INFERRING ANSWER QUALITY, ANSWERER EXPERTISE, AND RANKING IN QUESTION ANSWER SOCIAL NETWORKS

by

YUANZHE CAI

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON
May 2014

To my mother Yongping Wei and my father Jianping Cai
who set the example and who made me who I am.

Acknowledgements

I would like to express sincere gratitude to my advisor Sharma Chakravarthy for constantly motivating and encouraging me as well as providing valuable guidance and support during the course of my doctoral work. It is the same type of enthusiasm that he devoted into advising that helped in making me a true researcher. He has spent a amount of time training me in research, writing, and giving professional talks. His dedication, persistence, and sincerity in research deeply impressed me and have set up the high standards that I want to maintain in my future career.

I want to thank my thesis committee members Gautam Das, Chengkai Li and Leonidas Fegaras for their interest in my research, for taking time to serve in my dissertation committee and for their comments, suggestions, guidance and help at the time of need. I would also like to thank Ramez Elmasri and Bahram Khalili for all their support, encouragement and guidance during the years I have been in UTA as a graduate teaching assistant. In addition, I would like to acknowledge the assistance I received, from Pamela McBride, Sherri Gotcher, and Camille Costabile as well as the entire staff of the Computer Science and Engineering Department at UTA, during the course of my doctoral work.

I wish to thank all my colleagues (past and present) at Information Technology Lab (ITLab) for their support and encouragement and for making the stay at ITLab over the last five years thoroughly exciting. I am also grateful to my friends for their interest in my research and for their helpful discussions and invaluable comments. My appreciation especially goes to, for providing me with generous support and countless fun throughout the five years at UTA. I am grateful to all the professors and mentors who have helped me throughout my career, both in China and United States. Without their encouragement and endurance, this work would not have been possible.

Defense Time: April 18, 2014

Abstract

INFERRING ANSWER QUALITY, ANSWERER EXPERTISE, AND RANKING IN QUESTION ANSWER SOCIAL NETWORKS

YUANZHE CAI, Ph.D.

The University of Texas at Arlington, 2014

Supervising Professor: Sharma Chakravarthy

Search has become ubiquitous mainly because of its usage simplicity. Search has made great strides in making information gathering relatively easy and without a learning curve. Question answering services/communities (termed CQA services or Q/A networks; e.g., Yahoo! Answers, Stack Overflow) have come about in the last decade as yet another way to search. Here the intent is to obtain good/high quality answers (from users with different levels of expertise) for a question when posed, or to retrieve answers from an archived Q/A repository. To make use of these services (and archives) effectively as an alternative to search, it is imperative that we develop a framework including techniques and algorithms for identifying quality of answers as well as the expertise of users answering questions. Finding answer quality is critical for archived data sets for accessing their value as stored repositories to answer questions. Meanwhile, determining the expertise of users is extremely important (and more challenging) for routing queries in real-time which is very important to these Q/A services – both paid and free. This problem entails an understanding of the characteristics of interactions in this domain as well as the structure of graphs

derived from these interactions. These graphs (termed Ask-Answer graphs in this thesis) have subtle differences from web reference graphs, paper citation graphs, and others. Hence it is imperative to design effective and efficient ranking approaches for these Q/A network data sets to help users retrieve/search for meaningful information.

The objective of this dissertation is to push the state-of-the-art in the analysis of Q/A social network data sets in terms of theory, semantics, techniques/algorithms, and experimental analysis of real-world social interactions. We leverage “participant characteristics” as the social community is dynamic with participants changing over a period of time and answering questions at their will. The participant behavior seems to be important for inferring some of the characteristics of their interaction.

First, our research work has determined that temporal features make a significant difference in predicting the quality of answers because the answerer’s (or participant’s) current behavior plays an important role in identifying the quality of an answer. We present learning to rank approaches for predicting answer quality as compared to traditional classification approaches and establish their superiority over currently-used classification approaches. Second, we discuss the difference between ask-answer graphs and web reference graphs and propose the ExpertRank framework and several approaches using domain information to predict the expertise level of users by considering both answer quality and graph structure. Third, current approaches infer expertise using traditional link-based methods such as PageRank or HITS. However, these approaches only identify global experts, which are termed generalists, in CQA services. The generalist may not be the best person to answer *an* arbitrary question. If a question contains several important concepts, it is meaningful for a person who is an expert in these concepts to answer that question. This thesis proposes techniques to identify experts at the concept level as a basic building block. This is critical as it can be used as a basis for inferring expertise at different levels using the

derived concept rank. For example, a question can be viewed as a collection of a few important concepts. For answering a question, we use the ConceptRank framework to identify specialists for answering that question. This can be generalized using a concept taxonomy for classifying topics, areas, and other larger concepts using the primary concept of coverage.

Ranking is central to the problems addressed in this thesis. Hence, we analyze the motivation behind traditional link-based approaches, such as HITS. We argue that these link-based approaches correspond to statistical information representing the opinion of web writers for these web resources. In contrast, we address the ranking problem in web and social networks by using the ILP (in-link probability) and OLP (out-link probability) of a graph to help understand HITS approach in contexts other than web graphs. We have further established that the two probabilities identified correspond to the hub and authority vectors of the HITS approach. We have used the standard Non-negative Matrix Factorization (NMF) to calculate these two probabilities for each node. Our experimental results and theoretical analysis validate the relationship between ILOD approach and HITS algorithm.

Table of Contents

List of Illustrations

Figure Page

List of Tables

Table Page

CHAPTER 1

Introduction

Search engines (e.g., Google [2]) have become common mainly because they are easy to use and are accessible to the general public eliminating the need for a learning curve. Search has made great strides towards making targeted information gathering relatively easy. Although search has become indispensable, it has some limitations. For example, search outputs only individual pages containing search words (or combinations thereof). Currently, search does not combine/join contents from multiple pages. Notwithstanding ranking [3], search has the problem of *too many answers*. That is, a large number of ranked results (sometimes in the Millions) are returned making it impossible to browse more than a few of them.

Several improvements and alternatives have come about to overcome some of the above limitations. Advanced search mechanisms [4, 5, 6], or *meta-search engines* (e.g., Vivisimo [7]), post-process the output of one or more search engines to organize and classify the resulting sources in a meaningful manner. Faceted search [8, 9, 10] allows the user to zero in on their desired retrieval through a navigation process assisted by the system. Additionally, several domain/topic-specific retrieval systems (e.g., Google Base [11]) operate on a predetermined set of sources belonging to individual concepts and provide results based on well-understood user criteria such as cost, schedule, proximity, etc. A number of approaches, techniques, and systems such as Havasu [12], MetaQuerier [13], Ariadne [13], TSIMMIS [14, 15], InfoMaster [16], Information Manifold [17], Whirl [18], Tukwila [19], and others have addressed the problem of heterogeneous data integration. These architectures are modeled using different approaches (such as mediated [20], federated [21], ontological [22], query-based [23], warehousing [24], navigational [25], and object-oriented [26]), which together and independently work to tackle a disparate and varied set of challenges.

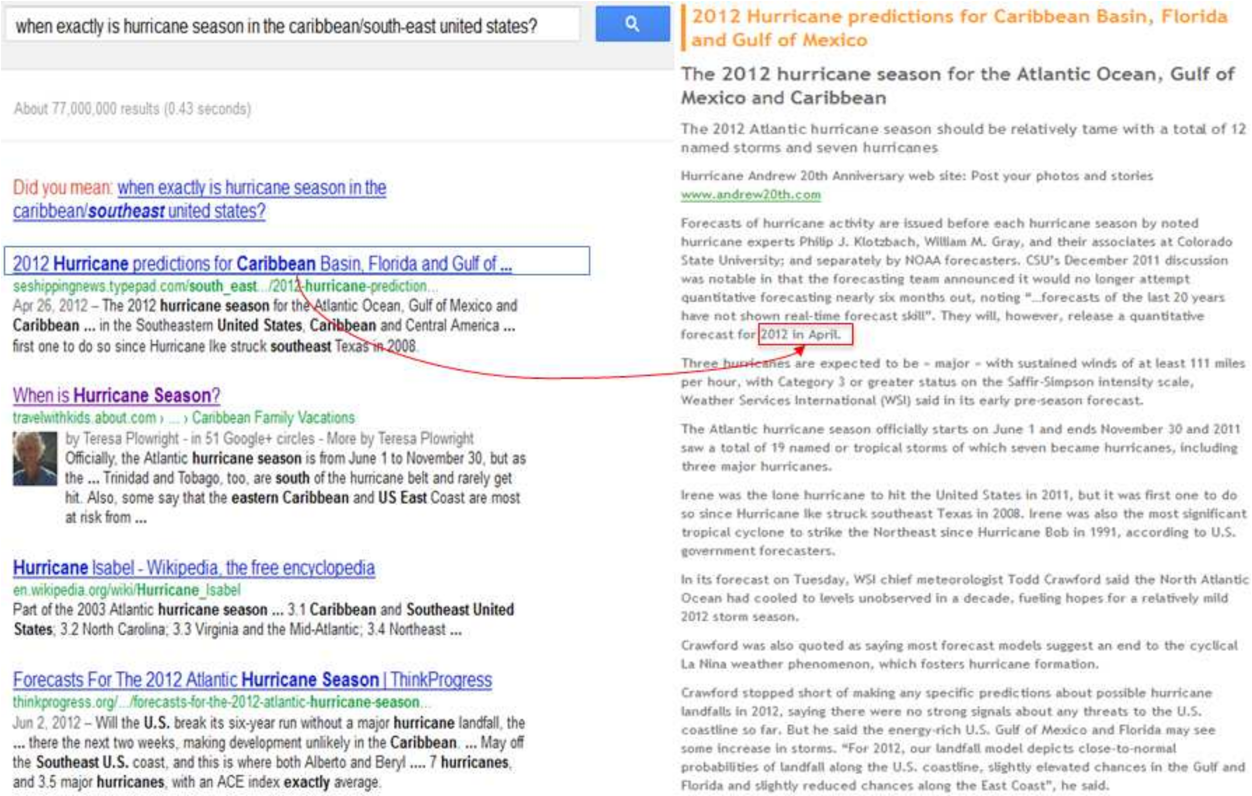
More recently, Liquid search [27, 28] has proposed multi-domain query answering by facilitating combining results from multiple pages/sites. Query-By-Keywords [29] and using search words and forms [8] have tried to map the search paradigm into unstructured and structured repositories by generating queries to get few but more specific answers.

Question answering services/communities (termed CQA services or Q/A social networks) have come about in the last decade as yet another alternative to search. It is seen as an alternative to *too many answers*. It avoids dealing with a large number of answers/results as well as the task of sifting through them (although ranked) to get at the desired information. Both general purpose and topic-specific communities are growing in number for posting questions and obtaining direct answers in a short period of time. Yahoo! Answers (`http://answers.yahoo.com/`) (Y!A), for example, provides a broad range of topics where as Stack Overflow (`http://stackoverflow.com/`) (SO), and Turbo Tax Live (`https://ttlc.intuit.com/`) (TT) are examples of focused and domain-specific question/answer social networks.

Compared with the traditional information retrieval technique (e.g., Google.com, Ask.com, Bing.com, etc.), question answering services provide more accurate results. For example, since a questioner wants to plan her honeymoon this fall, she wants to know when is the hurricane season in the southeastern United States. This questioner tried three ways to find an answer: 1. by inputting the question into the Google search engine and retrieves the results, 2. by using ask.com to find results, and 3. by posting this question in the Yahoo! Answers CQA service. Figures 1.1, 1.2 and 1.3, respectively, show the results returned by Google, Ask.com, and Yahoo! Answers. In

Figures 1.1 and 1.2, these traditional search systems typically return a long document (when the first url returned is clicked) to the questioner (e.g., more than 2000 words in Google and Ask.com) and questioners need to read all the documents to search for useful information. Meanwhile, the validity of results from Google and Ask.com cannot be confirmed. Therefore, the questioner needs to read more web pages to confirm that result. However, compared with the traditional results, Yahoo! Answers is likely to give more accurate, succinct, and clear result. In Figure 1.3, the answer obtained using Yahoo! Answers system is short (only about 17 words) and to the point. In addition, since other users in the community evaluate the answers and give their votes for all these answers, Yahoo! Answers system automatically selects the best answer from all the answers given to that question. Therefore, these question/answering services not only provide a direct answer, but also an accurate one.

The above communities allow questioners to post questions and others to provide answers. Figure 1.4 shows a "socket" question in C language and some of its answers from the Stack Overflow. Questioner "destructo_gold" posts this question and the other two answerers "Len Holgate" and "Clifford" answer this question. These communities have become quite popular in the last several years for a number of reasons. First, because of the targeted response from answers with knowledge and/or experience, these answers are likely to be more useful and easy to understand for the questioner. Second, the CQA communities also provide a consolidated communication environment where answers to related questions can also be viewed. This environment facilitates multiple answers (likely from different perspectives) and discussion (in the form of comments, threads) which can benefit the questioner (and others as well). It is also possible for the questioner to interact with the answerer (by email or other means) for clarification and advice. This paradigm, although quite different from the instantaneous search for stored information, is likely to provide the



when exactly is hurricane season in the caribbean/south-east united states?

About 77,000,000 results (0.43 seconds)

Did you mean: when exactly is hurricane season in the caribbean/***southeast*** united states?

2012 **Hurricane** predictions for **Caribbean** Basin, Florida and Gulf of ...
seshippingnews.typepad.com/**south_east**.../2012-**hurricane**-prediction...
Apr 26, 2012 – The 2012 **hurricane season** for the Atlantic Ocean, Gulf of Mexico and **Caribbean ...** in the Southeastern **United States**, **Caribbean** and Central America ... first one to do so since Hurricane Ike struck **southeast** Texas in 2008.

When is **Hurricane Season**?
travelwithkids.about.com › ... › Caribbean Family Vacations
by Teresa Plowright - in 51 Google+ circles - More by Teresa Plowright
Officially, the Atlantic **hurricane season** is from June 1 to November 30, but as the **...** Trinidad and Tobago, too, are **south** of the hurricane belt and rarely get hit. Also, some say that the **eastern Caribbean** and **US East** Coast are most at risk from **...**

Hurricane Isabel - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/**Hurricane**_Isabel
Part of the 2003 Atlantic **hurricane season ...** 3.1 **Caribbean** and **Southeast United States**; 3.2 North Carolina; 3.3 Virginia and the Mid-Atlantic; 3.4 Northeast **...**

Forecasts For The 2012 Atlantic **Hurricane Season** | ThinkProgress
thinkprogress.org/.../forecasts-for-the-2012-atlantic-**hurricane-season**...
Jun 2, 2012 – Will the **U.S.** break its six-year run without a major **hurricane** landfall, the **...** there the next two weeks, making development unlikely in the **Caribbean**. **...** May off the **Southeast U.S.** coast, and this is where both Alberto and Beryl **....** 7 **hurricanes**, and 3.5 major **hurricanes**, with an ACE index **exactly** average.

2012 Hurricane predictions for Caribbean Basin, Florida and Gulf of Mexico

The 2012 hurricane season for the Atlantic Ocean, Gulf of Mexico and Caribbean

The 2012 Atlantic hurricane season should be relatively tame with a total of 12 named storms and seven hurricanes

Hurricane Andrew 20th Anniversary web site: Post your photos and stories
www.andrew20th.com

Forecasts of hurricane activity are issued before each hurricane season by noted hurricane experts Philip J. Klotzbach, William M. Gray, and their associates at Colorado State University; and separately by NOAA forecasters. CSU's December 2011 discussion was notable in that the forecasting team announced it would no longer attempt quantitative forecasting nearly six months out, noting "...forecasts of the last 20 years have not shown real-time forecast skill". They will, however, release a quantitative forecast for 2012 in April.

Three hurricanes are expected to be – major – with sustained winds of at least 111 miles per hour, with Category 3 or greater status on the Saffir-Simpson intensity scale, Weather Services International (WSI) said in its early pre-season forecast.

The Atlantic hurricane season officially starts on June 1 and ends November 30 and 2011 saw a total of 19 named or tropical storms of which seven became hurricanes, including three major hurricanes.

Irene was the lone hurricane to hit the United States in 2011, but it was first one to do so since Hurricane Ike struck southeast Texas in 2008. Irene was also the most significant tropical cyclone to strike the Northeast since Hurricane Bob in 1991, according to U.S. government forecasters.

In its forecast on Tuesday, WSI chief meteorologist Todd Crawford said the North Atlantic Ocean had cooled to levels unobserved in a decade, fueling hopes for a relatively mild 2012 storm season.

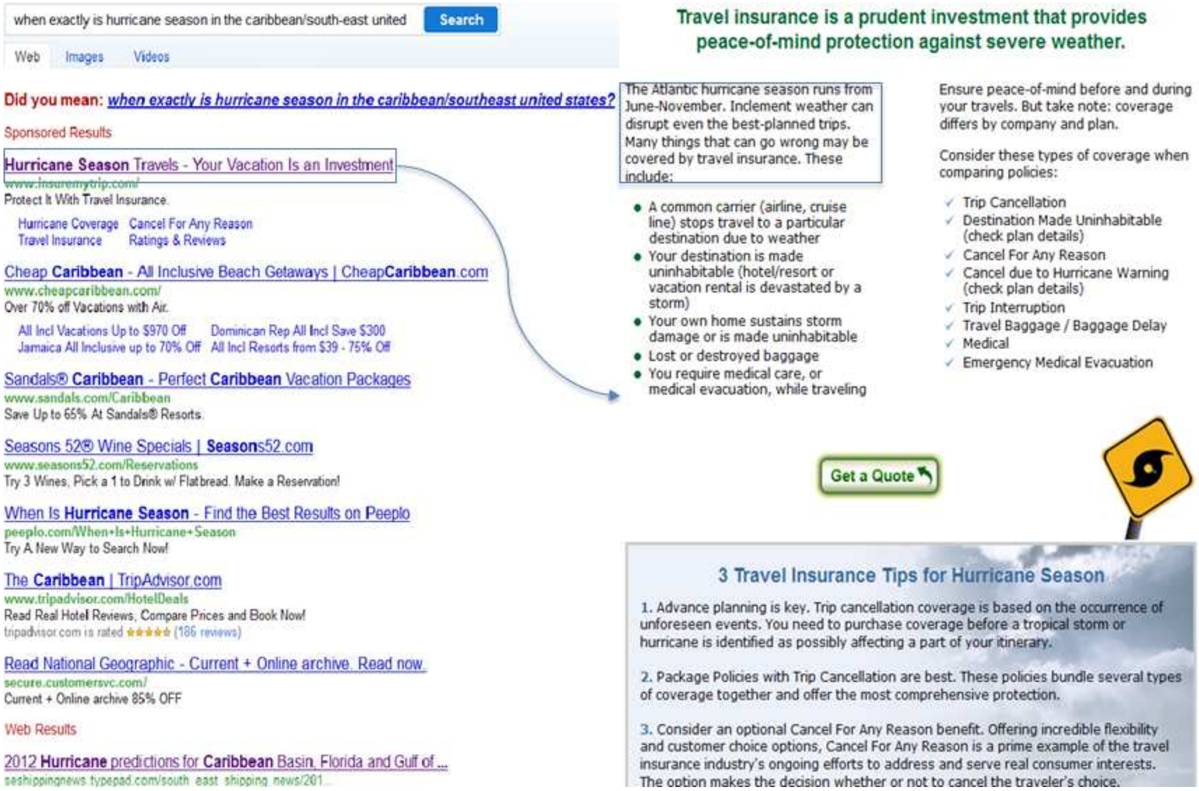
Crawford was also quoted as saying most forecast models suggest an end to the cyclical La Nina weather phenomenon, which fosters hurricane formation.

Crawford stopped short of making any specific predictions about possible hurricane landfalls in 2012, saying there were no strong signals about any threats to the U.S. coastline so far. But he said the energy-rich U.S. Gulf of Mexico and Florida may see some increase in storms. "For 2012, our landfall model depicts close-to-normal probabilities of landfall along the U.S. coastline, slightly elevated chances in the Gulf and Florida and slightly reduced chances along the East Coast", he said.

Figure 1.1: Results from Google Search Engine

questioner with useful answers. Finally, the forum provides an incentive for people to showcase their expertise and in the process get recognized by the community. For this reason, many CQA services allow the questioner to flag the *best* answer from the set of answers. Some CQA services have a voting mechanism to rank the responses. The notion of an expertise level exists in some services and is based on a number of factors, such as number of best answers given by a answerer, votes obtained for answers, etc.

The intent of CQA services is to provide good/high quality answers (for a question). This can also be accomplished by retrieving answers from a archived Q/A repository. However, to make use of these services effectively as an alternative to search, it is imperative that we develop approaches, algorithms and techniques as



when exactly is hurricane season in the caribbean/south-east united

Did you mean: when exactly is hurricane season in the caribbean/southeast united states?

Sponsored Results

Hurricane Season Travels - Your Vacation Is an Investment
www.insuremytrip.com/
Protect It With Travel Insurance.
Hurricane Coverage Cancel For Any Reason
Travel Insurance Ratings & Reviews

Cheap Caribbean - All Inclusive Beach Getaways | CheapCaribbean.com
www.cheapcaribbean.com/
Over 70% off Vacations with Air.
All Incl Vacations Up to $970 Off Dominican Rep All Incl Save $300
Jamaica All Inclusive up to 70% Off All Incl Resorts from $39 - 75% Off

Sandals® Caribbean - Perfect Caribbean Vacation Packages
www.sandals.com/Caribbean
Save Up to 65% At Sandals® Resorts.

Seasons 52® Wine Specials | Seasons52.com
www.seasons52.com/Reservations
Try 3 Wines, Pick a 1 to Drink w/ Flatbread. Make a Reservation!

When Is Hurricane Season - Find the Best Results on Peeplo
peeplo.com/When+Is+Hurricane+Season
Try A New Way to Search Now!

The Caribbean | TripAdvisor.com
www.tripadvisor.com/HotelDeals
Read Real Hotel Reviews, Compare Prices and Book Now!
tripadvisor.com is rated ★★★★★ (186 reviews)

Read National Geographic - Current + Online archive. Read now.
secure.customersvc.com/
Current + Online archive 85% OFF

Web Results

2012 Hurricane predictions for Caribbean Basin, Florida and Gulf of ...
seshippingnews.typepad.com/south_east_shipping_news/201...

Travel insurance is a prudent investment that provides peace-of-mind protection against severe weather.

The Atlantic hurricane season runs from June-November. Inclement weather can disrupt even the best-planned trips. Many things that can go wrong may be covered by travel insurance. These include:

- A common carrier (airline, cruise line) stops travel to a particular destination due to weather
- Your destination is made uninhabitable (hotel/resort or vacation rental is devastated by a storm)
- Your own home sustains storm damage or is made uninhabitable
- Lost or destroyed baggage
- You require medical care, or medical evacuation, while traveling

Ensure peace-of-mind before and during your travels. But take note: coverage differs by company and plan.

Consider these types of coverage when comparing policies:

✓ Trip Cancellation
✓ Destination Made Uninhabitable (check plan details)
✓ Cancel For Any Reason
✓ Cancel due to Hurricane Warning (check plan details)
✓ Trip Interruption
✓ Travel Baggage / Baggage Delay
✓ Medical
✓ Emergency Medical Evacuation

3 Travel Insurance Tips for Hurricane Season

1. Advance planning is key. Trip cancellation coverage is based on the occurrence of unforeseen events. You need to purchase coverage before a tropical storm or hurricane is identified as possibly affecting a part of your itinerary.

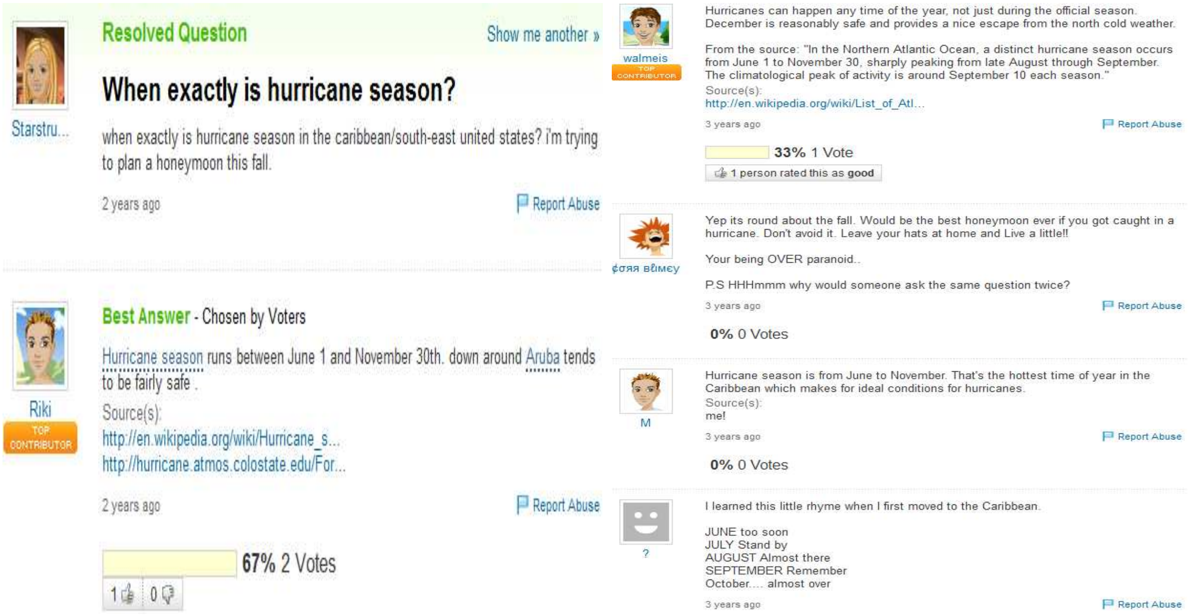
2. Package Policies with Trip Cancellation are best. These policies bundle several types of coverage together and offer the most comprehensive protection.

