Learning from the Past: Answering New Questions with Past Answers

Anna Shtok
Faculty of Industrial Engineering and Management
Technion, Israel Institute of Technology
Haifa 32000, Israel
annabel@techunix.technion.ac.il

Gideon Dror, Yoelle Maarek, Idan Szpektor
Yahoo! Research
MATAM, Haifa 31905, Israel
gideondr,idan@yahoo-inc.com, yoelle@ymail.com

ABSTRACT

Community-based Question Answering sites, such as Yahoo! Answers or Baidu Zhidao, allow users to get answers to complex, detailed and personal questions from other users. However, since answering a question depends on the ability and willingness of users to address the asker's needs, a significant fraction of the questions remain unanswered. We measured that in Yahoo! Answers, this fraction represents 15% of all incoming English questions. At the same time, we discovered that around 25% of questions in certain categories are recurrent, at least at the question-title level, over a period of one year.

We attempt to reduce the rate of unanswered questions in Yahoo! Answers by reusing the large repository of past resolved questions, openly available on the site. More specifically, we estimate the probability whether certain new questions can be satisfactorily answered by a best answer from the past, using a statistical model specifically trained for this task. We leverage concepts and methods from query-performance prediction and natural language processing in order to extract a wide range of features for our model. The key challenge here is to achieve a level of quality similar to the one provided by the best human answerers.

We evaluated our algorithm on offline data extracted from Yahoo! Answers, but more interestingly, also on online data by using three "live" answering robots that automatically provide past answers to new questions when a certain degree of confidence is reached. We report the success rate of these robots in three active Yahoo! Answers categories in terms of both accuracy, coverage and askers' satisfaction. This work presents a first attempt, to the best of our knowledge, of automatic question answering to questions of social nature, by reusing past answers of high quality.

Categories and Subject Descriptors

H.3.4 [**Systems and Software**]: *Question-answering systems*

Keywords

community-based question answering, automatic question answering

WWW 2012, April 16–20, 2012, Lyon, France.
ACM 978-1-4503-1229-5/12/04.

1. INTRODUCTION

In a number of situations, users prefer asking questions to other users on Community Question Answering (CQA) sites such as Yahoo! Answers, Baidu Zhidao, Naver Ji–Sik-in, or more recent ones such as StackOverflow or Quora, rather than issuing a query to a Web search engine. These situations arise for instance when users struggle with expressing their needs as a short query. Such needs are typically personal, heterogeneous, extremely specific, open-ended, etc. Indeed, question titles on Yahoo! Answers count on average between 9 and 10 words, and this without counting the additional "details" field, to which many paragraphs can be added, see Figure 1 as a common example of verbosity. In other cases, users assume that no single Web page will directly answer their possibly complex and heterogeneous needs, or maybe they perceive that real humans should understand and answer better than a machine [31].

While the social angle of interacting with humans is clearly important in CQA, as demonstrated by the recent success of Quora, a significant portion of questions is driven by a real need for which the user expects a single relevant answer [23]. Yet in spite of active participation in CQA sites, a significant portion of questions remain unanswered, a phenomenon we refer to as *question starvation*) [21]. In an analysis we conducted on Yahoo! Answers, one of the first CQA sites on the Web, we discovered that about 15% of the questions do not receive any answer and leave the asker unsatisfied. One approach to reduce the amount of unanswered questions is to pro-actively push open questions to the most relevant potential answerers [21, 17, 11]. Another approach is to attempt to automatically generate answers from external knowledge resources such as Wikipedia or the Web [23]. In this paper, we investigate a third approach, which is to answer new questions by reusing past resolved questions within the CQA site itself. This latter approach relies on the intuition that even if personal and narrow, some questions are recurrent enough to allow for at least a few new questions to be answered by past material.

To study the potential relevance of past answers to new questions, we conducted an analysis on three active categories of Yahoo! Answers, namely *Beauty & Style*, *Health* and *Pets*, and observed that they expose a relatively high percentage of recurring questions. More specifically, we considered all the questions asked in these categories during 3 months in 2011, and checked whether they had a "matching" past question (indicated by a cosine similarity of above 0.9 between question titles) within a repository of questions

asked during 11 consecutive months in 2010. It turned out that around 25% of these questions did have such a match on this limited corpus only, and this figure should be significantly higher over the full repository of questions over the years. Note however that the percentage of recurring questions should only be considered as an indicator of the existence of potentially reusable answers rather than as a target coverage. Indeed some similar questions will reflect a perfect similar intent, *e.g.* "`My eyeliner keeps smudging please help?`" and "`Why does my eyeliner keep smudging?`". Yet, other questions, such as "`what's wrong with me?`", which matches the title of more than 4,000 resolved questions in Yahoo! Answers, require better matching techniques to find a correct answer. Such superficially similar questions drastically differ in intent as evidenced by the "details" part of the question. Another source of possible mismatch is time sensitivity, as perfectly similar questions about sports results for instance, might have drastically different answers over time.

Even if we consider the full details of questions, we cannot limit ourselves to pure content similarity. Consider for example the two questions, "`How many pounds can you lose in 9 weeks?`" and "`Can I lose 9 pounds a week?`". A superficial analysis would indicate a high-level of content similarity, while their intent clearly differ. On the other hand, we cannot demand perfect question similarity between pairs of questions, as it would drastically diminish coverage. Indeed question formulation for similar needs can vary a lot among users, if only because of questions being significantly longer than queries as mentioned earlier. To address this phenomenon, we propose here to use the following two-stage approach. Given a fresh new question, we find the most similar (content-wise), yet not necessarily identical, past question. Then, in a second stage, we apply a classifier that estimates intent similarity and decides whether or not to serve the answer of this past question as a new answer. In order to ensure precision, the past answer will be utilized only when we are sufficiently confident with both content and intent similarities.

We embodied the above approach in a running system and selected Yahoo! Answers as our test site for the following two reasons. First, it currently holds the largest repository of answers on the Web with more than a billion posted answers[1] and thus has more chances to offer reusable past material. Second, its content is easily accessible to all, either by crawling or through its open API and RSS feed[2].

