Predicting Web Searcher Satisfaction with Existing Community-based Answers

Qiaoling Liu[§], Eugene Agichtein[§], Gideon Dror[†],
Evgeniy Gabrilovich[‡], Yoelle Maarek[†], Dan Pelleg[†], Idan Szpektor[†],
[§]Emory University, Atlanta, GA, USA
[†]Yahoo! Research, Haifa, Israel
[‡]Yahoo! Research, Santa Clara, CA, USA
{qliu26, eugene} @mathcs.emory.edu, {gideondr, gabr, yoelle, dpelleg, idan}@yahoo-inc.com

ABSTRACT

Community-based Question Answering (CQA) sites, such as Yahoo! Answers, Baidu Knows, Naver, and Quora, have been rapidly growing in popularity. The resulting archives of posted answers to questions, in Yahoo! Answers alone, already exceed in size 1 billion, and are aggressively indexed by web search engines. In fact, a large number of search engine users benefit from these archives, by finding existing answers that address their own queries. This scenario poses new challenges and opportunities for both search engines and CQA sites. To this end, we formulate a new problem of predicting the satisfaction of web searchers with CQA answers. We analyze a large number of web searches that result in a visit to a popular CQA site, and identify unique characteristics of searcher satisfaction in this setting, namely, the effects of query clarity, query-to-question match, and answer quality. We then propose and evaluate several approaches to predicting searcher satisfaction that exploit these characteristics. To the best of our knowledge, this is the first attempt to predict and validate the usefulness of CQA archives for external searchers, rather than for the original askers. Our results suggest promising directions for improving and exploiting community question answering services in pursuit of satisfying even more Web search queries.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Search process

General Terms

Algorithms, Experimentation

Keywords

Searcher satisfaction, community question answering, query clarity, query-question match, answer quality

SIGIR '11 Beijing, China

1. INTRODUCTION

"Just because Google exists doesn't mean you should stop asking people things." – Alexia Tsotsis [35]

Community-based Question Answering (CQA) sites, such as Yahoo! Answers, Baidu Knows, and Naver Ji-Sik-In, as well as more social-oriented newcomers such as Vark and Quora, have gained substantial popularity over the recent years, effectively filling a niche left by the mainstream Web search engines. People around the globe resort to community help for a variety of reasons, from lack of proficiency in Web search to seeking an answer "with a human touch". Although some of these sites allow for monetary payments in exchange for answering questions (*e.g.*, JustAnswer, or the now discontinued Google Answers), answerers are usually attracted by social reward and less tangible incentives, such as reputation or points, as demonstrated by Raban [23]. The CQA communities are mainly volunteer-driven, and their openness and accessibility appeal to millions of users; for example, the size of Yahoo! Answers surpassed 1 billion answers in 2010[1], and Baidu Knows had over 100 million answered questions as of January 2011[2].

To date, prior studies of community question answering have mainly considered first-order effects, namely, the satisfaction of the original question asker by the posted answers. However, we believe CQA has significant *secondary* benefits, whereas previously answered questions are likely to be useful for future askers of substantially similar or related questions. Indeed, today many additional users already benefit from the public accessibility of CQA archives via all the major web search engines.[3] Existing answers often satisfy information needs of users who submit queries to a Web search engine, obtain results from a CQA site, such as the ones shown in Figure 1, select one of these results, and finally reach a resolved question page on the CQA site, as illustrated in Figure 2.

This scenario poses new challenges and opportunities for both search engines and CQA sites. For search engines, it provides a unique (semi-structured) source of human answers, which are particularly useful for satisfying tail queries [10]. For CQA sites, it creates substantial incoming traffic, which has its own multiple benefits from growing the

[1]http://yanswersblog.com/index.php/archives/2010/05/03/1-billion-answers-served/
[2]`http://zhidao.baidu.com/`, visited on January 19, 2011.
[3]Unfortunately, specific numbers describing the amount of incoming traffic to CQA sites from search engines are not publicly available.

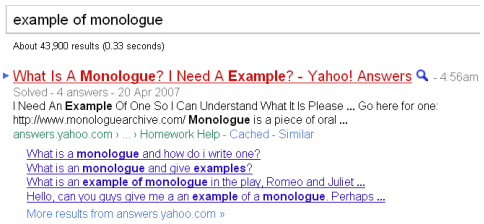


Figure 1: A subset of Google search results including resolved questions from Yahoo! Answers

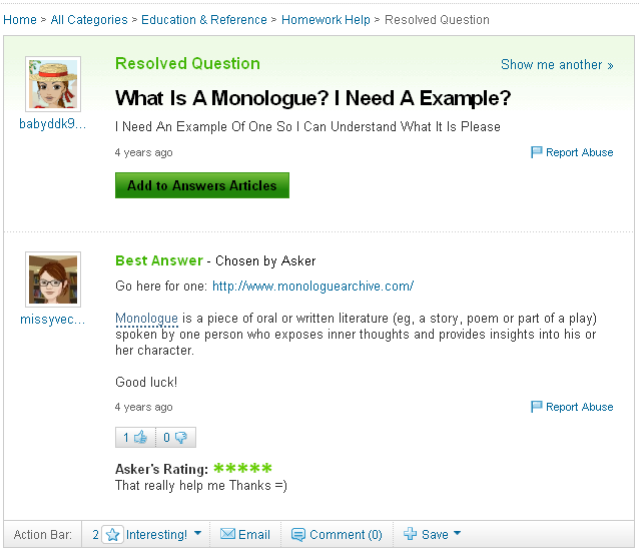


Figure 2: A resolved question on Yahoo! Answers

community to providing advertising revenue. More sophisticated symbiosis is also possible. For example, imagine if a search engine could detect that a user is struggling with a search session (e.g., by using techniques described in [7]). A search engine could then suggest posting a question on a CQA site, optionally suggesting relevant categories for such posting, and could even assist the user in transforming the query into an effective question. But in order to make this vision a reality, it is necessary to understand what it means for a searcher to be satisfied by an existing answer from a CQA archive, and to be able to predict this satisfaction.

This paper proposes and addresses this new problem of predicting the satisfaction of Web searchers with *existing* CQA answers. One way to approach this problem would be to define features of queries, questions, and answers (and possibly pairs thereof), and solve it within the machine learning paradigm. We call this a *direct* approach. This can be done by constructing a labeled dataset of queries and answers, tagged by human judges based on how well they believe a query is satisfied by a given answer.

