Figure 3). If anything, conciseness appears to be a virtue.

Using features from global topic models alone (i.e., likelihood of each of the 10 topics) achieved a performance (64.2%) close to using unigrams alone (65.8%); and when combined with content features improved the performance to over 70% in accuracy. This confirms our hypothesis that latent topics associated with a question can be predictive of the question quality. Table 3 shows some sample topics (the corresponding top words) learned by this model, and they do

seem to capture the aboutness of questions.

Local (sentence-based) topic models alone did not outperform the global model, but it is interesting that when combined with content features, the resulting system yielded a significantly better performance of 71.8% (+1.3% over content+global model and +10.7% over popularity). This demonstrates that internal (hidden) structural aspects (local properties) of a question are indicative of its quality.

Adding the global topic structure model on top of this system did not yield additional improvements for our task. But it is interesting to note that the topic structures learned by this model did conform to the desired discourse-level constraints: contiguous sentences shared the same topics and the model learned sparser topic sequences compared to its local topic model equivalent where we extracted the most likely topic associated with each sentence in the question to produce a final topic sequence. We believe that while it did not directly improve the performance of the system, the model can help uncover hidden structural aspects (e.g., what do good questions share in common?) that would be interesting to study in the future. This information could be contrasted to or used in conjunction with other sources such as answers or comments to better understand the dynamics of community question-answering sites from a content viewpoint.

We also trained nonlinear models using the technique described earlier. Overall, nonlinear models outperform their linear counterparts but we notice that the improvements from nonlinear models are higher when using (sparse) content features alone (71.0%) in comparison to content+topic model (72.1%). But nonlinear models learned using the latter (content+topic model features) yield the best performance on this task and the improvement is statistically significant ($p < 0.0001$) over the nonlinear models learned for content features.

Discussions

We now discuss other aspects of question quality: repeated questions, user engagement patterns, and question hardness.

Repeated questions. One way to quantify whether a question has done proper research is to examine whether similar questions have already been asked on the site. For each question, we computed its cosine similarity to all previously posted questions, picked its 10 nearest neighbors, and computed the average similarity to these questions. The lower this number is, the less similar this question is to existing questions. We experimented with using only the content of the title (Figure 4) and using both body and title of the question[14] and observed similar trends: as shown in Figure 4, while the difference is not large, we observe a consistent "left-shift" for good questions. While this feature alone is not a strong indicator of question quality, our intuition that good questions do not tend to repeat what has been asked before is validated.

[14]For efficiency reasons, for each question, we only computed its similarity with questions whose title share at least one (non-stopword) token. We then take the average of title-based similarity and body-based similarity as the combined similarity measure.

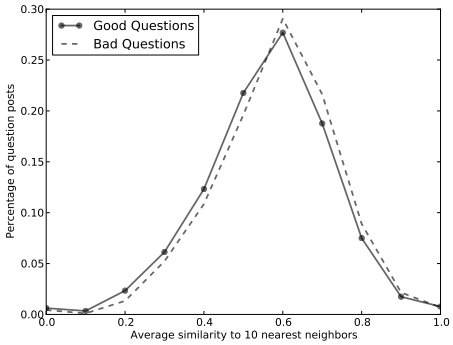

Figure 4: Distribution of average similarity to 10 nearest neighbors (among earlier questions) for good vs. bad questions. Good questions tend to be less similar to previous questions.

Answering patterns. Besides question content, question-answering sites sometimes offer additional sources of information that may not be available for newly posted question but evolve over time, for example answers and comments received in response to questions. It would be interesting to analyze if answer and comment content differ in any way depending on question quality. To study this, we performed a small experiment where we use content features from answers and comments (instead of question content) to predict question quality. We observe that using length features extracted from answers and comments yield an accuracy of 65.1% (much higher than the question length model = 55.5%) and adding text features yield some additional improvements (66.4% vs. 69.9% for question content model). This demonstrates that the answers/comments posted in response to high quality questions differ from those posted to low quality ones. Overall, good questions receive more answers (5 on average) compared to bad ones (2.5 on average) which shows that the question quality can have an effect on user engagement. Furthermore, answers to good questions tend to be longer (48.7 tokens per answer vs. 40.8 for bad questions). Lastly, we expected bad questions to receive more comments (asking for clarifications), but we observed the opposite: good questions receive 2.5 comments on average vs. 2.2 for bad ones.

