Copyright
by
Suratna Budalakoti
2009

**The Report Committee for Suratna Budalakoti
Certifies that this is the approved version of the following report:**

**Expertise Modeling and Recommendation in Online Question and
Answer Forums**

**APPROVED BY
SUPERVISING COMMITTEE:**

Supervisor:

K. Suzanne Barber

Aristotle Arapostathis

**Expertise Modeling and Recommendation in Online Question and
Answer Forums**

by

Suratna Budalakoti, B.E.

Report

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

December 2009

Dedication

To my parents, Shailendra and Nivedita Budhalakoti, and to my wife Sailaja.

Acknowledgements

I would like to express my appreciation for everyone who helped me in my research and academic life at the University of Texas at Austin. My advisor Dr. K. Suzanne Barber constantly encouraged me in my research while allowing the freedom to pursue the direction of my interest. Fellow members of the Laboratory for Intelligent Processes and Systems: Jaesuk Ahn, David DeAngelis, Karen Fullam, Chris Jones, Rajiv Kadaba, helped me refine and develop my ideas. Finally I would like to thank family, my parents and my wife Sailaja, for their constant support and patience.

November 30, 2009

Abstract

Expertise Modeling and Recommendation in Online Question and Answer Forums

Suratna Budalakoti, M.S.E.

The University of Texas at Austin, 2009

Supervisor: K. Suzanne Barber

Question and answer (Q&A) forums, as a way for seeking expertise on the Internet, have seen rapid growth in popularity in recent years. The expertise available on most such forums is voluntary, provided by individuals willing to invest their resources for no monetary remuneration. While these forums provide easy access to expertise, the expertise available is often lacking in quality and depth. Two major reasons for this are, the time investment required to participate in such forums, and the lack of a mechanism for identifying experts for specialized questions. We believe a Q&A recommender engine can ameliorate this problem significantly. The two primary contributions of this work are: a) a hierarchical Bayesian model based Q&A recommender, and b) a discussion of metrics to measure the performance of such a Q&A recommender. Two new metrics, responder load and questioner satisfaction, are suggested based on this discussion. These metrics are used to evaluate the performance of the recommender system on datasets harvested from the Yahoo! Answers website.

Table of Contents

List of Tables	viii
List of Figures	ix
List of Figures	ix
Chapter 1: Introduction	1
1.1 Contributions.....	1
1.2 Outline of Report	3
Chapter 2: Related Work	4
Chapter 3: Problem Description.....	6
Chapter 4: Expertise Modeling for Recommendation	7
4.1 Cosine similarity-based Approach	8
4.2 Directed Graphical Models for Topic Models	9
4.2.1 Learning parameters of the model (the EM algorithm)	12
4.3 Simple Document Clustering for Topic Models (k-means).....	13
Chapter 5: Evaluation Metrics	15
5.1 Responder Precision and Recall.....	16
5.2 Questioner Precision and Recall	17
5.3 Responder Load and Questioner Satisfaction	18
Chapter 6: Experiments.....	20
6.1 Analysis of Experiments	21
Chapter 6: Conclusions	24
Bibliography	25
Vita.....	27

List of Tables

Table 1: Metrics for Responder x	17
Table 2: Metrics for Questioner x	17
Table 3: Algorithm Accuracy Comparison: Astronomy and Space Dataset	23
Table 4: Algorithm Accuracy Comparison: Books and Authors Dataset	23
Table 5: Algorithm Accuracy Comparison: Wrestling Dataset	23

List of Figures

Figure 1: Generative Model for Q&A Forum Documents.....	10
---	----

Chapter 1: Introduction

Question and Answer (Q&A) forums have always been an important part of the World Wide Web, enabling users to seek expertise on questions where ordinary search engine results may not meet expectations. This may happen because the results are static documents not sufficiently customized to user needs, or because it might be difficult to express the question as a query string. Another common case is where users are looking for advice, or product recommendations, situations that might require various subjective tradeoffs best performed by human experts on the subject. Even in situations where Internet research might be able to answer a user question, quick access to an expert can often significantly cut down the time investment.

Recently, there has been a growing interest in such forums, both in the form of increasing participation in older forums such as Yahoo Answers, and the introduction of several more sophisticated Q&A portals such as Aardvark [Aardvark 2009] and Fluther [Fluther 2009]. This can be seen a combination of the increasing reliance on the Web in daily life, and the increasing importance of interactivity and communities in users' interaction with the Web.

However, most existing forums suffer from some serious drawbacks, particularly with respect to the quality and degree of specialization of the expertise available. Some reasons for this are: a) a significant time investment is required to participate in such forums, which deters many high quality experts, b) it is difficult for highly specialized experts to find questions that require their specific expertise, which discourages questions requiring significant skill or expertise from being asked or answered. A Q&A recommendation engine that identifies questions best suited to an expert's interest can help ameliorate these problems, both by reducing the time investment required to participate, and by enabling more specialized questions to be asked, by routing them to the appropriate experts.

