When Relevance is not Enough: Promoting Diversity and Freshness in Personalized Question Recommendation

Idan Szpektor, Yoelle Maarek, Dan Pelleg
Yahoo! Research
Haifa 31905, Israel
{idan,dpelleg}@yahoo-inc.com, yoelle@ymail.com

ABSTRACT

What makes a good question recommendation system for community question-answering sites? First, to maintain the health of the ecosystem, it needs to be designed around answerers, rather than exclusively for askers. Next, it needs to scale to many questions and users, and be fast enough to route a newly-posted question to potential answerers within the few minutes before the asker's patience runs out. It also needs to show each answerer questions that are relevant to his or her interests. We have designed and built such a system for Yahoo! Answers, but realized, when testing it with live users, that it was not enough.

We found that those drawing-board requirements fail to capture user's interests. The feature that they really missed was diversity. In other words, showing them just the main topics they had previously expressed interest in was simply too dull. Adding the spice of topics slightly outside the core of their past activities significantly improved engagement. We conducted a large-scale online experiment in production in Yahoo! Answers that showed that recommendations driven by relevance alone perform worse than a control group without question recommendations, which is the current behavior. However, an algorithm promoting both diversity and freshness improved the number of answers by 17%, daily session length by 10%, and had a significant positive impact on peripheral activities such as voting.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Apps—*Data Mining*

General Terms

Algorithms, Analysis, Experimentation, Human Factors

Keywords

Recommender Systems, User Models, Community Question Answering

1. INTRODUCTION

Community Question Answering (CQA) websites, such as Yahoo! Answers, Quora, Baidu Zhidao and WikiAnswers, offer a convenient means for Web users to address needs that Web search engines cannot satisfy. They range from complex, rare or heterogeneous needs, for which content does not exist yet on the Web, to socially oriented needs, for which the asker wants a human to provide a personal subjective opinion. Examples of such needs include homework help, *e.g.* "*how to integrate* e^{-x} *?*", opinion seeking, *e.g.* "*what would be the best kind of pet to have?*", recommendations, *e.g.* "*best place to hang out in Paris?*", etc. Answerers, in return, help their fellow users mostly for social reward [22], thus creating an askers/answerers ecosystem that all users, active or not, can benefit from.

In order to keep this ecosystem alive, new questions need to be answered constantly. It is therefore crucial for CQA sites to facilitate the answering task. One common way to do so is to recommend questions to potential answerers. Most previous studies on the topic have focused on presenting each question only to the best possible answerers, namely the "experts", so as to satisfy the asker's needs [13, 14, 17, 28]. However, to maintain the ecosystem, it is important to satisfy all potential answerers and not only a limited number of experts. A data analysis we conducted on a sample of 4 million answers from Yahoo! Answers revealed that level-1 users[1], *i.e.* the most junior users in the system (users with less than 250 points) generate almost a third of all answers, as shown in Figure 1. Furthermore, the graph also shows that level-1 and level-2 users together (users with less than 1,000 points) contribute almost 50% of all answers. It would therefore be a major mistake to ignore these "regular" answerers.

A key element to satisfying all types of users is to better understand what types of questions really attract them. The usual approach in question recommendation has been to focus on the relevance of the question to the user, that is, to what degree the question matches the user's tastes [21, 15]. Yet, CQA sites, like most user-generated content sites, are highly dynamic and constantly expose their users to diverse and fresh content originating from other users. Following this observation, we argue here that the engagement of users in CQA sites is driven not only by relevance but also by diversity and freshness needs. These needs were acknowledged in traditional recommender systems [30, 3], as well as in traditional Information Retrieval [8]. However, to the best of our knowledge, they have been ignored in current question recommendation research.

[1]See `http://answers.yahoo.com/info/scoring_system` for more details on levels and points in Yahoo! Answers.

WWW 2013, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2035-1/13/05.

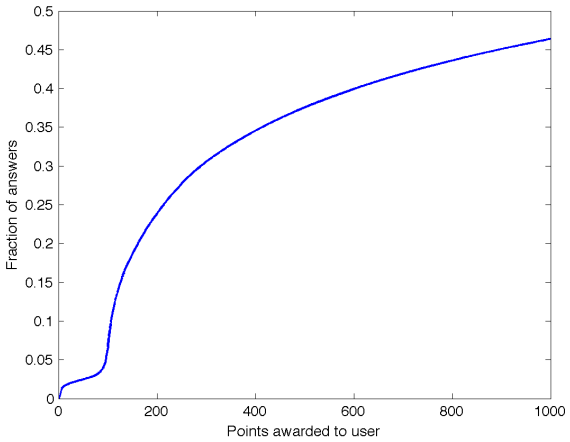

Figure 1: Cumulative rate of answers contribution, by level of activity in Yahoo! Answers

Following the above motivations, we introduce in this paper a novel question recommendation approach that is designed to meet three requirements: (a) questions need to be recommended for all types of users, from casual with minimal historical data to highly active experts, (b) questions have to be diverse and intriguing in order to keep the potential answerer engaged, and finally (c) recommendations need to be fresh and be served fast. The last requirement refers, among others, to the ability to serve questions as recommendations immediately after they have been posted, as well as to instantly adapting to users' changes in taste as they answer more questions. The above three requirements impose serious scalability constraints on both the serving and learning stages.

In our approach, a user is represented by a *profile* that is instantiated as soon as she answers her first question, allowing even new answerers to receive recommendations. The profile is then incrementally updated and immediately tuned for every new answer the user provides, getting richer with each additional answer. Similarly, profiles for questions are generated right after they are posted. Thus, questions immediately become candidates for recommendation, increasing their chances to be answered fast. Finally, recommendations are selected by relevance to the user, matching question profiles to the user profile. Yet, in addition to identifying personalized relevant questions, we also guarantee diversity by using a novel proactive *topic sampling* algorithm that enforces recommendations to match different topics within the user profile.

