DOI: 10.21917/ijsc.2011.0024

IMPROVISATION OF SEEKER SATISFACTION IN YAHOO! COMMUNITY QUESTION ANSWERING PORTAL

K. Latha[1] and R. Rajaram[2]

[1]*Department of Computer Science and Engineering, Anna University of Technology, Tiruchirappalli, India*
E-mail: erklatha@gmail.com
[2]*Department of Information Technology, Thiagarajar College of Engineering, Tamil Nadu, India*
E-mail: rrajaram@tce.edu

Abstract
One popular Community question answering (CQA) site, Yahoo! Answers, had attracted 120 million users worldwide, and had 400 million answers to questions available. A typical characteristic of such sites is that they allow anyone to post or answer any questions on any subject. Question Answering Community has emerged as popular, and often effective, means of information seeking on the web. By posting questions, for other participants to answer, information seekers can obtain specific answers to their questions. However, CQA is not always effective: in some cases, a user may obtain a perfect answer within minutes, and in others it may require hours and sometimes days until a satisfactory answer is contributed. We investigate the problem of predicting information seeker satisfaction in yahoo collaborative question answering communities, where we attempt to predict whether a question author will be satisfied with the answers submitted by the community participants. Our experimental results, obtained from a large scale evaluation over thousands of real questions and user ratings, demonstrate the feasibility of modeling and predicting asker satisfaction. We complement our results with a thorough investigation of the interactions and information seeking patterns in question answering communities that correlate with information seeker satisfaction. We also explore automatic ranking, creating abstract from retrieved answers, and history updation, which aims to provide users with what they want or need without explicitly ask them for user satisfaction. Our system could be useful for a variety of applications, such as answer selection, user feedback analysis, and ranking.

Keywords:
Social Media, Community Question Answering, Information Seeker Satisfaction, Ranking, History Updation

1. INTRODUCTION

Community Question Answering (CQA) [15] emerged as a popular alternative to finding information online. It has attracted millions of users who post millions of questions and hundreds of millions of answers, producing a huge knowledge repository of all kinds of topics, so many potential applications can be possibly made on top of it. For example, automatic question answering systems, which try to find the information to questions directly, instead of giving a list of related documents, might use CQA [15] repositories as a useful information source. In addition, instead of using general-purpose web search engines, information seekers now have an option to post their questions (often complex [17] and specific) on Community QA sites such as Naver or Yahoo! Answers [17], and have their questions answered by other users. These sites are growing rapidly. Also, Wiki Answers is a website that is an ad-supported website where knowledge is shared freely in the form of questions and answers (Q&A). Anyone can ask a question and anyone from anywhere in the world can answer it. This sharing of knowledge in turn becomes part of a permanent information resource. WikiAnswers.com leverages wiki technology and fundamentals, allowing communal ownership and editing of content. Each question has a "living" answer, which is edited and improved over time by the WikiAnswers.com community. WikiAnswers.com uses a System – where every answer can have dozens of different Questions that "trigger" it. However, it is not clear what information needs these CQA [15] portals serve, and how these communities are evolving. Understanding the reason for the growth, the characteristics of the information needs that are met by such communities, and the benefits and drawbacks of community QA over other means of finding information, are all crucial questions for understanding this phenomenon. As we will show, human assessors feel difficult in predicting [1] asker satisfaction, thereby requiring novel prediction techniques [16] and evaluation methodology that we begin to develop in this paper.

Not surprisingly, user's previous interactions such as questions asked and ratings submitted are a significant factor for predicting satisfaction. We hypothesized that asker's satisfaction with contributed answers is largely determined by the asker expectations, prior knowledge and previous experience which are used to update the taste of the asker (History updation) and the forth coming answers are given based on the past history (taste) and is not available in any of the CQA [15] portals. We report on our exploration of how to improve satisfaction prediction [16] that is, to attempt to predict whether a *specific* information seeker will be satisfied with any of the contributed answers. Based on the time spent by the asker in the particular session and askers voting, we can predict whether the asker is satisfied or not for a given question. If he is not satisfied, not voted within a span of time or may not have the prior knowledge (Background knowledge) about the answers, then our System can automatically rank the results with the help of ranking functions and assigns rank to the answers. Most of the askers may get irritated because of the more number of answers for a question and also go through only the first two or three answers for a given question. In this situation our Abstract Generation System can generate the gist (most important sentences) from all the answers in the asker's point of view.

2. LIFE CYCLE OF A QUESTION IN CQA

The process of posting and obtaining answers to a question is an important phenomenon in CQA [14]. A user posts a question by selecting a category, and then enters the question subject (title) and, optionally, details (description). For conciseness, QA will refer to this user as the asker for the context of the question, even though the same user is likely to also answer other questions or participate in other roles for other questions. Note that to prevent abuse, the community rules typically forbid the

asker from answering own questions or vote on answers. After a short delay (which may include checking for abuse, and other processing) the question appears in the respective category list of open questions, normally listed from the most recent down.

At the point, other users can answer the question, vote on other users' answers, or comment on the question (e.g., to ask for clarification or provide other, non-answer feedback), or provide various meta-data for the question. At that point, the question is considered as closed by the asker, and no new answers are accepted.

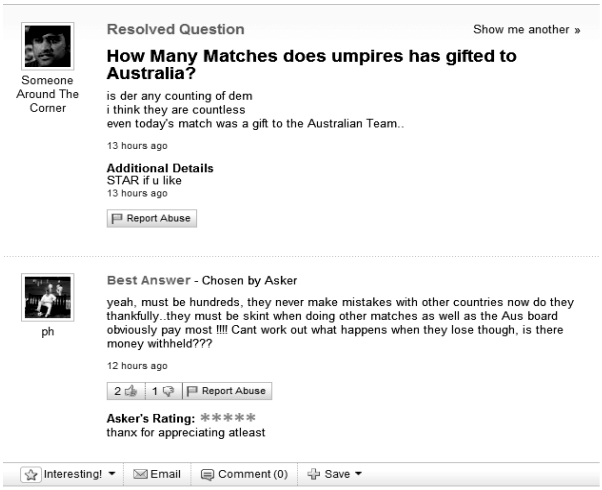


Fig.1. Example of "satisfied" question thread

QA believe that in such cases, the asker is likely satisfied with at least one of the responses, usually the one he chooses as the best answer.