1.1 CONTRIBUTIONS

There are two main contributions of this work:

1. We propose to model expertise in a Q&A system using higher-level concepts we call *expertise topics*, which are associated with distributions over words as well as experts. These distributions capture the historical word usage as well as participation patterns for users. We propose a directed hierarchical Bayesian model to discover the expertise topics, and estimate the parameters of the model using the expectation-maximization (EM) algorithm. The expertise topics are used to calculate the probability of an author having relevant expertise given a question.
2. We introduce three new metrics responder load, questioner satisfaction, and weak satisfaction, to characterize the quality of a Q&A recommendation system. These metrics are derived from precision and recall; standard metrics used in the information retrieval community. But they have been customized for the Q&A setup.

While question-answer systems already exist on the web such as Yahoo! Answers [YA 2009] and the now defunct Google Answers [GA 2006], in most current implementations, there is little or no modeling of user expertise, and hence an expert is expected to wade through many questions until finding one that is most suitable. By automating the process of finding an expert for a question, we remove the investment of time required by an expert to find a suitable question, thereby reducing the cost of participation and improve overall productivity.

An alternative to a graphical model based approach is to use a standard clustering algorithm such as the k-means algorithm, and assume the document clusters discovered correspond to topics. An even simpler alternative is to forgo the explicit definition of expertise topics. This is accomplished by associating words with authors based on historical usage and then recommending author responders based on the similarity between the words in the question and the responders' word usage. We intend to show that author-topic and word-topic models for describing Q&A behavior lead to better

responder recommendations than a traditional scheme of clustering words and authors into topics or simple author-word counts.

1.2 OUTLINE OF REPORT

This report is organized as follows: Section 2 provides an outline of the Related Work on the subject. Section 3 provides a description of the problem being approached in this report. Section 4 describes the three alternate approaches: the cosine-TFIDF approach, the graphical model approach, and the clustering-based approach, being compared in this document. Section 5 provides a discussion of suitable metrics for the problem, and defines two new metrics: responder load and questioner satisfaction. Section 6 provides the results of experiment with the three approaches described in Section 4, using the metrics described in Section 5. Section 6 provides the conclusion.

Chapter 2: Related Work

Peer production, the phenomenon where distributed users collaborate to contribute value to others without oversight or management by a business enterprise, is becoming an increasingly important part of the Internet. Contributors to peer production systems users, or agents, can be motivated by three types of rewards: monetary, intrinsic hedonic, and social-psychological [Benkler 2002]. Many successful examples of peer production, such as Wikipedia, question answer forums portals, and the series of games with a purpose [von Ahn and Dabbish 2008], forgo monetary rewards completely.

One popular example of a QA forum is Yahoo! Answers (YA). Whereas Google Answers provided a monetary incentive to answer questions, Yahoo! Answers relies on only intrinsic and social rewards. Adamic et al. have taken a close look at user behavior and content in Yahoo! Answers [Adamic, et. al. 2008]. They have characterized the distribution of question topics and demonstrated that responders who primarily respond to a small set of related topics have expertise in that topic and are therefore more likely to provide answers that are rated highly. The incentive mechanism used by YA is point-based; points are a non-monetary reward and they are assigned according to user behaviors such as asking a question, answering a question, providing an answer that is selected as the best answer. Jain et al. have performed a detailed analysis on the YA incentive mechanism and proven that the ‘best answer’ scoring rule does not always reward responders appropriately and they suggest approval voting or asker-distributed-the-points rules [Jain and Parkes 2009].

A recommendation system for Q&A forums can be seen as taking a different approach for motivating responses. Like other Q&A systems it assumes that responders gain some intrinsic or social reward for participating, but the framework is designed to lower the responders’ time investment and increase the questioners’ confidence in the provided answers. This recommender selects potential responders to answer a given question. This lowers responders’ time investment because they do not need to search through a list of questions to find one that they are willing and capable of answering. This

approach also increases questioner confidence in the responses because the recommendation algorithm is designed to identify the most appropriate experts (responders) given a question, thereby improving response quality over time.

Expertise discovery is fundamental to recommending the best responders. There are two sources information from which to model expertise: content-based and link-based [Adomavicius and Tuzhilin 2005]. Content-based expertise modeling analyzes the word usage by responders in order to build a model of the responders' expertise. This has been done in the context of clustering documents into topics by Griffiths and Steyvers [Rosen-Zvi, et. al. 2004]. Additionally, Zhang et al. have developed the QuME algorithm, which identifies expertise based on matching keywords [Zhang, et. al. 2007b]. Link-based expertise identification methods rely on evaluating the link structure between questioners and responders. Zhang et al. tested a number of network based ranking algorithms on data from the Java Forum, showing that link information can be used to identify expertise nearly as well as human raters [Zhang, et. al. 2007]. Jurczyk and Agichtein applied a variation of the HITS algorithm to the much broader Yahoo! Answers data set and were able to identify authoritative users, or expert responders [Jurczyk and Agichtein 2007]. The generative model approached used here provides a method to simultaneously leverage both content and link information in order to best identify expertise in a Q&A forum.

Chapter 3: Problem Description

The Q&A recommendation problem we propose can be described as follows: given a question by a user on a Q&A forum, identify users on the forum that are most capable of providing a satisfactory answer to the question. Previous historical information about interactions on the Q&A forum is available to train the system.