Hardness of questions. Ideally, we would like to infer the *hardness* of a question and incorporate this as a feature for the prediction task. For instance, a very difficult question, especially when it is about a very specific concept (e.g., "libx264 encoder") may not be considered as useful by the community at large, but among people who are interested in this subject, it might be appreciated with up votes. On the other hand, a beginner-level question on a hot topic may be useful to more people, but it also has a higher chance of having been asked before. As a result it may not receive up votes if the user has not done proper research. It is possible that different topics can have different inherent hardness levels, so that global topic models may have partially captured this notion. We leave a proper integration of this signal as

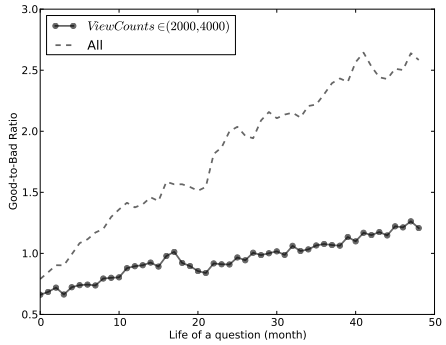

Figure 5: Temporal query distribution for good and bad questions. Good-to-bad ratio denotes the ratio between the number of queries leading to clicks on good questions vs. those leading to clicks on bad questions. Good questions tend to prosper as time goes on.

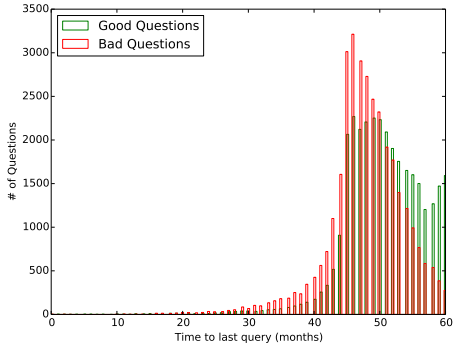

Figure 6: Life time histogram for good and bad questions.

interesting future work.

Quality vs. life of a question

Recall that only somewhat seasoned users with at least 15 points in reputation are capable of voting up. That is, a random user who stumbles upon this question page may contribute to its *ViewCount* but will not able able to affect its good-vs-bad label since she will not be able to vote. Does this mean our quantitative definition of question quality only reflects judgment of these seasoned StackOverflow users and potentially might not be meaningful to external users? In this section, we present a study that suggests otherwise. More specifically, we study the relation between the quality of a question and the temporal pattern of web queries that led to a click on it: do good questions tend to "live longer" in the sense that people consistently search for and click on such questions over time?

For each question in the StackOverflow dataset, we collected the set of web queries such that the corresponding StackOverflow question url was in the top three positions of the results for the query *and* the url was clicked. Each query was scrubbed for potential PII and is accompanied by a timestamp with no other meta-information. For privacy purposes, we only look at StackOverflow questions with queries from at least 50 distinct users. We extracted a total of 17 million web queries from a random sample of search log spanning 2008–2013.

We selected an equal number of good and bad questions posted in December 2008. Figure 5 shows how the number of queries leading to clicks on the two sets of questions change over time. The x-axis denotes time elapsed in months since a question was posted. The y-axis denotes the *good-to-bad ratio*, which is the ratio between the number of queries in each month leading to clicks on good questions vs. the number of queries leading to clicks on bad questions during the same month.

The curve labeled 'All' plots the ratio for all good and bad questions in this subset. The ratio is larger than 1.0 for all but the first few months and it is consistently increasing with time. This indicates that good questions receive more clicks on average than the bad ones and the gap widens across time. But good questions also tend to be more popular, which could have resulted from higher click traffic. To isolate this factor, we further restricted the set of good/bad questions to those with *ViewCount* between 2000 and 4000, and plotted the same ratio in Figure 5.