But in many cases the asker never closes the answer personally, and instead, after some fixed period of time, the question is closed automatically. The QA community has "failed" to provide satisfactory answers in a timely manner and "lost" the asker's interest. Question Answering communities are an important application by itself, and also provide unprecedented opportunity to study feedback from the asker. Furthermore, asker satisfaction plays crucial role in the growth or decay of a question answering community.

If the asker is satisfied with any of the answers, he can choose it as *best*, and provide feedback ranging from assigning *stars* or rating for the best answer, and possibly textual feedback. QA believe that in such cases, the asker is likely *satisfied* with at least one of the responses, usually the one he/she chooses as the best answer. An example of such "satisfactory" interaction is shown in Fig.1. If many of the askers in CQA are not satisfied with their experience, they will not post new questions and will rely on other means of finding information which creates asker satisfaction problems.

3. THE ASKER SATISFACTION PROBLEM

While the true reasons are not known, for simplicity, to contrast with the "satisfied" outcome above, we consider this outcome to be "unsatisfied."An example of such interaction is shown in Fig.2.

Fig.2. Example of "Unsatisfied" question thread

4. PROBLEM DEFINITION

We do not attempt yet to analyze the distinction between possibly satisfied and completely unsatisfied, or otherwise dissect the case where the asker is not satisfied. We now state our problem formally into four different angles.

4.1 ANSWER JUSTIFY PROBLEM

The asker may receive more number of answers for each question. Now the asker intended to read all answers and select one suitable answer for his question. Here the problem is, the asker may not know that which answer he has to choose?

"To overcome this problem we explore Automatic Ranking system to provide Rank for answers".

4.2 ANSWER UNDERSTANDING PROBLEM

How the asker can identify the objective of each answer?

"To avoid this problem Abstract generation providing a brief summary of answers and is often used to help the reader quickly ascertain the answer's purpose. When used, an abstract always appears at the beginning of all displayed answers, acting as the point-of-entry".

4.3 ASKER TASTE CHANGES

One important problem is to determine what an asker wants? What form of answer he expects?. It is crucial to determine what the user thinks in his mind?

"History Updation using distributed learning automata is a best solution to this problem. It is used to remember the information about the previous behavior of the asker who has selected answer in the past history and in order to show relevant answers from the learned behavior and it is updated in the asker's history".

4.4 TIME CONSUMING PROBLEM

To read all retrieved answers, the asker needs more time. Is the time factor affects the asker satisfaction?

"The time duration is computed by how long the asker viewing the displayed answers, and is used for predicting whether the asker is satisfied or unsatisfied".

5. METHODOLOGIES

5.1 AUTOMATIC RANKING BASED ON GENERALIZATION METHOD

The objective of applying learned association rules [10], [9] is to improve QA comparison by providing a more generalized representation. Good generalization [22] will have the desired effect of bringing QA that are semantically related closer to each other that previously would have been incorrectly treated as being further apart. Association rules [10] are able to capture implicit relationships that exist between features of QA. When these rules are applied they have the effect of squashing these features, which can be viewed as feature generalization.

Initially the most important features are extracted using Markov Random Field (MRF) [21] model. These features are used as the initial seeds for generalization [22]. Then association rule [10], [9] induction is employed to capture feature co-occurrence patterns.

It generates rules of the form H→ B, where the body B is a feature from answers, and the head H is a feature from a question. This means that rules can be used to predict the presence of the head feature given that all the features in the questions are present in the answer. This means that a rule satisfying the body, when the head feature is absent will not be considered.

The idea of feature generalization [22] and combining this with feature selection to form structured representation for ranking .Feature generalization [22] helps tone down ambiguities that exist in free text by capturing semantic relationships and incorporating these in the query representation. This enables a much better comparison of features in QA.

An interesting observation is that with feature selection and generalization a more effective ranking is achieved even with a relatively small set of features. Finally the retrieved features are used for ranking answers. This is attractive because smaller vocabularies can effectively be used to build concise indices that are understandable and easier to interpret.

5.2 ABSTRACT GENERATION

With the rapid growth of online information, there is a growing need for tools that help in finding, filtering and managing the high dimensional data. Automated text categorization is a supervised learning task, defined as assigning category labels to answers based on likelihood suggested by a training set of answers.

Real-world applications of text categorization often require a system to deal with tens of thousands of categories defined over a large taxonomy. Since building these text classifiers by hand is time consuming and costly, automated text categorization has gained importance over the years.

We have developed an automatic abstract generation [2] system for answers based on rhetorical structure extraction. The system first extracts the rhetorical structure, the compound of the rhetorical relations between sentences in answers, and then cuts out less important parts in the extracted structure to generate an abstract [2] of the desired length.

Abstract generation is, like Machine Translation, one of the ultimate goals of Natural Language Processing. This is realized as a suitable application of the extracted rhetorical structure. In this paper we describe the abstract generation system based on it.

5.3 RHETORICAL STRUTURE (RS)

Rhetorical structure represents relations between various chunks of answers in the body of each question. The rhetorical structure is represented in terms of connective expressions and its relations. There are forty categories of rhetorical relations which are exemplified in Table 1.