These users could then be contacted in some form, such as email, with the suggestion that the question might be of interest to them. The advantages of such a system are twofold: a) increase questioner satisfaction by providing them with prompter and more satisfactory answers, b) reducing the effort required on part of responders to find questions that match their interest and expertise. This might help increase participation on a forum. In the long run, such a system could also improve the average quality of responders by reducing the cost of participation in terms of time, and allowing users with highly specialized expertise to participate.

More specifically, we assume that the training data is available to us in the form of individual question-answers (QAs), and for each QA, we have the following information: a) a unique questioner id, b) text of the question, c) ids of all responders, d) the text of each responder's response, and e) some information indicating which answers were found satisfactory by the questioner. Then, in the test step, we are provided with the identity of the questioner, and the text of the question. The task is to suggest suitable responders for the question. Then the actual responders are revealed, along with the text of their answers.

In the case of a live test, it is possible to directly contact the suggested responders to judge their interest in the question. However, we currently test our system using historical data derived from Q&A forums. In this case, we evaluate the quality of our recommendations by how many of our recommended responders had originally answered the question.

Chapter 4: Expertise Modeling for Recommendation

Most approaches to the recommendation problem can be divided into two categories: a) content-based filtering, and b) collaborative filtering. In content-based filtering, users are modeled based on information associated with what they have been interested in, or liked, in the past. In collaborative filtering, new recommendations are made to a user based on the interests of other users identified as most similar to them. In the Q&A forum setup, we have, for each responder, information in the form of text of the questions he/she chose to answer, as well as the text used by him/her to answer these questions. We also have collaborative information about the other responders that chose to answer the same question as the responder, as well as the questioner.

A simple content-based approach to the expert recommendation problem would be to build a text-based profile, for example a term frequency-inverse document frequency (TFIDF) profile, for each responder. Then, when the system receives a new question, the question text could be compared to all the responder profiles using a similarity measure such as the cosine score, and the responders with most similar profiles could be recommended the question. This is a common approach with respect to expertise modeling and recommendation, taken by Zhang [Zhang, et. al., 2007] and Godil [Godil 2006].

However this approach, which treats the expertise identification problem as a document retrieval problem, suffers from a serious drawback: an expert is very different from a document in the sense the exact words used by a responder are heavily contingent on the questions he/she chose to answer, and do not cover all the information that a responder has, or the topic he/she may be knowledgeable about. For this reason a simple text profile based approach is not sufficient for the purpose of modeling human expertise.

To overcome this drawback, we introduce an alternate approach to expertise modeling, which models user expertise in terms of topics, instead of words. A topic can be seen as a higher-level concept over words, and is modeled as a distribution over words. Hence, two questions may belong to the same topic even though they may have

no words in common. Similarly, an expert may be recommended a question even though there is no match in terms of profile words, if the question is judged as belonging to a topic the expert is interested in. Two words that are synonymous will have similar distributions over topics for two reasons. First, they are likely to co-occur in the same document (question and corresponding answers). Secondly, they are likely to co-occur with many of the same words.

Section 4.1 below discusses in detail the cosine similarity based approach to expertise modeling. In Section 4.2, we discuss the Hierarchical Bayesian model for expertise identification.

4.1 COSINE SIMILARITY-BASED APPROACH

Cosine similarity is a popular similarity measure in the information retrieval community. It is based on a vector space model [Salton, et. al. 1975] of documents, where each document is modeled as a vector in an n -dimensional vector space, where n is the number of words (or terms) in the dataset. The magnitude of the document vector d in the k^{th} dimension depends on the occurrence count of the k^{th} word in the document d (where the words are numbered arbitrarily).

The magnitude of the vector in the dimension corresponding to any term/word ‘ k ’ is often calculated as the Term Frequency-Inverse Document Frequency score (TFIDF score). The TFIDF score of a term k for a document d , written as s_{kd} , is calculated as:

$$s_{kd} = \text{TF}_{kd} \cdot \text{IDF}_k$$

where TF_{kd} is the number of times the term k occurs in document d , and IDF is defined as follows:

$$\text{IDF}_k = \log\left(\frac{N_D}{N(k \in D)}\right)$$

Here, N_D is the total number of documents in the dataset, and $N(k \in D)$ is the number of documents that contain at least one occurrence of the term k . Then, a document d in the dataset can be represented by a vector $\mathbf{d} = [s_{1d}, s_{2d}, s_{3d}, \dots, s_{kd}]^T$. Then

gives two documents $\mathbf{d}_1$ and $\mathbf{d}_2$, the cosine similarity between the two documents is calculated as:

$$\text{sim}(\mathbf{d}_1, \mathbf{d}_2) = \mathbf{d}_1 \cdot \mathbf{d}_2 / (\|\mathbf{d}_1\| \|\mathbf{d}_2\|)$$

Here $\mathbf{d}_1 \cdot \mathbf{d}_2$ represents the dot product of the two document vectors, and $\|\mathbf{d}\|$ represents the magnitude of the vector $\mathbf{d}$. The cosine similarity, hence, gives the cosine of the angle between the two vectors. If the angle is small, this implies that documents are close to each other, and the cosine value will be large.