Since queries are often quite short, and questions, which we view as an intermediate link between queries and answers, are often not much longer, another way to approach the problem is through exogenous knowledge. To this end, we identify three key characteristics of searcher satisfaction, namely, query clarity, query-question match, and answer quality. We then collect separate human labels for each, and build regression models for predicting these characteristics. Learning from these task-specific labels explicitly makes use of domain knowledge about the problem structure, which is not available in the above direct approach. We then use the output of these individual regressors as features in a subsequent regression task, which aims to predict searcher satisfaction. We call this method a *composite* approach. This approach also allows us to better understand how much the performance in the main prediction task can be improved by improving each of the individual regressors (we model this by replacing the intermediate regression predictions with actual human labels). This additional *interpretability* of the model provides further insights into the problem.

We conduct our experimental evaluation using data from one of the leading CQA sites, namely, Yahoo! Answers. We collect human labels for each of the above tasks using Amazon Mechanical Turk. These include labeling searcher satisfaction with a given answer (our main prediction task), as well as intermediate labels for query clarity and query-question match. We gauge the answer quality by using a combination of signals shown to be highly correlated with it [1], namely, the answer length, and the asker and community ratings. The labeled data is publicly available through Yahoo's Webscope program[4].

Since we define a new task (predicting searcher satisfaction with CQA answers), it is difficult to identify a suitable baseline. The closest previously addressed problem, at least in spirit, is *asker satisfaction* - predicting whether the original question asker (on the CQA site) was satisfied with the answers [19]. Intuitively, searcher satisfaction should be related to asker satisfaction. However, as we show later, asker satisfaction and searcher satisfaction appear to be very weakly related, if at all. Hence, an entirely new method is needed to compute *searcher satisfaction*. It is essential to note that, in our scenario, Web search queries are different from the questions originally posted to CQA sites, and more importantly, these queries are issued by different users with different information needs. In fact, the users in these two scenarios are drastically different. Whereas users of the community site are willing to clarify their questions, provide additional details, and even provide feedback to the answerers, Web search users seek immediate satisfaction, and essentially treat existing answers as a static resource.

The main contributions of this paper are threefold. First, we formulate a new problem, namely, predicting *searcher satisfaction* with existing CQA answers. Second, we propose two methods for solving this problem, a direct method and a composite method, which uses the outputs of secondary regressors as features. Finally, we apply our methods to a standard ranking task, where we treat answers as a semi-structured document collection. We show that incorporating our predictor of searcher satisfaction leads to a significant improvement in ordering the answers for a given query. To the best of our knowledge, this is the first attempt to predict and validate the usefulness of CQA archives for external searchers, rather than for the original askers. Our results suggest promising directions for improving and exploiting community question answering services in pursuit of satisfying even more Web search queries.

2. BACKGROUND

Our work spans the areas of filtering, recommending and ranking Community Question Answering (CQA) content,

[4]http://webscope.sandbox.yahoo.com/catalog.php?datatype=l, Dataset L16 - Yahoo! Answers Query to Question.

and estimating searcher satisfaction with the retrieved CQA content. In this section we first give an overview of the CQA site we chose for our experiments, namely Yahoo! Answers, as it currently holds the largest repository of questions. Then related research will be discussed.

2.1 Yahoo! Answers

With millions of active users, Yahoo! Answers hosts over a billion answers on a wide variety of topics. The system is question-centric: users are interacting by engaging in multiple activities around a specific question. The typical lifecycle of questions is the following: users post new questions and assign them to a predefined category, such as '*Education & Reference > Homework Help*' in the example shown in Figure 2. The new question remains "open" for four days with an option for extension, or can be closed if the asker chooses a best answer earlier. If the question remains unresolved, its status changes from "open" to "in-voting", where other users can vote for a best answer until a clear winner arises. In addition to asking and answering questions, users can also provide feedback by "starring" interesting questions, and rating answers with "thumbs up" or "thumbs down", and voting for best answer as mentioned above. Users can also receive activity updates on questions and follow the activity of other users. Finally, the community is self-moderated, and users can report and block questions and answers that violate the community guidelines (inappropriate language, spam etc.).

We consider here only resolved questions, e.g., questions that have been answered and for which a best answer has been chosen, since they are well indexed by Web Search engines and often have better quality than open or in-voting questions. In addition, we consider only best answers, based on the intuition that an external searcher reaching a Yahoo! Answers page will in most case ignore the other answers. This simplifying assumption is based on the fact that the best answer is prominently displayed below the question (see Figure 2) and users rarely browse results below the fold [22]. A resolved question will thus be represented by a pair *(question, best-answer)* and their associated additional signals such as stars, thumbs-up etc.

An additional type of information that we make use of is the *click data-set*: pairs of queries and the related CQA pages on which users, who issued the queries, clicked within the search results for these queries. In our experiments we utilized a click data-set that consists of pages from Yahoo! Answers that were clicked on within Google's search engine (see Section 4.1.1).

2.2 CQA Quality and Asker Satisfaction

One of the main problems in Yahoo! Answers, and in CQA sites in general, is the high variance in the perceived question and answer quality. Recently, this problem attracted a lot of research attention. Some studies attempted to assess the quality of answers or users [17, 1, 28, 32, 12], or questions [1, 29, 31], and filter or rank them accordingly [33]. Most recently, Horowitz and Kamvar [14] attempted to match questions to possible answerers, aiming at improving the quality of generated answers. Another related effort was estimating the archival value of questions for subsequent recommendation and retrieval [11].

Relatively few studies addressed the satisfaction of a user from the service provided by CQA sites. Most closely related to our work, Agichtein et al. [19] attempted to predict whether the asker of a question will be satisfied with the received answers. Our work goes beyond previous efforts as we propose and empirically evaluate techniques to estimate the satisfaction of *searchers*, as opposed to the original askers and answerers in CQA.

2.3 Web Search Quality and Satisfaction

Significant research has been done on estimating the quality of web search results, and the satisfaction of searchers. Query difficulty has been actively studied in the IR community, [6, 40]. A related problem of query ambiguity can also affect search quality and searcher satisfaction [34]. We adapt these techniques as a first step in predicting searcher satisfaction in CQA, since the latter clearly depends on query interpretation.

Searcher satisfaction in web search was addressed in [15, 8, 13], which utilized query log information for the task, such as relevance measures, as well as user behavior during the search session, including mouse clicks and time spent between user actions. What makes this task such a challenging problem is the large diversity in user goals [26], with a different definition of satisfaction for each, which requires developing unique satisfaction prediction models for the respective information needs. In our work we focus on satisfying types of queries that arguably are the most difficult for a web search engine to satisfy and often require other people to answer [21]. Specifically, we argue that some of these needs can often be satisfied with *existing* answers from CQA archives. Hence, we aim at harnessing the unique structure of such archives for detecting web searcher satisfaction, which is not captured by standard query and session logs.