In order to measure the effectiveness of our system, we use a common offline evaluation method, using a dataset of several thousand *(new question, past answer)* pairs, annotated[3] by Mechanical Turkers[4]. More interestingly, we use an original online method, which relies on the most accurate evaluators possible, namely the real askers of new questions. Indeed, we defined three new users in Yahoo! Answers, nicknamed Alice, Jane and Lilly, who are in practice operated by our system[5]. We returned their answers to real askers,

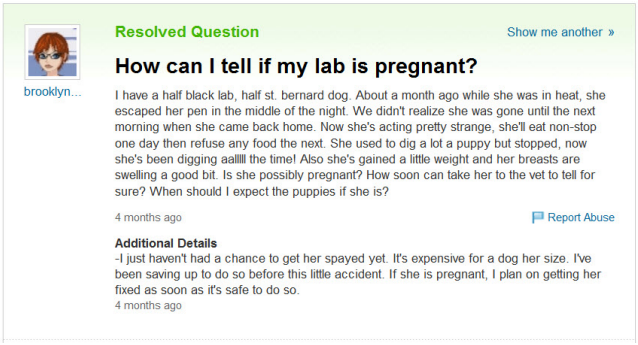

Figure 1: A typical question in Yahoo! Answers with specific details

making sure to remain ethical by pointing back to the original past question and answer and observed askers' reactions. One key challenge for such a system is to prioritize precision to avoid embarrassing cases, while not to the extreme as too few past answers would then be reused. This approach allowed us to verify whether we achieved our goal of high precision for the maximal level of coverage that is allowed for users in Yahoo! Answers.

2. RELATED WORK

Yahoo! Answers is a question-centric CQA site, as opposed to more social-centric sites such as Quora. Askers post new questions and assign them to categories selected from a predefined taxonomy, such as *Pets* > *Dogs* in the example shown in Figure 1. A question consists of a *title*, a short summary of the question (in bold at the top in Figure 1), and a *body*, containing a detailed description of the question and even additional details (See the paragraphs below the question title in Figure 1). The posted question can be answered by any signed-in user. It remains "open" for four days, or for less if the asker chose a best answer within this period. If no best answer is chosen by the asker, the task is delegated to the community, which votes for the best answer until a clear winner arises. Only then is the question considered "resolved." In case a question is not answered while "open" it is "deleted" from the site. Registered users may answer a limited number of questions each day, depending on their *level*. For example, first level users may answer 20 questions per day, and 60 when attaining third level. It takes 120 days for an average user to attain third level, but the activity of a user as well as the quality of his answers may substantially affect this number. A major problem in Yahoo! Answers, as well as other CQA sites, is question starvation [21]. Questions may be left unanswered for a number of reasons: unavailability of potential answerers while it is "open", the question being poorly formulated/uninteresting, or the sheer volume of incoming questions, which make it easy for potential answerers to miss questions they might have been interested in. One approach to tackling the latter case is to push open questions to relevant answerers [21, 17, 11]. This is quite a valuable approach that facilitates the task of answerers. Still, potential answerers may not always be available, or simply not

[1] `http://yanswersblog.com/index.php/archives/2010/05/03/1-billion-answers-served/`
[2] `http://answers.yahoo.com/rss/allq`
[3] We intend to make our two annotated datasets available to the research community in the next few months.
[4] `mturk.com`
[5] Alice's, Jane's and Lilly's actual activities can viewed at: `http://answers.yahoo.com/activity?show=SZRbSI3Uaa` `http://answers.yahoo.com/activity?show=hFEx4VF7aa` `http://answers.yahoo.com/activity?show=OzaaPpm8aa`

Figure 2: A detailed (truncated) answer for an advice seeking question

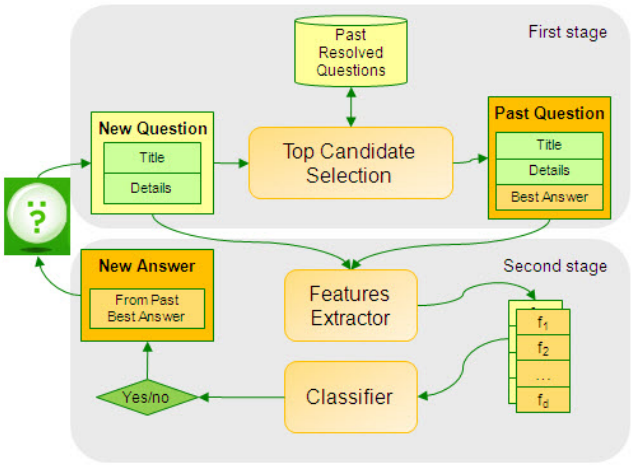

Figure 3: The Two-stage question-answering algorithm for CQA.

in the "mood" to answer a recommended question. A complementary approach is to automatically generate answers. Automatic question answering has been an active research field for several decades [32, 33, 28]. One common method used in this approach consists of first retrieving text passages that may contain the answer to the target question, then extracting candidate answers from the retrieved passages and rank them [22, 30, 8, 25, 24]. This is the preferred framework for factual unambiguous questions, such as "`Why is the sun bright?`", or "`Where is the Taj Mahal?`", for which a unique correct answer is expected. Yet it is not applicable to the often personal, narrow, ambiguous, open-ended or advice-seeking questions, such as "`Should I get lovebirds or cockateils?`", that often appear on CQA sites. Fewer efforts focus on the latter types of questions, [1, 27]. For such questions several answers may be valid and the answers may be quite complex. Figure 2 depicts such a detailed answer to the previous question.

A different type of effort proposes to identify within a collection of answers the most relevant ones to a given question. Bernhard and Gurevych [3] use translation models to this effect, and evaluate their approach on a small set of factual questions, for which answers are known ahead of time. Surdeanu et al. [29], combine translation and similarity features in order to rank answers by relevance to a given question, but focus only on *how to* questions. Both efforts share the same goal of identifying the existing most relevant answer from a collection and thus assume that such an answer does exist. In the same spirit, they need to know ahead of time what these answers are for a given number of questions in order to evaluate their algorithms. Neither these techniques nor their evaluation methodology are applicable in our case. Indeed our collection is too large and the questions too numerous and too various to know whether a valid answer even exists within the collection of past answers. In a similar vein, Bian et al. [4] attempt to rank past CQA question-answer pairs in response to factual questions. They utilize a supervised learning-to-rank algorithm to promote relevant past answers to the input question based on textual properties of the question and the answer, as well as indicators for the answerer's quality. Unlike Bian et al., we aim at detecting if a relevant answer even exists, and our scope is all questions posted to Yahoo! Answers, not just factual questions.