The TFIDF-based cosine similarity measure has been used for the task of expertise identification [Zhang, et. al., 2007] [Godil 2006]. This is usually done by treating each responder as a document in a document retrieval setup, with all the words ever used by the responder as the text of the document.

4.2 DIRECTED GRAPHICAL MODELS FOR TOPIC MODELS

Directed graphical models [Bishop 2007], also known as *Bayesian networks*, provide a tool to express the factorization properties of a joint probability distribution. In other words, graphical models enable us to formalize the independence assumptions we have with respect to random variables in a given dataset. These independence assumptions can be expressed in terms of a *generative model*, and represented visually as a directed acyclic graph (Figure 1), where an edge between two random variables represents a conditional dependence. A generative model represents an abstract model for how the data in a dataset was generated. The aim of a generative model is to represent the causal relationships between the various random variables.

A basic outline of a generative model for any online forum where people might gather for a discussion, or to exchange information, can be constructed as follows: at each timestep, a topic is generated from a distribution over topics, which might have its own prior distribution. Then, some (question) words are generated related to the topic. The topic distribution may or may not be independent of the original author of the post, depending on how closely people stick to their topic of interest. Following this, a set of responders is chosen from a distribution, based on the topic, and each of these responders

and the users $1, \dots, s$, arbitrarily. At each timestep, a topic t is generated from a multinomial distribution τ over topics, and a vector $\mathbf{w}^q = w_1 \dots w_p$ is generated, where w_i is the count of word labeled i in the generated words. The words are generated from ϕ_t , a multinomial distribution over words corresponding to topic t . Following this a responder vector $\mathbf{x} = x_1 \dots x_s$ is generated from θ_t , a multinomial distribution over users for topic t , where x_i is the number of time the user labeled i responded. Each of the users in $\mathbf{x}$ in turn generates words based on the topic t . Here, we make the important simplifying assumption that the words generated by a responder as part of the answer are drawn from the same distribution ϕ_t as the topic.

This assumption can be understood as saying that the words used in the answer to a question by a responder depend only on the topic of the question, and do not depend on any attributes of the responder. We believe this to be a reasonable assumption in Q&A forums where factual information is exchanged for the most part, or even in forums where personal opinions are expressed but the vocabulary used does not differ very much from user to user. It may not hold true in forums such as blogs or discussion forums, where responses to topics are much longer and more personal, and people may have favorite topics they might try to steer the topic toward. However, we believe that that level of model complexity is not required for Q&A forums.

Figure 1 displays the generative model described above in plate notation. The shaded variables are the observed variables, while the non-shaded variables are the hidden variables. Also, α , β and γ are symmetric Dirichlet priors, used for smoothing. We set $\alpha = \beta = \gamma = 1$. We also make the simplifying assumption that the total number of words and users generated for each question is independent of τ , θ and ϕ , and hence their randomness can be ignored in our discussion. Also, since we assume that the words generated by the responders depend solely on the topic, we can write the words generated by all responders as a vector $\mathbf{w}^r = w_1 \dots w_p$. We write $\mathbf{w} = \mathbf{w}^q + \mathbf{w}^r$.

4.2.1 LEARNING PARAMETERS OF THE MODEL (THE EM ALGORITHM)

It is easy to see from Figure 1 that the generative model is essentially a mixture model with a finite number of components, where the number of components is the number of topics we expect to see in the dataset. Let the number of such components/topics be g . Let the total number of unique Q&A interactions in the dataset D be n , where the j^{th} such interaction is referred to as d_j . Then, we assume, there is associated with each d_j a hidden vector z_j of length g , where $z_{ij} = 1$ if d_j is about topic i . We write $\boldsymbol{\theta} = \theta_1, \dots, \theta_g$, $\boldsymbol{\phi} = \phi_1, \dots, \phi_g$, and $\boldsymbol{\varepsilon} = (\boldsymbol{\theta}, \boldsymbol{\phi})$. Then, assuming we know z_j for all d_j , we can write the log likelihood of $\boldsymbol{\xi}$ given D as:

$$\log_D L(\boldsymbol{\varepsilon}) = \sum_{i=1}^g \sum_{j=1}^n z_{ij} \{ \log \tau_i + \log P(w_j | \theta_i) + \log(P(x_j | \phi_i)) \}$$

We use the expectation maximization (EM) algorithm [Dempster, et. al. 1977] to estimate the hidden variables z_i , and the parameters τ and ε . The derivation of the EM algorithm for the model is fairly straightforward. The E-step and the M-step equations are given below:

E-Step:

Given a guess for τ and ε , the expected value of z_{ij} is given by:

$$z_{ij} = \frac{\tau_i P(w_j, x_j | \varepsilon_i)}{\sum_{h=1}^g \tau_h P(w_j, x_j | \varepsilon_h)}$$

M-Step:

Given expected values of z_j , we can estimate τ , θ and ϕ as follows:

$$\tau_i = \sum_{j=1}^n \frac{\Psi + z_{ij}}{\Psi |T| + n}$$

$$\theta_{ik} = \frac{\alpha + \sum_{j=1}^n z_{ij} w_{jk}}{\alpha s + \sum_{j=1}^n \sum_{k'=1}^s z_{ij} x_{jk'}}$$

$$\phi_{ik} = \frac{\beta + \sum_{j=1}^n z_{ij} w_{jk}}{\beta p + \sum_{j=1}^n \sum_{k'=1}^p z_{ij} w_{jk'}}$$

The EM algorithm is guaranteed to converge to a local minimum. Once the algorithm converges, for each topic, the probability that a user will answer a question from the topic is given by ϕ .