2.4 Ranking and Recommendation in CQA

Searching CQA archives has been an active area of research, and several retrieval models specialized to CQA content have been proposed, [39, 4]. Such dedicated models are clearly deployed in practice, if only for specialized layout as demonstrated by Google specialized Yahoo! Answers snippets. Examples of other approaches include incorporating category information into retrieval [5], and exploiting the question-answer relationship [36]. While our main focus is on estimating the satisfaction with a *given* retrieved question-answer pair for a query, we adapt and extend these techniques for matching the query to the question and the answer content. Additionally, we show how our work could be applied for effective re-ranking of the retrieved question-answer pairs for a query, resulting in a significant improvement over a state-of-the-art baseline.

In the spirit of XML and semi-structured retrieval [2], it also makes sense to consider document structure in CQA, as has been done for Web Search [25], book retrieval [18] or sponsored search [3]. Thus in the case of Yahoo! Answers, we will consider the resolved question structure and distinguish between a question and its best answer (per our simplifying assumption, cf. Section 2.1), and their associated meta-data, while constructing features for our prediction tasks (see Section 3.2).

3. PREDICTING SEARCHER SATISFACTION

In this section we first introduce the task of predicting searcher satisfaction by a CQA page. Then, we propose approaches for representing and tackling this problem using regression algorithms.

3.1 Problem Description

We now define *searcher satisfaction by a CQA answer*:

Given a search query S, a question Q, and an answer A originally posted in response to Q on a CQA site,

predict whether A *satisfies* the query S.

For example, for a query "example of monologue", the best answer shown in Figure 2 is considered satisfactory because it clearly and comprehensively addresses the search intent.

Thus, instead of a Web search satisfaction task that examines *a (query,Web page)* pair, we consider a different tuple *(query,question,answer)*, where the *(question,answer)* pair has been extracted from the CQA page. The reason for using a more refined representation of *(question,answer)* rather than a full Web page (a Yahoo! Answers page in our case) is mostly for interpretability at a finer level. In practice, when experimenting with Yahoo! Answers in the remainder of this paper, we will use the simplification of considering only the best answer (cf. Section 2.1)[5] as A.

To solve our prediction problem, we propose to break it down into three sub-tasks: *query clarity*, *query-question match* and *answer quality*. More specifically:

• The *query clarity task*, which should not be confused with traditional query difficulty in IR, consists of estimating whether the query may be viewed, and understood, as a question. We hypothesize that if a query is not understandable or ambiguous, a CQA site is unlikely to have an existing satisfactory answer for this query.

• The *query-question match task* consists of estimating whether the question is driven by the same or by a similar enough information need as the query. This is a prerequisite for the answer to have a chance to address the query. Furthermore, since most search result snippets will only show the question title (such as shown in Figure 1), this match is a key driver for the searcher to select a specific CQA page: the question plays the role of a critical intermediary between the query and the answer.

• The *answer quality task* allows estimating the prior quality of the answer, with respect to the original question, and thus relates to the previously studied asker satisfaction task [19]. In our approach, answer quality characteristics are used not as the goal, but rather as additional input for our main task of predicting *searcher satisfaction*.

There are multiple advantages of breaking the main task into subtasks. First, we can better understand and analyze the problem structure, and devise more effective algorithms, as described next. Second, the resulting models become more *interpretable* and informative, by allowing us to analyze performance for each subtask. Finally, answer quality and related prior information (taking advantage of meta-information in particular) may be computed offline within the CQA site using methods such as described in [1].

The searcher satisfaction task seems to be better modeled as a graded task, since we found that it is easier for humans to judge satisfaction as a score within some range (see our human annotation in Section 4.1.2). Therefore, we treat the searcher satisfaction task as a regression problem. To this end, we need to define appropriate features for learning the regressor. We now describe the features used to represent the information in our task, and then formulate our direct and composite approaches.

[5]Note that the same model could be generalized to considering other, and not necessarily best, answers one at a time if the CQA site could isolate clicks or views of these answers. This is not the case with Yahoo! Answers where all answers are featured one after the other on the same page.

3.2 Features

By breaking down the main task into three subtasks, we distinguish between query clarity, query-question match, and answer quality features.

Since some of these subtasks were previously studied independently, such as [6, 34, 37] for query clarity, and [19, 1] for answer quality, we leveraged this prior work in the construction of our feature set. We now describe each of the feature groups, while the complete list of features is shown in Table 1.

• *Query clarity features* include query length, click statistics for the query, and query entropy computed based on the click dataset. We also compute a query clarity score based on a language model of the CQA collection, as well as an indicator whether the query starts with a "WH" or other question word.

• *Query-question match features* include match scores computed by popular retrieval models such as cosine similarity, TFIDF, BM25, and KL-divergence language model. For measuring these scores, we treat parts of a CQA page (the question title, question details and the best answer) as separate documents, and match each such part against the query. Additional features include measures of the overlap between the query and question, such as Jaccard coefficient and length ratios, and co-occurrence statistics between the query and the given question from the click data.

• *Answer quality features* are of two types. The first type of features deals with the quality of the answer, and is mainly based on the analysis given in [1]. The second type of features addresses the answer quality as predicting asker satisfaction, which directly maps to our third subtask. To this end, we mostly used the top performing features for predicting asker satisfaction as reported in [19].

Before using these features to build regression models, preprocessing was performed, as described in Section 4.2, to deal with missing values and to normalize the data.

3.3 Direct Approach: Logistic Regression

Our first approach to estimating searcher satisfaction, which we call the *direct approach*, consists of simply training a regressor over all the features defined for a given *(query, question, answer)* tuple. The rationale here is to rely on the power of discriminative learning to optimally use all available features to predict the final target.

While many regression algorithms could be employed for this task, our preliminary experiments with a wide range of models, including Linear Regression, Gaussian Processes, Ridge Regression and Random Forests, indicated Logistic Regression to be the most promising approach due to high variability and non-linear distribution of many of the input features. We now present our adaptation of the "classical" logistic regression algorithm to our problem.

Logistic regression uses the logistic function $f(t) = \exp(t)/(1 + \exp(t))$ to model an output variable restricted to the open set $(0, 1)$. This property makes the logistic function, properly scaled, a natural candidate for modeling our rating targets, all constrained to the range of $(1, 3)$ (see our human annotation in Section 4.1.2).