A related but slightly different approach proposes to find past questions that are similar to the target question, based on the hypothesis that answers to similar questions should be relevant to the target question. Carmel et al. [7] rank past questions using both inter-question and question-answer similarity, with response to a newly posed question. Jeon et al. [18, 19] demonstrate that similar answers are a good indicator of similar questions. Once pairs of similar questions are collected based on their similar answers, they are used to learn a translation model between question titles to overcome the lexical chasm when retrieving similar questions. Xue et al. [35] combine a translation model for question similarity and a language model for answer similarity as part of the retrieval model for similar questions. Duan et al. [12] retrieve questions with similar topic and focus on those that pertain to the target question. Wang et al. [34] identify similar questions by assessing the similarity between their syntactic parse trees. Our work belongs to the above school that seems the most promising given the huge repository of more than a billion answers in Yahoo! Answers today. Unlike the previous efforts in this school however, we try not only to retrieve semantically similar questions but further to find those that share a common need.

3. A TWO STAGE APPROACH

In this work, we investigate the validity of reusing past answers for addressing new questions in Yahoo! Answers. We propose to follow the example of factual question answering, and adopt its two-stage approach, which first ranks candidate passages and then extracts plausible answers from the best passages. Yet our approach, depicted in Figure 3, differs from factual question answering. Instead of ranking answer passages, in the first stage (upper part of the diagram) past questions similar to the new question are identified and ranked so as to produce one single resolved question candidate. As mentioned before, we follow Jeon et al.'s hypothesis [19, 35], which noted that similar questions should have similar answers. We therefore exploit the new-question/past-question similarity (which is less prone to lexical gaps than question/answer similarity) in order to identify this top candidate question. In the second stage (bottom part of the diagram), instead of extracting answers from passages, our

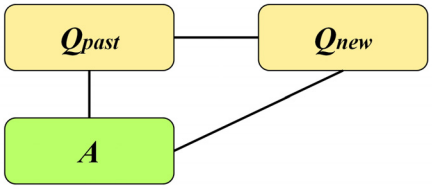

Figure 4: The three input components from which features are extracted: the new question Q_{new}, the past question Q_{past} and its best answer A.

algorithm assesses the best answer to the top candidate question as a plausible answer to the new question. The candidate answer is evaluated in order to verify whether it meets the underlying need of the new question. It is served to the asker only upon qualification. We next describe these two stages in detail. Note that while stage two considers only the highest ranking candidate, our approach could be adapted to assess any top N candidates and choose the best answer within them. We reserve this direction for future work.

3.1 Stage One: Top Candidate Selection

Since our purpose is to eventually serve a single answer that will satisfy the needs of the asker, we follow the definition of a *satisfying answer* given in [2], and limit our search space to past questions with answers that were marked as *best answer* by the askers, and were rated with at least three stars. Several works present effective similar question retrieval methods for CQA archives [34, 12, 35, 19]. However, these are mostly recall oriented, and relevance does not necessary imply a satisfying single answer. Hence, we adopt the more conservative traditional cosine similarity measure for ranking. The preference for short documents, usually treated as a weakness of cosine similarity, is actually of merit while searching for a candidate answer. Our observation is that shorter questions tend to be less detailed and complex and that their associated best answer tends to also satisfy similar questions.

As mentioned earlier, a question in Yahoo! Answers has two parts: a short title and a variably long body that contains the question description. Jeon et al. [19] demonstrated that using the question title for retrieval of similar questions is of highest effectiveness, while using the question body resulted in lower MAP. Following this finding, we rank similar questions for a target new question in two steps. In the first step, the similarity between the titles of the new question and the past questions is measured, keeping only past questions whose similarity score is above a threshold α. The remaining questions are then ranked by the similarity of both their title and body to those of the new question. Both 'title-only' and 'title+body' question representations are vector-space unigram models with TF-IDF weights.

Note that already in stage one, we may decide to leave the new question unanswered if no sufficiently similar question is found. If a top candidate is identified however, its best answer is considered in stage two.

3.2 Stage Two: Top Candidate Validation

In this stage, the candidate answer selected in stage one is assessed as a valid answer to the new question. To this effect, we consider the following *input triplet*: (Q_{new},Q_{past},A), where Q_{new} represents the new question, Q_{past} the top candidate past question and A the best answer for Q_{past}, as illustrated in Figure 4. We train a classifier that validates whether A can be served as an answer to Q_{new}. In order to feed the triplet to the classifier, we represent it as a feature set. We divide the features that are derived from the input into two types: features that quantify the "quality" of each entity and features that capture different aspects of similarity or relatedness between any two entities. The first type of features is applied to the elements of the triplet, represented as nodes in Figure 4, while features of the second type are applied to the edges.

Overall, we extracted 95 features using various lexical, natural language and query performance prediction considerations, in order to represent this triplet. We next detail these features, and then describe the classification model we selected for the task at hand.

3.2.1 Features

Surface-level Features.

Surface Level Statistics: We extract the following lexical level statistics from any given text: text length, number of question marks, stopword count, maximal IDF within all terms in the text, minimal IDF, average IDF, IDF standard deviation, http link count, number of figures. These features try to identify the focus, complexity and informativeness of the text. Various IDF statistics over query terms have been found to be correlated to query difficulty in ad-hoc retrieval [16, 15]. For example, low maximal IDF indicates a general, non informative question and thus we expect it would be harder to find a question with the same intent and provide a correct answer. Other features, such as the number of question marks, help identifying complex questions that cover multiple needs, while a lack of stop words indicates a low question readability. The features in this family are applied to the title and the body of Q_{new} and Q_{past} separately, and to A. Finally, we also measure the word-length ratio between Q_{new} and Q_{past}.

Surface Level Similarity: These features measure how similar the entities are in terms of lexical overlap. They include the cosine similarity between the TF-IDF weighted word unigram vector space models of any two entities, in the same spirit as the cosine similarity measure used in Section 3.1. We generate similarity scores between the titles of the two questions, the bodies of the two questions, and their entire title+body texts. Similarities between the entire text of each question and the answer are also calculated. Finally, we also measure the difference between the similarity score of (Q_{new},A) and of (Q_{past},A).