4.3 SIMPLE DOCUMENT CLUSTERING FOR TOPIC MODELS (K-MEANS)

A document clustering approach to topic modeling can be seen as a simpler less expensive alternative to graphical model based approaches. The idea here is to use a standard document-clustering algorithm such as the k-means [Bishop 2007] algorithm to cluster Q&A interaction documents into clusters. Each such cluster can be treated as a single topic. This approach differs from the graphical model based approach most significantly in that it ignores the user interaction information while forming topic clusters, and depends wholly on word occurrence information. The k-means algorithm seems a natural choice for the clustering algorithm, as it is fast, scalable, and naturally combines with the TFIDF-based vector space model of document.

Given a set of M documents embedded in an n -dimensional vector space, as described in Section 4.1, and a preset number of clusters k , the algorithm consists of the following steps:

1. Select k documents at random as cluster centroids.
2. Do till convergence:

- a. Compare each document to the k cluster centroids, assigning each document to the cluster corresponding to the centroid with the highest cosine similarity.
- b. For each of the k clusters, recalculate the centroid as follows: for each dimension, set the magnitude of the centroid vector as the mean of the magnitudes of all the documents in the cluster, along the given dimension.

The k -means algorithm is guaranteed to converge to a local minimum. Once the algorithm converges, we can estimate the probability that a responder will answer a question in a topic in a given topic (cluster) as the number of times the responder answered a question belonging to be topic, divided by the number of questions in the topic. These probabilities are used to predict the users for a given question, during the test phase.

Chapter 5: Evaluation Metrics

Any evaluation of Q&A recommender systems needs to take into account the experience of users from both perspectives: as questioners and as responders. As questioners, they would like highly satisfactory responses. As responders, they would like not to be overloaded with too many questions, or be recommended questions they are not interested in. In particular, if they are high quality experts, there is a risk that they might be recommended too many questions in a bid to provide satisfactory responses to questioners, which might result in reduced participation from them.

We introduce two new metrics, *responder load* and *questioner satisfaction*, to measure the quality of Q&A recommenders from both of these perspectives. Both of these metrics are variations of *precision* and *recall*, metrics commonly used in information retrieval [Singhal 2001] and recommender system research [Sarwar, et. al., 2000].

It is possible to define precision and recall for a user in a Q&A system from two perspectives: as a questioner and as a responder. So we can define four metrics of interest in total. However, the action governing the quality of all four metrics is the same: each time a new question is introduced in the system, the recommender makes a decision to contact a subset of responders in the system and recommend the question to them. The decision to answer the question is made by each individual responder and cannot be controlled by the recommender. So the decision made by the recommender to recommend a question has to take into account both the possible impact on questioner metrics as well as responder metrics.

5.1 RESPONDER PRECISION AND RECALL

Let all the participants in the QA forum be represented by X . Any $x \in X$ can be a questioner or a responder/answerer. Let the user x 's responder precision be written as Π_x^a , and responder recall as ρ_x^a .

Then,

$$\Pi_x^a = \frac{R_x^+}{R_x^+ + N_x^+}$$

$$\rho_x^a = \frac{R_x^+}{R_x^+ + R_x^-}$$

Here, the right-hand-side terms are as defined in Table 1. Table 1 can be understood as follows: R_x is the number of questions that the responder x liked, i.e., responded to. N_x is the number of questions the user x did not like, or in other words did not respond to. A superscript '+' refers to questions that were suggested to the user by the recommender system, while a subscript '-' refers to questions that were not recommended to the user. Thus, for example, R_x^+ is a count of the number of questions that are both liked by responder x , and recommended to x . Similarly, N_x^+ is the number of questions that were recommended to user x , but not answered, while R_x^- is the number of questions that were recommended to x , but not answered.

For a given responder, responder precision is the ratio of the number of questions that were recommended to a responder, that the responder answered, to the total number of questions recommended to the responder. We believe this to be an important measure of the quality of the recommender, as a recommender that recommends too many irrelevant questions will drive away responders.

	Liked question / Answered question	Did Not Like / Did not Answer
Recommended	R_x^+	N_x^+
Not Recommended	R_x^-	N_x^-

Table 1: Metrics for Responder x

	Liked question / Answered question	Did Not Like / Did not Answer
Recommended	U_x^+	I_x^+
Not Recommended	U_x^-	I_x^-

Table 2: Metrics for Questioner x

The recall for a responder measures how many of the questions the responder answered were recommended by the system. It is a measure of how well the recommender covers all the interests of the questioner, and while still important, we consider it relatively less significant.