3. Consider an optional Cancel For Any Reason benefit. Offering incredible flexibility and customer choice options, Cancel For Any Reason is a prime example of the travel insurance industry's ongoing efforts to address and serve real consumer interests. The option makes the decision whether or not to cancel the traveler's choice.

Figure 1.2: Results from Ask.com System

well as define some of the concepts more precisely for identifying/predicting quality of answers in addition to qualifying or verifying the expertise of answerers.

First, since some of the questions have so many answers, when browsing the answer, it is important and challenging to associate quality with answers. Second, when a question is posted to the CQA services, answerers randomly answer these questions; therefore, there are a lot of irrelevant answers. In order to route questions to the right answerers in the CQA services, identifying the expertise of answerers from these data sets becomes an important question. Third, for a particular question, the global experts from this CQA services may not be suitable to answer this question. For example, for a socket question in C language the answer of a "socket" expert is likely to be much better than the answer from a C expert. Therefore, identifying specialists



When exactly is hurricane season?

when exactly is hurricane season in the caribbean/south-east united states? i'm trying to plan a honeymoon this fall.

2 years ago

Hurricane season runs between June 1 and November 30th. down around Aruba tends to be fairly safe .

Source(s):
http://en.wikipedia.org/wiki/Hurricane_s...
http://hurricane.atmos.colostate.edu/For...

2 years ago

Hurricanes can happen any time of the year, not just during the official season. December is reasonably safe and provides a nice escape from the north cold weather.

From the source: "In the Northern Atlantic Ocean, a distinct hurricane season occurs from June 1 to November 30, sharply peaking from late August through September. The climatological peak of activity is around September 10 each season."

Source(s):
http://en.wikipedia.org/wiki/List_of_Atl...

3 years ago

Yep its round about the fall. Would be the best honeymoon ever if you got caught in a hurricane. Don't avoid it. Leave your hats at home and Live a little!!

Your being OVER paranoid..

P.S HHHmmm why would someone ask the same question twice?

3 years ago

Hurricane season is from June to November. That's the hottest time of year in the Caribbean which makes for ideal conditions for hurricanes.

Source(s):
me!

3 years ago

I learned this little rhyme when I first moved to the Caribbean.

JUNE too soon
JULY Stand by
AUGUST Almost there
SEPTEMBER Remember
October.... almost over

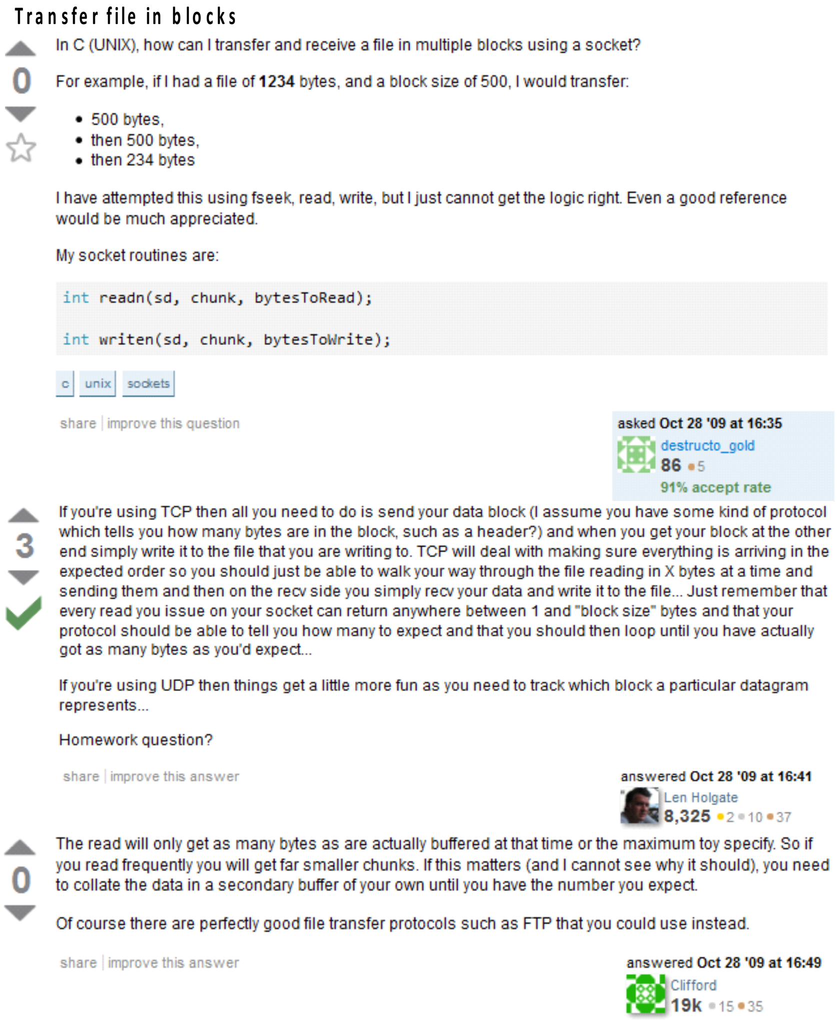
3 years ago

Figure 1.3: Answers from Yahoo! Answers

for a specific question would be much more beneficial and important than identifying a generalist in this CQA community. Finally, since our thesis mainly address the ranking problem in the social graph (especially CQA services), the motivation of these traditional linked-based approaches (e.g., HITS) is also analyzed in this thesis. We argue that these link-based approaches calculate statistics for the opinion of web writers for these web resources. Therefore, we address the ranking problem in web and social networks by using the in-link probability and out-link probability and use the standard Non-negative Matrix Factorization (NMF) to calculate these two probabilities for each node.

Specifically, for this dissertation we mainly focus on the following problems:

- Answer for Quality Analysis: Appropriate approaches/mechanisms to identify high quality answers (for a question) from a Q/A repository,
- Identifying Generalists (Global Expertise): Suitable techniques to provide/identfy experts from a Q/A repository,



Transfer file in blocks

In C (UNIX), how can I transfer and receive a file in multiple blocks using a socket?

For example, if I had a file of **1234** bytes, and a block size of 500, I would transfer:

- 500 bytes,
- then 500 bytes,
- then 234 bytes

I have attempted this using fseek, read, write, but I just cannot get the logic right. Even a good reference would be much appreciated.

My socket routines are:

```
int readn(sd, chunk, bytesToRead);

int writen(sd, chunk, bytesToWrite);
```

asked Oct 28 '09 at 16:35
destructo_gold
86 ●5
91% accept rate

If you're using TCP then all you need to do is send your data block (I assume you have some kind of protocol which tells you how many bytes are in the block, such as a header?) and when you get your block at the other end simply write it to the file that you are writing to. TCP will deal with making sure everything is arriving in the expected order so you should just be able to walk your way through the file reading in X bytes at a time and sending them and then on the recv side you simply recv your data and write it to the file... Just remember that every read you issue on your socket can return anywhere between 1 and "block size" bytes and that your protocol should be able to tell you how many to expect and that you should then loop until you have actually got as many bytes as you'd expect...

If you're using UDP then things get a little more fun as you need to track which block a particular datagram represents...

Homework question?

answered Oct 28 '09 at 16:41
Len Holgate
8,325 ●2 ●10 ●37

The read will only get as many bytes as are actually buffered at that time or the maximum toy specify. So if you read frequently you will get far smaller chunks. If this matters (and I cannot see why it should), you need to collate the data in a secondary buffer of your own until you have the number you expect.

Of course there are perfectly good file transfer protocols such as FTP that you could use instead.

answered Oct 28 '09 at 16:49
Clifford
19k ●15 ●35

0
c
unix
sockets
share | improve this question
3
share | improve this answer
0
share | improve this answer

Figure 1.4: A Sample Content from Stack Overflow

- Identifying Specialists (for Concepts): Appropriate techniques to identify experts for specific concepts, and
- NMF as an alternative approach to ranking: We propose an alternative way of analyzing ranking and proposing an NMF approach as an alternative to inferring hub and authority scores in HITS.

1.1 Answer Quality Analysis of Q/A Data Sets

A question and answer web site is purposefully designed to allow users to post questions and answer other user's questions on a range of topics. Based on our literature survey and observations, we classified the question answer web sites into three broad categories: (i) FQ/A Web Site: FQ/A web sites do not allow answerers to ask or respond to questions. In order to avoid answering similar questions again and again, technical support will post the frequent question answers (FQ/A) on the web site. HP Laptop Battery FQ/A (`http://laptopz.over-blog.com/`) is an example of such a web site. In an FQ/A web site, each question has only one answer, but the answer is generally given by the expert. (ii) Ask an Expert Web Site: This web site allows a questioner to pose questions to a real expert but only one expert will answer this question. These kinds of web sites have strict procedures for selecting experts. Because of the strict evaluation of an expert, these Q/A sites provide quality answers for a question. Examples of such communities include AllExperts(`http://www.allexperts.com/`), MadSci Network(`http://madsci.org`) and so on. (iii) Community Question Answer Web Site (also called community question answer server): Community question answer web sites allow a user to take part in the process of questioning and answering. One example of this Q/A server is Yahoo! Answers web site. In the Yahoo! Answers community, questioners post their questions and optional description on a specific topic category. The question then appears in the most recent open questions list and can be answered by any answers in this community. The questioner will choose the best answer or it is also possible for other users to vote for the best answer. Since both FQ/A web site and ask an expert forums do not need to identify the answer quality, this thesis mainly focuses on the community question answer web site.

Example 1.1 *Bad answers in Stack Overflow community:*

Q1: In C arrays why is this true? a[5] == 5[a]?

Answer: Good question but I do not know.

Q2: What is the best tool for creating an Excel Spreadsheet with C#?

Answer: You can search the results in the Google.

Q3: How can Inheritance be modeled using C?

Answer: Technically no. Practically yes.

After a questioner posts a question in the community question answer web site, a lot of answerers can answer this question. Therefore, a question may receive many answers. For example, in Stack Overflow community, some of the questions receive more than 100 answers. However, not all of the answers are useful. Example 1.1 illustrates a few bad answers from Stack Overflow community. Therefore, to help the user browsing these answers becomes an important question. One solution is to identify an answer's quality and then ranking these answers according to their quality. Most of the extant work for evaluating/predicting the quality of answers are based on a large number of features extracted from these data sets, and the use of traditional classification approaches for predicting the best answer. However, in our thesis we first argue that the currently-used classification approaches are not well-suited for this problem and propose learning to rank approaches to identify not only the best answer but also to rank all answers. Second, we propose a small set of temporal features and the establishment of their effectiveness for predicting the quality of answers. In our experiment section, a small set of temporal features performed much better than the other features proposed for this purpose in the literature.

1.2 Identifying Generalists (Global Expertise):

Community Question Answering services strive to provide users with meaningful information using the ask-answer paradigm. Hence, it would be helpful if useful inferences can be mined from these data sets so they can be employed to improve these services. For example, if we can infer or identify expertise of answerers' from these data sets, these questions can be routed to the right group of answerers.

Currently, Q/A communities mainly use two approaches for finding appropriate answerers to answer a question: (i) Questioner-Based Approach: the questioner is responsible for choosing an appropriate expert to answer his/her question, (i-i) Answerer-Based Approach: This approach allows answerers to answer questions that are of interest to them. However, both of these two approaches have a number of drawbacks. In the questioner-based approach, there are a large number of answerers in a Q/A community; hence, it is impractical to expect questioners to find an expert by browsing all the answerers' profiles. For example, in Yahoo! Answers community, as there are millions of answerers, it is impossible to ask the questioner to find suitable answerers. In the answerer-based Approach, although this approach encourages various users to answer questions, this method ignores answerer's quality. Example 1.1 shows a sample of irrelevant/bad answers. Therefore, in order to receive better answers, automatically identifying the answerer's expertise and routing the questions to proper answerers becomes an important problem.

To solve the above problem, researchers have mainly used information retrieval [30] and (extended) link-based methods [31, 32] for discovering experts from CQAs. However, these approaches do not seem to be appropriate for this problem. For example, similarity score (used from information retrieval) does not represent quality. Also, all the link-based methods only consider the graph structure without using the contents of questions or answers; but we believe that in Q/A communities the answer quality plays an important role to identify the answerers' expertise. Thus, taking into account both graph structure and domain information, we propose the ExpertRank framework to identify answerers' expertise in CQAs. In our experimental part, we have demonstrated the effectiveness of our approach by comparing them with traditional link-based approaches.

1.3 Identifying Specialists (for Concepts)

In CQAs, expertise can be classified into general and specific. A generalist has a broad knowledge over a topic or an area (similar to breadth of knowledge). Whereas a specialist, in our case, is identified at the level of a concept. S/he has a good understanding of the concept in questions, and hence, will be able to answer questions on that concept better than a generalist. Generalists are likely to choose to answer a broad spectrum of questions. Generalists may be good at answering many questions, but typically are not at the same expert level as a specialist. Since most questions are related to a specific area, specialists in that specific area are likely to give a better answer for these questions. However, all the link-based methods [31, 32] only focus on identifying the generalist. For CQAs, finding the specialist for answering a question is much more important since in normal case specialists' answer are better than that of a generalist.

Consider the following short example to illustrate the above observations. Figure 1.4 shows a socket question in C language and answerer Clifford and Len Holgate respectively answers this question. Figure 1.5 shows the characteristics of answerer Clifford and Len Holgate. In Figure 1.5, Clifford answers 286 questions in CQAs and his answered questions involve various aspects of C language (e.g., memory, thread, buffer, etc.), but Clifford only answered 4 "socket" questions; The other answerer Len

Holgate only answers 32 questions in CQAs, but 27 questions are related to “socket” problems. Clifford is a generalist in C language compared to Len Holgate, and Len Holgate is a specialist in “socket” questions. In Figure 1.4, since questioner destructogold asks how to transfer a file using socket functions (a “socket” related question), Len Holgate gives a better answer than Clifford (Len Holgate’s answer receives 3 votes which is higher than Clifford’s answer (0 score)). In other words, specialist usually provides a high quality answer than generalist for a specific question.

We observe that the answerer’s ability to answer a question is definitely decided by his/her understanding of these concepts. For example, “socket,” “transfer,” “file,” “block,” are the key words for this question. If a answerer is familiar with these concepts, this answerer is more likely to answer this question better. This entails developing a framework for obtaining expertise ranking for concepts. Once we have established concept-based expertise ranking, the rankings can be beneficially used for answering a question composed of many concepts. Therefore, the first step is to automatically extract meaningful concepts from all questions and build the answerer’s expertise score for each concept. Then, we analyze domain information from several data sets and indicate how they can be used to analyze the answerer’s expertise score. We also analyze the importance of each concept and set a different weight for each concept. Finally, using the Top-K search model, we combine the weight and answerer’s expertise score for each concept together to identify the specialist for each question. We present our framework along with the algorithms as well as extensive experimental analysis that indicates superiority of our approach as compared to other link-based methods. Detailed discussions and approaches are elaborated on Chapter 5.

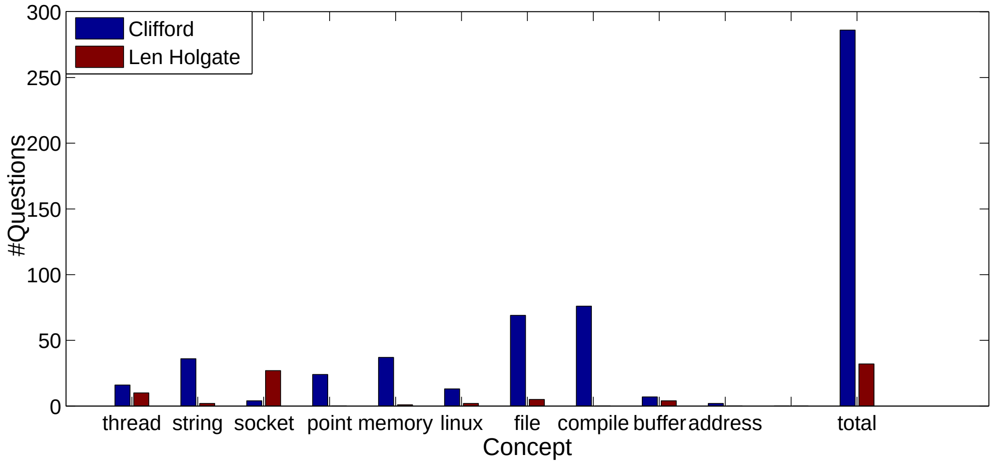


Figure 1.5: Characteristics of Answerer Clifford and Len Holgate in Stack Overflow: x axis describes 10 widely used concepts in C language extracted from Stack Overflow, and y axis describes the number of questions answered by an answerer which contains that concept. The column "total" in x axis describes the total number of questions answered by an answerer.

1.4 NMF as an Alternative Approach to Ranking:

Link-based ranking algorithms [2, 33] have been widely used for web and other networks to infer quality/popularity. Both PageRank and HITS were developed for ranking web pages from a web reference graph. Nevertheless, these algorithms have also been applied extensively for a variety of other applications such as question-answer services, author-paper graphs, and others where a graph can be deduced from the data set. The intuition behind HITS has been explained in terms of hubs and spokes as two values are inferred for each node. HITS has also been used extensively for ranking in other applications although it is not clear whether the same intuition carries over. It is essential to if we can understand these algorithms mathematically in a general manner so that the results can be better interpreted and understood for different applications.

For this work, we generalize the graph semantics in terms of two underlying concepts: In-Link Probability (ILP) and Out-Link Probability (OLP). Using these two concepts, the rank scores of nodes in a graph are computed. We propose the standard non-negative matrix factorization (NMF) approach to calculate ILP and OLP vectors. We also establish a relationship between HITS vectors and ILP/OLP vectors which enables us to better understand the HITS vectors associated with any graph in terms of these two probabilities. Finally, we illustrate the versatility of our approach using different graph types (representing different application areas) and validate the results. This work provides an alternative way of understanding HITS algorithm for a variety of applications. Details of this approach are presented in Chapter 6.

1.5 Contributions and Roadmap

The overall goal of this dissertation is to provide alternative mechanisms for searching the internet to get meaningful and useful answers without the user having to navigate and filter large numbers of web pages. In order to do so, we have chosen the Q/A networks as they provide abundant answers to a variety of questions from all walks of life. We have used Q/A features as well as participant characteristics (e.g., temporal) to assess answer quality using machine learning approaches.

Finding answer quality is critical for archived data sets which can be used as stored repositories to answer user searches. However, finding expertise of participants is extremely useful (and more challenging) for routing queries in real-time which is very important to these Q/A services. This problem boils down to an understanding of the characteristics of graphs inherent in these social networks, which have subtle differences from web reference graph, paper citation graph, and others. Hence, it is imperative to design effective and efficient ranking approaches for these Q/A networks to help users retrieve/search for meaningful information.

This dissertation consists of four goals - (i) to understand and design an effective machine learning approach to identify answer quality in CQAs (See Chapter 3), (ii) to understand and design an approach to identify generalists (See Chapter 4), (iii) to predict the expertise level of a user with respect to a concept (concept-based expertise) for a given CQA data set (See Chapter 5), and (iv) a different way of understanding traditional ranking (See Chapter 6). Related work and conclusions are shown in Chapters 2 and 7 respectively. We hope this dissertation motivates spawning of new ideas in these areas of research and paves the way for better search mechanisms than we have now.

CHAPTER 2

Related Work

Although Naver (`http://www.naver.com/`) was the first community question answering service (started in 2002), this phenomenon has grown significantly, and currently a large number of CQA services (both free and fee-based) exist. The fact that the CQA services have become so prolific in less than a decade is clearly indicative of its popularity and effectiveness as an *alternative to search*.

We review related work in this chapter under several topics that have close relationship with this work.

2.1 Answer Quality Analysis

We categorize previous work related to answer quality problem into two main categories: web page quality analysis and answer quality analysis.

2.1.1 Web Page Quality Analysis

Features have been used extensively for determining the quality of a web page and our problem is similar, but not exactly the same. In the context of the web, features have been classified by Strong et al. [34] into four categories: contextual, intrinsic, representational, and accessibility. Although link analysis [35, 36, 37] is widely used for ranking web pages, features have also been proposed to determine the quality of web pages by Zhu et al. [38]. In their approach, documents are first marked manually at different quality levels, such as "good," "normal" and "bad". Then, they build a classification model based on these features to predict the quality of other documents. Clearly, this classification evaluates the web page quality *globally*. However, our problem is slightly different as we need to assess quality of answers to each question.

2.1.2 Answer Quality Analysis

There is not much work on estimating answer quality in CQA services. Jeon et al. [39] is the first to describe the answer quality problem and propose a maximum entropy approach to predict the quality of answers using non-textual features. They do experiments using the *Naver* online community and demonstrate that it is possible to build a classification model to predict the answer quality. Shah et al. [40] use a number of automatically extracted features (most are meta-information features) from the *Yahoo! Answers* community to build a classification model. However, both of these papers consider only a single data set. Our focus is on identifying common features for multiple, diverse data sets with differing characteristics. We also show that our features can significantly improve upon earlier results. We also argue for a different approach for answer quality evaluation and establish the efficacy of our features. With a different focus, Harper et al. [41] discuss relationship between personal behavior and answer quality, and they conclude that a fee-based system receives better quality answers.

Other related work [42, 43] focus on finding relevant question-answer pairs from Q/A archives for a new query. Both papers integrate user feedback and interactions information (in addition to features) to predict the relevant question-answer pairs. Their focus is to identify similar questions along with their answers. Our research problem is somewhat different in that we are identifying the best answer from the answers given for that query. Our approach also differs in that we are interested in identifying generic features that can be used for diverse data sets. Moreover, none of the related papers propose the use of temporal features in the context of predicting or ranking answer quality.

Our focus is on automatically ranking answer quality for each question so that we cannot only infer the best answer but also rank all answers to facilitate retrieval of top-k answers. Our approach is also feature-based, but proposes generalization of features to include the temporal aspect as it seems important for these dynamic services. We also rank *all* answers using a learning to rank model and establish its appropriateness for this problem. (See Chapter 3)

2.2 Expertise Detection

With the widespread use of question answer services (or CQAs), the problem of identifying experts is becoming important. We categorize existing approaches related to this problem into three main categories: unsupervised expertise detection, semi-supervised expertise detection, and other approaches.

2.2.1 Unsupervised Expertise Detection

Some traditional IR techniques have been applied to CQA data sets to identify users' expertise. Efforts by [44, 30] build a term-based expertise profile for each user and rank expertise based on the relevance scores of their profiles for a question by using traditional IR models. However, since these IR techniques only identify related or similar users for this question, they do not properly capture the notion of expertise rank. Zhang et al. [31] propose four methods to identify expertise (or rank users in expertise order): number of answers given by a user (#Answers), Z_Score measure, PageRank, and HITS authority. Note that none of these methods use the contents of questions or answers. #Answers merely uses the number of answers given by a user as the quality score and Z_Score considers both the numbers of questions and answers given by a user to determine his/her expertise. Their experiments show that Z_Score has higher accuracy as compared to the other three methods. However, Z_Score is not very useful (and even representative) for CQA services because most of these are interested in a few top experts. Since these few top experts do not ask any (or very few) questions, the number of questions is not useful in determining users' expertise. Our experiments clearly show that rank results using Z_Score are very similar to #Answers. Zhang et al. [31] also use the PageRank [2] and HITS (authority) [33] score as user's expertise score. However, without considering the user's answer quality, the accuracy of these link-based algorithms is not likely to be high.

Jurczyk et al. [32] use HITS authority score as user's expertise score and compare evaluated user's expertise rank using several meta-data information from CQA data sets as the ground truth. Since we use these meta-data information as part of our approach, we cannot use their approach for the ground truth. Beyond the linkage mining techniques, other methods have also been used to identify experts. In [45], an entropy model using information theory is proposed to identify authoritative users from a graph. Expert users are those who have the most effect on the graph entropy when they are removed from that graph. The entropy of the whole graph is calculated and then the nodes in the graph are removed (one at a time) to test the change to the graph entropy. In the end this method will achieve a ranked list of node.