Recall that we picked questions with similar popularity. If we assume that web search click is proportional to *ViewCount*, then good and bad questions in this restricted set should receive a similar number of clicks in total, and the good-to-bad ratio cannot be consistently greater than one, so it is not interesting to look at the absolute value of this ratio. The important thing to note is that this ratio increases monotonically with time: good questions, even if they started out less popular, prosper as time goes on.

In other words, we argue that good questions have a longer life compared to bad questions of similar popularity. To give further evidence for this argument, we present in Figure 6 the histogram for the life times of good and bad questions. The x-axis denotes the life time of a question, which is defined as the time elapsed in months between the time when a question was posted to the time when it received its last query. The y-axis denotes the number of questions with a particular life time. (Recall that we use the same number of good and bad questions and hence the total area under the curves is the same for both.) The bad questions peak earlier, while good ones peak later, i.e., good questions are more likely to have a longer life time.

Conclusions

In this paper we addressed the problem of question quality. Based on a careful analysis of questions posted on a community question-answering site (StackOverflow), we developed a binary notion of question quality. We then proposed several models to predict question quality and showed that latent topical aspects shared between related questions are good predictors of question quality. Our best system achieved an accuracy of 72% using question content alone and outperformed strong baselines such the number of views by a significant amount. We also studied the dynamics of question quality from a different perspective: web search queries that

lead to clicks on question links. Our analysis showed that higher quality questions continue to garner interest over time in comparison to lower quality questions, thereby reinforcing the existence of such a quality notion.

References

Agichtein, E.; Castillo, C.; Donato, D.; Gionis, A.; and Mishne, G. 2008. Finding high-quality content in social media. In *WSDM*, 183–194.

Allamanis, M., and Sutton, C. 2013. Why, when, and what: analyzing stack overflow questions by topic, type, and code. In *MSR*, 53–56.

Anderson, A.; Huttenlocher, D. P.; Kleinberg, J. M.; and Leskovec, J. 2012. Discovering value from community activity on focused question answering sites: a case study of stack overflow. In *KDD*, 850–858.

Bian, J.; Liu, Y.; Zhou, D.; Agichtein, E.; and Zha, H. 2009. Learning to recognize reliable users and content in social media with coupled mutual reinforcement. In *WWW*, 51–60.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent Dirichlet allocation. *JMLR* 3:993–1022.

Brody, S., and Elhadad, N. 2010. An unsupervised aspect-sentiment model for online reviews. In *HLT/NAACL*, 804–812.

Chen, H.; Branavan, S.; Barzilay, R.; and Karger, D. R. 2009. Global models of document structure using latent permutations. In *HLT/NAACL*, 371–379.

Chiang, D.; Graehl, J.; Knight, K.; Pauls, A.; and Ravi, S. 2010. Bayesian inference for finite-state transducers. In *HLT/NAACL*, 447–455.

Daumé, III, H., and Marcu, D. 2006. Bayesian query-focused summarization. In *ACL*, 305–312.

Fligner, M., and Verducci, J. S. 1986. Distance based ranking models. *JRS(B)* 48(3):359–369.

Haghighi, A., and Klein, D. 2010. Coreference resolution in a modular, entity-centered model. In *HLT/NAACL*, 385–393.

Heilman, M., and Smith, N. A. 2010. Good question! Statistical ranking for question generation. In *HLT/NAACL*, 609–617.

Hoffman, M.; Blei, D. M.; and Bach, F. 2010. Online learning for latent Dirichlet allocation. In *NIPS*, 856–864.

Jeon, J.; Croft, W. B.; Lee, J. H.; and Park, S. 2006. A framework to predict the quality of answers with non-textual features. In *SIGIR*, 228–235.

Kozareva, Z., and Ravi, S. 2011. Unsupervised name ambiguity resolution using a generative model. In *Proc. 1st Workshop on Unsupervised Learning in NLP*, 105–112.