5.2 QUESTIONER PRECISION AND RECALL

Similarly, let the user x's questioner precision be written as Π_x^q , and questioner recall as ρ_x^q .

Then,

$$\Pi_x^q = \frac{U_x^+}{U_x^+ + I_x^+}$$

$$\rho_x^q = \frac{U_x^+}{U_x^+ + U_x^-}$$

Here the right-hand-terms are as defined in Table 2.

Questioner precision measures how many of the responders recommended by the system provided satisfactory answers. This is also relatively unimportant: a user will not usually mind extra answers so long as he/she is receiving a sufficient number of answers that are satisfactory. There may be problems in extreme cases, such as when a particular user is spammed, but this problem might be handled in other ways, such as allowing questioners to ban specific responders from their questions, or rank answers based on responder quality, or responder history.

Questioner recall measures how many of the answers/responders of interest to the questioner, the recommender was able to identify in advance. This metric is more important than questioner precision, as it indicates the degree to which questioners can depend on the recommender to find the most suited responders for a question.

We focus on Responder Precision and Questioner Recall when testing our recommender system. In the next section, we describe a small variation of Responder Precision, and rename Questioner Recall. These re-descriptions, we believe, make these metrics more intuitive to understand in terms of recommender system behavior.

5.3 RESPONDER LOAD AND QUESTIONER SATISFACTION

We define *Responder Load* λ_x for user x as the inverse of responder precision. However, to ensure a finite value, we perform Laplace smoothing on the data, that is, we assume that each user has received at least one question he/she responded to. Thus, the responder load λ_x is defined as follows:

$$\lambda_x^a = \frac{R_x^+ + N_x^+ + 1}{R_x^+ + 1}$$

The responder load is thus, the number of questions the responder would have to read through on average to find one of interest to him/her. The higher the value of λ_x , the greater the number of questions a responder has to read through to find questions of interest to him/her. In that sense, it is a measure of the load on the responder.

We define *Questioner Satisfaction* as $\sigma_x = \rho_x^q$, or:

$$\sigma_x = \frac{U_x^+}{U_x^+ + U_x^-}$$

σ_x measures what fraction of the answers found satisfactory by a questioner were from responders contacted by the recommender. It can be seen as a measure of how satisfied questioners will be with the Q&A system if the responders relied entirely on the recommender to provide them with interesting questions.

The overall quality of a QA recommender could be measured as $\sum_{x \in X} w_x \lambda_x$, and $\sum_{x \in X} w_x \sigma_x$. The value of w_x could be set based on some criteria, or set as $1/|X|$ to get the average values. In the experiments presented here, there is a tradeoff between λ and σ . For example, by recommending all responders in the system, we can move the satisfaction σ_x to 1 for all questioners. However, this will have an adverse impact on the questioner load, as most of the questions we suggest to a responder, he/she will not find interesting. Hence, an important challenge for a Q&A recommender system will be to provide quality answers to questioners, without overloading the better experts among responders, and to manage this tradeoff in a reasonable way.

Chapter 6: Experiments

We have performed experiments to demonstrate a working implementation of our recommender based QA framework. 10,000 questions and their answers from each of three different subject categories in Yahoo! Answers have been crawled, parsed, processed. These categories are Astronomy and Space, Books and Authors, and Wrestling. The collected data represents approximately two months of activity within each category. These categories were chosen to represent a broad spectrum of the type of content available on Yahoo! Answers. After the pages were stripped of html, we applied stopping [Fox 1989] and stemming [Porter 1980] algorithms to remove suffixes and unimportant words.

The experiments we performed compare the effectiveness of three different methods of identifying and modeling expertise. The first method uses a basic information retrieval (IR) algorithm that calculates the cosine similarity between the words used by an author in historical QA data and the words contained in a question. This is included as a baseline measurement. The second method uses k-means clustering [Hartigan 1975] to group question-answer documents into clusters. Users are then assigned a probability weight in each cluster based on the fraction of questions answered in the cluster. Given a new question, users are recommended based on the normalized weighted sum of the similarity of the question to each cluster centroid, multiplied by the probability of response of each user in the cluster.

The expertise models are built on a training set of the first of the two months of questions and answers. The remaining month of questions and answers form the test set. For each question in the test set, we recommend the top 15 responders most likely to respond, as predicted by each of the three algorithms. The recommender performance is analyzed using our metrics of responder load and questioner satisfaction. We introduce one more simple metric called *weak satisfaction* that indicates whether the recommender was able to select at least one of authors who actually responded.

We make a simplifying assumption that all actual responses are satisfactory. Yahoo! Answers collects best-answer information that indicates the single most satisfactory answer. However, the unpredictability of the dataset makes recommending the single most satisfactory response difficult, and also, marking a single response as satisfactory does not imply that all other responses were unsatisfactory. Table 3 displays the experimental parameters and results of our experiment to examine the impact of expertise modeling on recommendation.

6.1 ANALYSIS OF EXPERIMENTS

Yahoo! Answers is a noisy and unpredictable dataset. Over a training set of 5000 questions representing one month of activity in one category (Space and Astronomy), 11588 unique users participated. Of these users, 4890 responded to a question only once, and 4723 never responded to a single question, but only asked questions. In addition, many users leave the system after a short period of time.