We have evaluated our system in production in Yahoo! Answers, launching a "bucket" experiment over a percentage of its users. The users selected in the experiment were exposed to a new tab when visiting the site, labeled "recommended" as shown in Figure 2. In this specific example, the questions were personalized for a user who had answered a few questions in travel in France and in movies. Several configurations of our recommender system were evaluated over a period of two weeks, and the activities of users in each configuration were compared to a control group. The surprising results showed that based only on relevance, the recommender system actually had a negative effect compared

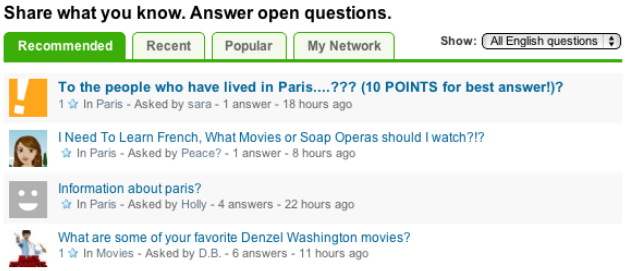

Figure 2: Personalized question recommendation shown in the online experiment in Yahoo! Answers

to the control group. However, when we added freshness, forcing recommendations to come from recently asked questions, the trend changed, and users answered 4% more when offered personalized *and* fresh questions. But, the highest user activity rates were achieved when diversification was added, even at the cost of reduced freshness. In this configuration, users provided 17% more answers than the control group, and the improvement was observed across all user levels. Furthermore, this successful question recommendation experience had an indirect positive effect on many other user activities. These include, among others, higher asking rates (+5%), voting rates (+19%) and longer dwelling times on the site (+10%).

2. RELATED WORK

With millions of active users, Yahoo! Answers connects between askers and answerers, who interact on a large variety of topics. Askers post questions by providing a title that specifies their core needs, which are often syntactically formulated as a question. They can then optionally add details in a body field. Finally, they assign their question to a specific category within a predefined hierarchy of categories. For example, the question "*how to stop my dog from barking?*" was assigned to the category '*Pets/Dogs*' (*e.g.* the sub-category "Dogs" under the parent "Pets"). Any new question remains "open" for answering for four days, or less if the asker chose a best answer within this period. Users can also rate answers and questions and vote for best answer. Once a question has been answered and a best answer has been chosen, the question is considered resolved.

Today, users in an "answering mood" typically scan a long and dynamic list of all open questions, looking for questions to answer. This list is ranked by recency, with the freshest questions at the top. It is also very diverse, since new questions are asked on different topics all the time. As a result, it is pretty tedious for users to find the questions they like to answer. Consequently, prior work on question recommendation mostly focused on personalizing suggested questions to the user by relevance. The efforts in this direction can be classified into two types, which we refer to as "question routing to experts" and "question recommendation to all".

The first type aims at satisfying first of all the asker, and to this effect attempts to identify the most qualified answerers, or "experts", in order to route questions to them [13, 29, 14, 16, 18, 17, 28, 23]. While this approach does have some benefits, it ignores new and casual users who are essential

elements of the ecosystem as discussed before, and represent the majority of users in Yahoo! Answers [11].

The second class of studies, to which this work belongs, aims at satisfying all potential answerers by recommending personalized open questions to them, and include [21, 15, 10]. Most of the existing solutions in this class however suffer from several limitations. First, these algorithms do not scale well to real-time ranking of millions of questions to hundreds of users per second, as required in a large site like Yahoo! Answers. One scalability limitation in these algorithms is the utilization of complex machine learning or time consuming feature construction [15, 10], which hinders fast searching within millions of open questions. Another scalability limitation lies in offline construction of user preferences [21], which prevents real-time response to new questions and answers. Second, the needs of new users with very little historical data are not addressed well. For example, an offline model construction algorithm, such as [21], cannot serve recommendations to brand new users who answer for the first time, since they have no preferences until the next offline construction round. Not only new users but also new questions are not modeled adequately in many cases. For instance, collaborative filtering approaches such as [15] are ineffective for both new users and new questions, as very little information pertaining to questions is available at posting time, as detailed in the next section.

Finally, all prior research on question recommendation focused only on relevance, ignoring the need for diversity in the recommendations as well as their freshness. Yet, research in other recommendation tasks and in Information Retrieval indicates that both diversity [30, 26, 6, 3, 7] and freshness [8, 9] are critical to user engagement and satisfaction. In this paper, we aim at bridging the gap between current question recommendation algorithms and on-line CQA requirements. We propose an algorithm that recommends questions to any user, taking into account relevance, diversification and freshness, as well as scalability and real-time requirements.

We next present our approach, which includes the representation of questions (Section 3) and users (Section 4) followed by the recommendation of questions to users while addressing relevance, diversification and freshness (Section 5).

3. QUESTION PROFILE

We represent each question by a *question profile*, which is the basic building block in our framework. We will discuss in the following section how *user profiles* are derived from question profiles. Thus, questions and users are represented in the same feature space in order to facilitate the matching between open questions and users for personalized recommendations, as done for instance by [13, 18, 10, 23].

We chose not to use a single feature space for all of questions but rather split it according to the 26 top categories in Yahoo! Answers (*Sports*, *Health*, *Pets*, *Travel* etc.), as they usually represent disjoint users' interests. Another advantage of this separation is an implicit word sense disambiguation. We follow Li et al. [17], who showed that language models that incorporate categories are beneficial when modeling answerers, yet we do not go as far as splitting at the category leaf level. Thus, a question posted in the nested category '*Travel/France/Paris*', will be modeled in the feature space associated with the top category level '*Travel*'. Our main rationale is twofold: first, we want to allow matching question and user profiles across leaf categories, and second, we want to avoid data sparseness in infrequent leaf categories. We do, however, encode the leaf category '*France/Paris*' as a feature in the question profile itself as discussed later.

LDA		Lexical		Category	
$topic_5$:	0.6	bark:	0.7	Pets/Dogs:	1.0
$topic_{30}$:	0.3	dog:	0.1		
		stop:	0.2		

Table 1: Example of question profile for "*how to stop my dog from barking?*" posted in top category '*Pets*'

Since we consider all open questions as recommendation candidates as soon as they are posted, the amount of available information at this time is limited. Indeed, new questions include no information about interactions with answerers, no click data, etc. We note that a consequence of this limited information is that traditional collaborative filtering methods for question recommendation, such as [15], are less appropriate for recommending fresh questions. We decided to take a simplifying approach and to only consider the question textual content (title and body) and its category when building the question profile, ignoring any additional question data that may be added later, as well as the asker id (which is provided at posting time). This is motivated by a previous study conducted by some of the authors of this paper. They showed that the prominent contributors for matching a question to a user are the text and the category of the question, and that user-user interaction does not improve the relevance of question recommendations [10]. We note that this is also the data choice in other studies that considered new questions for recommendation [13, 21, 16, 23]. An additional motivation for utilizing only the initial question data is that it allows for question profiles to be built only once, at posting time, without any updates later. This presents clear benefits in terms of scalability.

Given a freshly posted English question, we first conduct some basic preprocessing on its textual content, before building the actual profile. We concatenate the text and body contents and then apply tokenization, lemmatization, as well as domain-specific stop-word removal[2]. Once the question is preprocessed, we build its profile, which is represented by three vectors: (a) a **Latent Dirichlet Allocation (LDA)** [2] topic vector that represents the latent topics that are related to the question, (b) a **lexical** vector that represents the surface level textual content of the questions, and (c) a **category** vector that represents the category in which the question is posted. Each vector corresponds to a different question model, with its own separate feature space, as illustrated in Table 1.