Another effort [46] that uses the ask-answer paradigm is the exchange of emails in a group/community. The graph generated by emails describes the email communication relationship among users. Since users always ask or answer questions by email, finding expertise of users in an email graph is closely related to our work. Campbell et al. [46] and Dom et al. [47] use HITS and in-degree method to rank users' expertise. They apply link-based algorithms to both a synthetic graph and a small email graph to rank correspondents according to their out-degree of expertise on subjects of interest. In their experiments, HITS authority shows higher accuracy than in-degree algorithm.

However, all the link-based methods or entropy based approach only consider the graph structure without using the contents of questions or answers; however, we believe that in Q/A communities the answer quality plays an important role to identify the users' expertise. Thus, taking into account both graph structure and domain information, we propose the ExpertRank framework to identify users' expertise in CQAs. (See Chapter 4)

2.2.2 Semi-Supervised Expertise Detection

Liu et al. [48] use TrueSkill [49] and the SVM models [50] to rank users' expertise score. They assume that given a question, its best answerer b has a higher expertise level than its asker a and other answers. Thus, they extract pairwise comparison for each question as the training data set using $|A|$ as the number of answers for each question. Then, they train semi-supervised learning models, such as TrueSkill [49] and SVM [50], to detect users' ranking score. In addition, Bian et al. [51] propose a semi-supervised coupled mutual reinforcement framework for calculating user's reputation and content quality. However, these approaches have the following issues. First, as compared with unsupervised expertise detection, the time complexity of these semi-supervised approaches is very high making them not suitable for large data sets. Second, TrueSkill and SVM models study the $|A| + 1$ pairwise comparison from each question because their approaches only distinguish the rank order of asker, best answerer and other answerers. We believe that in Q/A communities the answer quality plays an important role to identify users' expertise. However, since in the CQAs each answer receives a different quality score by considering the contents of questions or answers, we can also compare each answer (See Table 4.1) so that there are $(|A|! + |A|)$ comparison pairs for each question. If a question receives a large number of answers (e.g., 635 answers for one question in Stack Overflow and 96 answers for the other question in Turbo Tax), the number of comparison pairs are extremely large so that TrueSkill and SVM models cannot be applied to these applications. Since these information in the link-based algorithms can be easily used, we only focus on link-based approaches in Chapter 4.

2.2.3 Other Approaches

There are also some research papers on expertise analysis which is not directly related to this work. Pal et al. [52, 53] extract six features from CQAs by considering the user's motivation and ability to help other users and build learning models, such as SVM and DTree, to predict *potential* experts. We want to use quality information rather than motivation and ability to help. Also, we mainly use link-based algorithms to identify user's expertise.

2.3 Concept Rank Analysis

For this topic, existing related work are categorized into two main categories: specialist vs. generalist analysis and expertise analysis in online community. Since expertise analysis in online community was addressed in Section 2.2, we show some related work for "specialist and generalist" problem in the Q/A community.

To the best of our knowledge, few paper discusses the specialist and generalist problem in Q/A community. However, this problem is widely discussed in medicine area since these researchers analyze the respective role of generalist and specialist physicians in the care of patients. Ayanian et al. [54] and Harrold et al. [55] conclude that cardiologist (specialist) is more certain about key advances in the treatment of myocardial infarction than are family and internists (generalist). However, Rose et al. [56] and Landon et al. [57] are demonstrated that generalists seems to provide care of equal quality to specialists. Until now, in the medicine area, there is no clear answer for this problem. In Q/A community, our conclusion is given a question, both specialist and generalist are useful to answer that question. If questioner ask a question in special area, specialist should be better than generalist to answer that question; however, if a questioner ask a general question, generalists' answers seem to be equal to or sightly better than specialists' answers. (See Chapter 5)

2.4 NMF as an Alternative Approach to Ranking

We categorize existing work related to our problem into two main categories: ranking approaches and non-negative matrix factorization.

2.4.1 Ranking Approaches in Graphs

PageRank [58] and HITS [33] algorithms are the most widely used approaches to measure a web page's quality for search. PageRank is an iterative algorithm and in each iteration PageRank simulates a web user randomly surfing the web page. The final PageRank score of a web page describes the probability of this surfer visiting a particular web page. HITS algorithm models the web as two types of pages: hubs and authorities. Hubs are web pages that link to many authoritative pages and authorities are web pages that are linked to by many hub pages. HITS is also an iterative algorithm that updates the hub and authority score of a page based on the scores of pages of its neighboring web page. These two algorithms are also widely applied to a number of applications. For example, the graph generated by a collection of emails describes the email communication relationship between users. Campbell et al. [59] and Dom et al. [60] apply HITS and in-degree approaches to a synthetic graph as well as a small email graph to rank correspondents according to their expertise on subjects of interest. In their experiments, HITS authority score shows higher accuracy than that of the in-degree algorithm. PageRank and HITS are also applied to the ask-answer graph [31, 32] to measure the users' expertise score and these two algorithms provide more accurate results than other algorithms. In addition, PageRank and HITS algorithms have also been extended to bring node's order to the biological graph [61], paper co-citation graph [62] and other social graphs. Although these rank algorithms are widely used for various applications, the intuition behind these rank algorithm for these applications is not as clear. Chapter 6 will revisit HITS algorithm for social graphs and provide an alternative intuition of HITS vectors.

2.4.2 Non-negative Matrix Factorization

Non-negative Matrix Factorization (NMF) has been widely studied in the data mining and machine learning areas since the initial work of Lee et al. [63]. It has been applied to a number of different areas such as pattern recognition [64], multi-media data analysis [65], text mining [66], and DNA gene expression analysis [67]. Extensions of NMF have also been developed to accommodate various cost functions as needed in different data analysis problems, such as classification [68], collaborative filtering [69] and clustering [70]. In Chapter 6, we will explore the standard NMF approach to solve the rank problem in a social graph context and to the best of our knowledge, this is the first paper to apply the NMF approach for the social graph rank problem. Extensions to this approach by using different cost functions to improve rank accuracy are also possible.

In Chapter 6, we establish a relationship between the HITS algorithm and the vectors of non-negative matrix factorization. We also show theoretically that the HITS hub and authority scores corresponding to the vectors obtained by non-negative matrix decomposition. This relationship provides another way of understanding the scores computed by HITS algorithm and provides an alternative intuition for many social network problems.

CHAPTER 3

Answer Quality Problem

Community Question Answering (or CQA) services (aka Q/A social networks) have become widespread in the last several years. Currently, *best* answers in CQA are determined either manually or through a voting process. Many CQA services calculate activity levels for users to approximate the notion of expertise. As large numbers of CQA services are becoming available, it is important and challenging to predict *best* answers (not necessarily answers by an expert) and rank *all* answers for a question using machine learning techniques. Work in this regard, typically, extracts a set of features (textual and non-textual) from the data set and feed them to a classification system to determine the *best* answer.

This chapter first identifies the importance and use of temporal features, different from the ones proposed in the literature, for predicting the quality of answers. The suitability of temporal features is based on the observation that these systems are dynamic in nature – in terms of the number of users and how many questions they choose to answer over an interval. We analyze a small set of temporal features and demonstrate that these features work better than the ones used in the literature using traditional classification techniques. Second, we also argue that the classification approaches measuring precision and recall are not well-suited as the CQA data is unbalanced, and quality of ranking of *all* answers need to be measured. We propose the use of learning to rank approaches, and show that the features identified in this work work very well with this approach. We use multiple, diverse data sets to establish the utility and effectiveness of features identified for predicting the quality of answers. The long-term goal is to build a framework for identifying experts, at different levels of granularity, for CQA services.

3.1 Introduction

Community Question Answering (or CQA also termed Q/A social networks) is gaining momentum in the last several years. It is seen as an alternative to search as it avoids dealing with large number of answers/results as well as the task of sifting through them (although ranked) to get at the desired information. Both general purpose and topic-specific communities are growing in numbers for posting questions and obtaining direct answers in a short period of time. Yahoo! Answers (`http://answers.yahoo.com/`) (Y!A), for example, provides a broad range of topics where as Stack Overflow (`http://stackoverflow.com/`) (SO), and Turbo Tax Live (`https://ttlc.intuit.com/`) (TT) are quite focused and domain-specific.

In contrast to the traditional search engines such as Google [71], CQA services provide an alternative paradigm for seeking targeted information. These communities allow questioners to post questions and others to provide answers. These communities have become quite popular in the last several years for a number of reasons. First, because of the targeted response from users with knowledge or experience, these answers are likely to be more useful and easy to understand for the questioner. Second, the question answering communities also provide a consolidated communication environment where answers to related questions can also be viewed. This environment facilitates multiple answers (likely from different perspectives) and discussion (in the form of comments, threads) which can benefit the questioner (and others as well). It is also possible for the questioner to interact with the answerer (by email or other means) for clarification and advise. This paradigm, although quite different from the instantaneous search for stored information, is likely to provide the questioner with useful information as a substitute for search. Finally, the forum provides an incentive for people to showcase their expertise and in the process get recognized by the community. For this reason, many CQA services allow the questioner to flag the *best* answer from the set of answers. Some CQA services have a voting mechanism to rank the responses. The notion of an expertise level exists in some services and is based on several factors: number of best answers given by a user, votes obtained for answers, etc.

Although Naver (`http://www.naver.com/`) was the first community question answering service (started in 2002), this phenomenon has grown significantly, and currently a large number of CQA services exist that support this paradigm. The fact that the CQA has become prolific in about a decade is clearly indicative of its popularity and effectiveness as an alternative to search. As the number of CQA services grows, they are also available as archives motivating new approaches for searching and selecting answers that best match a question. *In order to do this, it is critical to be able to automatically evaluate and predict the quality of existing answers with respect to a question whether in a focused topic or in a broader category.* This is even more important when we have to deal with a large number of answers. For example, in Y!A, some questions have large number of answers.

Most of the extant work for evaluating/predicting the quality of answers are based on features extracted from the data set, and the use of a traditional classification approaches for predicting the *best* answer. There are some efforts aimed at predicting information seeker satisfaction as well [72]. This paper does not address human aspects of this community. Jeon et al. [73] extract 13 *non-textual* features from the Naver data set and build a maximum entropy classification model to predict the *best* answer. Along the same lines, Shah et al. [1] extract 21 features (mainly non-textual) from Y!A and use the logistic regression and classification model to predict the *best* answer.

In this work, we propose a set of *temporal* features and show that they are effective for predicting the quality of answers. Based on our analysis of diverse data sets (Y!A, SO-C, SO-O), the answerer's current state seems to be important and has a bearing on answer quality. It is not difficult to discern that these communities are dynamic in nature: number of users vary over time, users' gain experience as they answer questions, and the current set of active users is relevant to determine the answer quality. We elaborate on the features, intuition for choosing them, and their extraction. We use both traditional classification and learning to rank models approaches to establish the effectiveness of these features. We compare our results with features and classification methods used in the literature.

We also argue that the classification approaches currently used are not well-suited for this problem. First, the data set is highly unbalanced with the ratio of best answer and non-best answer being very small (less than 0.1 in Y!A). This makes it difficult to build a good classification method to predict the *best* answer. Also, ranking of all answers for a question and its accuracy is equally important which is not captured by traditional classification approaches.

Finally, features [73, 1] available/used from the different CQA data sets are also different. For example, the Yahoo! Answers community has well-defined user levels whereas Stack Overflow data set does not have this information. Since learning to rank model can integrate different features into a unified ranking framework to retrieve high quality answers, we propose the use of learning to rank framework for CQA services. Based on the above observations, we argue that the learning to rank approaches are better-suited for this problem. This is further elaborated in Section 3.6.

The focus of this work is two fold: (i) to identify and analyze a set of features that can be used for multiple and diverse (in terms of topics covered and other characteristics) data sets for predicting answer quality and (ii) to propose and demonstrate the appropriateness and applicability of learning to rank models for evaluating Q/A data sets. We want to clarify that we are not proposing a new learning to rank model but argue for that approach and use one to substantiate our case.

Contributions: The contributions of this work are:

- Identification and justification of *temporal* features that are relevant to multiple, diverse data sets.
- Demonstrate superiority of *temporal* features for answer quality as compared to the features used in the literature using the same classification approach.
- Argue for the learning to rank model as a better approach for measuring the quality of answers for CQA data.
- Extensive experimental analysis of three different diverse data sets using the proposed features and a learning to rank model.
- Results confirming that the proposed features as well as the learning to rank model are effective in determining answer quality.

Road Map: Section 3.2 motivates and defines the problem of answer quality analysis. Section 3.3 introduces the need for temporal features, new features, and their computation. In Section 3.5, we compare our features with extant work for inferring answer quality. Section 3.6 introduces the need for learning to rank approach used along with detailed experimental analysis and discussion of results for three data sets. Section 3.8 has conclusions.

3.2 Problem Statement

Although there are many CQA services for the same purpose, the approaches taken for interaction and how users to communicate with each other vary considerably. This has a bearing on the features that can be extracted and hence is important to understand the differences to separate generic features from service-specific (or paradigm-specific) features.

3.2.1 Answer Quality Characterization

Based on our analysis of these communities, we can broadly categorize existing online CQA services and the interaction by the questioner and the answerer with the service as described below. Of course, the goal of each service is to provide high quality answers to the user.

Expert Selection Approach: This approach uses strict guidelines for adding a person as an expert to the Q/A community. Before a potential expert joins the Q/A community, s/he needs to write a detailed self-introduction that includes his/her credentials. The staff of the service evaluate each person's self-introduction, background, and questions answered to determine whether or not allow this person to join the Q/A community. Only after verification (as an expert), will this person be allowed to answer questions. In this environment, a question has only one answer and provide a good/quality answer for a question. Examples of such communities include: *AllExperts* (`http://www.allexperts.com/`) and *MadSci Network* (`http://madsci.org`). For the *Allexperts* web site, one is expected to fill an application form which asks for experiences, organizational affiliation, awards received, and publications in relevant areas. After choosing an expert, the community will further evaluate that expert from several aspects, such as knowledgeability, clarity of response, politeness, and response time. Based on this, a questioner can direct his/her questions to a real expert and receive (quality) answers from these experts. Furthermore, in order to retain these experts, these communities provide incentives in the form of bonus, or as in *Google Answers* (`http://answers.google.com/answers/`), the questioners can choose/quote his/her price for answers. This makes the service sustainable.

Wikipedia Approach: In this approach, for each question, the first contributor will answer the question and others are allowed to modify an earlier answer to add their revision/opinion. In this approach a question has only one answer but is the result of refinement by many answerers. This approach avoids information redundancy and is beneficial to the questioner as it provides a consolidated final answer *Answers* (`http://wiki.answers.com/`) is an example of this approach. In order to confirm the quality of an answer, after the other users revise the answer, *Answers* will permit users to give a trust score to the contributors. Greater trust is placed with the contributor with a higher trust score.

User Vote-Based Approach: As the name suggests, this approach evaluates the quality of an answer by the number of votes it receives. This method is widely used in the Q/A community, but different communities use different strategies. For the user vote-based approach, Yahoo! Answers uses a three step process to determine the best answer: i) post question and optional description to a specific topic category, ii) Answers are collected and listed in random order for voting (no additional answers are allowed at this stage), and iii) After some fixed period time, the answer with the highest number of votes is chosen as the *best* answer. In other Q/A communities, such as Stack Overflow, Blurtit (`http://www.blurtit.com/`), and Turbo Tax Live, Answerbag (`http://www.answerbag.com/`), there is no clearly-defined time period. A user can answer a question, vote on the answers and choose the best answer at the same time.

Questioner Satisfaction Approach: In this approach, only the questioner will decide an answer's quality. If the questioner is satisfied with the answer, s/he will choose it as the *best* answer, provide a feedback rating, and also include textual feedback. The *best* answer resulting from this approach can be quite subjective. This method is also used in Yahoo! Answers.

In many CQA services, the above-mentioned approaches are *not* mutually exclusive. For example, Yahoo! Answers uses both questioner satisfaction approach and user vote-based approach to ascertain the answer quality. Stack Overflow allows a user to vote the best answer for a question or modify an earlier answer to add their opinions. Many of these communities also enroll some real experts to periodically post questions and answer questions. In order to develop automated techniques for predicting the *best* answer, and ranking of all answers, it is important to understand the above differences and the basis used for determining *best* answers.

3.2.2 Problem Definition

This work focuses on the user vote-based approach as it seems to be widely used, is consensus-based, and easy to support using the web framework.

Given a question q_i, and a set of its answers $\{A_{i_1}, A_{i_2}, ..., A_{i_n}\}$, our goal is to calculate the answer quality for each answer using temporal (and other) features and choose the highest ranked one as the *best* answer. Ranking of all answers will allow us to order answers with respect to quality. We will use the vote-based approach for comparing our results with the actual votes given for each answer. In case of a tie, all tied answers are *not* considered as *best* answers. The original order of extracted answers is used. *We have purposely chosen this approach to show that even the worst case scenario results in good accuracy.*

It has been established in [1] that manual assessment of quality of answers using a number of subjective criteria is comparable to the vote-based best answer. Hence, we test the accuracy against the vote-based approach. Ranking of answers can be used for searching the archives to identify the best (or a few) answer that is consistent with the voted scheme.

3.3 Feature Set and Its Analysis

Features are widely used and have been shown to be effective for analyzing documents, images, and web pages, to name a few. So, it is not surprising that this approach has also been used for analyzing questions and answers for predicting not only answer quality but other aspects such as questioner satisfaction as well.

A number of features (mostly non-textual) have been identified in the literature for predicting answer quality. In [73], they show that non-textual features, such as answerer's acceptance ratio, questioner's self evaluation, number of answers, click counts, users' dis-recommendation, and others (there are a total of 13 feature which are extracted primarily from the best answer; some of these features are specific to the Naver data set) can be systematically and statistically processed to predict the quality of answers. They assume that the user will randomly generate a "Good" or "Bad" label to each answer. Thus, they build the maximum entropy and kernel density functions to predict these labels. Their experiments conclude that it is a possible to build a prediction model to predict the "Good" or "Bad" answers for the online Q/A community.

In [1], a number of features (again, most of them being non-textual) are used to train a number of classifiers to predict the best answer. They initially perform a manual assessment of answers using the Amazon Mechanical Turk (`https://www.mturk.com/`) and establish that the qualitative subjective criteria used for establishing the best answer using the Mechanical Turk is comparable to the best answer chosen in the service. Actually, they propose and extract 21 features for each question and answer in the Yahoo!Quest data set. Some of the features used are: length of question's subject, information from asker's profile, reciprocal rank of the answer in the time order of answers for the given question, and information from answerer's profile. As this research is the closest to our work, we compare our results with their results in Section 3.5.

3.3.1 Need For Temporal Features

Although most of the earlier work (discussed in Section 2) focused on non-textual features, our analysis of various CQA services and modality of their usage indicate that there is a strong temporal component that plays an important role and influences answer quality. The number of users, for example, as well as their activity level varies significantly over time. As an illustration of this point, consider Figure 3.1.

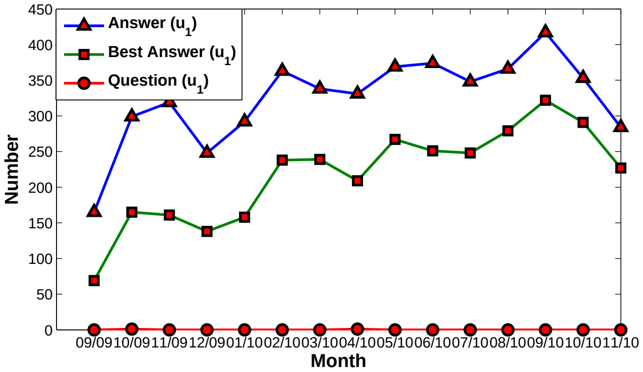


Figure 3.1: Monthly User u_1 Activity Level from SO Data Set

Figure 3.1 shows a single user u_1 activity from the Stack Overflow data set. This user, registered in February 2009, is a software engineer with 10 years of experience and is specialized in Java and Agile programming. Figure 3.1 shows, for each month, the total number of: answers, best answers, and questions by that user. The number of answers increased from 248 in September 2009 to 417 by November 2010. This user never asked a question in the time period shown. The same is true for the best answer. As can be seen, the activity level fluctuates considerably over time and this is to be expected (and is representative of) of a free CQA service where users provide answers at their convenience. Although we have shown a single user, we have observed this phenomena for a significant number of users in the data sets we are using. This seems to be true irrespective of the topic or the focus of the group.

Since it is difficult to show this statistic as an aggregate for all users in a data set, Figure 3.2 shows the maximum numbers of: answers, best answers and questions by any user for each month between September 2009 to November 2010. The fluctuation of these values over this period gives some indication of the widespread nature of user activity changes over a period of time.

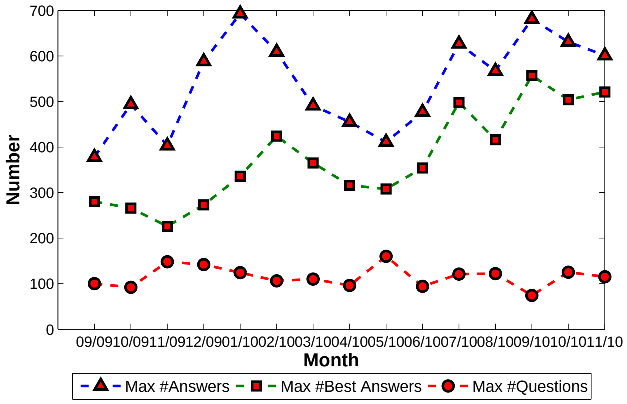


Figure 3.2: Monthly User Activity Level from SO Data Set

An important aspect of the above observation is how it affects the features extracted for a data set. To illustrate this, Figure 3.3 shows the feature – best answer ratio (tABA_Ratio) (elaborated below) for the *same* user u_1. It is easy to see that the best answer ratio tABA_Score (of an answerer) changes over a period of time (actually increases in this case as the user seems to acquire experience in providing better and easy to understand answers). Contrast this with the same feature value calculated for the entire period (as ABA_Ratio) instead for each month which is also shown as a constant line in Figure 3.3. Use of this feature value (instead of the temporal one) misses out on the subtle changes to the feature and hence to the accuracy computation.

In fact, temporal feature can be viewed as a generalization of the feature. Instead of computing a feature over the entire data set, it is now computed using smaller relevant intervals to reflect feature values more accurately. This is done for those features that are affected by the time component. Not all features have a temporal component (e.g., answer length and the similarity score between answer and question). In addition to activity levels, we have also observed user interest shift (or drift)[1] over time in all the three data sets. This is to be expected as the focus of questions is likely to change over a period of time. Also, what questions a user chooses to answer is also likely to change over a period of time (either based on topic drift or based on user interest drift or both).

Our data analysis has also indicated interest drift (or topic drift) for a user. The interest drift can be analyzed using a correlation function, $ID(s)$, to evaluate the relative overlap between words (or concepts) in answers by the same user for questions in the same topic category in two distinct intervals. We model an answerer's behavior

[1] One user in stack overflow community explains this as follows: *As I used Java language to build my web project in the last quarter of 2009, I am very familiar with that language; but I changed my job and in my new job I develop database applications using the C language. Now, I am interested in C programming questions.*

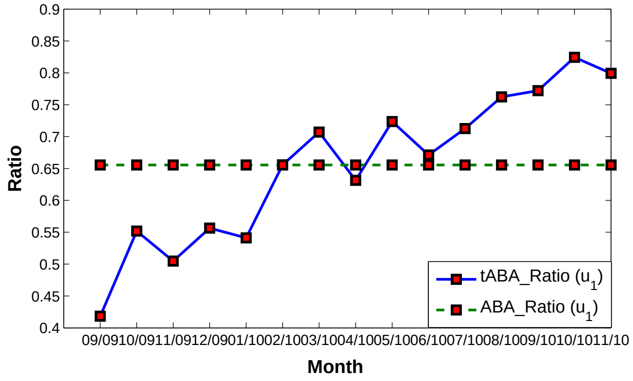


Figure 3.3: Monthly tABA_Ratio for a user u_1 ($\Delta t = 1$ month) vs. ABA_Ratio for a user u_1

as a vector $v(s, u_i)$. $v(s, u_i)$ is defined as $< c_1, c_2, ..., c_n >$, where c_j is the number of questions assigned to the word j from the answers given by the answerer u_i in the interval s. In our experiment, one interval is about month. s_1 is September 2009, s_n is November 2011 and s_i is one month between September 2009 and November 2011. $v(u_i, s_i)$ describes the vector given by the answerer u_i in September 2009. Thus, we define the interest drift for u_i between s_i and s_j.

$$ID(u_i, s_i, s_j) = \frac{v(u_i, s_i) \cdot v(u_i, s_j)}{\parallel v(u_i, s_i) \parallel \times \parallel v(u_i, s_j) \parallel} \tag{3.1}$$

where $v(u_i, s_i) \cdot v(u_i, s_j)$ is the dot product of the vectors for intervals s_i and s_j, $\parallel v(s_i) \parallel$ is the l_2 norm of vector $v(u_i, s_i)$, and $\parallel v(u_i, s_j) \parallel$ is the l_2 norm of vector $v(s_j)$. The l_2 norm of a vector is calculated as: $\parallel v(u_i, s_i) \parallel = \sqrt{\sum_{i=1}^{n} c_i^2}$.