Table.1. Example of rhetorical relations

Relations	Expressions
Confident <co>	I can
Example <eg>	For example
Recommend <rd>	Try…….this
Reason <re>	Because
Assumption <as>	I think
Plus <pl>	And
Specialization <sp>	Almost, most, always
Serial <sr>	Thus
Summarization <su>	After all, finally
Extension <ex>	This is, there
Suggestion <sg>	You can
Experience <ep>	I use, my experience, i used
Explanation <en>	So
Advice <ad>	You need, you would
Capture <ca>	Take
Appreciate <ap>	Good question
Next <ne>	Then
Simple <si>	Just, easy
Rare <ra>	Some time
Condition <cn>	If you
Negative <po>	But, i don't, not sure
Must <mu>	You should
Expectation <en>	Hope this….
Trust <tr>	I believe
Starting <st>	First of all
Doubt <dt>	May be
Accurate <ac>	Yes, no
Positive <po>	Why not?
Request <rq>	Please
Repeat <rt>	Again
Utilize <ut>	Use this
Direction <di>	Here is
While <wi>	Since
Memorize <me>	Remember
Question <qu>	Can you, are you
Same <sa>	Sounds like
Opinion <op>	Statement
Verify <ve>	Ask
Apology <ay>	Sorry, excuse.

wishes<wi>	All the best, welcome, best wishes, good luck

The rhetorical relation of a sentence, which is the relationship to the preceding part of the text, can be extracted in accordance with the connective expression in the sentence.

The rhetorical structure represents logical relations between sentences or block of sentences of each answer. Linguistic clues, such as connectives, anaphoric expressions, and idiomatic expressions in the answers are used to determine the relationship between the sentences In the sentence evaluation stage, the system calculates the importance of each sentence in the original text based on the relative importance of rhetorical relations. They are categorized into three types as shown in Table.2. For the relations categorized into Right Nucleus, the right node is more important, from the point of view of abstract generation [2], than the left node. In the case of the Left Nucleus relations, the situation is vice versa. And both nodes of the Both- Nucleus relations are equivalent in their importance. A sample Question & answer is considered, the rhetorical structure is built and shown in Fig.3.

Table.2. Relative importance of rhetorical relations

Relation type	Relation	Important Node
Right nucleus	Experience, negative, example, serial, direction, confident, specialization	Right node
Left nucleus	Especially, reason, accurate, appropriate, simple, rare, assumption, explanation, doubt, request ,apology, Utilize, opinion	Left node
Both nucleus	Plus, extension, question, capture, appreciate, next, repeat, many, condition, since, ask, same, starting, wishes, memorize, trust, positive, recommend, expectation, advice, Summarization,	Both node

5.4 HISTORY UPDATION BY USING LEARNING AUTOMATA (LA)

Based on asker's past history (already selected answer for his previous question) the taste of the asker can be updated and we can predict what kind of answer, the asker will choose for his current question.

Learning automata [10] are adaptive decision-making devices operating on unknown random environments. The automata [4] approach to learning involves the determination of an optimal action from a set of allowable actions. An automaton can be regarded as an abstract object which has finite number of possible actions. This action is applied to a random environment and is used by automata [4] in further action selection. By continuing this process, the automata learn to select an action with best grade. The learning algorithm [10] used by automata to determine the selection of next action from the response of the environment.

The proposed algorithm takes advantage of usage data and link information to recommend answers to the asker based on learned pattern. For that, it uses the rewarding and penalizing schema of actions which updates the actions probabilities in each step based on a learning algorithm. The rewarding factor for history updation is presented in equation

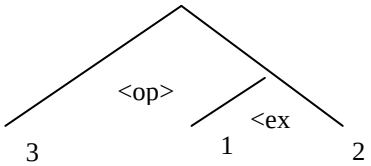
$$a = \omega + \lambda \quad (1)$$

where ω is a constant & λ is obtained by this intuition. If a user goes from taste i to taste j & there is no link between these tastes, then the value of λ is set to constant value; otherwise it is set to zero.

Question: does McDonald's veg burger in India contain egg?

Answer 1:Nope,In India its purely veg, I had taken one of my close associate who is purely veg and I discussed it with the Delhi shop and the manager confirmed and even wanted to give in writing. Made Indian food is my FAVVVV. I would be all over the street eating all the home cooked food out there I live USA and there's mD's on every block.

Thus the Rhetorical structure for answer 1 can be represented by a binary tree



This structure can also be represented as follows,

[[1<ex>2] <op> 3]]

Answer 2: No, way it's a guaranteed company <**co**>

Answer 3: I think yes. But you can ask the manager of McDonald's .Good Luck.

[[1<ad>2] <wi> 3]]

Finally the abstract from all the answers will be,

"I had taken one of my close associate who is purely veg and I discussed it with the Delhi shop and the manager confirmed and even wanted to give in writing- No, way it's a guaranteed company- you can ask the manager of McDonald's."

Fig.3. Abstract generation using rhetorical structure

If there is a cycle in users' navigation path, the actions in the cycle indicate the change of taste of the asker over a period of time or the dissatisfaction of asker from the previous tastes must

be penalized. The penalization increases with the cycle length. So, the parameter b which is penalization factor is calculated from the following equation

$$b = (Steps\ in\ cycle\ containing\ k\ and\ l) * \beta \quad (2)$$

where, β is a constant factor. The penalization factor has direct relation with the length of cycle traversed by the asker.

These navigational patterns are then used to generate recommendations based on the asker's current status. The answers in a recommendation list are presented according to their importance and similarities, which is in turn computed based on usage information.

5.5 DURATION

Time spent by the asker for viewing a page which contain answers as very important pieces of information in measuring the asker's interest on the page, and is defined in equation

$$\text{Duration} = \frac{\text{Total duration of the page}}{\text{number of answers}} \quad (3)$$

Based on the time spent by the asker in the particular page and number of received answers the time is calculated for each answer taken by the asker, here we can predict whether the asker is satisfied or not for a given question.

There may be many reasons why the asker never closed a question by choosing a best answer and closing a question with voting. Based on our exploration we believe that the main reasons are either

a) Closing a question within a minimum span of time and may not have interest in voting.
b) Closing a Question within a minimum span of time with voting
c) Never Closing a Question because the asker loses interest in the information
d) Never Closing a Question because none of the answers are satisfactory

In Option a) the true reasons are not known for closing a Question without voting. He might have read the best answer in the answer collection but not having interest in voting. In this juncture the time duration is calculated and based on this the automatic ranking is decided. So the Answers for Questions which are not voted can also be rated using our automated Ranking function which will be helpful for the forth coming askers.