In the question profile, the relative importance of each vector is left unspecified. Instead, it is tuned in a personalized way for each user in the user profile, as detailed in the next section. In addition, we chose to maintain question profiles under a probabilistic framework, in which each vector represents a question as a probability distribution over the model's feature space. This allows us to have a common comparable ground between models, and facilitates the diversification of recommendations as detailed in Section 5.2.

[2]For conciseness, we omit here the details of these stages.

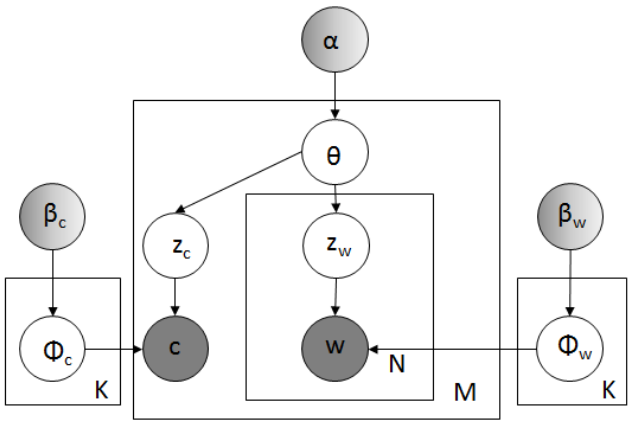

Figure 3: Plate notation of our LDA variant. Categories and words are respectively denoted as c and w. Gradient colored nodes are time-dependent parameters.

3.1 LDA Model

We were inspired by several studies that showed the benefit of utilizing latent topics in various CQA tasks, such as expert finding [13, 18, 23], question recommendation [21] and question retrieval [4, 24]. Following, we employ an LDA variant that explains the textual content of the question as well as its assigned category, as in [13, 4]. We note that the category information is particularly useful for topic inference when there are only few words in the question, as it acts as a constraint over the semantic domain of the question. For longer questions, the effect of the category diminishes. Another particularity of our LDA variant is the incremental time-dependent update of the β and α hyperparameters at training time given new questions, inspired by [1]. This design enables incremental training, and thus supports backward compatibility, since topics are only gradually shifted to include new evidence. As a result, given an incrementally trained model, the only update required in a live system is the reloading of the new LDA models, without the need to construct all question profiles and user profiles from scratch, which would not scale. Our LDA variant is depicted in Figure 3.

Following our top-category split, we learned a separate LDA model for each top category. For each model, we used as initial training set a random sample of up to 2 million resolved questions[3] published in the associated top category between 2010 and 2011. We trained our models using a sparse collapsed Gibbs sampler [25] with 1,000 rounds, learning 200 topics per model. We set β initially to 0.01 and α's sum to 4. After learning the initial model, we evolved it, through incremental learning, to incorporate a random sample of up to half a million questions per top category from the first half of 2012. This allowed us to verify that topics indeed remain stable after a large incremental update, in which each top-category lexicon was enriched by several thousand new words on average. At inference time, we apply 100 burn-in Gibbs rounds and then average 10 rounds with a gap of 10 rounds between each sample.

LDA inference provides a dense topic distribution, in which every topic has a non zero probability due to the Dirichlet smoothing. Even if only topics that were assigned to words are considered, the distribution is not sparse enough due to Gibbs assignment averaging. This is bad news when searching for the best questions that match a user, since many question profiles have to be considered. This is because their topic intersection with the user profile is not empty. Yet, we observed that most questions are short and focus on one theme that represents the asker's need. Therefore, one would expect that most of the probability mass will be assigned to one or two LDA topics. We empirically found it to be the case, and, following this observation, we filter the inferred LDA vector and retain only the topics that were assigned at least 10% of the probability mass. After this filtering, most questions are represented by no more than 3-4 topics that capture the essence of the question, a sparse representation that enables fast matching via an inverted index (see Section 5.1). We note that we do not re-normalize the topic probability after this filtering, but remain with a "discounted" probability distribution (left column of Table 1).

3.2 Lexical Model

We follow prior work in incorporating a unigram bag-of-words representation of a question. This model captures fine-grained word level interests. For example, an answerer may be interested not in haircuts in general, for which there is a specific LDA topic, but in Korean haircuts only. This refinement of the asker's need is addressed by the appearance of the word '*Korean*' in the lexical vector.

To this end, each word in the text of the question is assigned a simple tf·idf score. Note that each top category has its own idf scores, computed over all resolved questions posted in that category between 2010 and 2012. Once the tf·idf scores are calculated, they are $L1$ normalized, resulting in a probability distribution (middle column of Table 1). As for LDA, we support incremental updates to the idf scores.

3.3 Category Model

Finally, the category model is most straightforward as it assigns a probability of 1 to the category in which the question was posted. This model provides a rigid high-level representation of interests, which is useful when encountering fine-grained categories such as '*Paris*', but not as much for more generic leaf categories, such as '*Performing Arts*'.

4. USER PROFILE

In most question recommendation methods, users are represented by their interactions with the questions they answered in the past. More specifically, the user representation is generated by aggregating signals over these questions [13, 29, 21, 15, 16, 18, 17, 10, 28, 23]. We follow this paradigm by deriving *user profiles* from question profiles. In prior work, user profiles might include several models as we do, but the relative weights of models were learned globally, and this unique distribution of weights was applied to all users, as done in [29, 18, 16]. In this work, we take a more personalized approach, learning model weights as well as preferences over top categories for each user separately.

To support this level of personalization, we represent a user profile as a *probability tree*, in which each node consists

[3]All questions considered here have been classified as non-spam beforehand by the usual Yahoo! Answers mechanisms.