In what follows, we denote by $(x_i, y) \in R^n \times [1, 3], \quad i =$

Table 1: Features by subtask

Query clarity features (9 total)
• # of characters in the query. • # of words in the query. • # of clicks following the query. • # of users who issued the query. • # of questions clicked following the query. • Overall click entropy of the query [34]. • User click entropy of the query [37]. • Query clarity score computed based on the language model built with approximately 3 million questions (using title, details and best answer) posted in 2009-2010 [6]. • WH-type of the query (whether it starts with 'what', 'why', 'when', 'where', 'which', 'how', 'is', 'are', 'do').
Query-Question match features (23 total)
• Match scores between the query and the question title/details/best-answer using the cosine/TFIDF/KL-divergence/BM25 retrieval models. • The Jaccard/Dice/Tanimoto coefficient between the query and the question title. • Ratio between the number of characters/words in the query and that in the question title/details. • # of clicks on the question following this/any query. • # of users who clicked the question following this/any query.
Answer quality features (37 total)
• # of characters/words in the answer. • # of unique words in the answer. • Ratio between the number of characters/words of the question (including title and details) and the answer. • # of "thumbs up" minus "thumbs down", divided by the total number of "thumbs" received by the answerer. • # of "thumbs up" minus "thumbs down" received by the answerer. • # of "thumbs up/down" received by the answerer. • Match scores between the answer and the question title/details using cosine/TFIDF/KL-divergence/BM25 retrieval models. • Percentage of users who voted this answer as the best. • # of votes given by the voters for the answer. • Best answer ratio for the answerer. • Avg # of answers attracted by past questions of the asker. • # of answers received by the asker in the past. • Asker's rating of the best answer to her previous question. • Avg past rating by the asker. • Time passed since the asker registered in Yahoo! Answers. • # of previous questions resolved for the asker. • Avg asker rating for best answers in the category. • Avg voter rating for best answers in the category. • Time of day when the question was posted. • Avg # of answers per question in the category. • Time passed since the answerer with most positive votes registered in Yahoo! Answers. • Highest best answer ratio for any answerer of the question. • Avg best answer ratio for all answerers of the question. • Avg # of answers per hour in the category. • Whether the best answer is chosen by the asker. • Asker rating for choosing the answer as best answer.

$1, \ldots, l$, a generic example in the training set, where n is the number of features. We use $f_{w,b}(x_i) = 1 + 2\exp(w^T x_i + b)/(1+\exp(w^T x_i + b))$ as an estimate for the target variable y_i. The parameter vector of the model, w, and the scalar b are obtained by minimizing the squared loss

$$L(w,b) = \frac{1}{2}\sum_{i=1}^{l}\left(f_{w,b}(x_i) - y_i\right)^2 + \frac{\lambda}{2}\|w\|^2 \qquad (1)$$

The second term is introduced for regularization, where λ controls the strength of regularization.

Since there is no closed form solution for the parameters w and b that minimize Equation 1, we resort to Stochastic Gradient Descent [30], a fast and robust optimization method. It is an online algorithm where the parameters, initially random, are being updated using the gradient of the loss. In our case, the update is $b \leftarrow b + \Delta b$ and $w \leftarrow w + \Delta w$, where

$$\begin{aligned}\Delta b &= \eta\left(y_i - f_{w,b}(x_i)\right)\frac{\partial f_{w,b}}{\partial w}\\ \Delta w &= \eta\left(\left(y_i - f_{w,b}(x_i)\right)\frac{\partial f_{w,b}}{\partial w}x_i - \frac{\lambda}{l}w\right)\end{aligned}$$

We cycle through random permutations of the observations to achieve convergence. For the learning rate, we use a schedule of the form $\eta = \eta_t = \frac{\eta_0}{t+\tau}$ where $\tau > 0$, and t is the number of update steps taken thus far. The schedule satisfies the Robbins-Monro conditions [24], $\sum \eta_t = \infty$ and $\sum \eta_t^2 < \infty$, hence convergence is guaranteed. In light of the small number of examples in our dataset, we did not attempt to optimize the hyper-parameters of the model. Specifically, since the data is dense and the number of features is much smaller than the number of examples, we used weak regularization $\lambda = 0.01$ in all our experiments. We used moderate values for the learning rate schedule, $\eta_0 = 10$ and $\tau = 10$, and stopped iterating when the training set Mean Squared Error fell below a predefined threshold (0.0001).

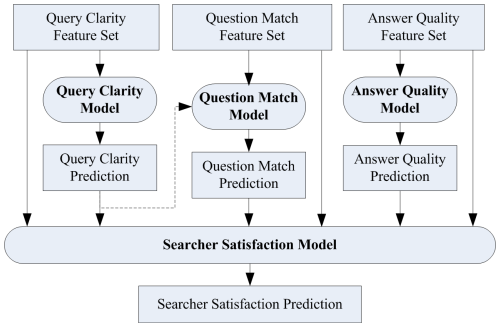


Figure 3: The *composite* approach.

3.4 Composite Approach

Our second approach, which we call the *composite approach*, first trains a separate logistic regression model for each of the three subtasks defined above, and then combines their results for main task (predicting searcher satisfaction). Figure 3 depicts the high-level workflow of this approach. In this approach, each regressor is trained using a subset of features relevant for each subtask. Considering that query clarity may affect the question match, we also added query clarity prediction as a feature to the query-question match regressor. Finally, the regression predictions for the three subtasks are provided as features for the final regressor to predict the overall searcher satisfaction.

This composite approach presents several advantages over the direct approach. First, it is more flexible in terms of feature selection. The individual regressors could be trained either on the same feature set, i.e., the large feature vector used in the direct approach, or on different feature sets selected by suitable feature selection methods for each subtask. More importantly, the composite approach can take advantage of the advances made by the research community in each of the sub-tasks, to improve the prediction of overall searcher satisfaction.