Linguistic Analysis.

Latent Topics: For each category in Yahoo! Answers we learn LDA topics [5] from the corpus of past resolved questions in that category. Then, for each input triplet we infer the distribution over topics for Q_{new}, Q_{past} and A separately. From these distributions, we generate topic quality features for each entity by measuring the entropy of the topic distribution and extracting the probability of the most probable topic. We also measure topic similarity between any two entities via both Jensen-Shannon and Hellinger divergences between the two topic distributions. Finally, we look at the

binary match/mismatch of the most probable topic in each distribution.

Lexico-syntactic Analysis: We parse each question title using the Stanford dependency parser[6] [10]. From the parse tree we extract the WH question type, if it exists, and the number of nouns, verbs and adjectives as question quality features. We also check for a match between the WH question type of Q_{new} and Q_{past}.

In addition, the main predicate and its arguments are extracted, either the main verb and its subject and object or the main noun or adjective for a non-verbal sentence. The main predicate is also marked for negation. For example, from "`Why doesn't my dog eat?`", we extract '¬eat' as the negated predicate and 'dog' as its subject. We then generate features that test for mismatch between the main predicates in Q_{new} and Q_{past}, and mismatch between the predicate arguments. These features help to identify semantic inconsistencies between questions even if the topic and lexical similarities are high. For example, they help identifying that "`Why doesn't my dog eat?`" and "`Why doesn't my cat eat?`" have different needs even though the LDA similarity and cosine similarity are high (since 'dog' and 'cat' have very low IDF scores).

Result List Analysis.

We adopt methods used in post-retrieval query-performance prediction, which estimates the quality of document retrieval to a given query, in order to measure the quality of the new and past questions and their similarity to the answer. Specifically, post-retrieval prediction methods exploit properties of the retrieved document list (the *result list*) to attest to query difficulty. Since our objective differs from that of traditional query-performance prediction, we present variants of two of these measures that can be effectively applied for the task question difficulty estimation in CQA.

Query Clarity is an effective measure for query ambiguity proposed by Cronen-Townsend et al. [9]. It measures the coherence of the result list with respect to the corpus via the KL-divergence between the language model of the result-list and that of the corpus. It relies on the premise that ambiguous queries tend to retrieve documents on diverse topics and thus the result list language model would be similar to that of the corpus. Since we rely largely on the question title when looking for a candidate answer, we utilize the Clarity measure to detect ambiguous titles of new questions. For instance, the result-list for the title "`What's wrong with me?`" contains questions on many diverse topics, resulting in low Clarity.

We adapt the traditional Clarity measure as follows: a) we obtain the result list for the title of the new question; b) the rank-based language model is constructed using the method described in [9], taking the entire question text (title+body) for each retrieved question; c) The KL divergence between this language model and the language model of the whole repository is calculated and serves as the Clarity score.

Query Feedback is a state-of-the-art performance predictor for ad-hoc retrieval, proposed by Zhou and Croft [36]. Query Feedback treats the retrieval process as a transmission of the query q over a noisy channel and the retrieved result-list L as the corrupted version of q. The quality of L is associated with the quality of the noisy channel. In order to measure that quality, query q' is distilled from L and a secondary retrieval is performed, resulting in list L'. The overlap between the document lists L and L' is an estimate for the quality of the channel, as it corresponds to the amount of information which we recover from L. The main idea behind Query Feedback is that informational similarity between two queries can be effectively estimated by the similarity between their ranked document lists.

We adapted the Query Feedback measure to our setup, in order to measure both the quality of the two queries and their similarity to each other and to the answer. Lets denote the list of retrieved questions to an input query q by $L(q)$, and the similarity (overlap) between two such lists by $sim(q, q') = overlap(L(q), L(q'))$. Following, we generate these features:

- Intra-question similarity: sim(title of Q_{new}, Q_{new}), sim(title of Q_{past}, Q_{past})
- Inter-question similarity: sim(title of Q_{new}, title of Q_{past}), $sim(Q_{new}, Q_{past})$
- Question-answer similarity: sim(title of Q_{new}, A), $sim(Q_{new}, A)$, sim(title of Q_{past}, A), $sim(Q_{past}, A)$

Intra-question agreement features capture the coherence of a question by identifying when the question title has little in common with its body. Inter-question similarities address the agreement on information need between the two questions, while question-answer similarities address the agreement between the information need and the information provided by the answer. It is important to point out that key point difference between two entities can be easily "missed" by surface level similarity measures. For example, given to a past question "`what dog food is the best?`", the sarcastic answer "`Other dogs`" has high surface level similarity. Yet, this answer is not informative and indeed its similar answers refer to questions that are unlikely to focus on dogs food, resulting in low Query Feedback based similarity.

We note that $L(A)$ is a ranked list of questions, and can be compared to the lists retrieved for the questions. Yet, we refrain from finding similar questions to the answer directly, because of the lexical gap between answers and questions. Instead, we follow the intuition that similar answers indicate similar questions [18]. Hence, $L(A)$ is generated by first retrieving a ranked list of answers from the best answer corpus, ordered by their similarity to the candidate answer. Then, we construct $L(A)$ from the questions whose corresponding best answers were retrieved, keeping the order of the retrieved answers.

Result List Length: While not a performance predictor, we also add as a feature the number of questions that pass the threshold α, which is described in Section 3.1. This feature helps capturing the level of uniqueness of the new question in the corpus.

3.2.2 Classification Model

We performed preliminary experimentation with four families of classifiers: Random Forest, Logistic regression, SVM and Naive Bayes, as implemented by the Weka machine learning workbench [14]. Using F_1 and Area under ROC curve as quality measures, the Random Forest classifier [6] showed consistently superior results and was thus selected as our classification model for stage two of the algorithm. There are two parameters controlling Random Forest: (1)

[6] `http://nlp.stanford.edu/software/lex-parser.shtml`

number of trees, (2) number of features used at node splitting. We use a conservative parameter setup #trees=50, #features = 7, and we note that our experiments showed very little sensitivity to these parameters.