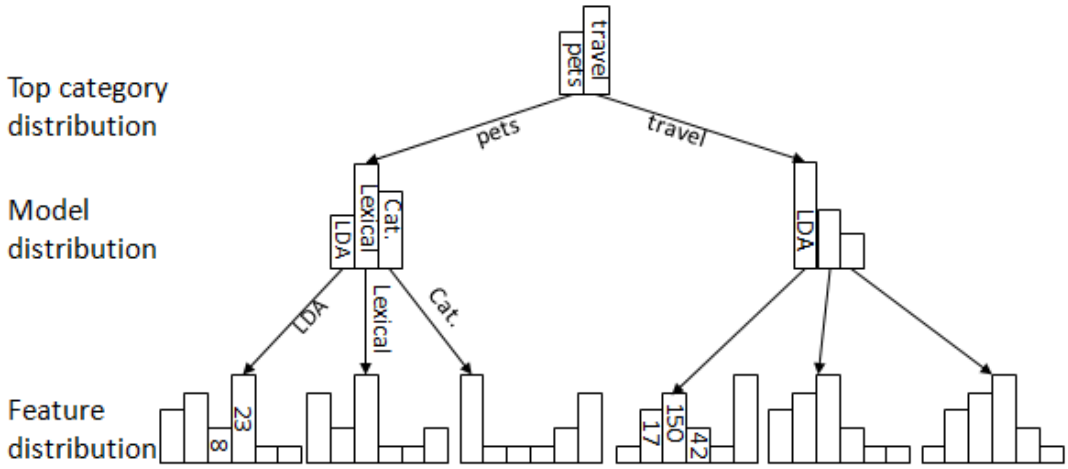

Figure 4: User profile as a probability distribution tree

of a probability distribution that stands over various elements of our question model. Figure 4 exemplifies one such probability tree. There are three levels in each probability tree, defined as follows:

top-category-distribution **level:** The root node consists of a distribution of preference probabilities over top categories. Thus, as illustrated in Figure 4 via histogram-like bars, this specific user has been more active in the '*Travel*' than in the '*Pets*' category and has ignored other categories. For each top category appearing in the root of this user profile, there is an edge (labeled with the top category node in the same Figure) that points to a second-level node.

model-distribution **level:** Each second-level node consists of a distribution of probabilities over the models that define any question profile, namely the LDA, lexical and category models. Each such node points to exactly 3 nodes at the third level of the tree, one for each model.

feature-distribution **level:** The nodes in the third level hold a distribution of probability over features, where features differ for each model. For the nodes reached by an LDA edge, each feature represents a latent topic, for those reached by a lexical edge, each feature is a word, and finally for those reached by a category node, each feature is a leaf category.

We explain next how this data structure is built for each user by aggregating the profiles of the questions the user answered. Specifically, whenever the user answers a question, her user profile is updated with the question profile in a multi-armed bandit fashion. This update changes the probability distributions along the relevant paths in the profile distribution tree that correspond to the question's top category and its models. At the first and third tree levels, representing top-category-distribution and feature-distribution respectively, the updates are rather straightforward, adding each distribution in the question profile to the distribution of the corresponding node in the user profile. One limitation of all prior question recommendation approaches to the best of our knowledge, is that they ignored the fact that users may change their answering tastes. In contrast, we introduce a decaying factor on past questions, reducing their effect on the user profile over time to enable the user to shift their answer tastes more rapidly. We use the following node update formula:

$$p_u = \frac{p_q + \alpha \cdot Z_c \cdot p_c}{1 + \alpha \cdot Z_c}$$
$$Z_u = 1 + \alpha \cdot Z_c$$

where p_q is a distribution in the answered question, p_c is the corresponding current distribution in the user profile, p_u is the updated distribution in the user profile, α is the decaying factor, and Z_c and Z_u are the current and updated normalizing values.

The second level, which represents a distribution over models, cannot be updated in a similar manner, since question profiles do not specify any preference over the LDA, lexical and category models. To overcome this, we first measure the similarity between the feature distribution of each model in the question and the corresponding feature distribution in the user profile. The more similar the vectors, the better this model captures the user's choice of answering this question. The similarity scores are then normalized to a probability distribution, which updates the corresponding second level node in user profile using the same update formula above. Any similarity function is valid, but since we want to promote models that should correlate with future relevant questions, we use here the same similarity function that was chosen also in the recommendation algorithm (see Section 5), which is a dot-product (between two L1-normalized feature vectors).

We discuss next how question and user profiles are used for recommending open questions to potential answerers.

5. QUESTION RECOMMENDATION

We next describe the key elements of personalized question recommendation, namely relevance, diversity and freshness, as captured within our recommendation algorithm.

Pets/LDA/topic$_3$	$\rightarrow$	$(q_7, 0.24), (q_{54}, 0.33), ...$
Travel/Lexical/bag	$\rightarrow$	$(q_2, 0.11), (q_{12}, 0.05), ...$
Travel/Category/France	$\rightarrow$	$(q_3, 0.25), (q_{25}, 0.15), ...$
...	$\rightarrow$	...

Table 2: Inverted index of feature to open questions

5.1 Matching Question and User Profiles

The core idea behind the recommendation algorithm is to return to any user a list of open questions ranked by a relevance score, which is calculated for the pair {*question profile,user profile*}. One important constraint here is to do this in an efficient manner since (1) open questions should be served immediately after having been posted, providing fresh results (this supports "competitive answering", as people prefer to answer fresh questions that have few or no answers), and (2) user profiles should be updated as soon as users post an answer (to support new users who answer for the first time, as well as versatile answerers).

To this effect, we apply an IR-like approach using a traditional vector-space model, in which the questions are seen as documents, and users as queries. To do so, we need to flatten both users and questions profiles into vectors. For question profiles, we first turn the three vectors forming the question profile into a single vector. To be consistent with our probability scheme, we multiply the probability of each feature by $\frac{1}{3}$ before storing it in the index, turning the flattened vector into a proper probability distribution over all features. We then index every question vector and build an inverted index[4], in which the key is the individual feature, namely an LDA topic, a single word from the lexical model or a category. Each key is qualified by a top category and points to a posting list of open question ids together with their adequate feature score, as illustrated in Table 2.

Next we need to turn the user profile into a single vector that can be queried against the inverted index. To do so, we traverse the user profile probability tree and consider as indexing units the individual leaves of the tree, qualified by the path that led to them (top category/model type), similar to the qualified indexed features of the question profile. We associate with each user feature a score that consists of the product of each probability score on the tree path that led to this feature.

Finally for ranking, we experimented with several measures of similarity, and chose a simple dot-product as there was no observed difference between them. We thus have at our disposal a "question retrieval engine" that takes as "query" a user vector u and returns the top ranked open questions $q1, \ldots, q_n$ that are the most relevant to the user.

5.2 Proactive Diversification

If we were to simply use our question retrieval engine for recommending questions, a user who answers mostly baseball questions and only occasionally questions about fast food would be offered only baseball questions. The reason is that most of the probability mass in his profile is centered around baseball features. Thus, baseball questions will receive a higher matching score and be ranked higher. Since baseball questions are abundant, they will populate all top ranked positions, from which recommendations are served. To compensate for the imbalance between different user interests, diversification needs to be promoted.

So far, diversification was not addressed in prior work on question recommendation, but it is an active research field in recommender systems. Typically, prior algorithms diversify recommended items as a response to a given recommendation request. Following, such methods either attempt to rerank the retrieved list of recommendations, [30, 26, 12], or apply algorithms for balancing between relevance and item similarity when constructing the recommendation list [3]. Similarly, diversification algorithms in IR attempt to rerank the already retrieved original result-set [5, 27, 6, 7].