4. EXPERIMENTAL SETUP

This section first describes how we assembled a dataset from a sample of Google queries and a log of visits to Yahoo! Answers. We then describe the rating procedure to

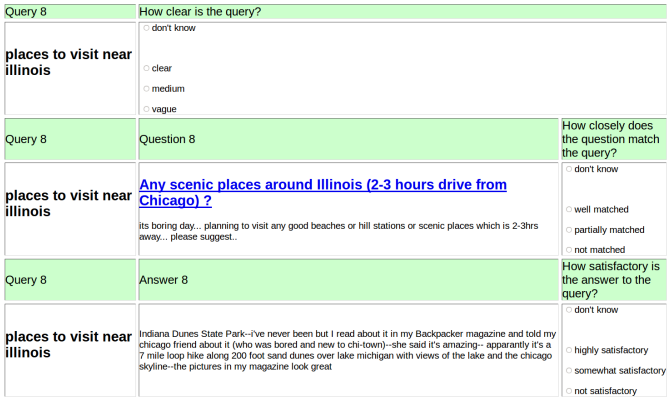


Figure 4: MTurk interface for human labeling.

acquire the "ground truth" for searcher satisfaction, and the characteristics of the resulting data.

4.1 Datasets

4.1.1 *Dataset Preparation*

To explore how searchers are satisfied with the existing answers in CQA sites, we used a large sample of queries issued to Google's search engine from Aug 24, 2010 to Aug 30, 2010 by users who selected as result (by clicking on it) a Yahoo! Answers link. This click dataset contains more than 37M clicks on 6M questions by 20M users following around 20M queries. By analyzing the distribution of this click data, we found that 86% of the queries are issued by only one user; therefore, most of the queries are tail queries.

Since it is hard for human to label searcher satisfaction for such a big dataset, we randomly sampled it to generate a smaller dataset consisting of 614 clicked questions following 457 queries issued by at least two users. These questions and corresponding answers may be biased to satisfy the searchers' information needs, as they are clicked from the search results. To correct this effect, we further issued a random sample of 118 queries to Google's search engine with site restriction to Yahoo! Answers and crawled the top 20 results (all question pages due to the site restriction). Only questions posted in 2009-2010 are kept based on the available associated meta-data. In total, our final dataset comprised of 1681 query-question pairs.

4.1.2 *Human Labeling*

Amazon Mechanical Turk (MTurk) was used to collect human judgments on how an answer satisfies a search query. To better understand the effects of query clarity and query-question match on searcher satisfaction with answers, we also asked the MTurk workers to label how clear the query is and how the question matches the query. Figure 4 shows the interface we used in MTurk. We used 3-scale rating method for all the rating tasks, {clear=1, medium=2, vague=3} for query clarity, {well matched=1, partially matched=2, not matched=3} for question match, and {highly satisfactory=1, somewhat satisfactory=2, not satisfactory=3} for searcher satisfaction. Each MTurk hit consists of 15 *(query, question, answer)* triples as shown in Figure 4, and each hit is labeled by 5-7 workers.

To validate the labels of MTurk workers, we also asked 6 researchers to label the query clarity for all the queries. Then we analyzed the agreement between the researchers and the MTurk workers. We first computed the average rating by researchers as well as by MTurk workers for each query, then used a threshold t to cast each average numerical rating nr into a binary rating br (if $nr <= t$ then br=clear, else br=not clear), and finally we computed the Fleiss's kappa coefficient[9] based on these binary ratings between the two sources. The highest kappa value 0.38 was achieved with a threshold of 1.3 (average agreement=0.70, average majority percentage=0.85). This analysis showed that the ratings from MTurk workers were reasonable.

For query-question match and searcher satisfaction, we only had ratings from the Mechanical Turk. So we used the same threshold strategy to cast each ordinal rating into a binary rating, then computed the Fleiss's kappa coefficient for each MTurk HIT, and finally computed the average kappa value. The highest kappa value (0.34) was achieved with a threshold of 2 for query-question match (average agreement=0.85, average majority percentage=0.91), and the highest kappa value (0.25) was achieved with a threshold of 2 for searcher satisfaction (average agreement=0.76, average majority percentage=0.84).

From the above agreement analysis, we can see that although the kappa coefficient among MTurk workers is not high, possibly due to the careless rating of some MTurk workers, the average rating by all the MTurk workers shows a moderate agreement with researchers. Therefore, we use the average rating by MTurk workers as our ground truth in order to evaluate the prediction of query clarity, query-question match, and searcher satisfaction with answers.

Figure 5 shows the distributions of the resulting ground truth set. The x axis represents the mean over the ratings by all MTurk workers, with 1 standing for the *highest* score, e.g., clear/well-match/highly satisfactory for respectively query clarity/query-question match/searcher satisfaction with answers, and with 3 standing for the *lowest* score for each. The y axis represents the frequency count of ratings in each bucket of x. We can see that all the distributions are skewed, especially the query clarity one due to the bias of the click data. Distribution for question match and searcher satisfaction are more balanced after we add the search engine results. To better understand the relations between the three variables, we computed the Pearson correlation between them and obtained the following result: the correlation between query clarity and searcher satisfaction is 0.1428, and the correlation between question match and searcher satisfaction is 0.6970.

4.2 Data Preprocessing

To make the data more amenable for modeling we used a three-stage preprocessing, performed on each feature:

1. null values have been replaced by the mean value. An example of a null value is the similarity between the query and the question body when the latter is empty;
2. features obtained by counting or summation, such as 'number of characters/words in the query' or 'number of "thumbs up/down" received by the answerer' were log transformed; specifically, we used $x = \log_2(1 + x_{raw})$ instead of the raw values x_{raw};
3. features were shifted and scaled to have a zero mean and a unit variance.

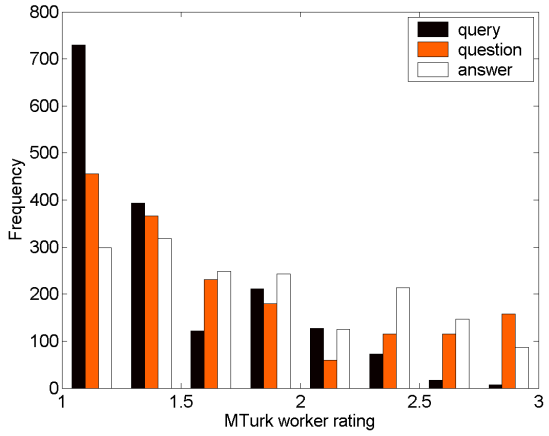


Figure 5: Distributions of the mean ratings of MTurk workers for query clarity, query-question match, and searcher satisfaction with answers.

4.3 Methods Compared

We consider four methods for estimating a searcher's satisfaction on Yahoo! Answers:

• **Google-derived baseline**: As described in Section 4.1.1, we crawled the top 20 results by submitting our queries to Google search, with site restriction to the Yahoo! Answers site. As a result, we obtained a ranked list of question pages for each query from the search engine. The rank of each question page indicates how well this page satisfies the query. Since a search engine ranks results by maximizing searcher's satisfaction with the overall result ordering, we use Google's ranking of question pages as our baseline.