6. EXPERIMENTAL SETUP

We present the experimental evaluation of our asker satisfaction phenomenon over the Yahoo! Answers [17]. We have addressed the concrete areas of question answering community portals by automatic ranking, history updation, abstract generation, and duration based problems. These areas tend to have significant interest among the askers and it was shown that our methodologies are outperforming, predicting [1] and presenting best results to the asker's point of view. Also our method solutions are evaluated by human and system judgement called Kappa Score [20] which is efficient in providing the correct score toward the relevancy of the answers.

We describe the baselines and our specific methods for predicting asker satisfaction. In other words, our "truth" labels are based on the rating subsequently given to the best answer by the asker himself. It is usually more valuable to correctly predict whether a user is satisfied (e.g., to notify a user of success). This section describes the experimental setting, datasets, and metrics used for producing our results in Section 7.

6.1 EVALUATION METRICS

We use three variants of standard Information Retrieval metrics such as Precision, Recall and F-Measure to examine the effectiveness of Yahoo! Answers [17] for answering questions: In our experiments the metrics are computed using relevance judgements given by the user and the system. In our automatic ranking system the results are computed by evaluating the answers for each question thread in decreasing order (top ranked answers for the question). This models a "naive" searcher that examines results in order. To determine whether the results given by the system, are producing sufficient information for a human to consistently gain knowledge from the answers according to our goal framework, a specialized score called Kappa [20] is used.

Precision: The fraction of the predicted *satisfied* asker information needs that were indeed rated satisfactory by the asker .And can also be defined as the fraction of the retrieved answers which is relevant. Precision at K for a given query is the mean fraction of relevant answers ranked in the top K results; the higher the precision, the better the performance is. The score is Zero if none of the top K results contain a relevant answer. The problem of this metric is, the position of relevant answers within the top *K* is irrelevant, while it measures overall use potential satisfaction with the top *K* results. We use the "best answer" tagged by the Yahoo! Answers [17] web site as the ground truth.

Recall*:* The fraction of all rated *satisfied* questions that were correctly identified by the system. And can also be defined as the fraction of the relevant answers which has been retrieved. This is used to separate high-quality content from the rest of the contents and evaluates the quality of the overall answer set. If more answers are retrieved, recall increases while precision usually decreases. Then, a proper evaluation has to produce precision/recall values at given points in the rank. This provides an incremental view of the retrieval performance measures. The answer set is analyzed from the top answers and the precision-recall values are computed when we find each relevant answer.

F measure: The weighted harmonic mean of precision and recall, the traditional F-measure or balanced F-score is:

$$F = \frac{2.\text{Pr}ecision.\text{Re}\,call}{(\text{Pr}ecision + \text{Re}\,call)} \quad (4)$$

This is also known as the F_1 measure, because recall and precision are evenly weighted. The general formula for non-negative real β is:

$$F = \frac{(1+\beta^2).(\text{Pr}ecision.\text{Re}\,call)}{(\beta^2.\text{Pr}ecision + \text{Re}\,call)} \quad (5)$$

Two other commonly used F measures are the F2 measure, which weights recall twice as much as precision, and the F0.5 measure, which weights precision twice as much as recall. Fβ

"measures the effectiveness of retrieval with respect to a user who attaches β times as much importance to recall as precision". It is based on van Rijsbergen's effectiveness measure E=1− (1/ (α/P+ (1−α)/R)). Their relationship is Fβ = 1 − E where α = 1 / (β2 + 1).

6.2 DATASETS

The data for this study comes from the resolved questions of Yahoo! QA service log, having different requirements on the questions associated with "games," "Food and Drinks", "Education & Reference," "computer & internet", "travel", "social culture", "family" and the "news and events". We have created a pool of 3568 QA Pairs drawn over 50 categories are considered as training data set among 5000 queries. The Question in QA pool is associated with minimum of 5 answers and maximum of 20 answers. In order for large-scale evaluation of interactive question answering to be practical, user–system interactions in Yahoo QA community are encapsulated in HTML pages called interaction forms— similar to clarification forms which focused on arbitrarily interface controls, that could appear on an HTML form—thumbs up, thumbs down, report abuse, Sliding bar, Stars for interestingness and comments .

6.3 METHODS COMPARED

In this section we describe the study of ranking the answers, beginning with details of ranking algorithms.

6.3.1 Vector Space Model (VSM):

VSM is an algebraic model for representing answers (and any objects, in general) as vectors [18] of identifiers, such as, for example, index terms. It is used in information filtering, information retrieval, indexing and relevancy rankings.

Questions and answers are represented as vectors [18].

$$a_j = (w_{1,j}, w_{2,j}, \ldots, w_{t,j})$$
$$q = (w_{1,q}, w_{2,q}, \ldots, w_{t,q})$$

Each dimension corresponds to a separate term. If a term occurs in the answer, its value in the vector is non-zero. Several different ways of computing these values, also known as (term) weights, have been developed. One of the best known schemes is tf-iaf [11] weighting Vector [18] operations can be used to compare answers with queries.

Relevancy rankings of answers in a keyword search can be calculated, using the assumptions of answer similarity theory [23]) [19], by comparing the deviation of angles between each answer vector and the original query vector where the query is represented as same kind of vector as the answers. Using cosine similarity [19], [23] between answer a and query *q* can be calculated using,

$$sim(a_j, q) = \frac{a_j . q}{|| a_j || || q ||} = \frac{\sum_{i=1}^{t} w_{i,j} * w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} * \sqrt{\sum_{i=1}^{t} w_{i,q}^2}} \quad (6)$$

where a_j is the j^{th} answer for the query q. $w_{i,j}$ is the weight of the i^{th} term in the answer j. and $w_{i,q}$ is the weight of the i^{th} term in the query q. A cosine value is zero if the question and answer are orthogonal and have no match (i.e. the question term does not exist in the answer being considered).