In this paper, we propose a different proactive diversification approach, which we call *thematic sampling*. In this approach, for each user vector u, we generate N query vectors $u_1, u_2, \ldots, u_N$, each with a different constraint that imposes one specific "theme" to be represented in all retrieved questions. These N queries are submitted to the question retrieval engine, and assuming the retrieved ranked lists are disjoined enough (as each is focused on a different theme), blending them together results in a final diverse list that incorporates questions from various themes (see Section 5.3).

We consider here two types of thematic constraints. The first type is a constraint over a specific top category, retrieving only questions that are assigned to categories underneath this top category in the category hierarchy. Since each question is posted in one single top category, the returned lists are disjoint. We randomly select top categories as constraints by sampling without repetition based on their distribution in the root node of the user's probability tree. This allows for diversification while still favoring themes the user typically answers in, since their corresponding top categories will be sampled more.

The second type of constraints is in the form of a specific LDA topic that must appear in all retrieved questions[5]. Since different LDA topics typically represent different interests, and our assignment of topics to questions is sparse (see Section 3.1), the returned ranked lists should have very little overlap. We randomly sample LDA topics without repetition from the user profile by traversing the probability tree based on the distributions in the first and third levels (at the second level we always choose LDA). As in the case of top categories, this allows for diversification while still favoring themes the user typically answers in.

The more queries a system can process per recommendation request the more diverse the results will be. To this end, we found that this stage can be sped up substantially, assuming each constraint relates only to a small subset of questions. This assumption is true for LDA constraints. Following, we maintain for each LDA topic the top (highest probability) questions related to it in a cache. In our implementation, the cache for each topic holds 200 questions. When an LDA topic is sampled, only its top questions are retrieved from the cache and are then reranked based on the complete query u_i. This saves searching over hundred of thousand of questions, resulting in over an order of magnitude speedup in serving time.

[4]Since question profiles never change, they are indexed only once and no update is necessary. Resolved questions are flagged for deletion and removed from the index via a lazy deletion process.

[5]This can be viewed as a "+" operator in a free text search, forcing the query term to appear in the retrieved documents.

Top Category	Top topic words	Score
Pets	00th, march, 0th, april, date	0.03
Sports	site, search, website, google	0.11
Games and Recreation	wanna, gonna, yeah, idk, xd	0.18
Environment	term, explain, word	0.42
Society and Culture	watch, movie, tv, film	0.53

Table 3: Examples of non-thematic LDA topics

Top Category	Top topic words	Score
Dining Out	pizza, hut, domino, papa, crust	0.64
Home and Garden	bug, bed, tiny, rid, black	0.82
Science & Math	nuclear, fuel, power, energy	0.95
Arts & Humanities	slave, black, american, african	0.99

Table 4: Examples of thematic LDA topics

5.3 Recommendation Merging

Once we have obtained these N retrieved lists, as described above, we merge them in order to return one single ranked diversified list to the user. We do so by applying a generic *blending algorithm* that takes as input the N lists, each list being associated with a probability score that represents the percentage of recommendations it will contribute to the final recommendation list. One blending step is then performed by sampling an intermediate list, based on the assigned probabilities, and removing one recommendation from the sampled list to be added at the end of the final list. This step is repeated until the final recommendation list is completed.

The blending algorithm does not specify how a recommendation is chosen from each sampled list. One option is to pick the top recommendation, since the lists are ordered. However, in order to increase diversity as well as reduce "question starvation"[6], we sample a question in the list using a mixture of a geometric distribution and a uniform distribution over the ordered items in the list.

The blending algorithm can merge as many intermediate lists as the system can provide. In our implementation we chose 4 latent topic constrained lists, 4 top category constrained lists and 4 top category constrained lists that are also restricted only to questions that were posted in the last 4 hours, to ensure the retrieval of fresh questions. We experienced with different probabilities scores assigned to the lists, in order to test the importance of freshness and diversity on top of relevance (see Section 6.2).

5.4 Non-Thematic LDA Topics

In our thematic sampling algorithm, we assume that each constraint query u_i returns only questions related to a specific theme, which represents one facet of the user's interests. Top categories can indeed be easily mapped into such high level themes. However, we wanted to verify whether this assumption also holds for LDA topics, namely would they all indeed stand for one easily interpretable theme.

Some prior studies that investigated the semantics of LDA topics indeed showed that some generated LDA topics might not be coherent enough [20, 19]. Yet, we observed that there are also coherent LDA topics that still might not be of practical value when identifying user interests. Examples of such topics, as illustrated in Table 3, include style, slang and other figure-of-speech lists (third row in Table), clusters of general terms, such as dates or colors (first row), and generic activities within specific top categories (second row). We refer to these LDA topics as "non-thematic topics", and discuss next how to handle them.

[6]Question starvation refers to the situation in which a question is not recommended to any user.

(a) Probability that the LDA topic will be assigned to at least one word in a question

(b) Average LDA topic probability within the topic distribution of each question

(c) Average LDA topic probability considering only questions with at least one word assignment for this topic

(d) Histogram Entropy: the entropy of the normalized histogram of leaf category co-occurrence with the topic, constrcucted by counting the assigned categories of questions that have at least one word assigned to the LDA topic

(e) Entropy over the histogram of the target topic co-occurrence with other topics, constraining that both topics are assigned to words in the question

(f) Bin features: We bucket the topic probability for each question (feature 1) into 10 bins ($0-0.1, 0.1-0.2$ etc.), counting the number of questions related to each bin. For each bin, we generate features that capture the probability of each bin as well as the average, variance and standard deviation for each range of bins (1, [1,2], [1,2,3]... [1,..,10])

Figure 5: Thematic features generated for each topic

In our framework, we learn LDA topics for each top category in Yahoo! Answers separately, ending with $5,200$ topics. We discovered that non-thematic topics emerge quite often in all top categories. Identifying them manually would be tedious and not scalable, therefore we decided to build a classifier that would differentiate between thematic and non-thematic topics. One additional requirement was for the classifier to be effective across all top categories, even though they differ in content, corpus size, style etc. This implies that the classification features need to be generic enough to consistently capture thematic behavior across different domains. Our main intuition here is that questions are focused on a narrow theme. Thus, the distribution of LDA topics should be centered around very few topics, which represent the concrete theme of the question (see also Section 3.1). Hence, topics that typically receive few word assignments are the non-thematic ones, as they only help in explaining the style and generic scenario of the question (slang, time, place). Following the same logic, non-thematic topics should co-appear with more leaf categories and other topics than thematic topics. Based on the above intuitions, we extracted the features detailed in Figure 5 for each LDA topic from the training set.