Figure 3.4 captures three users u_1, u_2, u_3's interest drift between the periods of October 2009 and November 2010[2]. Figure 3.4 indicates that (i) the user's interesting

[2] We select 57 users which answered more than 10 questions between October 2009 and November 2010. Figure 3.4 only shows three users u_1, u_2, and u_3's interesting drift. The difference of user u_2's interesting drift is sharpest around these 57 users; the difference of user u_3's interesting drift is

drift is different according to the different users (e.g., user u_2's interesting drift changes fast, but user u_3's interesting drift changes slow). (ii) User u_1's interest drift score which represented the common user's drift decreases from 0.98 to 0.90. This indicates that user's current interest is important to evaluate that user's answer quality.

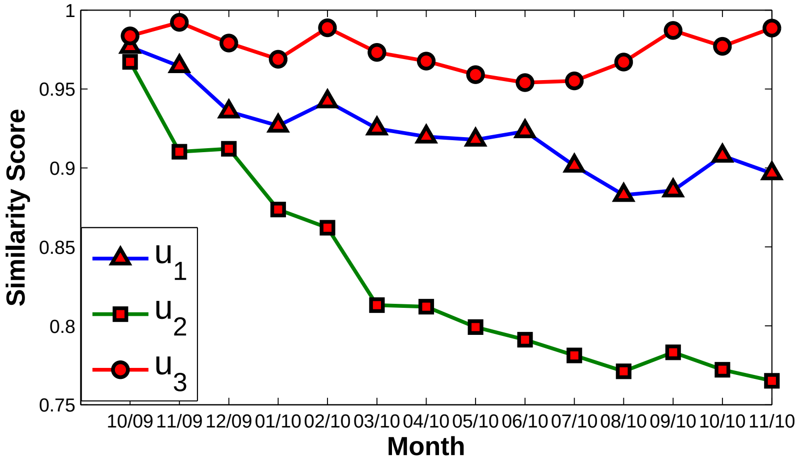


Figure 3.4: We calculate the similarity between s_i and s_1.

Based on the above observations, we believe that answerer's temporal characteristics can significantly contribute to the quality of an answer.

3.3.2 Need for Δt and its Computation

Below, we propose a number of features that are temporal in nature. All the temporal features use a time interval (termed Δt) over which the feature is computed from the data set. we first discuss the intuition behind each feature, its role, and how it is computed for a data set.

relative smallest; the difference of user u_1's interesting drift represent the difference of the common user's drift in stack overflow data set.

Table 3.1: Average Response Times of Data sets

Data set	Avg. time for First Answer	Avg. time for Best Answer	Avg. time for Last Answer
Y!A	00:21:15	21:09:27	2 days 12:57:50
SO-C	01:12:33	9 days 20:11:17	12 days 20:32:16
SO-O	01:16:12	9 days 21:57:17	13 days 16:42:18
TT	26:06:29	11 days 19:14:29	24 days 25:51:19

[1] SO-C involves all the questions tagged as “C”.

[2] SO-O involves all the questions tagged as “Oracle”.

Due to the dynamic nature of Q/A networks, we need to capture the activity levels of users' around the time when a question is asked and the time period over which that question is answered. Thus, our period of interest starts when a question is asked and ends at the time the last answer is given for that question. This interval is used for determining the quality of an answer given by a user. This is based on the intuition that the quality of answer varies over different periods of time even for the *same* answerer (as can be seen in Figure 3.1, 3.2, 3.3).

It is also important that this interval (Δt) is chosen properly and is relevant to a data set. Δt needs to capture the current flow of activity in the data set. Thus, for a given data set, we calculate Δt as the interval starting from the time at which the question is asked to the *average time* it takes to receive the last answer in that data set. Note that the average time is specific to a data set.

Table 3.1 shows the average response time for the three data sets for: the first answer, the last answer, and the best answer. Yahoo! Answers receives the last answer in about 2.5 days on the average. In contrast, Stack Overflow and Turbo Tax community take nearly 14 and 25 days respectively. Thus, we choose Δt to be 3, 14, and 25 days, respectively, for Yahoo! Answers, Stack Overflow, and Turbo Tax. It is interesting to note that the best answer seems to come in much earlier than the last answer.

The appropriateness of Δt is elaborated in Figure 3.5. For the Stack Overflow data set, Figure 3.5 plots the aggregate percentage of last answers received over a 30 day period (X-axis) for all questions in the data set. The Y-axis shows the percentage of last answers and the cumulative percentage of last answers for each day. We can see that more than 48% of last answers were received on the *first* day. As the time progresses, the last answer percentage for each day decreases significantly. At 14 days (average for that date set), we can see that nearly 70% of questions would have received the last answer.

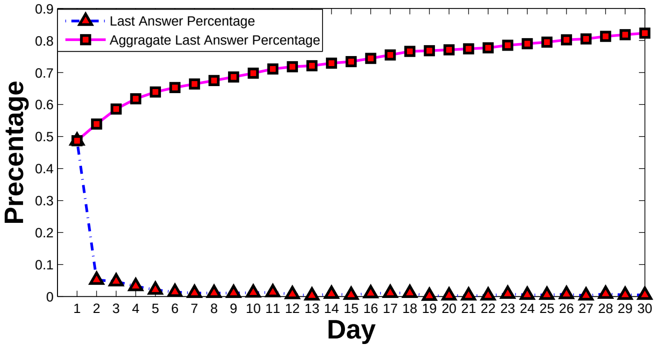


Figure 3.5: Last Answer Percentage for all questions (for 30 days, SO data set)

3.4 Proposed Temporal Features

The following temporal features are identified and computed for each data set. Below, we describe each feature, its relevance, and how it is computed. Features starting with a t are temporal features computed using the Δt discussed earlier.

tAA_Count(u_i,Δt): Number of answers given by u_i in the interval Δt.

tABA_Count(u_i,Δt): Number of best answers given by u_i in the interval Δt.

tAQ_Count(u_i,Δt): Number of questions asked by u_i in the interval Δt.

Best Answer Ratio (tABA_Ratio) for an answerer: For a given answerer u_i and an interval Δt, the tABA_Ratio is the number of best answers to the total number of answers given by that user in that interval. Formally,

$$tABA\_Ratio(u_i, \Delta t) = \begin{cases} 0 & tAA\_Count(u_i, \Delta t) = 0 \\ \frac{tABA\_Count(u_i, \Delta t)}{tAA\_Count(u_i, \Delta t)} & otherwise \end{cases} \tag{3.2}$$

where tABA_Count(u_i,Δt) is the number of best answers by user u_i during Δt and tAA_Count(u_i,Δt) is the number of answers by user u_i during Δt. tABA_Ratio value $\in [0, 1]$ captures the quality of user's answers. A tABA_Ratio of 1 indicates that each answer is a best answer and a tABA_Ratio of 0 indicates that none of his/her answers are best answers. Fluctuations in tABA_Ratio can also indicate user's effectiveness for quality of answers over a period of time or even interest drift (due to job change, etc.).

Question Answer Score (tAQA_Score) for an answerer: This measure classifies each user as: questioner only, answerer only, or a combination thereof. Again, this score can have significant influence over the quality of answers. For example, a user who is an answerer, has a high tABA_Ratio, is likely to provide a better answer. This score is computed as:

$$tAQA\_Score(u_i, \Delta t) =$$

$$\begin{cases} 0 & |A(u_i, \Delta t)|\, and\, |Q(u_i, \Delta t)| = 0 \\ \frac{|A(u_i,\Delta t)| - |Q(u_i,\Delta t)|}{\sqrt{|A(u_i,\Delta t)|^2 + |Q(u_i,\Delta t)|^2}} & otherwise \end{cases} \tag{3.3}$$

where A and Q indicate, respectively, answers and questions by that user in the specified interval and $|value|$ represents the cardinality of $value$. tAQA_Score describes the level of user participation. A tAQA_Score of -1 indicates a questioner whereas a +1 score indicates an answerer. Along the same lines, we define the rest of the temporal features as follows:

Normalization of the values of some of the above features to the range [0,1] is be discussed in Section 3.6.

Figure 3.6 illustrates the computation of temporal features using Δt. Figure 3.6 shows user u_1 asking a question q_i at t_1 and u_2 answers this question in the interval $t_1 + \Delta t$. In order to rank u_2's answer for question q_i, we compute all of the temporal features indicated above for u_2 in the closed interval $[t_1,\ t_1 + \Delta t]$. As an example, between t_1 and $t_1 + \Delta t$, if user u_2 asks 1 question and answers 3 questions, we calculate $tAQA\_Score(u_2, \Delta t)$ as $\frac{3-1}{\sqrt{1^2+3^2}} = 0.63$. In the same way, we calculate other temporal features for this user u_2.

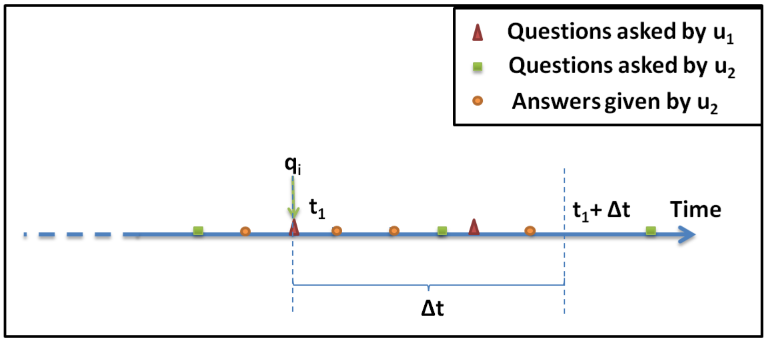


Figure 3.6: Computation of Temporal Features

3.4.1 Additional Features

In addition to temporal features, we also extract other features from the data sets. As these overlap with the features from the literature, we list all the features extracted and their descriptions in Table 3.2. These features are computed from the entire data set without using Δt. Temporal features start with t. Others can be understood as question- and questioner-related (start with a Q), answer- and answerer-related (start with an A), or both (starts with QA).

We extract a total of 22 features, 5 of which are temporal (for an answerer), 5 related to an answerer, 3 related to an answer, 4 related to a question and 5 to a questioner. Question and questioner features have also been used in the literature [1]. Our experiments show that they do not contribute to answer quality prediction as discussed in Section 3.6. We have shown in Section 3.6 that some of the features shown in the table (e.g., question and questioner) do not contribute to accuracy.

3.5 Evaluation

We analyze and use three diverse data sets (Y!A, SO-O, SO-C, TT) to establish the relevance of 22 features elaborated earlier. We do not use the question and questioner features for all experiments as they do not contribute to the accuracy as shown in Table 3.4 (See Section 3.6.2). We briefly discuss the data sets first to provide their unique characteristics. Table 3.3 shows some of the broader characteristics of the data sets used. A subset of these data sets are used for experiments as indicated below.

Y!A Data set: Y!A community contains 26 top-level topics and a number of sub-topics. The average number of answers for a question is about 10.11 across the entire data set (see Table 3.3). We use "Singles & Dating" category. The choice of this category is intentional to keep it significantly different from the categories of the other two data sets. In Y!A community any users can register a new user. After becoming a user in Y!A community, s/he can ask or answer questions in that community. In addition, once user's answer is posted in the community, only the Y!A staff can modify or delete that answer. In Y!A community the best answer have already been marked in every

Table 3.2: Summary of *all* Features

Feature	Description
Answerer Temporal Features	
tABA_Ratio	answerer's best answer ratio in Δt
tAQA_Score	answerer's question answer score in Δt
tAA_Count	answerer's number of answers in Δt
tABA_Count	answerer's number of best answers in Δt
tAQ_Count	answerer's number of questions in Δt
Answer Features	
A_Length	number of words in the answer
QA_Sim	cosine similarity between question and answer
E_Link	whether or not an embedded link is in the answer
Answerer Features	
AA_Count	number of answers given by an answerer
ABA_Count	number of best answers given by an answerer
AQ_Count	number of questions posted by an answerer
ABA_Ratio	answerer's best answer ratio
AQA_Score	answerer's QA score
Question Features	
QS_Length	number of words in question's subject
QC_Length	number of words in question's content
Q_Popular	number of answers for this question
Q_Comment	number of comments for this question
Questioner Features	
QA_Count	number of answers answered by questioner
QQ_Count	number of questions posted by questioner
QBA_Count	number of best answers answered by questioner
QQA_Score	questioner's question answer Score
QBA_Ratio	questioner's best answer ratio

Table 3.3: Complete Data set Characteristics

Data set	Questions	Answers	Users	Avg(#Answers)
Y!A	1,314,888	13,293,102	1,064,064	10.11
SO	1,467,066	3,469,270	559,676	2.36
TT	501,978	567,515	486,196	1.13

resolved question[3] so that we only choose resolved questions as the experimental data set.

SO Data set: This service is focused on computer programming topic. Unlike the Y!A service, SO allows a user to modify other user's answers. In another words, when an answerer wants to answer a question, s/he has two choices: modify former user's answer or provide a new one. This community has characteristics of both the wikipedia approach and user vote-based approach (discussed in Section 3.2). As a result, the average number of answers for each question is only 2.36 (See Table 3.3). In our experiments, we only consider the first user who posts the answer as the answerer, because in most cases the first user is likely to provide a larger contribution of the answer than those who revise. *This, again, corresponds to the worst case scenario to illustrate that our approach results in good accuracy.* Each question in this community is marked with a topic tag (e.g., "C" or "Oracle").

TT Data set: TT service only discusses tax-related issues. Since tax preparations are made mostly between January and April of each year, this community is very active during these months (that also explains the large average last answer time as our data spans more than one year.) TT community enrolls some real tax experts to answer questions. In this community, most of the users are mainly questioners, and are less likely to answer questions. Thus, the average number of answers for each question is only about 1.13 (See Table 3.3). Unlike other services, in this service, the same user may give more than one answer to a question. When the answerer gives an answer for a question, the questioner or others are allowed to write comments for this answer and the answerer may give another answer for the same question. This is

[3] "Resolved" is one kinds of label in Y!A community. If the question has been marked as "Resolved" label, the best answer has already been identified from this question (also see Section 3.2 to understand the process to identify the best answer from Y!A community).

in contrast to the other two data sets where one answerer can only give one answer to a question. For this data set we choose the answer which has the highest rating as the best answer.

For each data set, we randomly choose 1000 questions (from the relevant category) each of which has at least 5 answers. We use 100 of these as test data and the remaining 900 as training data. For each question, we retain the best answer and 4 other randomly selected answers. Thus, we use 1000 questions, 1000 best answers and 4000 non-best answers for our experiments as the test data set[4]. We perform 10 fold cross validation.

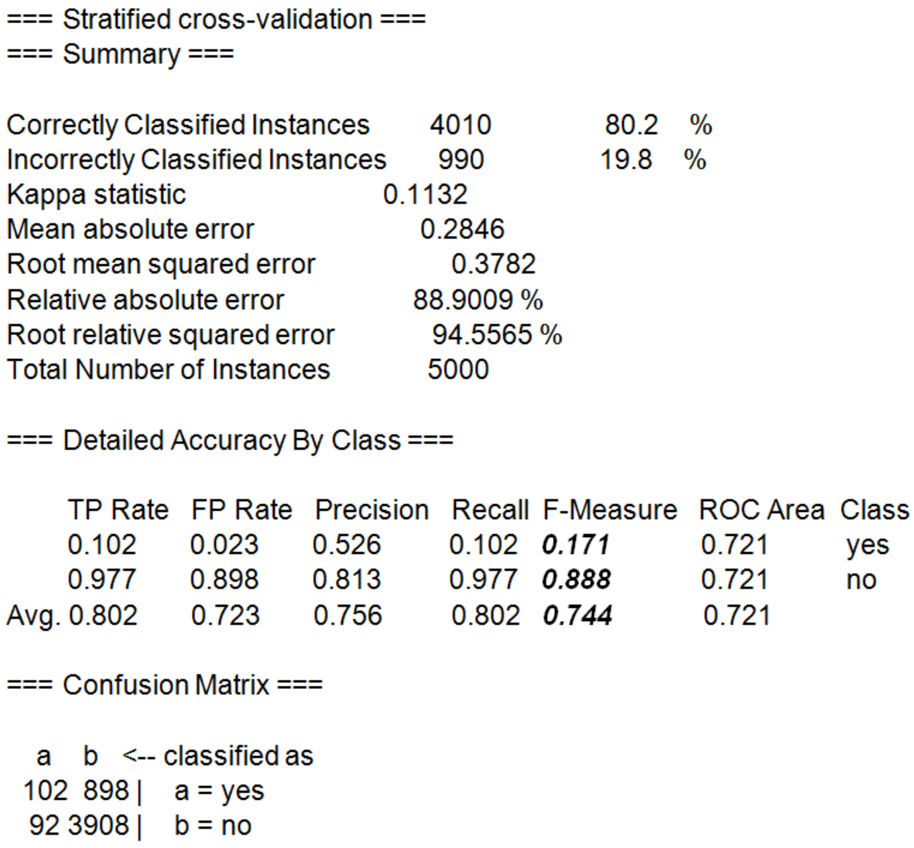
The above data sets serve the purpose of diversity – in terms of topics, mode of interaction, choice of best answer as well as the average number of answers per question. We believe that the diversity of our chosen data sets will stress the features for their effectiveness if they are to perform significantly better than the baseline.

3.5.1 Comparison with Earlier Work

In this section, we compare the prediction accuracy of answer quality using only temporal features with the features used in [1]. We use the same data set (Y!A) and classification approach. As described in their paper, we randomly choose 1000 questions in a topic category in which each question has at least 5 answers. For questions with more than 5 answers, we randomly remove non-best answers to bring the number of answers to 5. In Yahoo! Answers data set, each question-answer pair

[4] We have also done experiments by relaxing the 5 answer constraint and have obtained similar results (See Section 3.6). If we use a question which has 1 best answer and 4 non-best answer, we can easily analyze the accuracy of these approaches. Note that if we randomly choose an answer as the best answer, the accuracy will reach 20%. In this way, we can compare these approaches with the random approach.

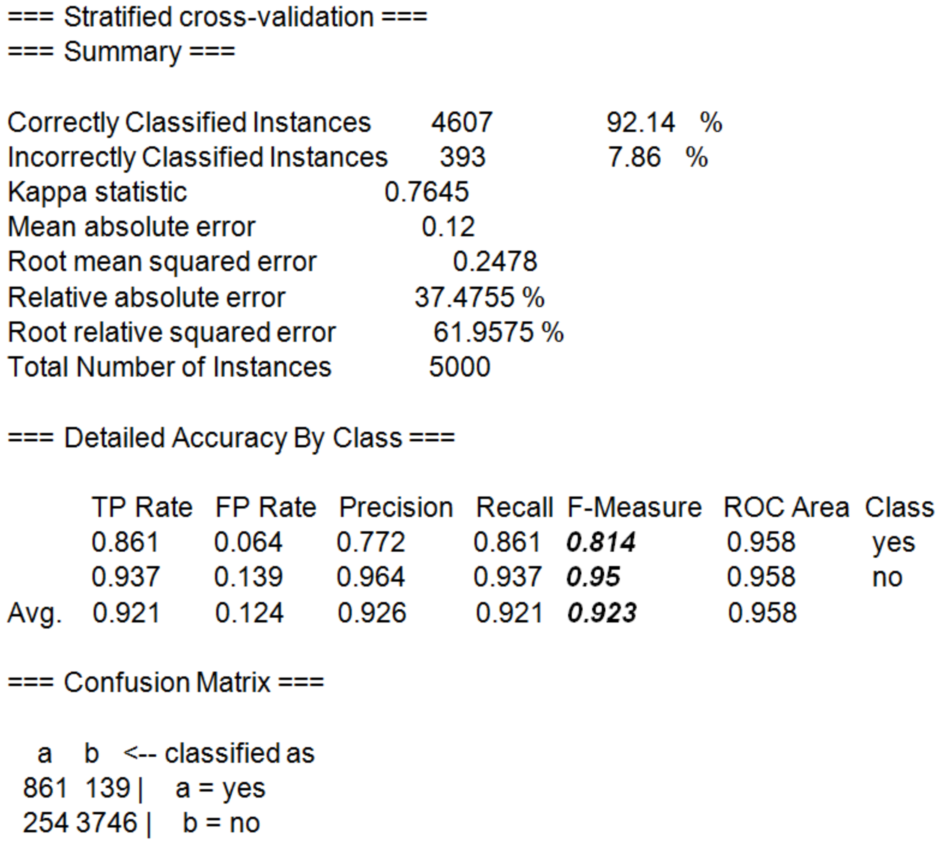
has been classified as “Best Answer” or “Non-best Answer”. Thus, we build the classification model to evaluate the accuracy of our proposed features. We extract all the 21 features reported in [1] for each question-answer pair and build the *same* logistic regression model using the Weka package (`http://www.cs.waikato.ac.nz/ml/weka/`). We use 10-fold cross-validation to calculate the accuracy for Yahoo! Answers data set.



```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        4010               80.2    %
Incorrectly Classified Instances       990               19.8    %
Kappa statistic                      0.1132
Mean absolute error                    0.2846
Root mean squared error                  0.3782
Relative absolute error               88.9009 %
Root relative squared error             94.5565 %
Total Number of Instances              5000

=== Detailed Accuracy By Class ===

       TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
       0.102    0.023    0.526      0.102   0.171      0.721     yes
       0.977    0.898    0.813      0.977   0.888      0.721     no
Avg.   0.802    0.723    0.756      0.802   0.744      0.721

=== Confusion Matrix ===

   a    b   <-- classified as
  102  898 |    a = yes
   92 3908 |    b = no
```

Figure 3.7: Accuracy using 21 Features in [1] on Y!A data set

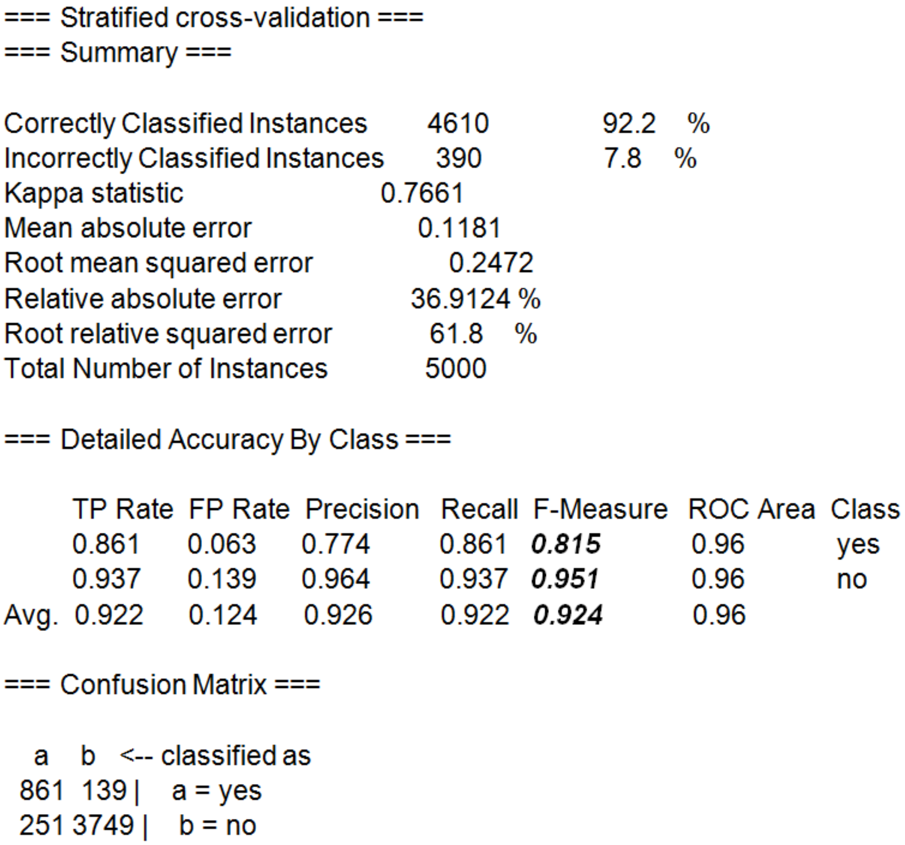
In Figure 3.7, for 21 features (12 features excluding 10 features from the Table 3.2 – five temporal features and QQA_Score, QBA_Score, AQA_Score, ABA_Score and QA_Sim plus the following nine features: reciprocal rank of the answer in the list of answers for the given question, answerer's star, answerer's point, answerer's level, the number of answerer's solved questions, questioner's point, questioner's star, questioner's level and the number of questioner's solved questions) used in [1], the experiment gives 0.744 classification accuracy (F-Measure Score) which is consistent with the results in [1] for 10-fold cross-validation. However, using *only the five temporal* features described in Section 3.3, the classification accuracy increases to 0.923 as shown in Figure 3.8. Our intuition about the importance of temporal features is validated by this comparison between temporal and previously proposed features using the same classification method.



```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       4607              92.14   %
Incorrectly Classified Instances      393               7.86   %
Kappa statistic                    0.7645
Mean absolute error                   0.12
Root mean squared error                 0.2478
Relative absolute error              37.4755 %
Root relative squared error            61.9575 %
Total Number of Instances             5000

=== Detailed Accuracy By Class ===

        TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
        0.861    0.064    0.772      0.861   0.814      0.958      yes
        0.937    0.139    0.964      0.937   0.95       0.958      no
Avg.    0.921    0.124    0.926      0.921   0.923      0.958

=== Confusion Matrix ===

   a    b   <-- classified as
  861  139 |    a = yes
  254 3746 |    b = no
```

Figure 3.8: Accuracy using only five Temporal Features on Y!A data set

In order to further understand the incremental effect, we combine the 21 features with the 5 temporal features and measure the accuracy using all 26 features. The accuracy improved only marginally (as compared with the temporal features) to 0.924 as shown in Figure 3.9. If we interpret this as adding five temporal features to the previously proposed features, accuracy has improved from 0.744 (Figure 3.7) to 0.924 (Figure 3.9), an improvement of 24.1%. On the other hand, if we are to interpret this as adding 21 features to the proposed five temporal features, there is *less improvement* (from 0.923 to 0.924) in accuracy. This seems to clearly establish the robustness and efficacy of temporal features on answer quality accuracy.



```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        4610            92.2    %
Incorrectly Classified Instances       390             7.8    %
Kappa statistic                      0.7661
Mean absolute error                      0.1181
Root mean squared error                      0.2472
Relative absolute error                 36.9124 %
Root relative squared error               61.8    %
Total Number of Instances               5000

=== Detailed Accuracy By Class ===

       TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
       0.861    0.063    0.774      0.861   0.815      0.96      yes
       0.937    0.139    0.964      0.937   0.951      0.96      no
Avg.   0.922    0.124    0.926      0.922   0.924      0.96

=== Confusion Matrix ===

   a    b   <-- classified as
  861  139 |   a = yes
  251 3749 |   b = no
```

Figure 3.9: Accuracy using five temporal + 21 Features on Y!A data set

We have also performed these experiments on the other two data sets and have observed similar accuracy improvements.