6.3.2. Indri Method:

Returns a ranked list of answers containing the important term and its term frequency.

$$H(p,q) = H(p) + D_{KL}(p \,||\, q) \quad (7)$$

Here H(p) is a entropy and Indri[7] handles ranking via KL – divergence / cross entropy for each answer.

$$H(P) = \sum_{i-1}^{n} p(x) \log p(x) \quad (8)$$

$$D_{KL}(p \,||\, q) = \sum_{i} p(x) \log \frac{p(x)}{q(x)} \quad (9)$$

where, P(x) is the probability of selecting an answer for the given query and q is the collection of answers. The lower the KL-Divergence value, the more similar are two distributions *P* and *Q*.

6.3.3. Lucene Ranking:

In Lucene Ranking algorithm**,** we found that the participants of QA Community benefited from a search experience where good answers were called out and bad ones were downplayed or filtered out. And we managed to achieve this with absolute threshold through careful normalization [6] (of a much more complex scoring mechanism). The sole purpose of the normalization is to set the score of the highest-scoring result. Once this score is set, all the other scores are determined since the ratios of their scores to that of the top-scoring result do not change. But this normalization [6] would not change the ranking order or the ratios among scores in a single result set from what they are now. It also uses term frequency and inverse answer frequency to calculate the score for each answer. The scores are intrinsically between 0 and 1. The top score will always be 1.0 assuming that the entire query phrase matches (while the other results have arbitrary fractional scores based on the tf iaf ratios) with the answers. Top score would be 1.0 or 0.5 depending on whether one or two terms were matched. We obtain the rank of each answers using,

$$score\_a = sum\_t((tf\_q * iaf\_t)/norm\_q) * ((tf\_a * iaf\_t)/norm\_a\_t)) \quad (10)$$

where, score_a : score for answer a
sum_t : sum for all terms t in answer
tf_q : the square root of the frequency of t in the question
tf_a : the square root of the frequency of t
iaf_t : log (numans/ansFreq_t+1) + 1.0
numans : number of answers in index
ansFreq_t : number of answers containing t
norm_q : sqrt(sum_t((tf_q*iaf_t)^2))
norm_a_t : square root of number of terms in a in the same field as t.

6.3.4. Mutual Information (MI):

Mutual information is a quantity that measures the mutual dependence of two terms in the question for the given answers.

Formally the mutual information [5] of two terms X and Y can be defined as:

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \frac{p(x,y)}{p_1(x) p_2(y)} \quad (11)$$

where X and Y are the selected terms from question .p(x, y) is the joint probability distribution of X and Y, and p(x), P(y) are the marginal Probability distribution of X and Y respectively.

Instinctively, mutual Information [5] measures the information that *X* and *Y* share. It measures how much, knowing one of these variables reduces our uncertainty about the other. For example, if *X* and *Y* are independent, then knowing *X* does not give any information about *Y* and vice versa, so their mutual information is zero. At the other extreme, if *X* and *Y* are identical then all information conveyed by *X* is shared with *Y*, knowing *X* determines the value of *Y* and vice versa. As a result, in the case of identity the mutual information is the same as the uncertainty contained in *Y* (or *X*) alone, namely the entropy of *Y* (or *X*: clearly if *X* and *Y* are identical they have equal entropy). Mutual information [5] quantifies the dependence between the joint distribution of *X* and *Y* and what the joint distribution would be if *X* and *Y* were independent. It is a measure of dependence in the following sense: $I(X; Y) = 0$ if and only if *X* and *Y* are independent random variables, then $p(x,y) = p(x)\, p(y)$, which is described by

$$\log \frac{p(x,y)}{p(x)p(y)} = \log 1 = 0 \quad (12)$$

Moreover, mutual information is nonnegative (i.e. $I(X;Y) \geq 0$) and symmetric (i.e). $I(X;Y) = I(Y;X)$).

6.3.5. Weight Calculation Method (tf*iaf):

The tf–iaf [11] weight (term frequency–inverse answer frequency) is a statistical measure used to evaluate how important a word is to an answer in a collection of answers. The importance increases proportionally to the number of times a word appears in the answer but is offset by the frequency of the word in the collections. One of the simplest ranking functions is computed by summing the tf-iaf for each query term; many more sophisticated ranking functions are variants of this simple model.

$$Weight(W) = tf * iaf \quad (13)$$

The term frequency (tf) in the given answer is simply the number of times a given term appears in that answer. This frequency is usually normalized to prevent a bias towards longer answers (which may have a higher term count regardless of the actual importance of that term in the answer) to give a measure of the importance of the term t_i within the particular answer a_j. Thus we have the term frequency as,

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad (14)$$

where $n_{i,j}$ is the number of occurrences of the considered term(t_i) in answer a_j, and the denominator is the number of occurrences of all terms in answer a_j. The inverse answer frequency is a measure of the general importance of the term (obtained by dividing the total number of answers by the number of answers containing the term, and then taking the logarithm of that quotient).

$$iaf = \log \frac{|a|}{|\{a : t_i \in a\}|} \quad (15)$$

|a| is the total number answers in the corpus and {a: $t_i \in a$} is a number of answers where the term t_i appears (that is $n_{i,j} \neq 0$). If the term is not in the corpus, this will lead to a division-by-zero. It is therefore common to use 1+| {a: ti ϵ a}|.

A high weight in tf–iaf [11] is reached by a high term frequency (in the given answer) and a low answer frequency of the term in the whole collection of answers; the weights hence tend t to filter out common terms. The tf-iaf value for a term will always be greater than or equal to zero.