• **Direct approach:** This method implements the logistic regression approach described in Section 3.3.

• **Composite approach:** This method implements the composite approach described in Section 3.4.

• **Composite upper-bound:** We also trained the composite approach with the intermediate predictions for the query clarity and query-question match subtasks replaced with their *human ratings*. Since human judgments are expected to be more reliable than the automatic predictions, this method serves as an upper bound for the possible performance of the fully-automatic composite approach.

4.4 Evaluation Metrics and Setup

Estimating searcher satisfaction: Our main prediction task estimates searcher satisfaction for a query with one given answer, independently of other query-answer pairs. Hence our main evaluation of the direct and composite approaches over all pairs is via root mean squared error (RMSE) and Pearson correlation between predictions and the human judgments. Both RMSE and Pearson correlation are standard performance estimators for regression problems.

When comparing our results to the Google-derived baseline described above, however, we could not use the above metrics, since Google does not divulge an independent score for each query-answer pair. Therefore, we propose to use two different metrics for ranking : (1) Kendall's tau (τ) correlation that has often been used in IR [27] to compare two ranked lists, and (2) the popular NDCG metric often used in IR evaluation [16]. As ground truth, we use the MTurk ratings described above to calculate these metrics as follows:

Kendall's τ: First for each search query s, we identify the set $Q(s)$ of all questions associated with s in our dataset. We then generate the following ranked lists.

1. The ground truth MTurk rating scores between each query s and each question q in $Q(s)$ are used to infer a ranked list L_M.

2. We identify the rank of each question q in $Q(s)$ in the ranked list of question pages featured in our previously described Google-derived baseline. The list of these ranks, that we refer to as L_G, gives us our baseline of ranking of questions by Google for a given query.

3. Similarly the direct and composite approach introduced in the previous section generate a score for each pair (s, q) and these scores induce two ranked lists that we note L_d for the direct approach and L_c for the composite approach.

Kendall's τ correlation is then computed between the ground truth list L_M and each of the system-generated lists L_G, L_d, and L_c. We process the answers $A(s)$ associated with s in the same way, and generate and evaluate the four respective lists in exactly the same manner.

NDCG metric: The ranked lists to compare are generated exactly in the same way as described above for Kendall's τ evaluation. Following Long et al. [20], we do not discretize the ground truth MTurk ratings, and use them directly (in place of relevance) to calculate the gain in the NDCG computation as follows:

$$DCG_p = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{log_2(1+i)}, nDCG_p = \frac{DCG_p}{IDCG_p}$$

where the relevance values are computed as $rel_i = (3 - r) \in [0, 2]$, where r is the average assessor rating of query-question match or searcher satisfaction, respectively.

Evaluation setup: For training and testing, we use stratified 10-fold cross-validation. This guarantees that the data distribution in each fold is similar to that of the entire dataset.

5. EMPIRICAL EVALUATION

We first present our results for the main task of predicting searcher satisfaction, and compare the direct and the composite approaches. Then, we analyze the performance of the proposed methods to identify key factors that affect the prediction accuracy. Finally, we present the results of applying our models to re-rank CQA answers in search engine results, showing significant improvements of our ranking over Google's.

5.1 Direct vs. Composite Comparison

Table 2 shows the results on predicting searcher satisfaction using our proposed direct and composite approaches. We report the mean ($\pm$ standard deviation) RMSE and Pearson correlation over the ten cross-validation folds.

In the first two rows of Table 2, we see that the composite approach performs better than the direct approach on both

Table 2: Regression results on searcher satisfaction.

Method	Correlation	RMSE
Direct	0.608±0.042	0.222±0.009
Composite	**0.618**±0.054	**0.217**±0.011
Composite upper-bound	0.773±0.029	0.178±0.010

Table 3: Regression results on individual sub-tasks.

Task	Correlation	RMSE
query clarity	0.713±0.028	0.151±0.005
question match	0.702±0.043	0.218±0.014
answer quality	0.213±0.057	0.478±0.015

correlation and RMSE[6] metrics. This difference is statistically significant according to the Wilcoxon two-sided signed ranks test at $p = 0.01$ [38]. This observation is quite intuitive, since the composite approach takes advantage of additional knowledge, which is learned from the human labels for the query clarity and query-question match sub-tasks.

Now consider the last row in Table 2, which reports the upper bound performance of the composite approach. To estimate the upper bound, we replace the individual regressors we trained for the query clarity and query-question match sub-tasks with the *actual* (average) human scores for those tasks, and plug these scores as features into the composite regressor. Evidently, the performance of the composite method can be improved dramatically if its components, namely, the query clarity and the query-question match predictors, are improved. We believe this flexibility of the composite approach constitutes a substantial advantage over the simpler direct approach.

5.2 Analysis and Discussion

Table 3 details the performance of the individual regressors that were combined in the composite approach. Here the query-question match regressor also uses the query clarity prediction as a feature, as explained in Section 3.4. The answer quality regressor is trained using the asker satisfaction ground truth [19] (An asker is considered satisfied iff he selected the best answer and gave at least 3 "stars" for the quality). Again, we report the mean (± standard deviation) RMSE and Pearson correlation over the ten folds.

By analyzing the predictions for searcher satisfaction by the composite regressor, we see that it can predict accurately both when the searcher is satisfied and not satisfied. We show a number of actual examples in Table 4. In the first example (E1), our method is able to correctly detect that the query is a little vague[7], the query-question match is low[8], and the answer is too simple to convince the searcher[9]. On the other hand, in the second example (E2), the query is quite clear and matches the question well, and the answer provides helpful advice to the searcher. Again, our method successfully predicts the overall searcher satisfaction with the answer, as well as individual sub-task scores (query clarity and query-question match).

To better understand the effectiveness of our methods, we also performed error analysis on the 30 cases where the difference between our prediction and the target was larger than 1. We found two cases, E3 and E4 (Table 4), for which our system predicted lower searcher satisfaction than the ground truth. In the other cases, our system predicted higher than actual satisfaction—average prediction of 1.66 versus average ground truth of 2.65. We believe the main reason for these large differences lies in the answer quality. In fact, more than half of the answers are not helpful at all (e.g., E5); other answers show negative opinions towards the askers, or contain ads. Thus, our error analysis confirms the importance of answer quality to searcher satisfaction, and also poses the challenge of more intelligent prediction of answer quality.