To train the classifier, we labeled 116 topics from 23 top categories, with 13 top categories having 3 or less labeled topics in the training set. This training set consists of 34% non-thematic topics. Obviously, there is redundancy among our features, so we applied forward feature selection. We then learned a logistic regression classifier on the training set, achieving 82% accuracy on 10-fold cross-validation. Examples for topics that are not in our training set, together with their classification scores, are presented in Tables 3 and 4.

We applied this classifier to provide a "thematic" probability to all $5,200$ topics, which is then used as a bias during thematic sampling. Specifically, before sampling topics,

the user profile is temporarily altered by multiplying each LDA topic probability by its thematic probability. This process promotes thematic topics over non-thematic ones for the sampling task.

6. EXPERIMENTS

To evaluate our question recommendation system, we conducted two experiments. The first experiment is an offline experiment, in which the ground truth is provided by the past questions answered by a sample of users. The closer to these questions our recommendations are, the more effective the algorithm is. While such an offline experiment can provide insights on the differences between algorithms, it still suffers from clear limitations, as we do not know how users would have reacted, had they been exposed to other recommended questions.

As in many other Web systems, a live experiment on a large set of real users provides a more realistic setup for evaluation. Thus, as our main evaluation we conducted a live online experiment comparing the activity of users who are exposed to question recommendations via a new "recommended" tab in the Yahoo! Answers home page, as compared with those who keep the traditional view.

6.1 Offline Experiment

The goal of our offline experiment was to compare the relative performance of each of our relevance models, namely LDA, lexical and category based, as well as their combination, ignoring freshness and diversification altogether. We evaluated their associated algorithms on 8 different top categories, considering both active users and new users.

In each top category we sampled several thousand *active users*, who answered at least 21 questions as of January 2011. We derived their user profiles from the first 20 questions, and then let the algorithm being evaluated rank the "next question" (namely the 21^{st} one) among other open questions. More specifically, the algorithm ranked only the questions in the top category that were open at the time the user answered the 21^{st} question, which typically includes tens of thousands of questions. Similarly, we sampled several thousand *new users*, who answered at least two questions as of January 2011, and conducted the same ranking experiment, now with a user profile derived from only the first question the user answered, and the "next question" to be ranked being the second one.

We used as metric the percentage of users whose "next question" was ranked within the top 100 questions recommended by the algorithm. This 100 cut-off is quite large as we do not expect recall to be high, mostly because we consider many questions that the users were actually not exposed to before choosing their "next questions". Yet, this metric should provide sufficient insight to compare the relative performance of models. The results for both active and new users are shown in Table 5.

As expected, the results for the combination of models (listed in the rightmost column of the Table) systematically outperformed each independent model. In addition, the LDA model did in general better than the lexical model. Interestingly, while both the LDA and lexical models perform better for active users than for new users, the combined algorithm's performance slightly declines for active users. One possible interpretation for this is that active users hold a more diverse set of interests, and thus guessing the topic of the next question the user would answer becomes harder. This is also reflected in the category model, which is the worst performing model. One possible reason for this is the coarseness of this model, since there are numerous open questions in each category, and the model picks one randomly. For active users, who answered in several categories, the category model performance is poor, which probably influences negatively the combined model. Yet, for new users it provides a good constraint, thus significantly boosting the combination. In future work, we plan to model answering sessions for improving recommendations for users with heterogeneous interests. One possible conclusion might be to lessen the influence of the category model as the user becomes more active.

	New Users			
	LDA	Lexical	Category	Combined
Beauty and Style	6.1	6.1	0.8	**7.9**
Food and Drink	14.7	13.1	11.2	**22.3**
Health	7.5	6.5	2.5	**10.5**
Home and Garden	16.6	14.3	5.7	**23.0**
Politics	10.2	8.1	5.8	**15.0**
Pets	13.4	14.5	4.9	**19.3**
Social Science	15.1	14.3	11.2	**24.7**
Sports	28.8	18.5	19.2	**37.7**
	Active Users			
	LDA	Lexical	Category	Combined
Beauty and Style	7.6	6.5	0.1	**7.9**
Food and Drink	17.6	15.8	6.3	**20.9**
Health	7.9	7.5	0.1	**9.3**
Home and Garden	21.7	19.0	0.7	**24.2**
Politics	8.3	7.6	0.3	**9.6**
Pets	17.7	16.6	0.5	**18.7**
Social Science	20.0	17.0	0.5	**21.5**
Sports	33.7	24.6	12.4	**35.0**

Table 5: Percentage of users for whom the next question they answer appears in their top 100 recommendations in the offline experiment

Given the results of the offline experiment, when considering relevance, we will refer by default to the combined relevance model. This holds in particular for the online experiment that is described next.

6.2 Online Experiment

In our main experiment, we utilized a common inter-subject design known as a controlled experiment (a.k.a. "bucket test" or "A/B test"). Our system ran on production in the live Yahoo! Answers website for the first fifteen days of October 2012, and was exposed in different configurations to a small random sample of US-based users, who visited the site at least once. All configurations were shown under the same treatment, with a new recommended tab (as shown in Figure 2), except for the control group, which was exposed to the regular view. Each sampled user was assigned and exposed to only one of the configurations under evaluation for the duration of the experiment.

The following configurations were compared using four different *buckets* of n users each as follows:

- **Control bucket**, **CTL** ($n = 25,093$): Users in this bucket are exposed to the regular user interface without recommendations. This bucket plays the role of the control group.

- **Relevance bucket**, **R** ($n = 5,359$): Users are shown in the recommended tab recommended questions ranked only by relevance, as in the offline experiment, without any diversity or freshness considerations.

- **Freshness bucket**, **F** ($n = 46,228$): Users are shown recommendations ranked by relevance, yet with 50% of them originating only from recent questions (opened in the last 4 hours), and 20% of them selected by thematic sampling.

- **Diversity bucket**, **D** ($n = 42,041$): Recommendations are ranked by relevance with 20% of them originating from recent questions, and 50% of them selected by thematic sampling for diversity.

In addition to monitoring the main activity, namely answering, we tracked the indirect influence of question recommendations on asking and more peripheral activities such as rating or voting, as well as dwell time on Yahoo! Answers. Figure 6 shows the results in a relative scale, in which the statistics of the control bucket CTL serve as the 100% reference point and the percentage difference of each bucket as compared to the CTL performance are displayed on the graph. Answering/asking activities are measured in counts (absolute number of answers and questions normalized by the size of the bucket). Dwell time is measured as the average time in minutes from first to last click on the same day. Other peripheral activities (voting, rating, and starring) are measured as the proportion of users who participated in the activity at least once. Note that the most important metric in the figure is the number of answers (represented in the bottom bar), since our main goal is to increase the number of answers.