6.3.6. Markov Random Field (MRF):

MRF ranks the answer in response to a query that focuses on textual features [13] defined over query/answer pairs. Thus, the input is a query/answer pair and the output is a real value. The MRF [21] model generalizes various dependence models and is defined by

$$P(A/Q) = \sum_{c \in C(G)} \lambda_c f(c) \quad (16)$$

where P(A/Q) is the probability of choosing the answer A for the given query Q. λ_c is iaf (inverse answer frequency), and $f_c(c)$ is a feature value[13]from answers calculated using BM 25.

Okapi **BM25** is a ranking function used by MRF to rank answers according to their relevance to a given search question.BM 25 is a bag of words that ranks a set of answers based on the query terms appearing in each answer, regardless of the inter-relationship between the query terms within a answer. It is not a single function, but actually a whole family of scoring functions, with slightly different components and parameters. One of the most prominent instantiations of the function is as follows,

Given a query Q, containing keywords $q_1, \ldots, q_n$, the BM25 score of an answer is:

$$f_{T,}(q_i, a) = \frac{(k_1+1) tf_{w,a}}{k_1(1-b) + b \frac{|a|}{|a|_{avg}} + tf_{w,a}} \log \frac{N - af_w + 0.5}{af_w + 0.5} \quad (17)$$

where $f(q_i,a)$ is q_i is a term frequency in the answer a, $|a|$ is the length of the answer a in words, (tf_w,a) is the number of times the term w matches in answer a and $|a|_{avg}$ is the average answer length in words. Here k_1 and b are free parameters, usually chosen as $k_1 = 2.0$ and $b = 0.75$, N is the total number of answers and af_w is the total number of answers that have at least one match for the term w.

7. EMPIRICAL RESULTS

In this paper, we focus our analysis on askers' satisfaction prediction [16] in CQA. The number of newly posted questions and answers over a period remains steady but satisfaction level varies inherently with respect to the mentality of the asker. If the askers are continuously posting questions, but not selecting answers that introduces a complicated situation for the forth coming users to select answers without any background knowledge. Our experiment result shows that the level of asker

satisfaction is excellent for our proposed method than the traditional methods. This implies that instead of just posting the questions, we satisfy and encourage the Yahoo! participants to select best answer (highly correlate with asker's question) for his question.

This Section shows the results of satisfaction prediction level by our proposed methods. From the collected Yahoo! Answers [17] snapshots, 70% of data is considered as a training set and the rest for testing.

Table.3. Precision recall F measure for ranking algorithms, abstract generation and History updation

Type	Method	Precision	Recall	F-measure
Ranking Method	Generalization	0.9223	0.8730	0.8972
	VSM	0.851	0.771	0.809
	Indri	0.8029	0.7134	0.756
	MRF	0.889	0.800	0.842
	Weight	0.8432	0.749	0.793
	Lucene	0.8376	0.7615	0.798
	MI	0.9201	0.8630	0.891
Abstract Generation	RS	0.9056	0.7813	0.844
History Updation	LA	0.9178	0.824	0.869

In this paper Automatic ranking, abstract generation and history updation are the contributions to the CQA [15] which is not available in any of the CQA portals and we have proved that the highest precision and recall levels are attained with above contributions and the results are reported in Table.3.

The first set of experiments investigates the answers obtained from different ranking algorithms. Algorithms that use a bag of words approach such as Vector Space Model [18], Lucene and Weight calculation are producing fair results compared to others. Interestingly our proposed method called "generalization" produces higher precision of 0.9223 than other ranking methods. Adding the best feature selection method (BM 25) with MRF slightly improves the performance and generates the good precision of 0.889. Also an algorithm (Indri [7] method) generates less precision value of 0.8029 that uses cross entropy. We observe that, performance of ranking algorithms is very similar to each other.

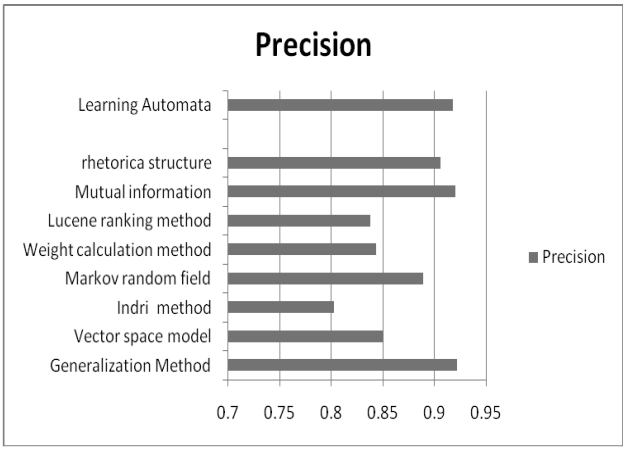


Fig.4. Satisfaction prediction accuracy for methods

Fig.4. demonstrates the satisfaction prediction accuracy for various methodologies and highlight the importance of generalization method [22] for ranking answers to improvise the performance of Yahoo! Answers community.

We have established promising preliminary results on asker satisfaction even with relatively simple models. An Algorithm with tf and iaf (vector Space Model [18], Weight calculation Method, and Lucene algorithm) achieves moreover same precision value**.**

Human judgment often has wide variance on what is considered a "good" summary [13], which means that making the evaluation process automatic, is particularly difficult. Manual evaluation can be used, but this is both time and labor intensive as it requires humans to read not only the summaries [13] but also the source answers.

The metric used here is Kappa score [20] in which our abstract generation [2] system submits the results to the human experts and it is evaluated by them. Our system generates summaries [13] automatically and compared with the human generated summaries [3]. It is proved that there is a high overlap between the two summaries indicate that a high level of shared concepts between them.

The generated abstracts were evaluated from the point of view of key sentence coverage. In Table.4, Samples of 15 questions are selected from Food & Drink, Sports and Home &Garden categories which present short answers of 6 or 7 sentences. Seven subjects judged the key sentences and four judged the most important key sentence of each answer. As for the Questions 9 & 10, the average correspondence rates of the key sentence and the most important key sentence among the subjects are 86% and 100% respectively.