Langford, J.; Li, L.; and Strehl, A. 2007. Vowpal Wabbit, `http://hunch.net/~vw`.

Lebanon, G., and Lafferty, J. 2002. Cranking: Combining rankings using conditional probability models on permutations. In *ICML*, 363–370.

Li, B., and King, I. 2010. Routing questions to appropriate answerers in community question answering services. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, 1585–1588. New York, NY, USA: ACM.

Li, S., and Manandhar, S. 2011. Improving question recommendation by exploiting information need. In *ACL*, 1425–1434.

Li, B.; Jin, T.; Lyu, M. R.; King, I.; and Mak, B. 2012. Analyzing and predicting question quality in community question answering services. In *WWW Companion*, 775–782.

Li, B.; King, I.; and Lyu, M. R. 2011. Question routing in community question answering: Putting category in its place. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, 2041–2044. New York, NY, USA: ACM.

Liu, Y.; Bian, J.; and Agichtein, E. 2008. Predicting information seeker satisfaction in community question answering. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, 483–490. New York, NY, USA: ACM.

Meila, M.; Phadnis, K.; Patterson, A.; and Bilmes, J. 2007. Consensus ranking under the exponential model. In *UAI*, 285–294.

Navalpakkam, V.; Jentzsch, L.; Sayres, R.; Ravi, S.; Ahmed, A.; and Smola, A. J. 2013. Measurement and modeling of eye-mouse behavior in the presence of nonlinear page layouts. In *WWW*, 953–964.

Qu, M.; Qiu, G.; He, X.; Zhang, C.; Wu, H.; Bu, J.; and Chen, C. 2009. Probabilistic question recommendation for question answering communities. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, 1229–1230. New York, NY, USA: ACM.

Reisinger, J., and Paşca, M. 2009. Latent variable models of concept-attribute attachment. In *ACL/IJCNLP*, 620–628.

Riahi, F.; Zolaktaf, Z.; Shafiei, M.; and Milios, E. 2012. Finding expert users in community question answering. In *Proceedings of the 21st International Conference Companion on World Wide Web*, WWW '12 Companion, 791–798. New York, NY, USA: ACM.

Ritter, A.; Mausam; and Etzioni, O. 2010. A latent Dirichlet allocation method for selectional preferences. In *ACL*, 424–434.

Shah, C., and Pomerantz, J. 2010. Evaluating and predicting answer quality in community QA. In *SIGIR*, 411–418.

Smola, A. J., and Schölkopf, B. 2000. Sparse greedy matrix approximation for machine learning. In *ICML*, 911–918.

Sung, J.; Lee, J.-G.; and Lee, U. 2013. Booming up the long tails: Discovering potentially contributive users in community-based question answering services. In Kiciman, E.; Ellison, N. B.; Hogan, B.; Resnick, P.; and Soboroff, I., eds., *ICWSM*. The AAAI Press.

Szpektor, I.; Maarek, Y.; and Pelleg, D. 2013. When relevance is not enough: promoting diversity and freshness in personalized question recommendation. In *Proceedings of the 22nd international conference on World Wide Web*, 1249–1260. International World Wide Web Conferences Steering Committee.

Tian, Q.; Zhang, P.; and Li, B. 2013. Towards predicting the best answers in community-based question-answering services. In Kiciman, E.; Ellison, N. B.; Hogan, B.; Resnick, P.; and Soboroff, I., eds., *ICWSM*. The AAAI Press.

Titov, I., and McDonald, R. T. 2008. Modeling online reviews with multi-grain topic models. In *WWW*, 111–120.

Wu, H.; Wang, Y.; and Cheng, X. 2008. Incremental probabilistic latent semantic analysis for automatic question recommendation. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys '08, 99–106. New York, NY, USA: ACM.

Zhou, T. C.; Lyu, M. R.; and King, I. 2012. A classification-based approach to question routing in community question answering. In *Proceedings of the 21st International Conference Companion on World Wide Web*, WWW '12 Companion, 783–790. New York, NY, USA: ACM.