Tables 3,4 and 5 contain the accuracy results for the three datasets: Astronomy and Space (S&A), Books and Authors (B&A), and Wrestling (WR). The results show that the ‘k-means clustering’ and ‘EM algorithm’ based approaches outperform the basic information retrieval algorithm according to questioner satisfaction in nearly every case. This evidence supports the use of a topic-based approach for Q&A recommendation, and demonstrates that a word-matching based approach is not sufficient. As can be seen from the tables, topic-based model outperform word-matching approaches by a huge margin. The chief reason for this is that word matching approaches model expertise only at the level of keywords, and have no higher-level concepts such as topics. As a result of this dependence on exact word matches between question words and user profiles, they are much more likely to miss potential matches than topic based approaches. In topic-based approaches an expert can be identified as suitable for a question even if he/she has not used any words in the question, as long as he/she is active in the topic the question is identified as belonging to.

Another way of looking at this is that topic-based approaches are able to use word co-occurrence information that word matching algorithms are unable to. As a

hypothetical example, suppose our topic learning algorithm identifies a ‘Shakespeare’ topic that contains high probabilities of keywords Hamlet and Macbeth. In this case, a question about Macbeth could be directed to an expert on Hamlet, simply because keywords Hamlet and Macbeth belong to the same topic, and there is a fair chance that an expert on Hamlet also knows something about Macbeth. Human expertise is ‘deep’ in this way that documents are not: a document about Hamlet may contain little or no information about Macbeth. That is why information retrieval approaches, tailored to document modeling, do not work well for modeling humans.

Also, for the S&A dataset, our system was able to correctly recommend at least one responder more than half the time (weak questioner satisfaction) while maintaining a load of $\lambda < 21$. This means that of every 20 recommended questions to a user, on average a user responded to one. This might be a tolerable level of load for an expert even for a live system.

The results are less promising the B&A and WR dataset. There are two likely reasons for this: a) as pointed out by [Adamic, et. al. 2008], the S&A group is much more expertise-oriented, while the B&A and WR groups function more as discussion groups, and there tend to be user-based cliques, rather than topic-based cliques, and b) topic trends move faster in these groups, making an offline learner that does not update along with current trends unlikely to do well. Explanation a) is supported by the fact that the EM algorithm based approach, which takes into account user interaction information as well as word usage data, outperforms the word-based k-means algorithm by a much larger margin for the B&A and WR dataset, than it does for the S&A dataset.

Given the bulletin board structure of YA, it is certain that responders rarely see every available question. In a live recommender test, we can expect a much higher percentage of responses from recommended responders because we can assume the responder sees the question and knows it has been recommended based on his/her expertise. Also, with a live test it would be possible to measure the questioner's satisfaction with any given response, leading to a more accurate measure of the recommender performance regarding questioner satisfaction.

Model	Astronomy and Space Dataset		
	Resp. Load	Ques. Satisfaction	Weak Satisfaction
Cosine-TFIDF	22.41	11.87%	35.70%
K means Algorithm	20.37	26.82%	62.60%
EM Algorithm	20.67	33.90%	65.67%

Table 3: Algorithm Accuracy Comparison: Astronomy and Space Dataset

Model	Books and Authors Dataset		
	Resp. Load	Ques. Satisfaction	Weak Satisfaction
Cosine-TFIDF	70.92	6.86%	11.81%
K means Algorithm	62.86	9.82%	14.24%
EM Algorithm	49.36	13.36%	19.51%

Table 4: Algorithm Accuracy Comparison: Books and Authors Dataset

Model	Wrestling Dataset		
	Resp. Load	Ques. Satisfaction	Weak Satisfaction
Cosine-TFIDF	72.81	4.72%	9.72%
K means Algorithm	66.40	7.82%	11.05%
EM Algorithm	51.58	12.39%	18.24%

Table 5: Algorithm Accuracy Comparison: Wrestling Dataset

Chapter 6: Conclusions

Question and Answer forums provide an important service on the Internet, by providing a service complementary to search engines, answering queries that are best answered by humans. They have been growing in importance in recent years, as Internet users grow in sophistication, and come to rely more and more on the Internet. However, most of these forums use relatively primitive technology, relying on experts to identify questions of interest to them. We hypothesize that this reduces the quality of interactions on these forums, and a Q&A recommender system that tries to match experts with questions best suited to them, will help improve both the quality and the level of specialization for questions on such forums.

The two primary contributions of this work are: a) a novel graphical model based approach for making recommendations in a Q&A forum and, b) the design of two new metrics for characterizing the performance of such a recommender. These metrics: responder load and questioner satisfaction, are used to evaluate the performance of our recommender system on data harvested from Yahoo! Answers. Three methods of constructing expertise models are compared: a simple TFIDF-cosine similarity based approach, a word clustering based topic discovery approach, and a hierarchical Bayesian model based approach. Experiments across several topic domains demonstrate our system's ability to predict responder identities and suggest new responders.