The key sentence coverage increases with the abstract [2] word count (WC). The reason is the less word count answers contain only less rhetorical expressions. That is they provide less linguistic clues and the system cannot extract the rhetorical structure exactly. The average length ratio (abstract/original) is reduced to 36.2 % (Question 5) to make the length of the abstract shorter.

7.1 HUMAN JUDGEMENT

To complement the asker ratings the human judgements are obtained from users of Yahoo! Answers [17]. Here Cohen's kappa score [20] is used to evaluate human judgement.

7.2 KAPPA SCORE (K)

Cohen's kappa measures [20] the agreement between the two raters who each classify answers into two mutually exclusive categories (satisfied and unsatisfied).Kappa score is defined by

$$k = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)} \qquad (18)$$

where Pr(a) is the relative observed agreement among raters, and Pr(e) is the hypothetical probability of chance agreement, using the observed data to calculate the probabilities of each observer randomly saying each category.

Table.4. Key sentence coverage of the abstract

Mate -rial	No of Ans	Word count (WC)	Abstr - act WC	Len ratio	Cover ratio	
					Key sen	Most impt Sen
Q 1	5	19	8	0.421	0.429	0.75
Q 2	7	23	11	0.478	0.429	0.50
Q 3	6	29	13	0.448	0.571	0.75
Q 4	11	37	14	0.378	0.571	0.75
Q 5	8	58	21	0.362	0.714	0.75
Q 6	14	60	29	0.483	0.714	1
Q 7	13	67	32	0.478	0.857	1
Q 8	16	74	35	0.473	0.714	0.75
Q 9	10	85	41	0.482	0.857	1
Q 10	14	93	43	0.462	0.857	1
Q 11	17	102	49	0.480	0.714	1
Q 12	19	114	52	0.456	0.571	0.75
Q 13	20	138	54	0.391	0.714	0.75
Q 14	18	152	70	0.461	0.857	1
Q 15	21	161	73	0.453	0.857	1

If the raters are in complete agreement then $\kappa = 1$. If there is no agreement among the raters (other than what would be expected by chance) the score is ≤ 0. Kappa score [20] for various methodologies are shown in Table.5 & 6. Surprisingly our proposed methods abstract generation, history updation and automatic ranking using generalization [22] are highly correlated but not exceeding with the human judgments.

Table.5. Human judgments for ranking methods

Method	**Ranking algorithms**					
	VSM	**MRF**	**Lucene**	**Weight**	**Indri**	**MI**
Kappa	0.8743	0.8853	0.8413	0.8659	0.8292	0.9049

Table.6. Human judgments for proposed methods

Method	**Generalizatio n method**	**Abstract generation**	**History updation**
Kappa Score	0.9267	0.9218	0.9289

Because it is very difficult to predict what the user exactly thinks in his mind and also the taste of the human continuously changes based on the environmental factors.

8. BACKGROUND WORK AND RELATED WORK

Community Question Answering is rapidly growing popularity. However, the quality of answers, and the user satisfaction with the CQA [15] experience, varies greatly which has recently become a viable method for seeking information online.

Question answering over community QA archives is different from traditional TREC QA, and applying QA techniques over the web. The most significant difference is that traditional QA operates over a large collection of documents (and/or web pages) whereas we are attempting to retrieve answers from a social media archive with a large amount of associated user generated metadata. This metadata (such as explicit user feedback on answer quality) is crucial due to the large disparity of the answer quality, as any user is free to contribute his or her answer for any question.

The previous research results in this area can help filter low-quality content from CQA archives. Jeon et al tried to estimate CQA [15] answer quality. They used 13 non-textual features and trained a maximum entropy model to predict answer quality. Their results showed that retrieval relevance is significantly improved when answer quality or question utility is integrated in a log likelihood retrieval model. Later, Agichtein et al. explored a larger range of features including both structural, textual, and community features. He has proposed the identification of question quality as well as answer quality. In addition to those above, Song et al. has proposed a measure called 'question utility' used to evaluate question quality. Question utility can be estimated by either a language model based method or a LexRank based method.

Unlike in question answering, the goal is not to develop a better algorithm for retrieving and extracting answers, but instead to enable the exchange of high-quality, relevant information between community participants. Finding such quality information, in QA communities varies significantly which provides a unique challenge, which recently has been addressed.

Zhao et al. (2007) proposed to utilize "user click logs from the Encarta web site to identify question paraphrases". Jeon et al. (2005) employ a related method, in that they identify similar answers in the Naver Question and Answer database to retrieve semantically similar questions, while Jijkoun and deRijke (2005) include the answer in the retrieval process to return a ranked list of QA pairs in response to a user's question.

Lytinen and Tomuro (2002) suggest yet another feature to identify question paraphrases, namely question type similarity [19], which consists in determining a question's category in order to match questions only if they belong to the same category.

Other previous work on CQA[15] can be categorized into three major areas:(1) how to mine questions and answers and how to find related questions given a new question, (2) how to find experts given a community network and (3) how to predict users' satisfaction.

While automatic complex QA [17] has been an active area of research, ranging from simple modification to factoid QA technique [Soricut and Brill 2004] to knowledge intensive approaches for specific domains [Demner-Fushman and Lin 2007], the technology does not yet exist to automatically answer open domain, complex [17] and subjective question. Recent efforts at automatic evaluation show that even for well-defined, objective, complex questions [17], evaluation is extremely labor-intensive and has many challenges.

Our work is related to, but distinct from interactive Question Answering. In particular, we can directly study the satisfaction from information seeker perspective. We believe that our proposed methods can contribute the understanding of asker satisfaction prediction [16]. To our knowledge, this paper is the

study of real user satisfaction with variety of satisfactory parameters. Hence, our paper focuses on important manifestation of social media community question/answering sites, and our work draws on significant amount of prior research on Yahoo! Answers [17].