3.6 Evaluation by Learning to Rank Approach

We argue that the classification method used for comparison of features in Section 3.5 is not well-suited for the answer quality problem. First, it is difficult to build a good classification model as these data sets are unbalanced. In other words, there is only one best answer for each question, but several none-best answers. For example, in Y!A data set the best answer to non-best answer ratio is about one to nine. It is widely recognized in the research community that building a classification model for unbalanced data set is a real challenge. Therefore, it is difficult to find a good traditional classification model for this problem. To illustrate this point, for the classification method used for evaluating features in Section 3.5, if we categorize *all* the answers as non-best, the accuracy can easily reach 80% (at least four out of five answers are correctly classified); hence, this approach does not seem to be meaningful for answer quality ranking.

Second, the best answer is not an absolute best answer in this scenario. In other words, only the answers available for a given question are used for choosing the best answer. Therefore, the best answer choice is relative to other answers. In addition, an answer may be chosen as the best answer due to clarity of expression although it may not be the best answer technically. However, when we build the classification model as described in Section 3.5, an assumption is made that this question-answer pair is the absolute best answer as compared to all the other pairs. However, this assumption is not generally true for CQA data sets. In contrast, as the learning to rank models build the model for *each* question, this model is better suited.

Finally, for automated learning of answer quality, it is not enough to identify only the best answer; a ranking of all answers is needed for each question. This becomes important for the long-term goal of identifying experts in the system based on the aggregate quality of answers provided by a user (over a period of time). However, the classification method used earlier does not provide answer ranking as it only decides whether or not an answer is the best answer. On the other hand, learning to rank models provide a qualitative value for each answer.

Based on the above observations, we propose the use of learning to rank models to construct answer quality ranking from the question-answer pairs. As the data sets being analyzed can be very large, we choose the *RankSVM* [74] algorithm which has good accuracy in addition to computational efficiency for large data sets. The principle of the RankSVM model is to learn a binary classifier which can discriminate a better answer given a pair of answers for the same question.

We now briefly describe the use of the *RankSVM* algorithm for our problem. Learning to rank approaches require normalized feature values in the range [0, 1]. As some of our features are not in this range (e.g., tAQA_score), we need to normalize those features that are not in this range. For example, we normalize tAQA_score which is in the range [-1, +1] by adding 1 and dividing by 2. For other features which are not in the [0, 1] range, we normalize them as follows:

$$RF\_Score(q_i, v_j) = \frac{F\_Score(q_i, v_j)}{max\{F\_Score(q_i, v_j) | j = 1, ..., n\}} \tag{3.4}$$

where n is the number of values for a feature v_j (e.g., number answers by a user or number of best answers by a user) for question and $RF\_Score(q_i, v_j)$ is the relative feature score and $F\_Score(q_i, v_j)$ is the extracted feature score for that feature with respect to the question q_i. After normalization, each feature value is between 0 and 1.

For example, to calculate the feature tAA_count for a query q_1 in an interval Δt, we count the number of answers given by different users in that interval. The maximum of those answers form the denominator. The same approach is used for others.

We use pair-wise inputs to RankSVM. For each query q_i, we extract all the answers to form question-answer pairs (we have 5 for each question). For each pair, we extract all the features listed in Table 3.2. We input these extracted features for each question-answer pair along with rank as 1 for the best answer and rank as 0 for non-best answers. Based on this RankSVM derives a model of ranking which maximizes Kendall's τ [75] which is defined as

$$\tau(r^c, r^s) = \frac{P - Q}{\frac{1}{2}n(n-1)} \tag{3.5}$$

where r^c is the computed rank and r^s is the input rank for the training data set. n is number of elements in rank r^s (which is 5 as we have 5 answers for each question) and $\frac{1}{2}n(n-1)$ describes the number of rank pairs. P is the number of concordant pairs. A concordant pair is one where ranks of pairs from r^c and r^s agree. Similarly, Q is the number of discordant pairs. A discordant pair is one where ranks of pairs from r^c and r^s disagree. Kendall's τ has a value of +1 if the two ranked lists totally agree; Kendall's τ is -1, if the two ranked lists totally disagree; and if two ranks are independent, Kendall's τ score is 0. This is used for maximizing the objective function[5].

Because of the use of the learning to rank approach, we have a ranking of all answers instead of *best* and *non-best* answers. we can compute the accuracy of any answer with respect to the baseline. This approach will allow us to provide top-k answers to the questioner which is likely to be more useful.

[5] We are using the objective function used by RankSVM without any changes. It is also possible to explore alternative objective functions that are better-suited for this application

3.6.1 Evaluation Measures

Predicting the *best* answer with good accuracy is important. At the same time, it is equally important to predict the answer quality for all answers and compare that with the voting (or service-specific) approach. Conventional precision and recall does not seem to be appropriate for this approach. Thus, we choose precision at top one (P@1) [76] and the mean reciprocal rank (MRR) methods [76]. For each question q_i, we sort on the predicted ranking values. We pick the top answer and assert this answer as the predicted best answer.

We use 10-fold cross-validation to calculate P@1 and MRR for each data set and every experiment has been run five times and the average value is reported. Note that we can also compute P@n as we are using a learning to rank model as opposed to the traditional classification approach.

For the accuracy analysis, we use three baselines as we do not have a well-defined baseline for this problem in the literature.

Random Approach (Baseline 0): We randomly choose an answer as the best answer. This baseline indicates the worst accuracy to predict the best answer and we believe that any approaches should be better than this approach.

QA_Sim (Baseline 1): We use the cosine similarity between a question and its answer to rank the answer. This baseline is widely used in the traditional information retrieval [77] and also used in [30] to search for related answers in the CQA services.

Shaw's 21 Features (Baseline 2): In order to show the effectiveness of temporal features, we also use the feature set used in [1] as one of the baselines. In Y!A data set, we use 21 features as in [1] to identify the answer quality (see Section 3.5.1). Since SO data set does not have four of those features, such as answerer star, we only use 17 features in our experiments. This baseline has been included to show that temporal and learning to rank approach will provide significant improvement over the traditional features. As many of the features listed in Table 3.2 are part of these 21 features, we compare baseline 2 only with proposed five temporal features.

3.6.2 Experimental Analysis

The accuracy of evaluation measures, MRR and P@1, is related to the number of answers for each question. We extract five small data sets from Y!A data set and each data set has 1000 questions with the number of answers from one to five. We use all 22 features listed in table II to calculate the P@1 and MRR for these five data sets respectively (See Figure 3.10). With the increase of the number of answers, the accuracy of this ranking model drops quickly (P@1 from 1 to 0.81 and MRR from 1 to 0.87). The reason can be explained as with the increase of the number of answers for each question, it become more and more difficult to predict the best answer. For example, if every question has only one best answer, the worst accuracy of any ranking model will reach 100%; however, if every question has more than 1 answer, the worst accuracy of these ranking model should be lower than 100%. Therefore, in our experiment if we directly choose 1000 questions without describing the number of answers for each question, it is difficult to evaluate these temporal features. In our experiment, we randomly choose 1000 questions from each data set which has at least 5 answers for each question. For each question, we retain the best answer and 4 other randomly selected answers. Thus, we use 1000 questions, 1000 best answers and 4000 non-best answers for our experiments. Clearly, in these data sets, if we randomly choose an answer as the best answer, P@1 will be 0.2 and MRR will be 0.46. Any meaningful ranking models should be better than this accuracy.

In the literature [1] question and questioner features have been used for predicting answer quality. Thus, we also wanted to consider the question and questioner features for the CQA data sets. Question and questioner features are, respectively,

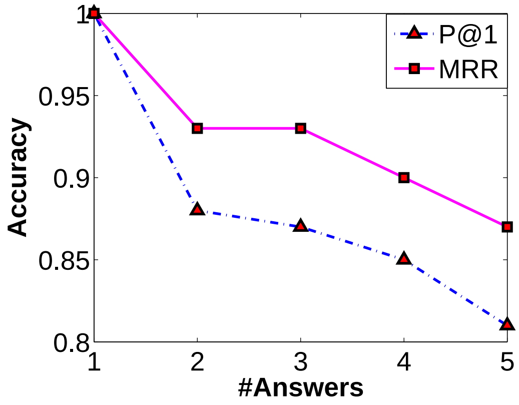


Figure 3.10: Accuracy with Different Number of Answers for Y!A Data set

the features related to the question and the questioner. The description of all the features is shown in Table 3.2. Table 3.4 tabulates the accuracy of features for the same data sets with and without question and questioner features. We can clearly observe that the accuracy does not change at all. This indicates that question and questioner features do not contribute at all to the accuracy so that we do not use these 9 question or questioner related features in our experiments shown below.

Table 3.4: With and Without Question/Questioner Features

Data set	With Q Features		Without Q Features	
	P@1	MRR	P@1	MRR
Y!A	0.810	0.877	0.810	0.877
SO-C	0.535	0.611	0.535	0.611
SO-O	0.536	0.612	0.536	0.612
TT	0.484	0.672	0.484	0.672

In the first experiment, we calculate the difference between the average score for the *best* answer and non-best answers and rank each feature by the deviation. Clearly, if the deviation is large, that feature will discriminate better between the best-answer and the non-best answer. Table 3.5 shows the details for the Y!A data set. Moreover, we also show the feature rank for the other two data sets in Table 3.6. The results of this experiment indicate: (i) Features tABA_Ratio and tABA_Count rank quite high in Table 3.5 indicating their discriminating power for answer quality; in fact, tABA_Ratio ranks as number 1 for all data sets (also see Table 3.6), (ii) AQ_Count and tAQ_Count are ranked at the bottom for all data sets and hence does not seem to be very useful for calculating answer quality. This seems logical as users who ask a lot of questions do not seem to contribute to the quality of answers, (iii) As can be seen from Tables 3.5 and 3.6, the ranking of features is *different for each data* set. This can be attributed to the different characteristics (topic, and others) of the data set, (iv) A_length comes out as an important feature for deciding the answer quality (e.g., top 3 in Y!A, top 6 in SO-C, top 4 in SO-O data set and top 4 in TT data set). This also indicates that a good answer is likely to be longer as the answerer can explain clearly, and finally (v) E_Link, proposed in [78], does not seem to be a good feature for distinguishing the best answer from the non-best answers. This feature seems to be applicable for Wikipedia style service than the ones used in this work. In summary, many of the temporal features are ranked high across data sets. Some non-temporal features (A_Length, ABA_Ratio, for example) also rank high in some data sets.

Next, we use the RankSVM learning to rank model for determining answer quality as explained earlier. The random approach (baseline 0) is shown in the first row of Table 3.7. We discuss the results with baseline 1 (second row of Table 3.7) as it is better. It can retrieve less than 25% in top one rank of the correct answer and MRR shows that the correct answer is less than 49%. This baseline only reflects whether this answer is related to this question, but it is difficult to distinguish the answer's quality; hence, the accuracy of baseline 1 is somewhat similar to baseline 0. We compare our temporal and other features with these two baselines in Table 3.7.

Table 3.5: Feature Analysis of Y!A Data set

Feature	Mean			
	Best A.	Non-Best A.	Diff.	Rank
tABA_Ratio	**0.8849**	**0.0681**	**0.8168**	**1**
tABA_Count	**0.7561**	**0.156**	**0.6001**	**2**
A_Length	0.5037	0.2628	0.2409	3
AQA_Score	0.2723	0.1191	0.1532	4
ABA_Ratio	0.2029	0.0807	0.1222	5
ABA_Count	0.1586	0.0812	0.0774	6
QA_Sim	0.349	0.2772	0.0718	7
AA_Count	0.1509	0.0854	0.0655	8
tAA_Count	**0.1625**	**0.1006**	**0.0619**	**9**
tAQA_Score	**0.2016**	**0.1451**	**0.0565**	**10**
E_Link	0.0547	0.0177	0.037	11
tAQ_Count	**0.0972**	**0.078**	**0.019**	**12**
AQ_Count	0.1435	0.1243	0.0192	13

Table 3.6: Feature Analysis of SO-C, SO-O and TT Data sets

Data set	SO-C		SO-O		TT	
Rank	Feature	Diff.	Feature	Diff.	Feature	Diff.
1	**tABA_Ratio**	**0.3768**	**tABA_Ratio**	**0.3329**	**tABA_Ratio**	**0.3192**
2	**tABA_Count**	**0.2013**	**tABA_Count**	**0.181**	ABA_Count	0.1588
3	ABA_Ratio	0.154	ABA_Ratio	0.1481	ABA_Ratio	0.1486
4	**tAQA_Score**	**0.0802**	A_Length	0.0673	A_Length	0.1485
5	QA_Sim	0.0661	**tAQA_Score**	**0.0524**	AQ_Count	0.1428
6	A_Length	0.0519	QA_Sim	0.0512	**tAQA_Score**	**0.1069**
7	**tAA_Count**	**0.0455**	**tABA_Count**	**0.1041**	**tABA_Count**	**0.1041**
8	AQA_Score	0.0401	AQA_Score	0.0332	**tAA_Count**	**0.0321**
9	ABA_Count	0.0321	ABA_Count	0.0291	AA_Count	0.0934
10	AA_Count	0.0128	E_Link	0.0111	QA_Sim	0.0284
11	E_Link	-0.005	AA_Count	0.002	E_Link	0.013
12	**tAQ_Count**	**-0.012**	AQ_Count	**-0.0317**	**tAQ_Count**	**-0.0358**
13	AQ_Count	-0.043	**tAQ_Count**	**-0.0525**	AQA_Score	-0.0645

As each features is added one at a time in *rank order* (note that the rank order is data set dependent), one can observe consistent improvement in accuracy for all data sets. The process is initialized with the QA_Sim (baseline 1) and the learning to rank model incrementally adds the features and computes the P@1 and MRR values. In Table 3.7, for SO-C, SO-O and Y!A data sets we can observe that if we only consider four features, tABA_Ratio, tABA_Count, ABA_Ratio, A_Length, our learning to rank model will achieve significant accuracy improvement and the other features seem only to improve the accuracy marginally. The results of this experiment shown in Table 3.7 for all 3 data sets indicate the following: (i) The accuracy for the best answer has improved **significantly** (from 0.262 to 0.810, 209%) for the Y!A data set, (from 0.29 to 0.535, 84%) for SO-C data set and (from 0.272 to 0.536, 97%) for SO-O data set, (ii) although the baseline is similar for the SO data set, the improvement in accuracy is still significant when all the features are included. This, we believe, is due to the data set characteristics of answers containing program code in SO-O and SO-C; we are further exploring ways to improve this.

In addition, we also compared the five temporal features with baseline 2 which has already used a number of features. The baseline 2, shown in the first row of Table 3.8, has shown substantial improvement, as expected, over baselines 1 and 0. It indicates that the best answer can be found more than 40% of the times in top one rank and MRR shows that the correct answer is in the first two answers. Then, as each temporal feature is added one at a time, one can observe consistent improvement in accuracy for all data sets (See Table 3.8). In the end, the best answer accuracy has improved significantly (from 0.552 to 0.813, 47%) for the Y!A data set, (from 0.401 to 0.539, 34%) for the SO-C data set and (from 0.395 to 0.537, 36%) for the SO-O data set. This seems to clearly establish the robustness and efficacy of temporal features on answer quality accuracy.

Table 3.7: Accuracy Values to Compare with Baseline 0 and 1

Features	Y!A		SO-C		SO-O		TT	
	P@1	MRR	P@1	MRR	P@1	MRR	P@1	MRR
Baseline 0	**0.2**	**0.456**	**0.2**	**0.456**	**0.2**	**0.456**	**0.2**	**0.456**
Baseline 1	**0.262**	**0.490**	**0.29**	**0.440**	**0.272**	**0.600**	**0.407**	**0.600**
"+"1	0.791	0.864	0.503	0.587	0.493	0.581	0.407	0.602
"+"2	0.798	0.868	0.505	0.591	0.502	0.585	0.402	0.596
"+"3	0.805	0.871	0.503	0.592	0.506	0.591	0.401	0.595
"+"4	0.811	0.877	0.529	0.607	0.526	0.611	0.478	0.664
"+"5	0.809	0.875	0.531	0.609	0.526	0.611	0.487	0.676
"+"6	0.809	0.876	0.526	0.607	0.526	0.611	0.478	0.669
"+"7	0.809	0.876	0.531	0.613	0.527	0.613	0.491	0.674
"+"8	0.805	0.874	0.529	0.610	0.529	0.614	0.488	0.674
"+"9	0.805	0.874	0.534	0.618	0.533	0.615	0.478	0.669
"+"10	0.809	0.876	0.534	0.617	0.535	0.616	0.477	0.667
"+"11	0.807	0.875	0.534	0.613	0.534	0.616	0.478	0.669
All features	**0.810**	**0.877**	**0.535**	**0.611**	**0.536**	**0.612**	**0.482**	**0.670**

[1] We add one feature at a time in their rank order listed in Tables 3.5 and 3.6 to the baseline 1 for each data set. Because *QA_Sim* is our baseline 1, we just need to add the other 12 features.

Our experimental results provide confirmation of: (i) learning to rank approach is flexibility to compute accuracy for the desired combination (e.g., best answer, MRR, or others), (ii) accuracy of features has improved significantly better when used with RankSVM as compared to traditional approaches, (iii) temporal features tABA_Ratio and tABA_Count come out to be two important features that can discriminate the best answer and answer quality even in large and noisy online Q/A data sets, (iv) there is no need to include a large number of features if we choose the set judiciously. From our experiments, tABA_Ratio, tABA_Count, ABA_Ratio, and A_Length come out as critical features, and finally (v) some features, such as tAQ_Count, AQ_Count, and E_Link, do not seem to be useful.

We also do experiments by relaxing 5 answer constraint. We randomly choose 1000 questions from each data set without any constraint as the experiment data

Table 3.8: Accuracy Values to Compare with Baseline 2

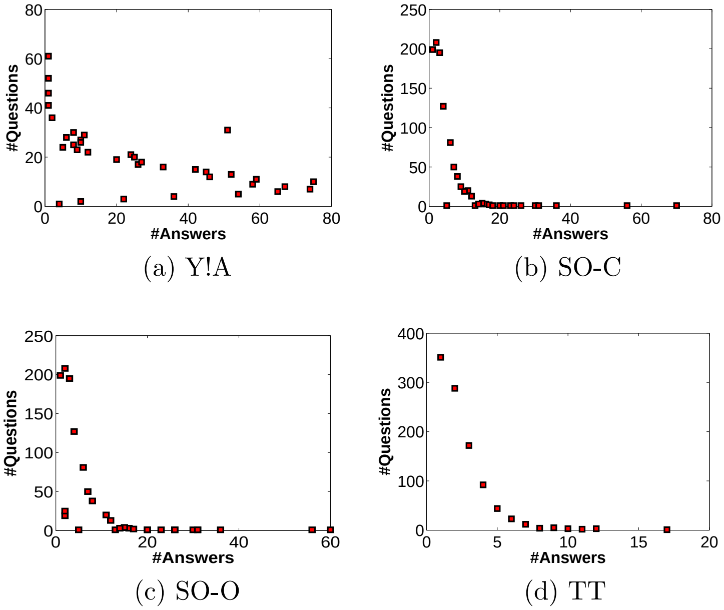
Features	Y!A		SO-C		SO-O		TT	
	P@1	MRR	P@1	MRR	P@1	MRR	P@1	MRR
Baseline 2	**0.552**	**0.640**	**0.401**	**0.510**	**0.395**	**0.511**	**0.461**	**0.659**
“+”1	0.805	0.872	0.532	0.606	0.533	0.604	0.463	0.663
“+”2	0.811	0.874	0.535	0.591	0.532	0.607	0.473	0.667
“+”3	0.811	0.875	0.534	0.592	0.536	0.613	0.479	0.670
“+”4	0.813	0.878	0.537	0.612	0.538	0.612	0.481	0.671
All features	**0.812**	**0.878**	**0.539**	**0.612**	**0.537**	**0.612**	**0.482**	**0.671**

[1] We add 5 temporal feature one at a time in their rank order listed in Tables 3.5 and 3.6 to the baseline for each data set.

sets. The distribution of answers for these three experimental data sets are shown in Figure 3.11a, 3.11b, 3.11c and 3.11d. The results of this experiment shown in Table 3.9 indicate the following: (i) Compared with Table 3.7, the accuracy of these three data sets without constraint are better than the data sets with 5 answer constraint. Especially for the TT data set, P@1 score increases from 0.482 to 0.961 and MRR score increases from 0.670 to 0.983. The reason can be explained as in Q/A community most of questions receive less answers so that the accuracy to predict the best answer is much higher than the data set with 5 answer constraint. For the TT data set, almost 90% questions has only one answer. (ii) Similarly, temporal features tABA_Ratio and tABA_Count also come out to be two important features that can discriminate the best answer and answer quality. (iii) tABA_Ratio, tABA_Count, ABA_Ratio, and A_Length come out as critical features to predict the answer quality.