Bibliography

- [Adamic, et. al. 2008] L.A. Adamic, J. Zhang, E. Bakshy, and M.S. Ackerman. (2008) “Knowledge sharing and yahoo answers: everyone knows something”, WWW '08: Proceeding of the 17th international conference on World Wide Web, Beijing, China, pp. 665—674.
- [Adomavicius and Tuzhilin 2005] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: a survey of the state-of-the-art and extensions,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, no. 6, pp. 734–749, June 2005.
- [Benkler 2002] Y. Benkler, “Coase's Penguin, or, Linux and the Nature of the Firm”, *Yale Law Journal*, vol. 112, no. 3, pp. 367–445, 2002.
- [Bennett and Lanning 2007] J. Bennett and S. Lanning, “The netflix prize,” in *Proceedings of KDD Cup and Workshop*, vol. 2007, 2007.
- [Dempster et. al. 1977] A. Dempster, N. Laird, and D. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society, Series B (Methodological)*, pp. 1–38, 1977.
- [Godil 2006] H. Godil, *Finding Experts by Modeling Domain Expertise*. University of Texas at Austin, 2006.
- [Hartigan 1975] J. Hartigan, *Clustering algorithms*. John Wiley & Sons, Inc. New York, NY, USA, 1975.
- [Jain and Parkes 2009] S. Jain and D. Parkes, “Designing Incentives for Online Question and Answer Forums,” in to appear in *Proceedings of the 10th ACM conference on Electronic commerce*, 2009.
- [Jurczyk and Agichtein 2007] P. Jurczyk and E. Agichtein, “Discovering authorities in question answer communities by using link analysis,” in *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. New York, NY, USA: ACM, 2007, pp. 919–922.
- [Pouwelse et. al. 2008] J. Pouwelse, P. Garbacki, D. Epema, and H. Sips, “Pirates and Samaritans: A decade of measurements on peer production and their implications for net neutrality and copyright,” *Telecommunications Policy*, vol. 32, no. 11, pp. 701–712, 2008.

- [Rosen-Zvi, et. al. 2004] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth, “The author-topic model for authors and documents,” in AUAI ’04: Proceedings of the 20th conference on Uncertainty in artificial intelligence. Arlington, Virginia, United States: AUAI Press, 2004, pp. 487–494.
- [Sarwar, et. al. 2000] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Analysis of recommendation algorithms for e-commerce,” in EC ’00: Proceedings of the 2nd ACM conference on Electronic commerce. New York, NY, USA: ACM, 2000, pp. 158–167.
- [Ahn and Dabbish 2008] L. von Ahn and L. Dabbish, “Designing games with a purpose,” Commun. ACM, vol. 51, no. 8, pp. 58–67, 2008.
- [Singhal 2001] A. Singhal, “Modern information retrieval: A brief overview,” IEEE Data Engineering Bulletin, vol. 24, no. 4, pp. 35–43, 2001.
- [Zhang, et. al. 2007] J. Zhang, M. S. Ackerman, and L. Adamic, “Expertise networks in online communities: structure and algorithms,” in WWW ’07: Proceedings of the 16th international conference on World Wide Web. New York, NY, USA: ACM, 2007, pp. 221–230.
- [Zhang, et. al. 2007b] J. Zhang, M. S. Ackerman, L. Adamic, and K. K. Nam, “QuMe: a mechanism to support expertise finding in online help-seeking communities”, in UIST ’07: Proceedings of the 20th annual ACM symposium on User interface software and technology. New York, NY, USA: ACM, 2007, pp. 111–114.
- [Salton, et. al. 1975] G. Salton , A. Wong , C. S. Yang, “A vector space model for automatic indexing”, Communications of the ACM, v.18 n.11, p.613-620, Nov. 1975.
- [Bishop 2007] C. Bishop, “Chapter 8: Graphical Models”, Pattern Recognition and Machine Learning, Springer 2007, pp. 359-422.
- [Porter 1980] M.F. Porter, “An Algorithm for Suffix Stripping”, Program 14(3) 1980.
- [Fox 1989] C. Fox, “A Stop List for General Text”, ACM SIGIR Forum 24, 1-2 (Sep. 1989), 19-21
- [Aardvark 2009] Aardvark, www.vark.com, 2009.
- [Fluther 2009] Fluther, www.fluther.com, 2009.
- [YA 2009] Yahoo! Answers, <http://answers.yahoo.com>, 2009.
- [GA 2006] Google Answers, <http://answers.google.com>, 2006.

Vita

Suratna Budalakoti was born in Gorakhpur, India, on 15 September 1979, to his parents Shailendra Budhalakoti and Nivedita Budhlakoti. He completed his secondary education at Army School, Ambala Cantt., India, in 1997. In 2001, Suratna earned the degree of Bachelor of Engineering in Computer Science at the National Institute of Technology, Rourkela, India. He joined the CS graduate program at the University of California Santa Cruz in 2004. He transferred to the Ph.D. program in Electrical and Computer Engineering at the University of Texas at Austin in 2006, where he is currently a member of the Laboratory for Intelligent Processes and Systems.

Permanent address: C-1038, Indiranagar, Lucknow, Uttar Pradesh, India, 226016.

This report was typed by the author.