[6]Note that lower RMSE values reflect better performance.
[7]Higher values mean *lower* query clarity.
[8]Higher values reflect *poorer* query-question match.
[9]Higher values mean *lower* searcher satisfaction.

Table 5: Mean Kendall's τ and NDCG results on ranking questions and answers for queries.

	Query-question match		Searcher satisfaction	
	τ	NDCG	τ	NDCG
Google	0.359	0.939	0.307	0.905
Direct	–	–	0.434(+41%)	**0.928**(+2.5%)
Comp.	0.301	0.919	**0.437**(+42%)	**0.928**(+2.5%)

5.3 Answer Ranking for Queries

One possible application of predicting searcher satisfaction is using this prediction for ranking CQA pages in Web search. To this end, in Table 5 we compare the quality of ranking produced by our methods to that of Google's ranking of results retrieved from the Yahoo! Answers site. We compared the entire ranked lists of results returned by our methods and by Google to the ranking induced by human (MTurk) labels, therefore, we report different metrics than above, namely, NDCG and Kendall's τ. Our prediction of searcher satisfaction results in improvements over Google's on both metrics. All improvements are statistically significant according to the Wilcoxon double-sided signed ranks test at $p = 0.01$ [38]. Interestingly, Google's ranking of the *questions* (as opposed to answers) for a query is superior, which is to be expected due to additional information Google maybe used for ranking the questions (such as link structure and clicks)—whereas our work focuses on predicting searcher satisfaction with the *answers*, where indeed our methods perform better.

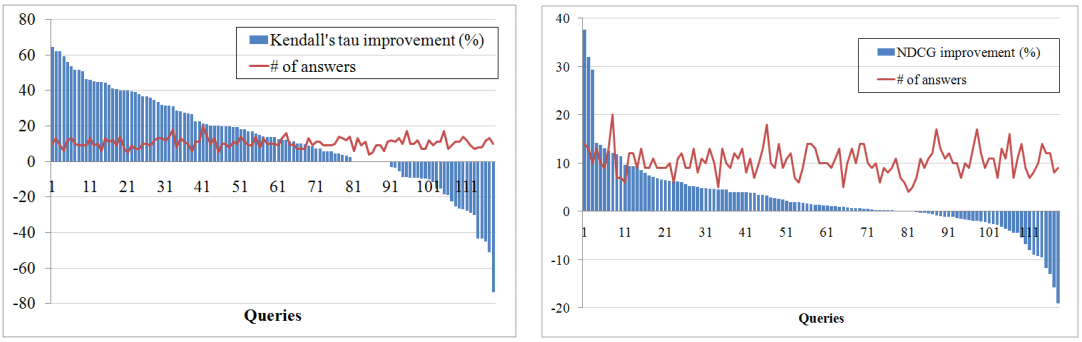
We further analyze these results by plotting the improvements over the Google baseline, for individual queries (Figure 6). Interestingly, it appears that the results do not depend on the number of answers to re-rank. In another experiment (omitted for lack of space), we found that the improvements are not correlated with query length. These results suggest that our methods are robust for a wide range of queries, and are likely to remain stable for other conditions. In summary, our results show that our satisfaction prediction allows our re-ranker to consistently outperform the state-of-the-art Google baseline, and could provide valuable input for other tasks, as we plan to explore in the future.

6. CONCLUSIONS

In this paper we formulated a novel task of predicting searcher satisfaction with answer pages from CQA sites. Prior research mainly concentrated on the first-order effects of community question answering, studying satisfaction of original askers of questions or potential answerers (both on the CQA site itself). In contrast, we study *second-order* effects of CQA archives, which repeatedly benefit many more

Table 4: Sample (query, question, answer) tuples, with predictions and ground truth labels. In the last three columns, lower values are *better* (higher clarity, better match, higher satisfaction).

No	Query	Question	Answer	Query clarity prediction (ground truth)	Query-question match prediction (ground truth)	Search satisfaction prediction (ground truth)
E1	mexican halloween	Is dressing up like a mexican on cinco de mayo or halloween degrading to mexicans?	Not really	1.74 (1.96)	2.04 (1.86)	2.70 (2.71)
E2	how to stop loving someone you shouldn't	How do you stop loving someone you're really not suppose to love?	...Keep your mind occupy with work or school. whatever u can to keep your mind busy...	1.09 (1.14)	1.20 (1.0)	1.16 (1.14)
E3	dtunes source	Ipod touch jailbreak dtunes and installous?	Installous is currently incompatible with the safari download plugin, which is required for dTunes to work...	1.94 (2.03)	2.11 (2.0)	2.01 (1.0)
E4	catsup vs ketchup	The condiment "KETCHUP" where did the name come from?	The most popular theory is that the word ketchup was derived from "koe-chiap" or "ke-tsiap" in the Amoy dialect of China...	1.67 (1.29)	1.85 (2.43)	2.17 (1.14)
E5	how much does it cost to send a letter to canada	How much does it cost to send a letter to canada?	Go to your local post office and ask them.	1.22 (1.26)	1.24 (1.4)	1.89 (3.0)



(a) Kendall's τ (b) NDCG

Figure 6: Kendall's τ (a) and NDCG (b) relative improvements of the composite approach over Google's baseline on ranking answers for queries.

Web users when these answers are included in Web search results for a variety of queries. We utilize the unique structure of CQA pages as well as all available community signals (e.g., ratings, thumbs-up) to improve the quality of matching between these pages and Web search queries.

We proposed to break the task of predicting searcher satisfaction into three sub-tasks, namely, predicting query clarity, query-question match, and answer quality. We then formulated two methods for solving the main prediction task. Our *direct* method simply uses all the available features in a single regression model. Our *composite* method first learns three separate regressors for each of the three sub-tasks, and then uses their predictions as features for solving the main task. Predictably, the performance of the composite method is statistically significantly superior to that of the direct method. This can be explained due to its use of additional exogenous knowledge, which is learned from the human labels for each of the sub-tasks while training the three individual regressors. Furthermore, the composite approach is more flexible, and it can immediately benefit as the predictions in individual sub-tasks are improved. Indeed, when we replace the predictions in each sub-task with *actual* human labels, the performance of the composite regressor is dramatically improved.