4. AN OFFLINE EXPERIMENT

As our first experiment, we wanted to evaluate our algorithm, and especially our two stage classifier. To that end, we set up a question-answering system based on our algorithm and constructed an annotated dataset for evaluating it. We next describe our system and experimental setup, the construction of the annotated dataset, the training of the classifier and the evaluation results

4.1 Experimental Setup

In order to build a working system, we needed a repository of past questions whose answers may serve as candidates for answering new questions. We collected resolved questions and their best answers from three active top-level categories in Yahoo! Answers: *Beauty & Style*, *Health* and *Pets*. These categories discuss very different topics, and contain diverse types of questions, from factoid and advice (*e.g.* "`fastest way to get rid of a cold sore?`") to opinion and social (*e.g.* "`what is my face shape?`"). As mentioned in Section 3.1, since we aim at providing high quality answers, we included in the repository only best answers chosen by the askers, and excluded best answers chosen by the community as well as those that received fewer than three stars. The repository was constructed from resolved questions dated between Feb and Dec 2010. After the above filtering, the repository included 305,508, 449,890 and 201,654 examples for *Beauty & Style*, *Health* and *Pets*, respectively.

Query indexing and retrieval was performed using the Lemur Toolkit[7]. We used Lemur's implementation of *query clarity* with the unsmoothed ranked list language model described in [9], trimmed to 100 terms. We learned 200 LDA topics per category, with all hyper-parameters set to 0.1. The *query feedback* overlap parameter was empirically set to 5,000, 8,000 and 2,000 for *Beauty & Style*, *Health* and *Pets* respectively.

4.2 Dataset Construction

To train and evaluate our classifier, we used pairs of the form *(new_question, past_answer)* where *past_answer* belongs to the repository while *new_question* does not. Each such pair was associated with a label, *valid* for past answers that satisfy the new question, and *invalid* otherwise. To generate such pairs for each tested category, we first randomly sampled 1,200 questions that were posted between Jan and Mar 2011. We then found for each sampled question a candidate answer by utilizing stage-one of our algorithm. The only free parameter to set at this stage is the threshold α, which is used for filtering out past questions that are not similar enough to the new question. To choose the value of α, we conducted a preliminary analysis of the percentage of new and past questions that have identical information need, as a function of the cosine similarity between their titles. We annotated 276 pairs[8] and found that about 80% of the pairs with shared need exhibited a title cosine similarity above 0.9. Therefore, we chose $\alpha = 0.9$, which provides precision oriented high threshold yet manages to maintain reasonable recall.

[7] `www.lemurproject.org`

[8] These pairs are different from the ones in the dataset.

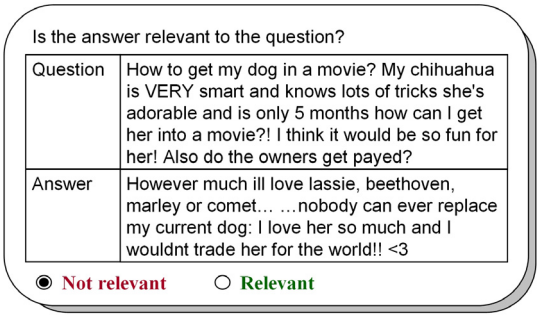

Figure 5: Mechanical Turk interface for labeling matching candidate answers to new questions

4.3 Human Labeling of the Dataset

In order to annotate the pairs consisting our dataset, we utilized Amazon's Mechanical Turk (MTurk). Each MTurk worker received a set of HITs (Human Intelligence Task, as defined by Amazon). Each HIT consists of 22 pairs, and for each pair the worker had to decide whether the answer was relevant to the question or not. Figure 5 shows the interface we used with an example pair. Given that the task is somewhat subjective, we assigned seven workers to each HIT. Additionally, out of the 22 pairs in each HIT, two had known labels and were introduced as traps for quality control (the traps were not included in the constructed dataset). Failing to answer both traps correctly resulted in discarding the entire HIT for the worker. This procedure proved itself very significant in terms of data quality.

To assess the quality of the workers, we calculated the inter-worker agreement in each category using Fleiss' kappa[9] [13, 26]. Table 1 summarizes the calculated kappa values and their standard errors. The agreement is fair for all three categories, with somewhat lower agreement for *Health*. Indeed, *Health* examples were more difficult to evaluate, mainly because such questions tend to be long and detailed, and sometimes extremely sensitive as in the example below:

- **Question:** *Could I get sent to a mental hospital for this? I'm just wondering . . . I do not like myself . . . I have been cutting myself . . . Is any of this really normal . . . my grany sometimes says if i dont stop cutting, she will send me to one . . .*

- **Answer:** *My close friend had to go through the same thing; they very much can. She couldn't return to school until after she went to the mental hospital . . .*

This finding expressed itself also in our attempts to learn a supervised model for this dataset. Still, all three kappa values are statistically significant, with p-values practically zero in all three cases.

In the final step of annotating the dataset, we relied on the MTurk labels for each pair in order to decide whether the pair represents a positive or a negative example. Since our

[9] Fleiss' kappa expresses the degree to which the observed amount of agreement exceeds the expected agreement in case all raters made their ratings randomly.

Category	#examples	% pos	kappa	Δ kappa
Beauty	1,200	59.2	0.337	0.006
Health	1,220	53.0	0.224	0.007
Pets	1,198	51.4	0.372	0.006

Table 1: Annotation statistics: the number of annotated pairs per category, the fraction of positive pairs, Fleiss' kappa and the kappa's standard error

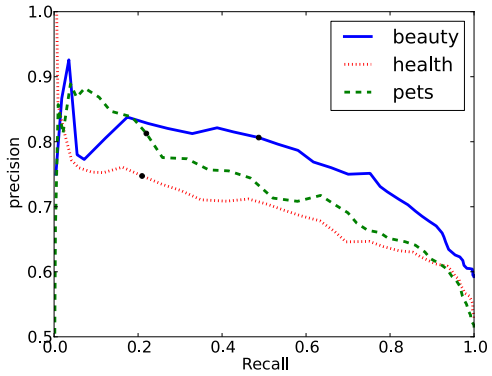

Figure 6: Recall precision curves for the Random Forest model by category (the black dots correspond to the thresholds selected to setup our system in the online experiment see Section 5)

primary goal is to achieve high precision, we chose to label an example as positive only if it was annotated as 'Relevant' by at least five out of seven workers; otherwise, the pair was labeled as negative. This methodology aims at generating a relatively reliable positive set of examples, while the negative set might include also controversial positive examples. Table 1 details the results of the labeling task, producing a nearly balanced dataset.