In the end, we also do experiments on the large scale data sets. We respectively extract 1000, 2000, 3000, 4000, 5000 questions with 5 answer constraint from Y!A, SO-C and SO-O data sets and 1000, 1500, 1897 questions from TT data set[6]. We only test these four features, such as tABA_Ratio, tABA_Count, ABA_Ratio, and

[6] In TT data set, there are only 1897 questions which have more than or equal to 5 answers



(a) Y!A

(b) SO-C

(c) SO-O

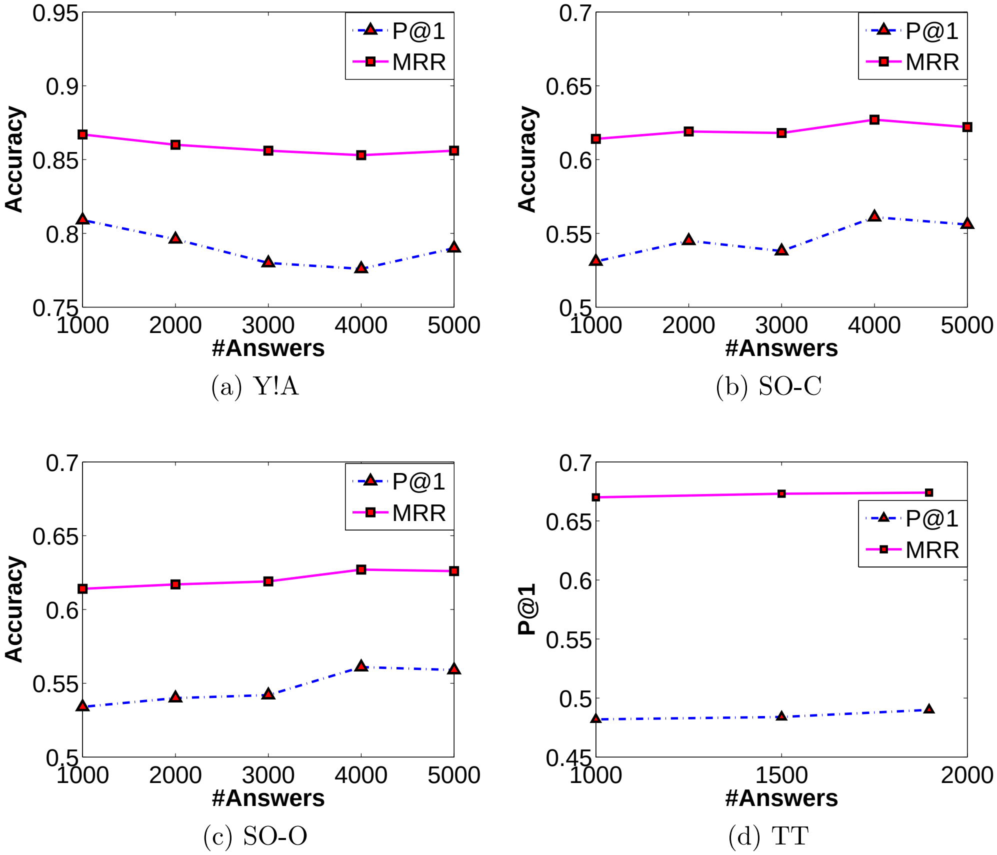
(d) TT

Figure 3.11: #Questions vs. #Answers in Four Data sets

Table 3.9: Accuracy Values for all Data sets without Answer Constraint

Features	Y!A		SO-C		SO-O		TT	
	Top@1	MRR	Top@1	MRR	Top@1	MRR	Top@1	MRR
Baseline 1	**0.273**	**0.501**	**0.428**	**0.562**	**0.447**	**0.581**	**0.957**	**0.980**
(QA_Sim)								
"+"1	0.809	0.939	0.626	0.681	0.646	0.715	0.961	0.982
"+"2	0.816	0.939	0.629	0.707	0.641	0.723	0.961	0.983
"+"3	0.816	0.944	0.632	0.724	0.647	0.742	0.961	0.983
"+"4	0.827	0.948	0.664	0.729	0.681	0.751	0.961	0.983
"+"5	0.827	0.946	0.658	0.729	0.675	0.747	0.961	0.983
"+"6	0.826	0.947	0.657	0.721	0.677	0.749	0.961	0.983
"+"7	0.825	0.947	0.651	0.728	0.682	0.751	0.961	0.983
"+"8	0.82	0.944	0.649	0.731	0.667	0.748	0.961	0.983
"+"9	0.821	0.945	0.653	0.735	0.671	0.752	0.961	0.983
"+"10	0.822	0.944	0.653	0.733	0.672	0.751	0.961	0.983
"+"11	0.823	0.944	0.654	0.733	0.674	0.752	0.961	0.983
All features	**0.825**	**0.949**	**0.662**	**0.734**	**0.681**	**0.761**	**0.961**	**0.983**

[1] Similarly, we add one feature at a time in their rank order which is listed in tables 3.5 and 3.6 to the base line for each data set.



(a) Y!A

(b) SO-C

(c) SO-O

(d) TT

Figure 3.12: Accuracy Analysis for Large Scale Data sets

A_Length since we believes that these four features are critical to identify the answer quality. The experimental results shown in Figure 3.12a, 3.12b, 3.12c and 3.12d indicates the following: (i) Using more data as the training data does not improve the accuracy of our ranking model. In Figure 3.12a the accuracy of Y!A data sets drops sightly with an increase in the number of questions, but in Figure 3.11b and 3.11c the accuracy of SO data sets increases sightly. The reason can be explained as in Y!A data set more data are used as the training data so that this RankSVM model is over-fitting [79] with this data set. (ii) These four features, such as tABA_Ratio, tABA_Count, ABA_Ratio, and A_Length, also effectively identify the answer quality in these large scale data sets.

3.7 Discussion

In this chapter, we stress an important characteristic - "user's behavior" in the social community. Users take part in the social community and communicate with each other in these communities. Therefore, no matter what kinds of problems we want to address in the social community, we should consider these users' behaviors to design our methods. Q/A community is one kinds of social community. In these Q/A communities, users post the questions and the other users answer these questions. Because most of Q/A communities are open communities, every users can post their questions and their answers in these communities. Therefore, these users' current status plays an important role to evaluate their answer quality. Image that John, an super expert, provide a lot of good answer in SO community. However, during these days John are very busy to write his PH.D thesis, he does not have enough time to answer questions in SO community. Therefore, his answer quality becomes low since user needs enough time and energy to write a clear answer. These temporal features are useful to identify these kinds of user's behavior. However, we also point out that temporal features are not appropriate for all the Q/A communities. Some Q/A communities, such as TT community, enroll experts to answer questions12. Because these communities pay money to these experts, every day these experts answer questions regularly. Since their salary are also related to their answers' quality, they should give good answers to these questions. Therefore, these temporal features are useless for these kinds of Q/A communities (See Section 3.5). The idea of temporal features mainly comes from understanding of the user's behavior in these social community.

We also believe that we can propose some other useful features to identify the answer quality by deeply understanding the user's behavior in these Q/A communities.

3.8 Conclusions

In this work, based on the dynamic nature of CQA services, we proposed a set of temporal features for predicting answer quality in CQA services. For these services, user characteristics was better captured with temporal features than the traditional ones proposed in the literature (both textual and non-textual). Further, we demonstrated the effectiveness and superiority of temporal features by comparing our features with the features and the classification approach used in the literature on multiple diverse data sets. To the best of our knowledge, this is the first time temporal features are proposed/used for answer quality prediction (although they was used in other applications).

We also argued for learning to rank approach as a more appropriate model for predicting accuracy of answer quality as it pertains to CQA services. Using the RankSVM learning to rank approach, we performed extensive experimental analysis on diverse data sets to demonstrate that the proposed features work well for predicting the *best* answer as well as *non-best* answer quality.

CHAPTER 4

Expert Finding Problem

Community Question Answering services (CQAs) have become ubiquitous, and are widely used in the last several years. Hence, it would be beneficial if we can mine useful inferences from these data sets so that they can be employed to improve these services. For example, inferring relative quality of answers for a question can help search archives to retrieve good (or top k) answers. As another example, if we can infer or identify expertise of users' from these data sets, we can route questions to the right group of people. With the identification of expertise, number of experts needed to cover a set of topics (in a CQA service) can also be optimized. Our research is geared towards the above problems, and this chapter addresses the problem of inferring expertise.

Current approaches infer expertise using traditional link-based methods such as PageRank or HITS, and others such as number of answers given by a user or the Z_Score. Although an ask-answer graph can be generated for a CQA service based on the ask-answer paradigm (who answers whose questions), this graph is different, in its semantics, from the traditional web graphs. Hence, directly applying link-based methods for CQAs may not be the best approach for identifying expertise (or ranking of users based on their *cumulative* answer quality). Intuitively, the web surfer model need to be enhanced for the ask-answer graph derived from a CQA data set to bring in quality associated with answers in some form. Hence, we posit that both graph structure and domain information related to a user (e.g., answer quality) is critical for inferring the expertise of users. Based on the above observation, we propose the

ExpertRank framework to compute users' expertise. We show that different kinds of domain information can be used to enhance the graph and the information used has a bearing on the accuracy of results. We present our algorithm along with extensive experimental analysis that compares our approach with traditional link-based and other methods. We believe that this framework can be beneficially used or extended to other applications (e.g., analyzing news articles, blogs, etc) as well.

4.1 Introduction

Community Question Answering services (CQAs) strives to provide users with meaningful information using the *ask-answer* paradigm. Briefly, these communities allow users to post questions and other users to answer these questions. When a user posts a question using a CQA service, different approaches are used to find appropriate users to answer this question. Current online communities mainly use the following approaches.

Questioner-based Approach: In this approach, the user who asks the question is responsible for choosing an appropriate expert to answer his/her question. For example, AllExpert (`http://www.allexperts.com/`) provides a number of statistics for each expert in their list including number of questions answered, publications, awards received, and honors in relevant areas. After a questioner browses this information, s/he can direct his/her question to one of the experts.

Answerer-based Approach: This approach allows answerers to select questions that are of interest and answer them. Users post their questions to the community and answerers will browse these questions at their pleasure and choose the ones to answer. This method is mainly used in many different CQAs, such as Yahoo! Answers (`http://answers.yahoo.com`), AnswerBag (`http://www.answerbag.com/`), Stack Overflow (`http://stackoverflow.com/`), etc.

Relationship-based Approach: Users in CQAs build various kinds of relationships among each other, such as friendship (e.g., Quaro community), contract-ship (e.g., Yahoo! Answers community), fans-ship (e.g., Yahoo! Answers community), follow-ship (e.g., Quaro community), etc. If user A builds a contract-ship with user B, user B is A's contractor. When a user posts a question, the service will automatically post this user's question to his/her contractors' web page and their contractors will help answer this question. Similarly, users in Quaro (`https://www.quora.com/`) build follow-ship with other users.

However, these three approaches have a number of drawbacks. In the questioner-based approach, there are a large number of users in current CQAs and hence it is impractical to expect questioners to find an expert by browsing users' profiles. Although the other two approaches encourage various users to answer questions, these methods ignore the answerer's quality. Interestingly, if you browse these web sites, you will find a good number of unfriendly and nonsense answers, such as "You can search the result on Google," "I don't know," etc. If it is possible to assess the expertise of users in an acceptable manner, one can automatically identify a small set of users to answer a question. The questioner will not only be relieved of this burden but is also likely to receive better answers. This will also result in less number of exchanges. This chapter mainly focuses on the problem of automatically identifying experts in CQAs.

Several methods have been proposed to identify experts in the Q/A community. Information retrieval techniques have been used to discover experts from CQAs. Littlepage et al. [30] describes a user's expertise as a term vector extracted from all of his/her perviously answered questions and calculates the cosine similarity (widely used in information retrieval) between a question vector (a term vector extracted from this question) and this user's vector as the expertise score. However, this results in a user list with no clear quality measure associated with it. Zhang et al. [31] and Jurczyk et al. [32] extend traditional link-based algorithms such as PageRank and HITS to compute (global) expertise rank in CQAs. The intuition behind these link-based methods is that if B can answer A's question, and C can answer B's question, C's expertise rank should be boosted because C is able to answer a question of someone who has some expertise. This assumption is perhaps true if we are considering narrow, topic-based services such as C or Java, but may not be true if questions are asked on diverse topics and there is considerable overlap among the users answering questions. Even within C or Java, questions can be on many narrow concepts and the answerer may not be familiar with all of them. A more important and critical observation is that the link-based methods do not use (or need to use) the quality of contents of a web page. We strongly believe that not using contents is not an option for Q/A services as the very notion of an expert depends upon the quality of answers (in addition to other things) given by that user.

Consider the following short example to illustrate the above observations. Table 4.1 shows a few questions and some of their answers from the Stack Overflow service for "C" language. Figure 4.1 shows the ask-answer graph for Table 4.1 using the ask-answer paradigm. A directed edge is drawn from user u_1 to user u_2 if user u_2 answered one or more questions of u_1. Table 4.2 shows the ranking result for this graph using the PageRank algorithm. In Table 4.1, user B and D have the same PageRank score (expertise score), but user B's expertise should be higher than user D because user B's answer is much better than user D's answer (as shown in Table 4.1). This is also reflected by the voted score[1] for B as shown in Table 4.1. Since users' expertise is definitely decided by their answer quality, this chapter proposes the

[1] In CQAs, the voted score of an answer is the *sum* of all votes received for that answer. A user can give a +1 for a good answer and a -1 otherwise.

Table 4.1: a Sample Content from Stack Overflow

User	Votes	Content
Questioner A		In C arrays why is this true? a[5] == 5[a]
Answerer B	330	Because a[5] will evaluate to: *(a + 5) and 5[a] will evaluate to: *(5 + a)
Answerer D	-6	You can search the result on the Google.
Questioner B		What is the best tool for creating an Excel Spreadsheet with C#?
Answerer C	144	You can use a library called Excel Library. It's a free, open source library posted on Google Code.
Questioner E		How can Inheritance be modeled using C?
Answerer A	0	See also: http://stackoverflow.com/questions/351733/can-you-write-object-oriented-code-in-c

incorporation of answer quality information into the graph for computing expertise (or authority) score.

We believe that global user expertise ranking is important because a lot of Q/A services can benefit from this. For example, in Aardvark community (`https://twitter.com/vark`), users can ask questions and Aardvark will pass their questions to its members who may know the answer. Later, the questioner will get answers via IM, Email, or Twitter. Thus, finding a set of people to answer this question becomes extremely important for these communities. In addition, other Q/A communities such as Stack Overflow, Yahoo! Answers, Blurtit (`http://www.blurtit.com/`) and Answerbag (`http://www.answerbag.com/`) can also benefit from this automated ranking approach to improve their answer quality by forwarding questions to better answerers instead of allowing any user to answer questions. Of course, inferring expertise is also of interest from a machine learning viewpoint.

Different domain-specific information can be extracted from CQAs to identify answer quality. The voting information (shown in Table 4.1) is one piece of information that can be used to evaluate answer quality. Another feature that represents

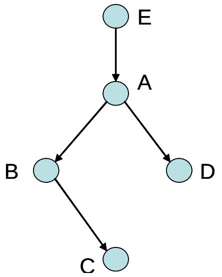


Figure 4.1: Ask-Answer Graph for Table 4.1

Table 4.2: PageRank score for Figure 4.1

Node	PageRank
A	0.21
B	**0.20**
C	0.28
D	**0.20**
E	0.11

answer quality can be "the length of answer" used in [80] because a long answer is likely to explain the question clearly. In addition, we can also use question/answer similarity score to approximate the answer quality of a user. In this chapter, our goal is to identify and analyze various measures of quality of answers that can be gleaned from the data set and evaluate their impact on expertise identification.

Contributions:

- In this chapter, we discuss the differences between the web reference graph and the ask-answer graph derived from CQAs. We argue why traditional link-based ranking algorithms cannot be directly used for the ask-answer graph. This understanding is important as it can be used to extend link-based algorithms appropriately for various applications.
- We propose a framework using which data set specific information can be incorporated into the ask-answer graph. We analyze domain information from several data sets and indicate how they can be added to the graph.

- We present the *ExpertRank* approach which is based on the Katz index algorithm [81] to measure users' expertise and rank them. A tunable parameter α, called an attenuation factor in Katz index, is used to control the transitivity aspect of the ask-answer graph.
- Extensive experimental analysis is performed on multiple, diverse data sets to show how the proposed algorithm and the domain information provide more accurate results than traditional link-based and other approaches. Choice of parameters values such as α using the characteristics of the data set are presented.

Road Map: Section 4.2 defines the problem and motivates our approach. Section 4.3 describes our contributions in detail along with the ExpertRank algorithm and alternative approaches to using domain information. Analysis of parameter values and their choice is also presented. Section 4.4 shows extensive experimental results on three diverse data sets and their analysis. Section 4.5 has conclusions and future work.

4.2 Problem Statement

This chapter focuses on the general problem of mining expertise of users' from a CQA data set and rank them. As there are many paradigms used for interaction in ask-answer graph, different types of domain information are available. This chapter evaluates the effectiveness and utility of these domain information for the purpose of ranking.

Given a CQA data set consisting of users $u_1, u_2, u_3, ..., u_n$ along with questions, answers, and available domain information (e.g., vote information), our goal is to infer users' expertise and rank them. Ranking of users will facilitate: (i) to provide a list of ranked experts for a topic category, (ii) to direct questions to small set of experts, and (iii) to optimize on the number of experts as needed. We would like to point out that expertise rank order is quite subjective and can vary with respect to the evaluator. Furthermore, no real expertise rank exists in the CQA data set itself. Thus, it is really difficult to find a standard (or a baseline) to compare the evaluated rank order. As a result, one has to resort to manual evaluation as is commonly done by researchers on this topic [31, 46]. In this chapter, we use a manually evaluated rank order as the standard for comparing our automated prediction of ranking results. For each data set, two experts analyze the questions and answers of a small number of users (50 in our case) and manually rank the expertise of these users.

4.2.1 Motivation for Our Approach

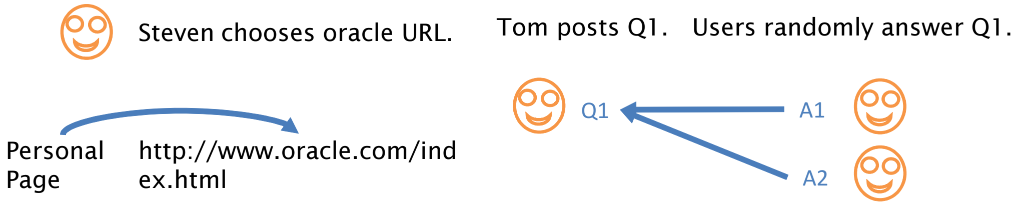
Traditional web page graphs capture citation or reference relationships. Consider a web designer Steve who is building his own personal web page. Because he works at Oracle, he wants to use a reference to Oracle in his personal web page. Table 4.3 lists several alternatives to introduce his company. In the end, after considering these candidate web pages, Steve chooses the Oracle main page for his personal web page because Steve believes that Oracle's main page is a better web page to introduce his company than others. In other words, in the web design process this web writer is likely to consider many candidates which can be used to describe his/her anchor text, and then chooses the best one (in his/her opinion) to include as a URL in his/her page. Figure 4.2a describes this web design process. In a web reference graph, the number of times a web page is referenced can be used to evaluate the web page's quality because if so many web writers believe this web page to be a good web page, this web page should rank high in quality. Citation algorithm (in-degree method) [82], PageRank, and HITS are all based on this motivation. In other words, the link-based approaches rely on web designers' choice of reference pages and use that as a statistic.

Table 4.3: Possible URLs to Introduce Oracle

Description	URL
Oracle Main Page	`http://www.oracle.com/index.html`
Oracle Wikipedia	`http://en.wikipedia.org/wiki/Oracle`
Oracle On Twitter	`http://twitter.com/#!/oracle`
Oracle Audio	`http://www.oracle-audio.com/`
...	...

4.2.1.1 Characteristics of Ask-Answer Graphs

In contrast, an ask-answer graph does not have the same intuition as the web reference graph. First, in CQAs, any answerer, no matter good or not, can give an answer to a question. Thus, in an ask-answer graph questioners, typically, cannot choose the best answerer. Figure 4.2b shows a small ask-answer graph. Recall that the direction of the edge is from the questioner to the answerer. Since, in an ask-answer graph neither asker nor answerer ensure the quality of the links, we cannot directly apply these linked-based rank algorithms to this ask-answer graph. Thus, we propose approaches that include quality aspect into an ask-answer graph using domain information in various ways. We propose four different approaches to include quality information for links in an ask-answer graph using domain specific information in CQAs. Second, the ask-answer relationship is also different from the web page reference relationship. Using a URL in one page, we can directly extract web page reference relationship. However, we cannot directly extract the ask-answer relationship from the Q/A data set because between two users there may be a number of ask-answer relationships (each with varying quality). Therefore, combining/aggregating these relationships together to describe the ask-answer relationship becomes an important problem. Previous work in this area (discussed in Section 2) do not seem to make this distinction. Based on our observation, we believe that for an ask-answer


Steven chooses oracle URL.
Personal Page
http://www.oracle.com/index.html
Tom posts Q1.
Users randomly answer Q1.
Q1
A1
A2

(a) Web Design Process

(b) Ask Answer Process

Figure 4.2: Process to Construct a Graph

graph we cannot directly use traditional link-based ranking algorithm to identify users' expertise.

4.3 ExpertRank Framework

Based on our earlier observations, the ask-answer graph derived from a CQA data set is enhanced. First, a weight is associated with each edge (or link) using some domain information that reflects quality of answers corresponding to that edge. As an edge may reflect more than one answer given by a user, it is not sufficient to use the number of answers given by a user as it can be misleading. Qualitative value of all the answers need to be aggregated to reflect the edge semantics. Otherwise, some spam users who give *bad or irrelevant* answers will reach a high authority score. In an ask-answer graph, as the questioner will not control link quality (as may be true for traditional web pages), we leverage available domain information in the CQAs to infer link quality. Since an edge from user u_i to user u_j in an ask-answer graph indicates that user u_j answered one or more questions of user u_i, the weight of the edge between users u_i and u_j takes into account: the quality of user u_j's each answer for u_i's question, and the fraction of u_i's questions answered by u_j.

Second, the transitivity relationship used in most of the link-based approaches also need to be adjusted for each data set. As we have indicated earlier, this may depend on the characteristics of the data set. For example, for a homogeneous data set (i.e., a single focused topic such as C language) transitivity may be valid. However, this may not be entirely true for other data sets where *overlapping* users answer questions on different *unrelated* topics. Eventually, we want to derive this from the data set itself. In this chapter, we have a variable α (called the attenuation factor in the Katz index) that can be adjusted to reflect this transitive relationship. We believe that this is important for these applications.

In the following sections we discuss the use of different kinds of domain information, their relevance, and the computation of edge weights for an ask-answer graph. In order to demonstrate the effect of domain information on expertise ranking accuracy, we have chosen the traditional similarity used in information retrieval to illustrate how such a simplistic measure will not provide good accuracy. We compare this to other domain information that captures answer quality better as well as other approaches.

4.3.1 Voted Score Approach (VS)

A voted score for an answer is the cumulative score of that answer given by users in the community. Voted score is used by many CQA services (e.g., Yahoo! Answers) and can be given by a user for an answer. A score, given by a user for an answer, is either a +1, or -1 reflecting positive, or negative answer quality. Although this approach is widely used in Q/A community, different services use different approaches to provide a voted score. For example, in the Yahoo! Answers community no additional answers are allowed at the voting stage. The answers are listed in random order and other users (than the questioner and answerers) can vote for an answer. After some fixed period, the question is closed and the answer's quality will be identified by the number of votes. In other Q/A communities, such as Stack Overflow, Blurtit (http://www.blurtit.com/), Turbo Tax Live (https://ttlc.intuit.com/), and Answerbag (http://www.answerbag.com/), there is no clearly-defined time period. A user can answer a question and vote on the answers at the same time.

A voted score is meant to reflect the quality of an answer. The voted score seems to be a better indication of the quality of an answer (especially if a good number of high quality users [2] take part in the voting process) as indicated by a few questions and answers shown in Table 4.1. As can be seen, since answerer B's answer is much better than answerer D's answer, answerer B'answer receives higher voted score than answerer D's answer. We directly use the voted score to identify answer quality of a question answered by user u_1 for a question from user u_2. if less users take part in the voting process, the voted score may not reflect the answer quality well. We take this into account in our Hybrid approach.

$$A\_Score(u_i, u_j, q_k) = VS(u_i, u_j, q_k) \tag{4.1}$$

where $A\_Score(u_i, u_j, q_k)$ is the answer score for u_j's answer to u_i's question q_k and $VS(u_i, u_j, q_k)$ is the voted score of u_j's answer for u_i's question q_k. This approach utilizes service-specific information to identify answer quality. We still need to combine quality scores from different answers given by the same user. Since this step is common to all the approaches it is discussed in Section 4.3.5 after describing the approaches.

[2] In Yahoo! Answers only the user whose level is higher than 2 can vote for an answer and in Stack Overflow only the user whose reputation is more than 15 can vote for an answer.

4.3.2 Ranked Answer Approach (RA)

Voted score is directly available in *some* services and can be used for answer quality. However, we also need an approach for services that may not have a voted score or where the voted score may not be reliable. Many services mentioned earlier do not have a specific voting phase. And if an answer in these communities is posted earlier, this answer has a greater chance of receiving a higher voted score. Also, many a times answers do not receive enough voted score thereby making the voted score less reliable and not representative of the answer quality. In order to overcome the limitations of voted score as a quality measure and have a safer alternative for arriving at the answer quality, we propose the approach of ranking answers for quality as described below.

Shah et al. [1] extract 21 features from Yahoo! Answers community and use the logistic regression model for predicting answer quality in CQA data sets. Since their classification approach only identifies the best answer for a question, this classification approach cannot be directly used by us as we need ranking for each answer for a given question. We need rank of answer quality for each question (rather than binary) to better formulate the weight of an edge in the ask-answer graph. Therefore, we use a learning to rank model to rank answers. Briefly, a learning to rank model uses three steps to evaluate the answer quality. First, features are extracted and used to identify the answer quality. Because Shah's 21 features are specific to Yahoo! Answers data set, we only use 16 general features (that are common to most data sets) in CQAs to predict answer quality. These 16 features include 2 answer features, 5 answerer features, 5 questioner features and 4 question features. Second, we use 1000 questions as the training data set and derive a learning to rank model from these 1000 questions and their answers. As each question dose not have an answer ranking order, we use the following objective rank order: Top 1 is the best answer[3] and the other answers are randomly ranked in the rank list. In Stack Overflow community administrators identify the best answer and mark them; in the Turbo Tax community administrators identify a high quality answer by marking them as a helpful answer (although in the Turbo Tax community some answers are marked as the best answer, as very few are marked as such, we use the helpful answer as the high quality answer). Third, to calculate the quality for a new answer, we extract the same 16 features from these new answers and use this learning model to rank them. In our experiments, we use RankSVM [74] model to build our ranking model because RankSVM is shown to be effective (and also efficient) for large data sets. We use the score obtained for each answer as its quality score.

$$A\_Score(u_i, u_j, q_k) = RankSVM(u_i, u_j, q_k) \tag{4.2}$$

where $RankSVM(u_i, u_j, q_k)$ describes the ranking score of u_j's answer for u_i's question q_k. The accuracy of ranked answer approach is mainly decided by the training data set and the features used. We will discuss this in section 4.4.