From these results, we see that users in the Relevance bucket **R**, with recommendations driven purely by relevance, underperformed as compared to those in the control bucket. This confirmed the major conjecture of this work, namely that in question recommendation, relevance is not enough. Our interpretation here is that users found the recommendations more annoying than useful, left the site early and were less engaged, as reflected by a decline in most metrics. Users did voice their frustration in the Yahoo! Answers blog[7] and complained about not wanting to answer four days old questions. Interestingly, they did not complain about the lack of diversity but in our opinion it is mostly because it is more difficult to perceive. It is thus expected that the response to the Freshness bucket **F** is positive, with a 4% increase in number of answers (not significant, $p > 0.05$), and a general increase in all activities by the users in the bucket. This shows that it is important for users to respond quickly to open questions, one of the reasons being the wish to be one of the first answerers.

Yet, the clear winner in this experiment is the Diversity bucket **D**. In this bucket, recency was significantly reduced in favor of diversity. While bucket users did not explicitly state their need for diversification, the results of this bucket show that they significantly prefer more diverse recommendations, not only over best matching ones, but also over recommendation of recent questions. In terms of number of answers, users in bucket **D** contribute 17% more answers on average than those in the control bucket ($p < 10^{-5}$), and a

[7] `http://yanswersblog.com/`

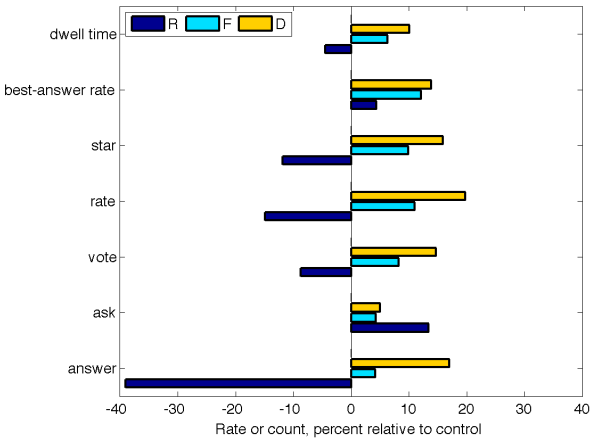

Figure 6: Comparison of user activities in all buckets in percentage, relative to CTL (the 0 median)

relative increase of 12% as compared to Freshness bucket **F** ($p < 10^{-5}$). This is a surprising result that points at the importance of diversification in question recommendation, which was unexplored so far.

Users in the Diversity bucket **D** also significantly increased their peripheral activities as compared to the control group as well as to Freshness bucket **F**, as shown by an improvement in all metrics in Figure 6. For example, users in bucket **D** increase, on average, their daily time spent on the site by 10% compared to CTL ($p < 10^{-6}$), their best-answer rate by 14% ($p < 10^{-5}$), and their rating volume (marking other answers with "thumbs-up" or "thumbs-down") by as much as 20% (not significant, $p > 0.05$). A possible interpretation for this behavior is that many of these activities are correlated and the improved experience in getting personalized questions increases satisfaction, and consequently deepens and prolongs users' engagement.

Taking a closer look at the increase in number of answers, it can be caused either by an increase in number of answers per each individual active user, or (non exclusive or) by a larger number of users contributing answers. This motivated us to go one level deeper and partition our results by the tenure of users on the site. It was shown before that the amount of activity of users on Yahoo! Answers depends on their tenure, [11]. The usual behavior is initial enthusiasm in the first few weeks, which often declines at a later stage, down to churning for a certain percentage of the population.

Figures 7 and 8 respectively show the participation rate and the average number of answers per user split by bucket and user tenure. More specifically, the x-axis refers to the tenure by listing the month on which the user joined Yahoo! Answers, with the "All" category referring to all users independently of tenure. As per the upper part of Figure 7, new users who started in October show low answering participation rates, as expected from users who are just trying out the site. Activity peaks for September users, who, with 2 to 6 weeks seniority, are in their "honeymoon" period with the site. Rates then drop off as the tenure of the user increases. This trend is consistent across all buckets. The lower part of Figure 7 indicates that, overall, the Diversity bucket encourages the highest answering participation rates compared to

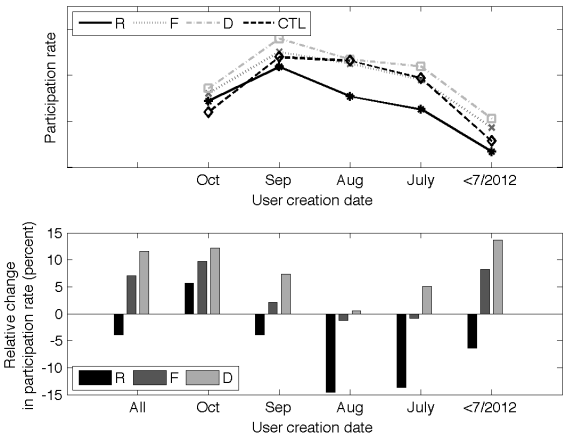

Figure 7: Answering participation rate over time

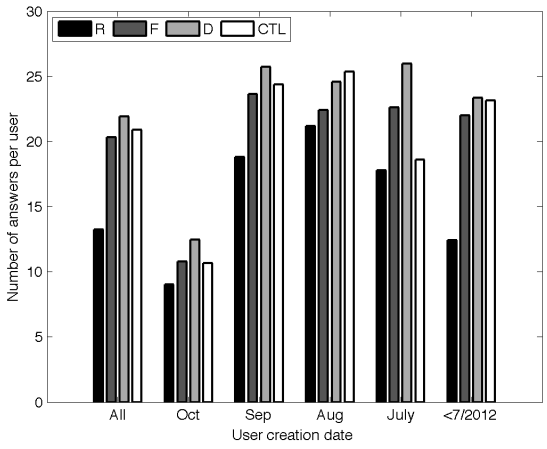

Figure 8: Average number of answers per user by user creation date

all other buckets, with a relative increase of 11% in average participation rate over the control bucket. Yet, it also boasts the highest participation rate for every user tenure group, indicating that diversity is appealing across all types of populations.

Finally, Figure 8 shows that the Diversity bucket also outperforms the other buckets in terms of the number of answers per user, both in the "All" category and at different tenure stages, with the single exception of August. In contrast the Freshness bucket performs better than the control bucket, but only in terms of answering participation rate. It fails do so for the answer rate. It might be interpreted by the fact that users might be attracted by fresh questions for a shorter period of time if they are not diverse enough, hence answering fewer questions.