9. CONCLUSION

This paper describes our work on seeker satisfaction prediction in Yahoo! Answers. We introduced and formalized automatic Ranking algorithm, abstract generation and history Updation to improve asker satisfaction. Also our results on satisfaction prediction [15] demonstrate significant accuracy improvements using "Generalization" ranking methodology, "rhetorical structure" and "Learning Automata technique [10]". Our proposed techniques work well with crucial problems like answer understanding, answer justifying, and asker's taste changes over a time period. Thus this paper outlines a promising area in the general field of modeling user intent, expectations, and satisfaction, and can potentially result in practical improvements to the effectiveness and design of question answering communities.

10. FUTURE WORK

In terms of future work, for some of the technical questions we can't expect answers with technical terms instead it may be collocial and general opinion. One of the crucial problems is that an answer may be fully relevant to the question according to the ranking system, but not to the asker's point of view, because the system can't fully predict what the asker really wants and it cannot understand on what context the user expects the answer.

Rhetorical structure is applicable only for a particular domain and it will detect only 40 categories. In future more number of categories can be added irrespective of the domain. The gist generated by the above system (without replication) is not preferable by some of the users because they may confirm the answer for a particular question, from the repeated answer (duplication) obtained from different answerers as the correct one.

Duplication in answer has both positive and negative vice versa and on the other hand redundancy makes answer prediction easier. If an asker has missed one answer, may be it has the other and a replica can be viewed. On the other hand from the point of view of CQA, storing duplicate content is a waste of resources. But from some asker's point of view getting duplicate answer from the response to a query is a nusense. The primary reason for duplication on the QA is a systematic replication of content across different answers. It is estimated that at least 10% of the answers are mirrored.

In future, we plan to address the problem of predicting satisfaction of new users who has no previous experience with Yahoo! Answers [17] .Also exploiting new user's interest and other interaction information with CQA remains a promising direction of future work.

REFERENCES

[1] E. Agichtein, E. Brill, S.Dumais, and R.Ragno, 2006, "Learning user interaction models for Predicting web search result preferences", In Proc of IGIR, pp. 3-10.

[2] Kazuo Sumita., Seiji Miike., kenji ono, and Tetsuro Chino, 2007, "Automatic abstract generation based on Document structure analysis and its evaluation as a Document retrieval presentation function", Journal of systems and computers in Japan, Vol.26, Issue 13, pp.32-43.

[3] W. Lehnert, 1980, "Narrative Text Summarization", in Proc. of AAAI, pp.337-339.

[4] H. Beigy, and M.R Meybodi, 2002, "A new distributed Learning automata based algorithm for solving stochastic shortest path problem", in Proc of the sixth international joint conference on information science,Durham, USA, pp.339-343.

[5] Syandra Sari, and Mirna Adriani, 2008, "Using Mutual Information Technique in Cross-Language Information Retrieval", Springer link –Volume 5362, pp. 276-284.

[6] C. Singhal Buckley and M. Mitra, 1996, "Pivoted document length normalization", in Proc of the 19th annual International ACM SIGIR Conference on research and development in information retrieval, pp. 21-29.

[7] T. Strohman, D. Metzler, H. Turtle, and W.B. Croft, 2005, "Indri: A language model-based search engine for complex queries", in Proc of the International conference in intelligent analysis.

[8] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. Verkamo, 1995, "Fast discovery of association rules", in advances in knowledge Discovery and Data Mining, AAAI/MIT, pp.307-327.

[9] W. Alvarez, and C. Ruiz, 2000, "collaborative recommendation via adaptive association rule mining", in Proc of the International Workshop on web mining for E-Commerce, pp.35-41.

[10] M. A. L, and R. Harita Bhaskar, 1987, "Learning automata with changing number of actions", IEEE Transactions on Systems Man and cybernetics, pp.1095-1100.

[11] G. Salton, and C. Buckley, 1988, "Term weighting approaches in automatic text retrieval" Information Processing and Management, pp. 513-523.

[12] R. Cohen, 1987, "Analyzing the Structure of Argumentative Discourse Computational Linguistics" Vol.13, pp. 11–24.

[13] Donald Metzler, 2007, "Automatic Feature selection in the Markov Random Field Model for Information Retrieval", CIKM '07, November pp. 6-8, Lisboa Portugal.

[14] E. Agichtein, C. Castillo., D. Donato., A. Gionis, and G. Mishne, 2008, "Finding High Quality Content in Social Media with an Application to Community Based Question Answering", in Proc .of WSDM, pp. 183-194.

[15] Y. Liu, and E. Agichitein, 2008, "you've got answers: Towards personalized models for predicting success in community question answering", in Proc of the 46th Annual Meeting of the Association for Computational Linguistics (ACL).pp. 97–100.

[16] J. Lin, and D. Demner-fushman, 2006 "Methods for automatically evaluating answers to complex questions", Inform. Retrieval. 9, 5, pp. 565-587.

[17] Y. Liu, and E. Agichtein, 2008, "On the evolution of the Yahoo! answers qa community", in Proceedings of the 31st Annual International ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR), pp.737-738.

[18] M. Mitchell, Lapata. 2008, "Vector-based models of semantic composition", in Proc of ACL, pp. 236-244.

[19] K. Erk, 2007, "A simple, similarity-based model for selectional preferences", in Proc of ACL, pp. 216-223.

[20] D.V Cicchetti, and Feinstein A.R. 1990 "High agreement but low kappa: II", volume 43, pp. 551-558.

[21] D. Metzler, and W.B. Croft 2005, "A Markov random field model for term dependencies", in Proc.28th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval, pp.472–479.

[22] S.D Pietra, V.D Pietra, and J. Lafferty, 1997, "Inducing features of random fields", IEEE Transactions on Pattern Analysis and Machine Intelligence", 19(4): pp. 380–393.

[23] P. Resnik, 1995, "Using information content to Evaluate semantic similarity in taxonomy", in Proc of IJCAI-95, pp.448-453, Montreal, Canada.