4.4 Results

For each of the three tested categories we trained a different classifier based on the Random Forest model described in Section 3.2.2. The model turned out to be quite effective in discriminating positive from negative examples. Figure 6 depicts the recall-precision curves of the classifiers for each of the three tested categories, obtained by 10-fold cross validation. The graph shows that high quality answering can be achieved for a substantial 50% recall in *Pets*, and even higher in *Beauty & Style*. High quality answers for *Health* may be maintained for a somewhat lower recall level of about 20%, possibly due to the lower quality of *Health* examples (an error analysis of the classifiers is discussed in Section 5.3).

We analyzed the contribution of the various features to the model by measuring their average rank across the three classifiers, as provided by the Random Forest. Table 2 presents the 15 most informative features to the model. It is notable that various *query feedback* features play a major role in identifying valid answers. This occurs with respect to the quality of past questions (# 9 in the table), but mainly for measuring agreement in information need between the questions and the answer (# 1, 2, 4, 5, 7). Indeed, our adapted *query feedback* features capture similarities beyond mere lexical overlap, but without the coarser overview of topic models. Yet, other feature families help too. Both latent topic similarity (# 8, 10) and surface level cosine similarity (# 3, 6, 13, 14) add reliability to the agreement on similar topics in the questions and the answer. Finally, surface level features (# 11, 12, 15) help quantify the quality and focus of the new question and the candidate past answer. In general, we note that the top features address all aspects of entity "quality" and entity similarity, as illustrated in Figure 4, which indicates that all these aspects contribute to the answer validation task.

Table 2: The 15 most informative features, sorted by their mean rank. We write CS for 'Cosine Similarity' in short

#	Feature Name	Mean Rank
1	Query feedback: Q_{new} vs. A	1.3
2	Query feedback: title of Q_{new} vs. A	2.0
3	CS: Q_{new} vs. A	2.6
4	Query feedback: title of Q_{past} vs. A	5.0
5	Query feedback: Q_{new} vs. Q_{past}	7.0
6	CS: body of Q_{new} vs. body of Q_{past}	7.3
7	Query feedback: Q_{past} vs. A	9.0
8	Jensen-Shannon (LDA): Q_{new} vs. Q_{past}	11.6
9	Query feedback: title of Q_{past} vs. Q_{past}	12.3
10	Hellinger (LDA): Q_{new} vs. Q_{past}	14.0
11	Answer length	15.6
12	Answer stopword count	17.3
13	CS: Q_{past} vs. A	17.6
14	Question-answer CS difference	18.6
15	Average IDF score in title of Q_{new}	20.6

Category	Precision	Coverage
Beauty	80.7%	9.5%
Health	74.7%	2.6%
Pets	81.3%	2.5%

Table 3: Expected performance of online evaluation, based on offline results, the coverage with respect to the all incoming questions

In summary, the recall precision curves of all three categories present negative slopes, as we hoped for, allowing us to tune our system to achieve high precision.

5. AN ONLINE EXPERIMENT

5.1 Online Evaluation Setup

We wanted to further test our question answering system in the most realistic environment possible. To this end, we created three new user accounts Yahoo! Answers, nicknamed Alice, Jane and Lilly, each being in fact a robot operated by our system and automatically answering, each in its own category, respectively *Beauty & Style*, *Health* and *Pets*. The robot works as follows: it inspects incoming questions in its category and posts an answer to it if our answering system does return one. The robot also adds a reference to the past question from which the answer was recycled.

The system training and parameter setup for each category is based on the results of the offline experiment (see Section 4.4). The only free parameter of the online evaluation is the threshold on the classifiers output in stage two, which decides which answers qualified to be served. Since any user in Yahoo! Answers has a limited number of questions she can answer online per day, we set this threshold to achieve high precision with respect to this limited coverage

Category	By Asker		Total	
	Robot	Avg. User	Robot	Avg. User
Beauty	*28.8%	7.4%	30.3%	17.8%
Health	20.9%	8.3%	30.6%	19.6%
Pets	12.6%	5.3%	19.2%	11.6%

Table 4: Best answer rates out of all resolved questions: system vs. average human answerer. All improvements are statistically significant at the 0.001 level, except the starred one, which is statistically significant at the 0.05 level.

Category	Dataset Properties		System Performance	
	Total	Positive	Precision	Recall
Beauty	198	60.1%	75.0%	45.4%
Health	195	57.9%	86.4%	16.8%
Pets	198	55.0%	65.9%	26.6%

Table 5: Annotated dataset properties and the resulting system performance.

level. The three dots in Figure 6 correspond to this chosen threshold in each category. Since the performance on *Health* is substantially inferior, we were forced to aim at a lower precision of 75% to reach the permitted coverage, while in the other two categories we targeted a higher 80% precision. Table 3 summarizes the expected precision and coverage for each category at its selected threshold.

5.2 Online Evaluation Results

Alice, Jane and Lilly, our robots actively answered in their respective categories for about a week. An interesting fact is that the absolute majority of other (real) Yahoo! Answers users were not aware that these users are robots (even if the robots posted back references), and viewed them as other humans. This property is reflected both by discussions between answerers, as well as the fact the both Jane and Alice have by now several fans who follow their actions. This behavior is important, since we wanted to compare the performance of the robot users to that of an average user in each tested Yahoo! Answers category. In practice, the robot was always the first to post an answer, which attests to the high efficiency of our system, relatively to an average answerer.

To this effect, we used the best-answer feedback in Yahoo! Answers as our comparison criterion. We first calculated the average best-answer rate achieved by users in each category for questions posted between Feb and Dec 2010. We then measured for each robot the rate of: (1) best answers that were chosen by the askers, which is, as mentioned before, an explicit evidence of asker's satisfaction, and (2) all best answers whether by asker or by the community. In this analysis we took into account only resolved questions, for which a best answer was chosen. The results are presented in Table 4, showing that our robots and consequently our algorithm substantially outperform the average user in all three categories with respect to both types of best answer rates. It shows an increase of more than 50% in rates for total best answers, but more importantly it more than doubles the rates for best answers by askers. While it is no secret that the answers our system provides are typically of high quality since they were chosen as best answers before, this result is a good indication of the quality of our two-stage matching approach, as well as of the potential reuse of past answers.