In conclusion, the online experiment verifies that the Diversity bucket **D**, which promotes diversity spiced with freshness, is the winning bucket, feeding a higher number of overall answers to the site as well as a highest answering participation rate for the different types of users.

7. CONCLUSIONS

In this paper we described a question recommendation approach designed to satisfy all types of answerers in CQA systems. This approach differs from most prior work, which targeted only expert answerers, with the single objective of improving asker satisfaction. The two key novel aspects of our work consist of not being exclusively driven by relevance but also by freshness and diversity. We were motivated by the intuition that CQA systems would follow the same drives that exist in other user-generated content sites. These two additional considerations also imposed further real-time and scalability constraints on our approach, on top of those that are derived from designing a live online recommender system.

We introduced a probabilistic representation of questions and users that incorporates several relevance models of user interests at different levels of granularity. User profiles are derived from the profiles of the questions they answered and are represented as a hierarchical data structure that capture personalized preferences. Each user profile is also incrementally updated as the user keeps answering. We serve question recommendations in an efficient manner using a "question retrieval engine", which given a user profile returns the most relevant questions to answer. Incorporated in our recommendation algorithm is a novel proactive approach for promoting diversity within the returned results, which we coin *thematic sampling*, as well as tunable preferences for fresh results.

We implemented our system for Yahoo! Answers, one of the largest and earliest CQA sites. We conducted an offline experiment to verify the best relevance models for question recommendation and as expected, a combination of three relevance models achieved the highest recall for both active and new users. We then presented our main online experiment on a sample of more than a hundred thousand of Yahoo! Answers users, splitting them into four buckets.

Our online live experiment validated our intuition that relevance was not enough in question recommendation. It even surprised us, as it showed that using only relevance for ranking actually discouraged users from answering questions as compared to the control bucket. In contrast, users answered 4% more questions than the control group when fresh questions were promoted. However, it was the incorporation of diversification that was the most appealing to answerers, even at the cost of reduced freshness. Indeed, users that were shown diverse recommendations answered 17% more questions than the control bucket. Furthermore, we observed indirect benefits to overall user activity, such as an increase in voting (+20%) and longer dwelling times on the site (+10%). These results indicate the importance of integrating diversification and freshness for question recommendation in CQA sites, aspects that were ignored in prior work. The algorithm described in this paper is currently deployed in production on Yahoo! Answers.

Acknowledgments

We wish to thank Gideon Dror for his help in this research and the Yahoo! Answers team at Bangalore for their help in the evaluations.

8. REFERENCES

[1] A. Ahmed, Y. Low, M. Aly, V. Josifovski, and A. J. Smola. Scalable distributed inference of dynamic user interests for behavioral targeting. In *Proceedings of KDD*, 2011.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.

[3] R. Boim, T. Milo, and S. Novgorodov. Diversification and refinement in collaborative filtering recommender. In *Proceedings CIKM*, 2011.

[4] L. Cai, G. Zhou, K. Liu, and J. Zhao. Learning the latent topics for question retrieval in community qa. In *Proceedings of IJCNLP*, 2011.

[5] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of SIGIR*, 1998.

[6] B. Carterette and P. Chandar. Probabilistic models of ranking novel documents for faceted topic retrieval. In *Proceedings of CIKM*, 2009.

[7] V. Dang and W. B. Croft. Diversity by proportionality: an election-based approach to search result diversification. In *Proceedings of SIGIR*, 2012.

[8] A. Dong, Y. Chang, Z. Zheng, G. Mishne, J. Bai, R. Zhang, K. Buchner, C. Liao, and F. Diaz. Towards recency ranking in web search. In *Proceedings of WSDM*, 2010.

[9] A. Dong, R. Zhang, P. Kolari, J. Bai, F. Diaz, Y. Chang, Z. Zheng, and H. Zha. Time is of the essence: improving recency ranking using twitter data. In *Proceedings of WWW*, 2010.

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

[11] G. Dror, D. Pelleg, O. Rokhlenko, and I. Szpektor. Churn prediction in new users of Yahoo! answers. In *Proceedings of CQA2012 workshop*, 2012.

[12] M. Drosou and E. Pitoura. Search result diversification. *SIGMOD Rec.*, 39(1):41–47, Sept. 2010.

[13] J. Guo, S. Xu, S. Bao, and Y. Yu. Tapping on the potential of q&a community by recommending answer providers. *Human Factors*, pages 921–930, 2008.

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

[15] Y. Kabutoya, T. Iwata, H. Shiohara, and K. Fujimura. Effective question recommendation based on multiple features for question answering communities. In *Proceedings of ICWSM*, 2010.

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

[17] B. Li, I. King, and M. R. Lyu. Question routing in community question answering: putting category in its place. In *Proceedings of CIKM*, 2011.

[18] M. Liu, Y. Liu, and Q. Yang. Predicting best answerers for new questions in community question answering. In *Proceedings of WAIM*, 2010.

[19] D. Mimno, H. M. Wallach, E. Talley, M. Leenders, and A. McCallum. Optimizing semantic coherence in topic models. In *Proceedings of EMNLP*, 2011.

[20] D. Newman, J. H. Lau, K. Grieser, and T. Baldwin. Automatic evaluation of topic coherence. In *Proceedings of HLT*, 2010.

[21] M. Qu, G. Qiu, X. He, C. Zhang, H. Wu, J. Bu, and C. Chen. Probabilistic question recommendation for question answering communities. In *Proceedings of WWW*, 2009.

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

[23] F. Riahi, Z. Zolaktaf, M. Shafiei, and E. Milios. Finding expert users in community question answering. In *Proceedings of WWW companion*, 2012.

[24] A. Shtok, G. Dror, Y. Maarek, and I. Szpektor. Learning from the past: answering new questions with past answers. In *Proceedings of WWW*, 2012.

[25] L. Yao, D. Mimno, and A. McCallum. Efficient methods for topic model inference on streaming document collections. In *Proceedings of KDD*, 2009.

[26] C. Yu, L. V. S. Lakshmanan, and S. Amer-Yahia. Recommendation diversification using explanations. In *Proceedings of ICDE*, 2009.

[27] C. X. Zhai, W. W. Cohen, and J. Lafferty. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *Proceedings of SIGIR*, 2003.

[28] T. C. Zhou, M. R. Lyu, and I. King. A classification-based approach to question routing in community question answering. In *Proceedings WWW companion*, 2012.

[29] Y. Zhou, G. Cong, B. Cui, C. S. Jensen, and J. Yao. Routing questions to the right users in online communities. In *Proceedings of ICDE*, 2009.

[30] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *Proceedings of WWW*, 2005.