4.3.3 Information Retrieval Approach (Sim)

Traditionally, cosine similarity value is used in document retrieval and search. Although we believe that this is not a very good measure for answer quality, we use this to demonstrate the effect of content information for CQA data sets. This measure certainly will give low score to "bad" answers where the similarity of an answer with a question is likely to be low or none. For more reasonable answers, the similarity is likely to be better. If the similarity between an answer and a question is low, we

[3] In Turbo Tax and Stack Overflow community, the answer which receives the highest voting score is the best answer for this question.

interpret this answer having low quality since this answer is unrelated to the question. For example, since in Table 4.1 user D gives a "bad" answer for user A's question, the similarity score between D's answer and A's question is 0.

After removing stop words from a question and its answers in CQAs and stemming them, we build a question vector (a term vector extracted from each question) and an answer vector (a term vector extracted from each answer for that question) and then use VSM [83] model to calculate cosine similarity between question and answer. We use the following equation to compute the answer quality score.

$$A\_Score(u_i, u_j, q_k) = \frac{V(u_i, q_k) \cdot V(u_j, q_k)}{||V(u_i, q_k)|| \times ||V(u_j, q_k)||} \tag{4.3}$$

where $V(u_i, q_k)$ describes the term vector of user u_i's question q_k and $V(u_j, q_k)$ describes the term vector of user u_j's answer for question q_k. $||V(u_i, q_k)||$ and $||V(u_j, q_k)||$ are the normal scores of these two vectors respectively.

4.3.4 Hybrid Approach (Hyb)

In order to improve accuracy, it is possible to combine multiple answer quality scores into a composite score. Since we believe that the similarity score is not as good as the others for measuring quality, we propose to combine the voted score with the ranked score. We use a parameter γ to adjust the contribution of these two scores. We use the following formula to calculate the hybrid answer quality score.

$$A\_Score(u_i, u_j, q_k) = \gamma VS(u_i, u_j, q_k) + (1 - \gamma) RankSVM(u_i, u_j, q_k) \tag{4.4}$$

In Section 4.4, we discuss the choice of γ for real data sets.

4.3.5 Computing Edge Weights

The voted score is the aggregate of the number of votes assigned to an answer. Since this score varies significantly from answer to answer, it needs to be normalized. We normalize the voted score using the minimum and maximum values of the scores for *each* answer. $minA\_Score(u_i, q_k)$ $(maxA\_Score(u_i, q_k))$ as the minimum (maximum) score of all answers to u_i's question q_k are computed as follows:

$$\begin{aligned} minA\_Score(u_i, q_k) &= min\{A\_Score(u_i, u_j, q_k)|j = 1, ..., n\} && (4.5) \\ maxA\_Score(u_i, q_k) &= max\{A\_Score(u_i, u_j, q_k)|j = 1, ..., n\} && (4.6) \end{aligned}$$

and the normalized A_Score is computed as

$$NA\_Score(u_i, u_j, q_k) = \frac{\frac{A\_Score(u_i,u_j,q_k)-minA\_Score(u_i,q_k)}{maxA\_Score(u_i,q_k)-minA\_Score(u_i,q_k)} + \epsilon}{1 + \epsilon} \qquad (4.7)$$

which is in the range $[\frac{\epsilon}{1+\epsilon}, 1]$. ϵ is used to adjust the lower bound of normalized values. We do not want to set the normalized value for the answer receiving the lowest voted score to 0 because these answers have been assessed for quality as opposed to answers that have not been voted upon (which receive a normalized score of 0). Thus, if an answer receives the lowest voted score, its normalized quality score is equal to $\frac{\epsilon}{1+\epsilon}$. In our experiments, ϵ is set to 0.1. ϵ value greater than 0.1 or 0.15 does not make sense as it is a compensatory value. We use the average score of all answers as the answer quality score between two users. That is,

$$A\_Quality(u_i, u_j) = \frac{\sum_{j=1}^{|QA(u_i,u_j)|} NA\_Score(u_i, u_j, q_k)}{|QA(u_i, u_j)|} \qquad (4.8)$$

where $A\_Quality(u_i, u_j)$ describes the answer quality and $|QA(u_i, u_j)|$ is the number of u_i's questions answered by u_j. Usually, if $A\_Quality(u_i, u_j)$ is high, user u_j has been able to answer user u_i's questions well.

In addition to answer quality, the fraction of u_i's questions answered by u_j captures whether or not user u_j is familiar with user u_i's questions. Hence, u_j's quality of answers need to be tempered by the fraction of u_i's questions answered by u_j and is calculated as

$$Q\_Factor(u_i, u_j) = \frac{|QA(u_i, u_j)|}{|Ques(u_i)|} \tag{4.9}$$

where $|QA(u_i, u_j)|$ is the number of u_i's questions answered by u_j and $|Ques(u_j)|$ is the total number of user u_i's questions. $Q\_Factor(u_i, u_j)$ is in the range of $[0, 1]$. The quality of u_j's answers to u_i's questions combines these two factors. Thus, the weight of the edge between u_i and u_j is computed as:

$$QA\_Quality(u_i, u_j) = A\_Quality(u_i, u_j) \times Q\_Factor(u_i, u_j) \tag{4.10}$$

where $QA\_Quality(u_i, u_j)$ captures the quality of u_j's to answer u_i's questions. As the edge in this graph captures the quality of answers to question, we term this graph a weighted ask-answer graph. Users and relationships in CQAs are modeled as a directed graph $G = (V, E)$, where a node in V represents a user in this Q/A community and a directed edge $< u_i, u_j >$ from u_i to u_j indicates that u_j answered one or more of u_i's questions. The weight of the edge $< u_i, u_j >$ captures the quality of u_j's answers u_i's questions. For a user u_i in this graph, we denote $Q(u_i)$ and $A(u_i)$, respectively, as the set of questioners (in-neighbors) and answerers (out-neighbors). The k^{th} questioner for user u_i are denoted as $Q_k(u_i)$, for $1 \leq k \leq |Q(u_i)|$, and individual k^{th} answerers of user u_i are denoted as $A_k(u_i)$, for $1 \leq k \leq |A(u_i)|$. Table 4.4 shows a small sample of questions and answerers from the Stack Overflow community and Figure 4.3 shows the weighted ask-answer graph for this sample. We

Table 4.4: A Sample Question Answer Community

Post 1			Post 2			Post 3			Post 4		
Q_1	U_4	A_S	Q_2	U_5	A_S	Q_3	U_5	A_S	Q_4	U_3	A_S
A_1	U_1	2.7	A_1	U_2	2.1	A_1	U_1	1.1	A_1	U_1	1.0
A_2	U_2	0.6	A_2	U_3	1.1	A_2	U_3	0.8	A_2	U_2	0.7
A_3	U_3	-0.2	A_3	U_4	0.3	A_3	U_2	0.2	A_3	U_4	-0.5

[1] In this example, A_Score is calculated by the RA approach.

use matrix U to store this graph. $U(u_i, u_j)$ describes the quality of answers by u_j to u_i's questions.

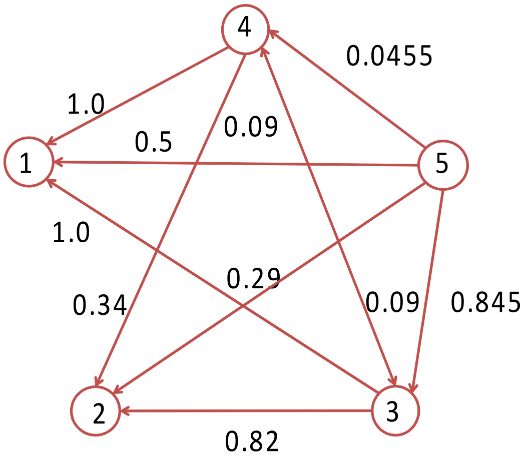


Figure 4.3: Weighted Ask-Answer Graph for Table 4.4

4.3.6 ExpertRank Algorithm

Section 4.3.5 constructs the weighted ask-answer graph and associates user's quality score $QA\_Quality(u_i, u_j)$ with each edge in the graph. We discussed earlier the transitivity relationship in ask-answer graphs derived from Q/A data sets. (in Section 2) We formalize this as the transfer probability $P(u_i \rightarrow u_k | u_i \rightarrow u_j, u_j \rightarrow u_k)$, which describes that the probability for u_i to answer u_k's question. Usually, if u_j

only asks and answers questions in the same area, the transfer probability $P(u_i \to u_k | u_i \to u_j, u_j \to u_k)$ should be high. However, if user u_j asks and answers questions in different areas, this transfer probability $P(u_i \to u_k | u_i \to u_j, u_j \to u_k)$ should be low. Although theoretically this transfer probability can vary between any three connected users, in order to simplify the approach and computation, we use a single transfer probability α for a data set. Thus, the expertise rank of a user u_i ($ER(u_i)$) is the sum of user u_i's ability score with respect to u_i's questioners. Thus, we have our iterative equations to compute $ER(u_i)$ as:

$ER_k(u_i)$ gives the expertise score for user u_i on the k^{th} iteration and we successively compute $ER_{k+1}(*)$ based on $ER_k(*)$. We start with $ER_0(*)$ where each $ER_0(*)$ is equal to 0, which is the lower bound on the actual expertise score $ER(u_i)$:

$$ER_0(u_i) = 0 \tag{4.11}$$

To compute $ER_{k+1}(u_i)$ from $ER_k(*)$, we use the following equation:

$$ER_{k+1}(u_i) = \sum_{j=1}^{|Q(u_i)|} QA\_Quality(Q_j(u_i), u_i) + \alpha \sum_{j=1}^{|Q(u_i)|} ER_k(Q_j(u_i)) \tag{4.12}$$

For iteration $k+1$, we update user u_i's expertise scores of his/her neighbors from the previous iteration k.

Algorithm 1 outlines *ExpertRank* computation. It takes one argument U. In line 1, *ExpertRank* algorithm initializes variables and sets all the user's expertise rank score as 0. Lines 2-8 implements iterative equation to calculate each user's expertise score. Line 6 is used to *normalize* the user's expertise score in each iteration. Lines 2 and 7 are used to stop this iterative algorithm. Although the convergence of iterative expertise rank algorithm can be guaranteed in theory, practically a tolerance factor ϱ is used to control the number of iterations performed. It is recommended to set

Algorithm 1 *ExpertRank*

```
Require:
    User Weighted Matrix $U$;
Ensure:
    Expertise Rank Vector, $ER$;
1: $ER_0 \leftarrow 0$;
2: Do $k = 0$ to Max-Iteration $K$
3: For each element $ER(u_i)$
4: $ER_{k+1}(u_i) = \sum_{j=1}^{|Q(u_i)|} U(Q_j(u_i), u_i) + \alpha \sum_{j=1}^{|Q(u_i)|} ER_k(Q_j(u_i))$;
5: End For
6: Normalize(ER);
7: If ( $max(ER_{k+1}(u_i) - ER_k(u_i)) < \varrho$ ) go to line 9;
8: End Do
9: return $ER$;
```

$\varrho = 0.001$, same as the one used in PageRank [2]. The terminating condition of the iterative algorithm is:

$$max(ER_{k+1}(u_i) - ER_k(u_i)) < \varrho \tag{4.13}$$

The algorithm stops if the maximal change of rate of expertise rank score between two consecutive iterations for all the users is smaller than the threshold ϱ. In all of our experiments we have seen rapid convergence with the relative expertise ranking score stabilizing in 40 iterations. Hence, we have fixed the number of iterations (k) to be 40.

we also analyze the time and space complexity of this algorithm. Because the weighted ask-answer graph is a very sparse matrix, we need to store only the edges for the weighted graph; Therefore, the space required is $O(e)$, where e is the number of edges in this graph. Let d be the average of $|Q(u_i)|$ over all the users u_i. The time complexity of this algorithm is $O(kdn)$, since in each iteration, expertise rank score of u_i is updated with values from this user's questioners. n is the number of nodes in the graph. As d is the average of $|Q(u_i)|$ over all the users u_i, it can be treated as a constant as it is not likely to increase with n.

4.3.7 Discussion of α

Recall that α describes the transfer probability in the Q/A community. If α is small, ExpertRank will only consider local information (a small scope of graph); if α is large, ExpertRank will consider global information (a large scope of graph). For example, Figure 4.4 is a sample ask-answer weighted graph and Table 4.5 shows ExpertRank's results. It is clear from the graph that users d and f answer questions, and e further answers questions of d and f. If the entire graph is considered, e should come out with the highest expertise score followed by d and e. If only local information is used, d and e should come out as equal experts. This translates to the values of α as follows. If α is 0.5, node e receives the highest expertise score. However, if α is 0.05, node d and f receive the highest expertise score. Table 4.5 also shows ranking results of PageRank and In-Degree. PageRank considers global graph information to rank each node; In-Degree just considers its neighbor nodes to rank the node. In this example, when α is 0.5 (large), ExpertRank has the same rank order as PageRank; when α is 0.05 (small), ExpertRank has the same rank order as In-Degree. This example clearly shows that ExpertRank's results shift from local to global with the increase of α.

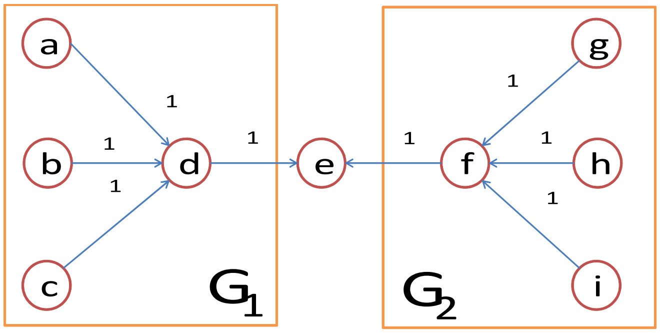


Figure 4.4: A Sample Ask-Answer Graph

Table 4.5: Results of Rank Algorithms for Figure 4.4

Nodes	ER(0.5)	ER(0.1)	ER(0.05)	PageRank	In-Degree
a	0.18	0.30	0.32	0.05	0
b	0.18	0.30	0.32	0.05	0
c	0.18	0.30	0.32	0.05	0
d	**0.45**	**0.39**	**0.37**	**0.18**	**3**
e	**0.63**	**0.38**	**0.35**	**0.34**	**2**
f	**0.45**	**0.39**	**0.37**	**0.18**	**3**
f	0.18	0.30	0.32	0.05	0
g	0.18	0.30	0.32	0.05	0
h	0.18	0.30	0.32	0.05	0

[1] ER(0.5) means that α in ExpertRank is 0.5.

[2] ϱ in all the algorithms is 0.001.

4.4 Experimental Analysis

We use three different data sets to test our approaches.

Stack Overflow (SO) Data set: This service focuses on computer programming topics. Unlike other traditional Q/A services, SO allows a user to modify other user's answers. In other words, when an answerer wants to answer a question, s/he has two choices: modify an existing answer or provide a new answer. As a result, the average number of answers for each question is only 2.36. We only consider the first user who posts the answer as the answerer, because, in most cases, the first user is likely to provide a larger contribution than other users. Each question in this community is marked

Table 4.6: Complete Data set Characteristics

Data set	#Questions	#Answers	Avg (V)	#Qs (1A)
SO-C	25,942	91,615	2.6	5,576
SO-O	8,644	21,879	1.6	2,811
TT	232,411	257,113	1.3	215,163

[1] Avg (V) is the average number of votes for each answer.

[2] #Qs (1A) is the number of questions having 1 answer.

with a topic tag (e.g., "C" or "Oracle"). We use questions marked as "C" as SO-C data set and questions marked as "Oracle" as SO-O data set. Broader statistical characteristics of these two data sets are shown in Table 4.6.

Turbo Tax (TT) Data set: TT service only discusses tax-related issues. This community enrolles many experts to answer questions, so most of the users are mainly questioners and are less likely to answer questions. Thus, the average number of answers for each question is only about 1.11. We only choose questions between January and April 2009 for our experiments as this community is very active in that period. Table 4.6 also shows TT data set characteristics.

4.4.1 Evaluation Method

For these studies, baseline or a standard with which to compare results is extremely important. For some scenarios it is easy to find or derive a baseline. Similarity is one such standard that is widely used in information retrieval. We believe that similarity is not a very good measure for our problem. Our problem for finding a standard for comparison is exacerbated by the fact that the notion of expertise itself can be quite subjective. Hence our choice of data sets to areas where that issue is minimized. There is no user expertise rank information in the data sets nor can it be derived from the data sets. Hence, as has been done by other researchers (e.g, [31]), we have used human experts to manually evaluate the expertise of users in each data

Table 4.7: Characteristics of 50 Random Users

Data set	#Questions	#Answers
SO-C	134	1,492
SO-O	246	2,408
TT	17	23,453

Table 4.8: Five Levels of expertise rating

Level	Meaning	Description
5	Top Expert	Knows core theory and advanced topics
4	Professional Expert	Can answer most questions and knows one or more sub topics well.
3	General Expert	Knows some advanced concepts in these topics.
2	Learner	Knows some basic concepts
1	New Recruit	Just starting to learn these topics

set. Due to the large number of users (as can be see from Table 4.6) it is impossible to manually rate all users in the data set. Hence, we randomly choose 50 users in each data for manual evaluation. We only choose users who have answered at least 10 questions to ensure enough content for manual evaluation. Five levels of expertise (again commonly used in these studies) as shown in Table 4.7 were used.

We have used two independent experts who are very familiar with the "C language" and the "Oracle database" from the computer science department to evaluate the two SO data sets. We have used two experts from the business department to evaluate the "Turbo Tax" data set. *None of these experts take part or are associated with this research.* After each expert evaluated the data sets independently, for sanity check, we use Kendall's τ [84] score to compare these two users' rank lists. The Kendall's Tau distance between two raters is 0.741 for SO-C, 0.761 for SO-Oracle, and 0.711 for TT. In order to maintain consistency of evaluation, we remove users from our evaluation whose score differs by more than 1 level in Table 4.8. Based on this criteria, we have removed 3 users from the SO-C, 2 users from SO-O, and 5 users from TT data set. After this, the Kendall's Tau has improved to 0.793 for SO-C, 0.783 for SO-O), and 0.788 for TT. As we add the rating of two raters, there are a total of 10 categories.

The metric used for comparison is also important. We believe that identifying a user's expertise rank with good accuracy is important. In the information retrieval area, researchers use a number of measures to evaluate the rank list's accuracy; one of them is the DCG (Discounted Cumulative Gain) score [84]. Intuitively, the DCG score evaluation method penalizes experts with a higher rank if they appear lower in the list. Hence, this evaluation metric matches well with our application requirement[4]. Since DCG score is not between 0 and 1, we use the Normalized DCG (or NDCG) [84] to evaluate the ranked list. If NDCG@n is large, this algorithm's rank order will match well with the manual standard; If NDCG@n is small, this algorithm's rank order does not match well with the standard.

4.4.1.1 Methods used for comparison

In the literature, four methods have been used for predicting the expertise of users. In this chapter, we have proposed four ExpertRank-based approaches – ER (VS), ER (RA), ER (Sim), and ER (Hyb) – for doing the same. Our analysis will compare these eight methods for accuracy.

- HITS: Jurczyk et al. [32] use the HITS authority score as the expertise score to identify users' expertise.
- PageRank: Zhang et al. [31] use the PageRank score as the expertise score. In our experiments, the parameter d (the damping factor) of PageRank is 0.85 (default).

[4] Kendall's Tau is a measure for the entire list where as NDCG can be calculated for various positions.

- #Answers: Zhang et al. [31] use the number of questions answered (or number of answers) as users' expertise score.
- Z_Score: Considering both the number of questions and answers, Zhang et al. [31] use Z_Score to identify users' expertise.
- Four approaches discussed in this chapter: ER (VS), ER (RA), ER (Sim), and ER (Hyb).

4.4.1.2 Intuitive Analysis

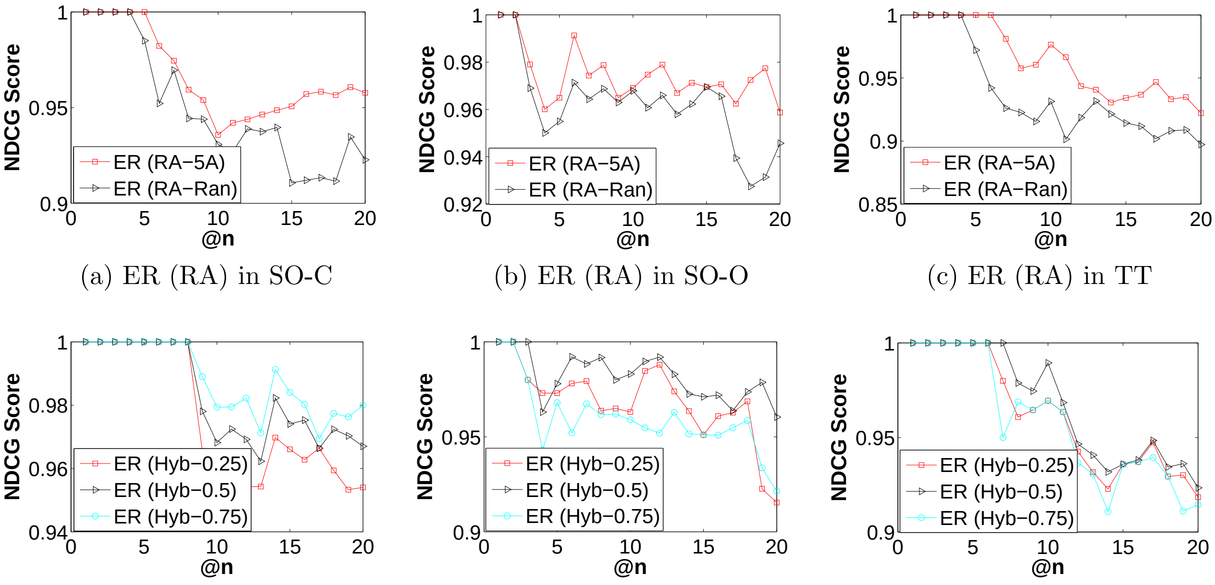
Our premise is that there is a need for quality information in addition to structure to predict expertise. We have used four different pieces of domain information for answer quality. Another component is the transfer probability. We expect both ER (VS) and ER (RA) approaches to do better than methods that do the not use the above information (PageRank and HITS). Z_Score and #Answers have been shown to be better than the link-based methods in [31]. Our experiments indicate the same. Intuitively, we do not expect the similarity approach to do better than any of our approaches as merely the similarity information is not sufficient to infer answer quality. Finally, we expect the ER (Hyb) to do much better than all others.

4.4.2 Experimental Results

Parameter α affects the accuracy of ExpertRank score directly. This parameter is application dependent. In order to study the effect of this parameter, for the SO-C data set, we compared accuracy with the standard by changing the α value and found as expected that it is high for the α value of 0.1. We have used this value for all data sets. This also indicates that although we choose the questions in a specific domain, the transitivity of the ask-answer graph is low. In other words, questions for each user cover a large scope. Eventually, this parameter needs to be tied to the characteristics of the data set.

4.4.2.1 Analysis of ER (RA)

The accuracy of ER (RA) is mainly decided by the training data set and the list of features used. In order to study the effect of the training data set, we chose the training data set (i.e., 1000 questions) in two ways. The first choice included 1000 questions each of which has more than 5 answers. In the second alternative, we randomly chose 1000 questions without imposing any constraints. We show the results of these alternatives as ER (RA-5A) and ER (RA-Ran) respectively. Figures 4.5a, 4.5b, and 4.5c show accuracy results of ER (RA-5A) and ER (RA-Ran). Intuitively, more answers provide better quality values and hence should perform better than random selection. Indeed ER (RA-5A) shows better NDCG curve as compared to ER (RA-Ran) method for all the data sets. The reason can be explained as follows. Some of these CQAs data sets have a lot of questions that have only one answer (See Table 4.6). RankSVM cannot build a very good ranking model for these questions because the rank is the comparison for at least two elements. Because ER (RA-Ran) randomly chooses questions from the CQAs, training data set will contains a lot of questions which have only one answer; Therefore, the ER (RA-Ran) model receives a lower accuracy than ER (RA-5A). Moreover, Figures4.5a, 4.5b, and 4.5c clearly indicate that in the TT data set the accuracy improvement from ER (RA-5A) to ER (RA-Ran) is much higher than the other two data sets. This is because in the TT data set more than 90% of questions has one answer. As ER (RA-5A) is better in general, we only compare ER (RA-5A) with the other algorithms.