Category	Passed Stage I	Passed Stage II
Beauty	22.7%	6.2%
Health	18.6%	2.8%
Pets	17.8%	3.0%

Table 6: Coverage of online system, detailing the fraction of all questions that pass the first and second stage.

Category	Unanswered Rate	Unanswered Sample Size	Answered by System
Beauty	15.4%	10,000	6.4%
Health	15.3%	10,000	1.9%
Pets	5.8%	4,291	2.7%

Table 7: Unanswered rate in each category, the sample size of unanswered questions and the percentage of sampled questions that could have been answered by our system.

To further assess our robots' performance, we randomly sampled from each category 200 new questions that passed the first stage of our algorithm and thus were considered by our stage-two classifier for answering online. These 600 questions and their candidate answer were evaluated by three external human annotators, who were asked to label each question-answer pair as relevant or not. The annotators were given detailed and strict guidelines for labeling examples. Pairs that an annotator could not decide how to label were eventually discarded from the dataset. Fifty pairs were shared by the annotators for agreement assessment (with majority over labels chosen as final labeling).

Table 5 details the annotated dataset and the system performance on it. There are two main reasons for the differences between the precision and recall in this table and the expected values detailed in 3. First, the latter were obtained using cross-validation, whereas the former by using a model trained on the whole training set. It is a well known fact that the cross validation performance assessment is slightly biased [20]. Second, the two datasets were labeled by different annotators, with different qualifications and motivation. Finally, the different time-frames from which the examples for the two datasets were taken may also contribute to small changes in performance. Taking these effects into account, together with the inherent error bars of both analyses, these estimates of precision and recall in the offline and the online experiments are in good agreement. Inter-annotator agreement was calculated via Fleiss' Kappa. We obtained kappa 0.57 (0.08), with p-value practically zero, which indicates a moderate agreement level, and is higher than that obtained by the MTurk workers (see Section 4.2).

We were interested in the difference in performance of our system on unanswered questions, which is one of our research motivations. To this end, we first measured our actual system coverage on the answers our robots provided (Table 6). We then sampled 10,000 unanswered questions in each category between Jan and Mar 2011 (in *Pets* we only found 4,291 in total) and attempted to "answer" them using our system with the same settings, measuring our system coverage on the sample (Table 7). Comparing between Table 6 and Table 7, we see that our system coverage for unanswered questions is in accordance with that measured for all incoming questions. This result may indicate that unanswered questions are at large not harder to answer than answered

questions, but are just missed or ignored by potential answerers.

To conclude, this experiment showed that askers posting highly personal questions in Yahoo! Answers, for which an "objective" answer is difficult to find in search-engines, may benefit from high quality answers given to similar past questions. Furthermore, our results show the potential for performing this task with automatic answering techniques. While there is a lot of room for improvement, our current system already outperformed the quality of average human answerers in terms of best answer rates.

5.3 Discussion

To get a better insight of our current system performance, we qualitatively analyzed several dozens false positives and false negatives in each tested category.

We found that while short questions might suffer from vocabulary mismatch problems and sparsity, the long, cumbersome descriptions introduce many irrelevant aspects which can hardly be separated from the essential question details (even for a human reader). Thus, two questions that are viewed as dissimilar based on the entire text may actually express very similar intent, and conversely, apparently similar questions might have very different intents. An illustrative example of dissimilar questions sharing the same intent is given below:

Q_{new} *I need help choosing a pet? Hello, I really, really would like a pet. However, I'm not sure . . . I would love a dog but . . . Also I don't not have much room for a cage . . . any ideas . . . ? Also, I have had hamsters . . . and I don't like them at all! . . . I would like something different and qwerky*

Q_{past} *I need help choosing a pet? I love rats but unfortunately my mom hates them! . . . I while ago I was asking for a bunny . . . she said: ". . . " The day after . . . I accidentally clicked Guinea pig . . . my heart just soared! . . . Unfortunately I can't have both . . . when i think about the guinea pigs my heart starts pumping. But i still want a rabbit very much! . . .*

One possible direction for reducing false negatives (dissimilar questions with similar intent) is to use ad-hoc retrieval techniques, when comparing between long documents, for instance by incorporating passage similarity or other term-proximity related measures. There is an inherent difficulty in detecting the central aspect of a question when it is masked by the large amount of surrounding text in long descriptive questions. We found that terms that are repeated in the past question and in its best answer should usually be emphasized more as related to the expressed need. Our current approach misses this insight, since it addresses the past question and the past answer independently. In future work we plan to better model this linguistic inter-dependency between the past question and its answer.

Another more problematic source of errors are the false positive cases (similar content with dissimilar intent). Our classifier while focused on eliminating such cases, can be easily tricked by time-sensitive questions for instance. We therefore took the simplifying assumption on not considering highly time-sensitive categories such as Sports or Politics and reserve this area of research to future work.

Moreover, various social and demographical aspects interfere with the traditional notion of intent and user satisfaction, affecting the dynamics of CQA. These aspects, while important, are out of the scope of this work. Additionally, CQA suffers from a large amount of noise and variability (similarly to other user generated content on the Web), such as invalid language usage, heterogeneous styles and plain spam, which largely affects the ability to perform high quality deep analysis of the text. Even extracting the question type and focus from the title may require discourse analysis, *e.g.* in "`I only eat dinner. Is that bad?`".

Despite all the above, we have demonstrated that, in practice, reusing previous answers results in relatively high user satisfaction, based on the "best-answer" rates achieved by our robots. We may attribute this to two key observations: (1) two questions do not have to express the exact same intent in order for the answer to satisfy both (2) as discussed above, in some questions the information need is limited, while the social need is more central. In the first scenario, a *general informative answer* can satisfy a number of topically connected but different questions. For example, a detailed answer on the habits of gerbils may answer any specific question on these pets. In a different tone, answering the question template "`I think I broke my <?>, what should I do?`" with the answer "`go see a doctor`" is typically viewed as valid.