We believe that modeling the searcher satisfaction with CQA answers has multiple benefits. For example, if a search engine detects that a user is struggling with a search session, it could suggest posting a question on a CQA site, offering help with formulating a natural language question and choosing an appropriate category. On the search engine side, an accurate predictor of searcher satisfaction can be used for improved ranking of CQA results in Web search. Indeed, our results show that this can be achieved. To this end, we compared the quality of ranking of CQA answers produced by our methods with, and demonstrated significant improvements over the quality of the ranking provided by Google's search engine (both compared to the "ideal" ranking generated from the ground truth labels provided by humans).

In our future work, we plan to further investigate the types of queries that are likely to be satisfied by CQA pages. We also plan to improve query-question matching using (monolingual) machine translation models. Another brunch of potential work is to develop semi-supervised or unsupervised methods to predict searcher satisfaction, as large numbers of human labels are hard to obtain. Finally, we intend to study searcher satisfaction with other types of community-generated Web pages that possess interesting structure, such as Facebook pages (subject to appropriate privacy policies).

7. ACKNOWLEDGMENTS

This work was supported by the National Science Foundation grant IIS-1018321 and by the Yahoo! Faculty Research and Engagement Program.

8. REFERENCES

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

[2] S. Amer-Yahia and M. Lalmas. XML search: languages, INEX and scoring. *SIGMOD Rec.*, 35:16–23, December 2006.

[3] M. Bendersky, E. Gabrilovich, V. Josifovski, and D. Metzler. The anatomy of an ad: Structured indexing and retrieval for sponsored search. In *WWW'10*, April 2010.

[4] J. Bian, Y. Liu, E. Agichtein, and H. Zha. Finding the right facts in the crowd: factoid question answering over social media. In *WWW '08*, 2008.

[5] X. Cao, G. Cong, B. Cui, C. S. Jensen, and C. Zhang. The use of categorization information in language models for question retrieval. In *CIKM*, 2009.

[6] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. Predicting query performance. In *SIGIR*, 2002.

[7] D. Donato, F. Bonchi, T. Chi, and Y. Maarek. Do you want to take notes?: identifying research missions in Yahoo! Search Pad. In *WWW '10*, 2010.

[8] H. Feild, J. Allan, and R. Jones. Predicting searcher frustration. In *Proc. of SIGIR*, pages 34–41, 2010.

[9] J. L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378 – 382, 1971.

[10] S. Goel, A. Broder, E. Gabrilovich, and B. Pang. Anatomy of the long tail: Ordinary people with extraordinary tastes. In *WSDM*, 2010.

[11] F. Harper, D. Moy, and J. Konstan. Facts or friends?: distinguishing informational and conversational questions in social Q&A sites. In *CHI*, 2009.

[12] F. M. Harper, D. Raban, S. Rafaeli, and J. A. Konstan. Predictors of answer quality in online q&a sites. In *CHI*, pages 865–874, 2008.

[13] A. Hassan, R. Jones, and K. Klinkner. Beyond DCG: User behavior as a predictor of a successful search. In *Proc. of WSDM*, pages 221–230, 2010.

[14] D. Horowitz and S. Kamvar. The anatomy of a large-scale social search engine. In *WWW*, 2010.

[15] S. B. Huffman and M. Hochster. How well does result relevance predict session satisfaction? In *SIGIR*, pages 567–574, 2007.

[16] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20:422–446, October 2002.

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

[18] G. Kazai and A. Doucet. Overview of the INEX 2007 book search track: Booksearch '07. *SIGIR Forum*, 42(1):2–15, 2008.

[19] Y. Liu, J. Bian, and E. Agichtein. Predicting information seeker satisfaction in community question answering. In *SIGIR*, 2008.

[20] B. Long, O. Chapelle, Y. Zhang, Y. Chang, Z. Zheng, and B. L. Tseng. Active learning for ranking through expected loss optimization. In *SIGIR*, pages 267–274, 2010.

[21] M. Morris, J. Teevan, and K. Panovich. A Comparison of Information Seeking Using Search Engines and Social Networks. In *ICWSM*, 2010.

[22] J. Nielsen. User interface directions for the web. *Commun. ACM*, 42:65–72, January 1999.

[23] D. Raban. Self-presentation and the value of information in Q&A web sites. *JASIST*, 60(12):2465–2473, 2009.

[24] H. Robbins and S. Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951.

[25] S. Robertson, H. Zaragoza, and M. Taylor. Simple BM25 extension to multiple weighted fields. In *Proceedings of CIKM*, pages 42–49, 2004.

[26] D. E. Rose and D. Levinson. Understanding user goals in web search. In *WWW*, pages 13–19, 2004.

[27] M. Sanderson and I. Soboroff. Problems with kendall's tau. In *SIGIR*, 2007.

[28] C. Shah and J. Pomerantz. Evaluating and predicting answer quality in community QA. In *SIGIR*, 2010.

[29] Y.-I. Song, C.-Y. Lin, Y. Cao, and H.-C. Rim. Question utility: A novel static ranking of question search. In *AAAI*, pages 1231–1236, 2008.

[30] J. C. Spall. *Introduction to Stochastic Search and Optimization*. John Wiley & Sons, 2003.

[31] K. Sun, Y. Cao, X. Song, Y.-I. Song, X. Wang, and C.-Y. Lin. Learning to recommend questions based on user ratings. In *CIKM*, pages 751–758, 2009.

[32] M. Surdeanu, M. Ciaramita, and H. Zaragoza. Learning to rank answers on large online qa collections. In *ACL*, pages 719–727, 2008.

[33] M. A. Suryanto, E. P. Lim, A. Sun, and R. H. L. Chiang. Quality-aware collaborative question answering: methods and evaluation. In *WSDM*, 2009.

[34] J. Teevan, S. T. Dumais, and D. J. Liebling. To personalize or not to personalize: modeling queries with variation in user intent. In *SIGIR*, 2008.

[35] A. Tsotsis. Just because Google exists doesn't mean you should stop asking people things. TechCrunch, Oct 2010. http://techcrunch.com/2010/10/23/google-vs-humans/.

[36] X. Wang, X. Tu, D. Feng, and L. Zhang. Ranking community answers by modeling question-answer relationships via analogical reasoning. In *SIGIR*, 2009.

[37] Y. Wang and E. Agichtein. Query ambiguity revisited: clickthrough measures for distinguishing informational and ambiguous queries. In *ACL*, 2010.

[38] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1:80–83, 1945.

[39] X. Xue, J. Jeon, and W. Croft. Retrieval models for question and answer archives. In *SIGIR*, 2008.

[40] E. Yom-Tov, S. Fine, D. Carmel, and A. Darlow. Learning to estimate query difficulty: including applications to missing content detection and distributed information retrieval. In *SIGIR*, 2005.