Finally, our second observation above concerns human interaction aspects, where askers are driven or influenced by social needs such as empathy, support and affection. In fact, the second observation is that a *general social answer*, may often satisfy a certain type of questions. For example, in our analysis we found that, not surprisingly, a substantial amount of incorrect answers are provided to questions containing links, since our robots answer them "blindly", without the capability of analyzing the links at this stage. Yet, quite surprisingly, we also found that our robots do manage to successfully answer many questions with links when using general social answers. A typical example is the question "`Am I pretty?`", which calls for the (usually satisfying) answer "`Yes, you are gorgeous!`", regardless of any link or additional description in the question body.

6. CONCLUSIONS

Askers visiting a CQA site need to have a sufficient level of confidence that their questions will be answered, in order to come to the site in the first place. Consequently, it is critical to keep the rate of unanswered questions to a minimum. We proposed here to help alleviating this problem by using an automatic question answering system. Unlike automatic answering systems that synthesize new answers, our approach here remains in the human-generated content spirit of the CQA and reuse past answers "as is". We presented a two-stage question answering algorithm that first identifies the resolved past question that is most similar to a target new question, and then applies a statistical classifier in order to determine whether the corresponding past answer meets the new question needs.

We evaluated our algorithm offline on an annotated dataset, showing that high precision answering is possible while maintaining non-negligent coverage. More interestingly, we tested our algorithm by building a live system that operates three robots, Alice, Jane and Lilly, who behave as real users on Yahoo! Answers and answer, when confident enough, live questions in three active categories. Our analysis showed that these robots outperform the level of the average answerer

in terms of asker's satisfaction, thus meeting our high precision goal, but still posted a reasonable amount of answers, reaching their daily answering limits.

In future work, we want to improve the quality of our robots in terms of precision and coverage by harnessing the askers' feedback on site via active learning. In addition, we would like to better understand time-sensitive questions, such as common in the *Sports* category, as their answers should be very carefully reused, if at all.

7. ACKNOWLEDGMENTS

The authors would like to thank Elad Yom Tov for helpful discussions and ideas, and to Savva Khalaman and Shay Hummel for their assistance in the manual evaluation.

8. REFERENCES

[1] E. Agichtein, S. Lawrence, and L. Gravano. Learning search engine specific query transformations for question answering. In *WWW*, 2001.

[2] E. Agichtein, Y. Liu, and J. Bian. Modeling information-seeker satisfaction in community question answering. *ACM Trans. Knowl. Discov. Data*, 3, 2009.

[3] D. Bernhard and I. Gurevych. Combining lexical semantic resources with question & answer archives for translation-based answer finding. In *ACL*, 2009.

[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*, 2008.

[5] D. M. Blei, A. Y. Ng, M. I. Jordan, and J. Lafferty. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:2003, 2003.

[6] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[7] D. Carmel, M. Shtalhaim, and A. Soffer. eresponder: Electronic question responder. In *CooplS*, 2000.

[8] A. Corrada-Emmanuel, W. B. Croft, and V. Murdock. Answer passage retrieval for question answering, 2003.

[9] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. Precision prediction based on ranked list coherence. *Information Retrieval*, 9(6):723–755, 2006.

[10] M.-C. de Marneffe, B. MacCartney, and C. D. Manning. Generating typed dependency parses from phrase structure parses. In *LREC*, 2006.

[11] G. Dror, Y. Koren, Y. Maarek, and I. Szpektor. I want to answer; who has a question?: Yahoo! answers recommender system. In *KDD*, 2011.

[12] H. Duan, Y. Cao, C.-Y. Lin, and Y. Yu. Searching questions by identifying question topic and question focus. In *ACL*, 2008.

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

[14] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, 2009.

[15] C. Hauff, D. Hiemstra, and F. de Jong. A survey of pre-retrieval query performance predictors. In *CIKM*, 2008.

[16] B. He and I. Ounis. Inferring query performance using pre-retrieval predictors. In *SPIRE*, 2004.

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

[18] J. Jeon, W. B. Croft, and J. H. Lee. Finding semantically similar questions based on their answers. In *SIGIR*, 2005.

[19] J. Jeon, W. B. Croft, and J. H. Lee. Finding similar questions in large question and answer archives. In *CIKM*, 2005.

[20] R. Kohavi. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In *IJCAI*, 1995.

[21] B. Li and I. King. Routing questions to appropriate answerers in community question answering services. In *CIKM*, 2010.

[22] X. Liu and W. B. Croft. Passage retrieval based on language models. In *CIKM*, 2002.

[23] E. Mendes Rodrigues and N. Milic-Frayling. Socializing or knowledge sharing?: characterizing social intent in community question answering. In *CIKM*, 2009.

[24] J. M. Prager. Open-domain question-answering. *Foundations and Trends in Information Retrieval*, 1(2):91–231, 2006.

[25] I. Roberts and R. Gaizauskas. Evaluating passage retrieval approaches for question answering. In S. McDonald and J. Tait, editors, *Advances in Information Retrieval*, volume 2997 of *Lecture Notes in Computer Science*, pages 72–84. Springer Berlin / Heidelberg, 2004.

[26] J. Sim and C. C. Wright. The kappa statistic in reliability studies: Use, interpretation, and sample size requirements. *Physical Therapy*, March 2005.

[27] R. Soricut and E. Brill. Automatic question answering: Beyond the factoid. In *HLT-NAACL*, 2004.

[28] T. Strzalkowski and S. Harabagiu. *Advances in Open Domain Question Answering (Text, Speech and Language Technology)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[29] M. Surdeanu, M. Ciaramita, and H. Zaragoza. Learning to rank answers on large online QA collections. In *HLT-ACL*, 2008.

[30] S. Tellex, B. Katz, J. Lin, A. Fernandes, and G. Marton. Quantitative evaluation of passage retrieval algorithms for question answering. In *SIGIR*, 2003.

[31] A. Tsotsis. Just because google exists doesn't mean you should stop asking people things, October 2010. Techcrunch.

[32] E. M. Voorhees. The trec-8 question answering track report. In *Text REtrieval Conference*, 1999.

[33] E. M. Voorhees. Overview of the trec 2003 question answering track. In *Text REtrieval Conference*, 2003.

[34] K. Wang, Z. Ming, and T.-S. Chua. A syntactic tree matching approach to finding similar questions in community-based qa services. In *SIGIR*, 2009.

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

[36] Y. Zhou and W. B. Croft. Query performance prediction in web search environments. In *SIGIR*, 2007.