(a) ER (RA) in SO-C (b) ER (RA) in SO-O (c) ER (RA) in TT

(d) ER (Hyb) in SO-C (e) ER (Hyb) in SO-O (f) ER (Hyb) in TT

Figure 4.5: NDCG score for ER (RA) and ER (Hyb)

4.4.2.2 Parameter γ in ER (Hyb)

ER (Hyb) uses γ to combine ER (VS) and ER (RA). Figures 4.5d, 4.5e, and 4.5f show the accuracy for different γ values (0.25, 0.5, and 0.75). When $\gamma = 0.75$, ER (Hyb) reaches the highest accuracy in SO-C; When $\gamma = 0.5$, in both SO-O and TT data sets ER (Hyb) reach the highest accuracy. To understand this, take a look at the average number of votes for each answer in the SO-C data set which is 2.6. In contrast, in SO-O and TT data sets the average number of votes for each answer is 1.6 and 1.3 respectively. (See Table 4.6). More voting results in a better quality assessment and hence weighing it higher is more useful for the hybrid approach. Based on this analysis, γ can be chosen based on the characteristics of the data set which are easy to compute. A similar analysis can be performed for the accuracy of the features and the training data set to further determine the value of γ in ER (Hyb).

4.4.2.3 Accuracy Analysis

In these experiments, an NDCG score of 1 is desirable for as many values of n as possible. A score of 1 up to an n indicates that the predicted expertise matches the baseline completely up to n experts. Hence, the goal is to improve the n value for which the NDCG score is 1. Even improvements by a small n is significant when identifying top k experts. For example, extending the NDCG score of 1 from @2 to @3 is an improvement of 50%. We have obtained an improvement of 40% for the TT data set and 160% for the SO-C data set. For the SO-O data set, our approach has achieved an NDCG score of 1 for @3 as compared to none of the earlier approaches even reaching an NDCG score of 1 for any n.

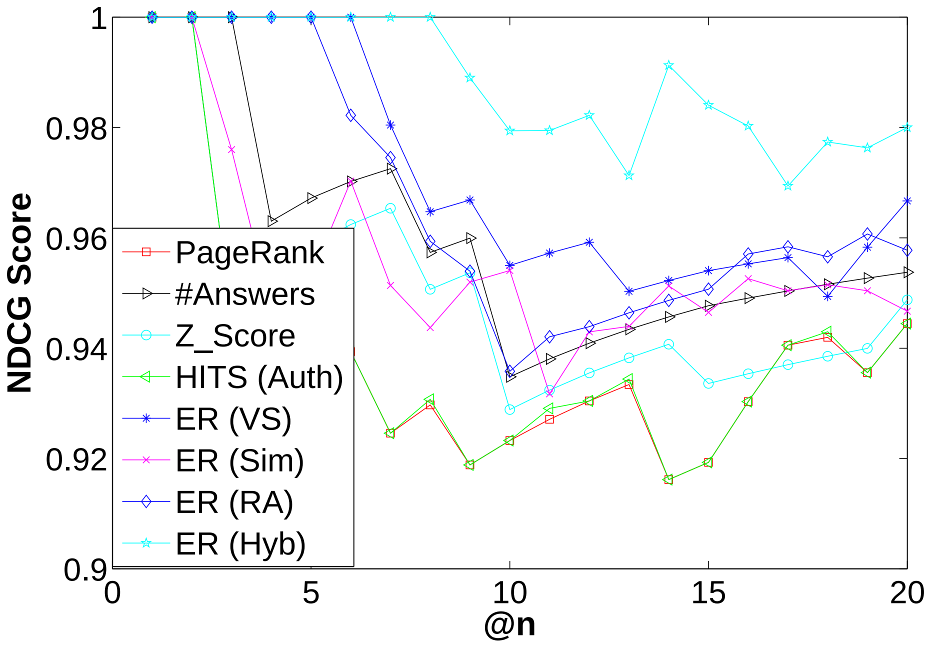


Figure 4.6: NDCG score for SO-C data set

Figures 4.6, 4.7, and 4.8 show the comparison of 8 approaches for all three data sets. Our experimental results clearly indicate: 1. in general, our approaches

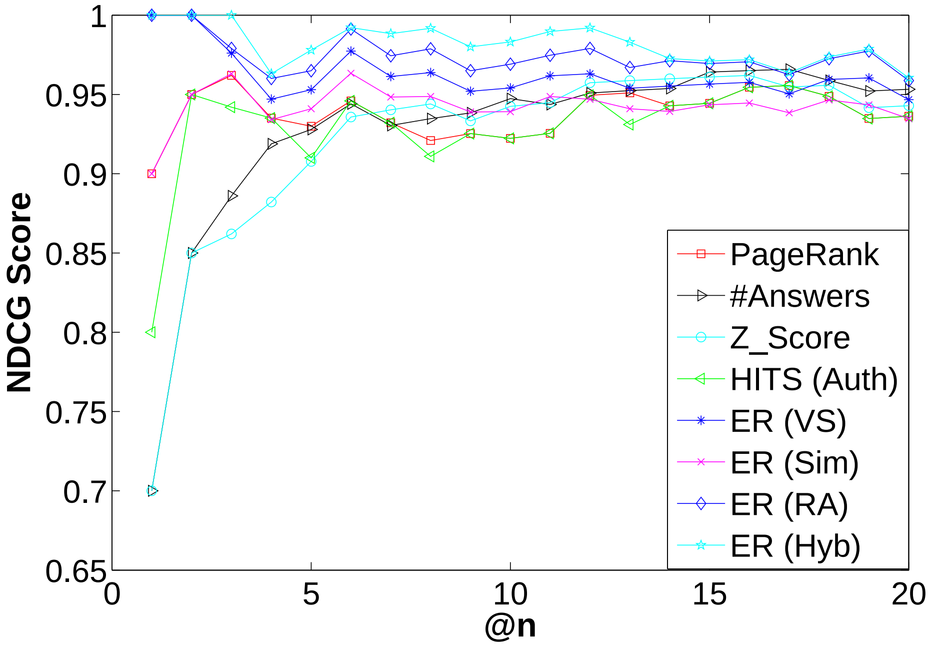


Figure 4.7: NDCG score for SO-O data set

(ER (Sim), ER (VS), ER (RA), ER (Hyb)) have better NDCG scores (for more n) as compared to the other four methods proposed in the literature. We believe this is due to the inclusion of answer quality and α, 2. #Answers and Z_Score methods have better NDCG curves than PageRank and HITS. The reason can be explained as the transitivity relationship in the Q/A community is much weaker than web page graphs, 3. Z_Score reaches similar accuracy as #Answers. In the Q/A community a user who answers a lot of questions is likely to ask few questions. In our test data sets these 50 users answer more than 10 questions and they ask very few questions (See Table 4.7). Thus, Z_Score and #Answers have similar NDCG curves, 4. PageRank and HITS have similar results because both of these two algorithms consider the transitive property as we discussed in Section 2, 5. ER (Sim) receives the lowest accuracy compared with the other three approaches, namely ER (VS), ER (RA) and ER (Hyb). The reason can be explained as similarity score only identifies the most

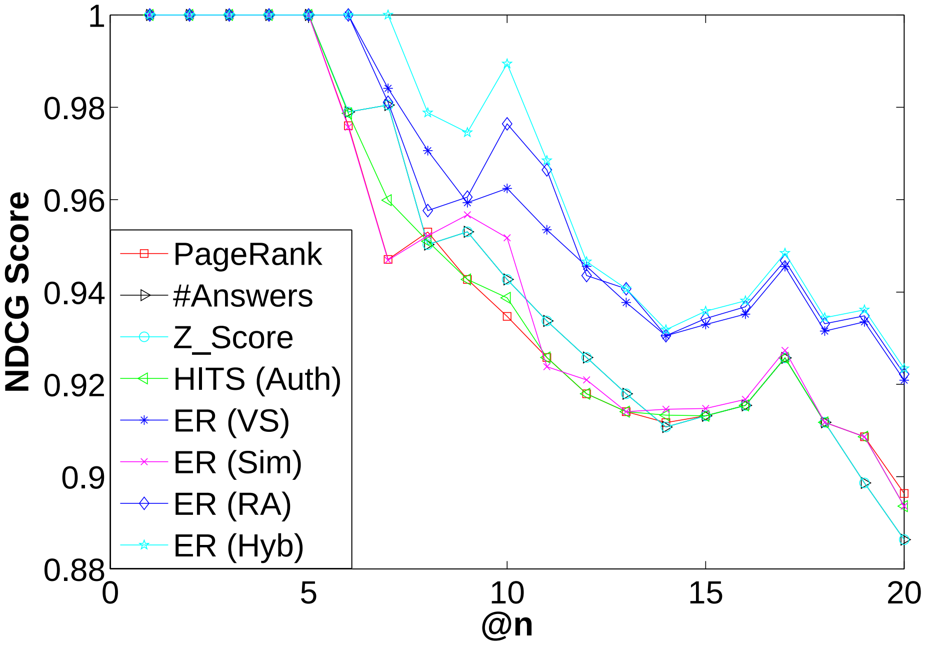


Figure 4.8: NDCG score for TT data set

related answer for a question but does not discriminate quality, 6. in SO-C data set ER (VS) is better than ER (RA); in SO-O and TT data set ER (RA) is better than ER(VS). The reason is that in SO-C data set each answer receives enough votes and hence ER (VS) is much more effective in identifying the users' expertise, 7. in SO-O data set, only ER-based approaches show a NDCG score of 1, and finally 8. as expected, ER (Hyb) reaches the highest accuracy in *all* three data sets because this method combines both ER (VS) and ER (RA) together with the right value for γ.

In summary, out initial hypothesis that both structure and answer quality are needed for these applications is borne out by the experimental results. Further, we have shown how to use alternative and available domain information beneficially. We have tied the values of weights to data set characteristics so that they can be determined easily.

4.5 Conclusions

In this chapter, we analyzed the ask-answer graphs generated from a CQAs and identified subtle differences with the conventional web graphs. Based on these differences, we chose domain information useful for mining expertise rank from CQA data sets. We proposed the ExpertRank framework and several approaches to predict the expertise level of users by considering both answer quality and graph structure. We argued and demonstrated why structure information alone is not sufficient for these applications and why domain specific quality information is important. We demonstrated the effectiveness of our approach using several large diverse data sets by comparing with traditional link-based approaches.

We also plan on doing additional larger-scale experiments using the Amazon Mechanical Turk on more data sets. In addition to global expertise, we plan on inferring concept-based expertise which can be used to extract quality answers for specific questions (based on the concepts used in the question) as well as identify users to answer specific questions.

CHAPTER 5

Identifying Specialists (for Concepts)

Community Question Answering services (CQAs) have become common and widely used. Hence, it would be beneficial to mine useful inferences from these Q/A data sets that support these services to improve their usability and confidence level. For example, if we can infer or identify the expertise of answerers for a specific concept from these data sets, we can route questions (topics) to the right answerers. With identification of expertise, the number of experts needed to cover a set of topics (in a CQA service) can also be optimized. This chapter addresses the problem of inferring expertise at the granularity of concepts so that it can be used as a building block for deriving other kinds of expertise such as expertise for answering a question, expertise on a topic, expertise in an area, etc.

Current approaches infer expertise using traditional link-based methods, such as PageRank, HITS, or other features (e.g., number of answers given by a answerer or Z_Score). However, these approaches mainly only identify the global experts called generalists in CQA services. However, for answering a specific question, these global experts are not as suitable as those who are experts in the concepts that prompted the question. By identifying experts for each concept, we propose a framework to identify specialists for answering a particular question. We first automatically extract all meaningful concepts from the question and build the answerer's expertise score for each concept. Then, we analyze domain information from several data sets and indicate how they can be used to analyze the answerer's expertise score. We also associate an importance measure for each concept and set weights based on that. Finally, using the Top-K search model, we combine the weight and answerer's expertise score for each concept together to identify the specialist for each question. We present our algorithm along with extensive experimental analysis that indicates superiority of our approach as compared to previously-proposed link-based methods.

5.1 Introduction

Community Question Answering services (CQAs) strives to provide users with meaningful information using the *ask-answer* paradigm. In short, these communities allow questioners to post questions and answerers to answer these questions. When a questioner posts a question using a CQA service, different approaches [31, 32, 48, 51] are used to find appropriate answerers to answer this question. Current online communities allow answerers to select questions that are of interest and answer them. Questioners post their questions to the community and answerers browse these questions at their pleasure and choose the ones to answer. This method is mainly used in many different CQAs, such as Stack Overflow (SO, `http://stackoverflow.com`), Yahoo! Answers (Y!A, `http://answers.yahoo.com/`) and AnswerBag (`http://www.answerbag.com/`), etc.

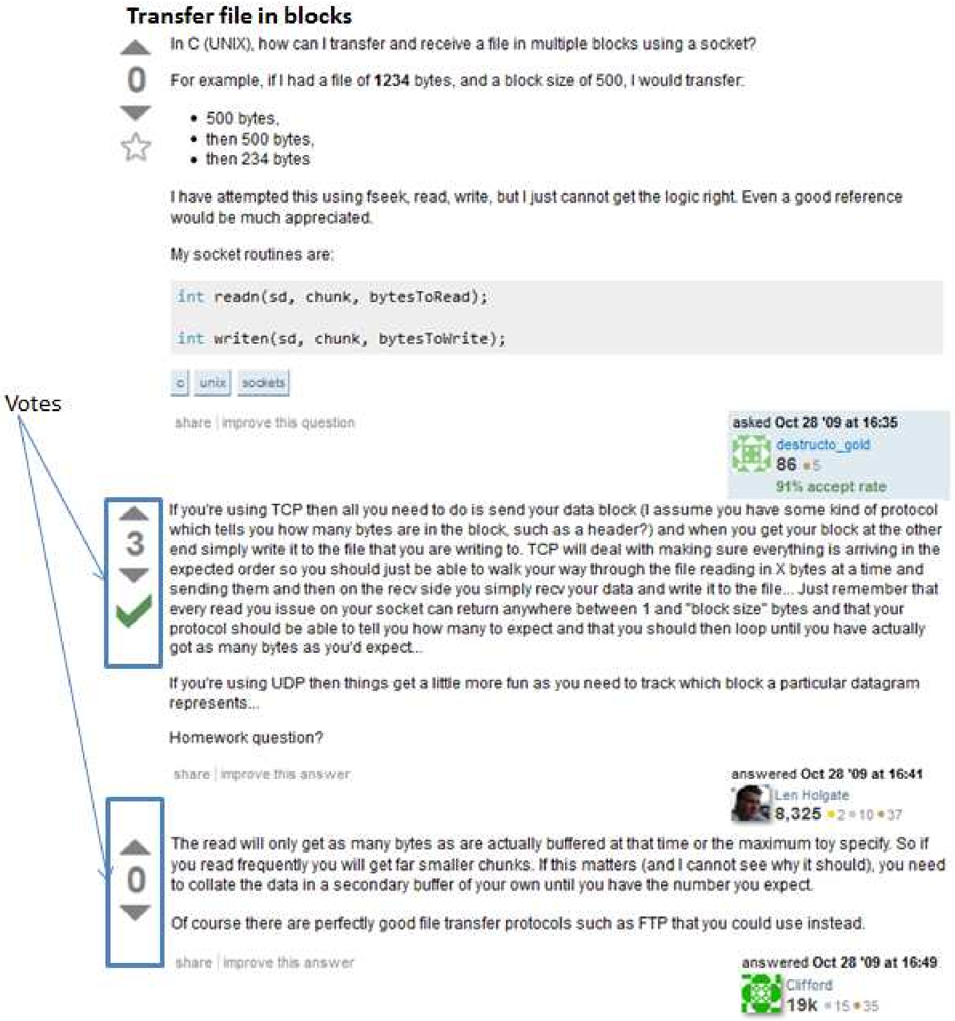
However, these widely used approaches completely ignore the answerer's quality. Interestingly, if you browse these communities, you will find a great number of unfriendly and nonsense answers, such as “You can search the result on Google,” “I don't know,” etc. If it is possible to assess the expertise of answerers in an acceptable manner, one can automatically identify a small cadre of answerers to answer these questions. The questioner will not only be relieved of this burden but s/he is also likely to receive better answers. This chapter mainly focuses on the problem of automatically identifying the best experts for answering a specific question in CQAs.

Several methods have been proposed to identify experts in the Q/A community. Information retrieval techniques have been used to discover experts from CQAs. Littlepage et al. [30] described a answerer's expertise as a term vector extracted from all of his/her perviously answered questions and calculated the cosine similarity (widely used in information retrieval) between a question vector (a term vector extracted from this question) and this answerer's vector as the expertise score. However, this results in a answerer list with no clear quality measure associated with it. Zhang et al. [31] and Jurczyk et al. [32] extended traditional link-based algorithms, such as PageRank and HITS, to compute (global) expertise rank in CQAs. The intuition behind these link-based methods is that if B can answer A's question, and C can answer B's question, C's expertise rank should be boosted because C is able to answer a question of someone who has some expertise. Liu et al. [48] extracted an implicit pairwise comparison from the best answer selection for each question and used the competition model to rank the answerer's expertise score. In addition, Bian et al. [51] proposed a semi-supervised reinforcement framework for calculating a answerer's expertise score. In sum, all of these approaches use some statistic model (e.g. competition model, random walk model, #Answers, Z_Score) to calculate the global answerer's expertise score in that CQAs. Intuitively, a answerer who gives good answers for a lot of questions should receive the highest expertise score in the CQAs.

An expert in CQAs can be classified into two categories: specialist and generalist. Specialists have great depth of experience in one or more concepts. They can easily give a good answer for a question that includes concepts in which they are experts. Generalists, on the other hand, choose to offer answers to a broad spectrum of questions, or they may not be considered as having developed expert-level skill on specific concepts. Generalists may be good at answering many questions, but typically are not at the same expertise level as that of a specialist. If questions are related to a particular area, specialists in that area can give a better answer for these questions.

Consider the following illustrative example. Figure 5.1 shows a "socket" question in C language and answerers Clifford and Len Holgate answer this question. Figure 5.2 shows the characteristics of answerers Clifford and Len Holgate. In Figure 5.2, Clifford answers 286 questions in Stack Overflow community and his answers involve various aspects of "C" language (e.g., memory, thread, buffer, etc.), but Clifford only answers 4 "socket" questions; The other answerer Len Holgate only answers 32 questions in CQAs, but 27 questions are "socket" questions. Clearly, in this example, Clifford is a generalist in C language and Len Holgate is a specialist in "socket" questions. In Figure 5.1, questioner destructo_gold asked how to transfer a file using socket functions, and Len Holgate, a socket expert, gave a better answer than Clifford (Len Holgate's answer received three votes which is higher than Clifford's answer (0 votes)). In other words, a specialist usually provides a higher quality answer than a generalist. However, since all current approaches (e.g., [31, 32, 48, 51]) only calculate users' global expertise score in CQAs, these approaches seem to identify the generalists rather than specialists. For example, in Figure 5.2, Clifford's expertise score (286 answered questions) will be much higher than user Len Holgate's score (32 answered questions) if #Answers [31] is used to calculate the answerer's expertise score. Therefore, these global expertise rankings are not very useful for selecting a user to answer a specific question. Since the user's ability to answer a question is definitely decided by his/her understanding of these concepts, [1] this chapter proposes Concept-Rank approach to identify specialists for each concept and specialists for answering a specific question.

[1] Concept can also be considered as a general idea, or something conceived in the mind; it can also be considered as a term occurring in a question.



Votes
0
3
0
c
unix
sockets
share | improve this question
share | improve this answer
share | improve this answer

Transfer file in blocks

In C (UNIX), how can I transfer and receive a file in multiple blocks using a socket?

For example, if I had a file of **1234** bytes, and a block size of 500, I would transfer:

- 500 bytes,
- then 500 bytes,
- then 234 bytes

I have attempted this using fseek, read, write, but I just cannot get the logic right. Even a good reference would be much appreciated.

My socket routines are:

```
int readn(sd, chunk, bytesToRead);

int writen(sd, chunk, bytesToWrite);
```

asked Oct 28 '09 at 16:35
destructo_gold
86 ●5
91% accept rate

If you're using TCP then all you need to do is send your data block (I assume you have some kind of protocol which tells you how many bytes are in the block, such as a header?) and when you get your block at the other end simply write it to the file that you are writing to. TCP will deal with making sure everything is arriving in the expected order so you should just be able to walk your way through the file reading in X bytes at a time and sending them and then on the recv side you simply recv your data and write it to the file... Just remember that every read you issue on your socket can return anywhere between 1 and "block size" bytes and that your protocol should be able to tell you how many to expect and that you should then loop until you have actually got as many bytes as you'd expect...

If you're using UDP then things get a little more fun as you need to track which block a particular datagram represents...

Homework question?

answered Oct 28 '09 at 16:41
Len Holgate
8,325 ●2 ●10 ●37

The read will only get as many bytes as are actually buffered at that time or the maximum toy specify. So if you read frequently you will get far smaller chunks. If this matters (and I cannot see why it should), you need to collate the data in a secondary buffer of your own until you have the number you expect.

Of course there are perfectly good file transfer protocols such as FTP that you could use instead.

answered Oct 28 '09 at 16:49
Clifford
19k ●15 ●35

Figure 5.1: A Sample Content from Stack Overflow

In our work, a questioner poses a question in a CQA community. We assume that a answerer, who masters/understands the concepts in that question very well, has a greater chance of providing a better answer for that question. Thus, we propose an off line process (See Figure 5.3) during which we extract the needed concept(s) from each question by considering the characteristics of that term. In our approach, we choose nouns and foreign words to describe concepts in a question as that increases the accuracy of the approach as compared to choosing other alternatives (e.g., adjectives,

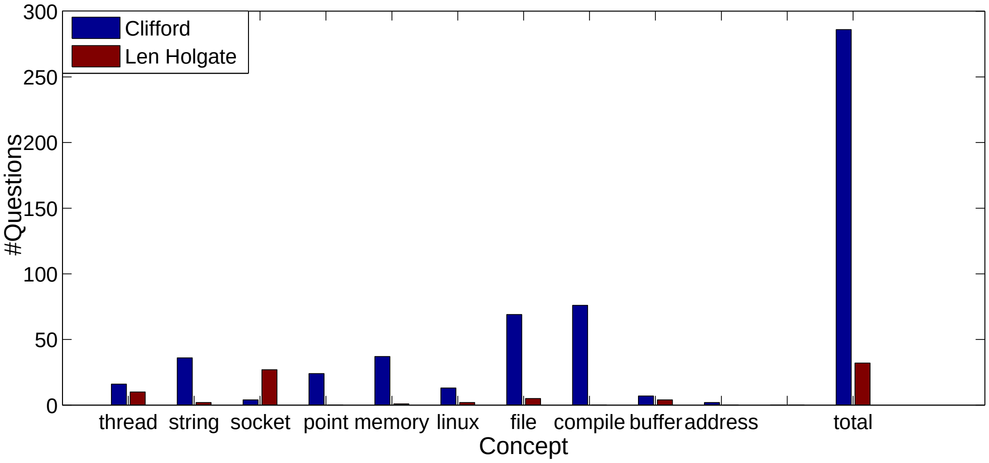


Figure 5.2: Characteristics of Answerer Clifford and Len Holgate in Stack Overflow: x axis describes 10 widely used concepts in C language extracted from Stack Overflow, and y axis describes the number of questions answered by a answerer which contains that concept. The column "total" in x axis describes the total number of questions answered by a answerer.

verbs, adverbs, etc.) This seems to match our intuition that concepts correspond to nouns and are also borne out experimentally (the accuracy reaches more than 80% as elaborated in Section 5.4). The need for foreign words is also clear as many concepts in programming languages such as C and Java are not commonly used words in English. Second, in order to calculate the answerer's expertise score for each concept, we propose five approaches by considering different domain-specific information. The voting information (See Figure 5.1) is one piece of information that can be used to calculate answer quality. Some link-based approaches (e.g., HITS, PageRank, etc.), which describes the user's ability to help other users in CQAs, are also used to compute the answerer's expertise score for each concept. Then, we also consider the weight of each concept by analyzing the importance of a concept (See Section 5.3). Different from the traditional tf-idf model [85], a concept which is widely used is set a higher weight. During an online process (See Figure 5.3), considering both answerer's expertise score and weight score for each concept in that question, we can obtain answerer's ranking score for each question.

This approach, although aimed at identifying an expert to answer a question, can also be used to identify experts for a sub-area of interest, a topic, and even for breadth by including a large number of concepts from related or unrelated areas. This is a generalization of the way in which we can identify experts at different levels by using the notion of concept rank.

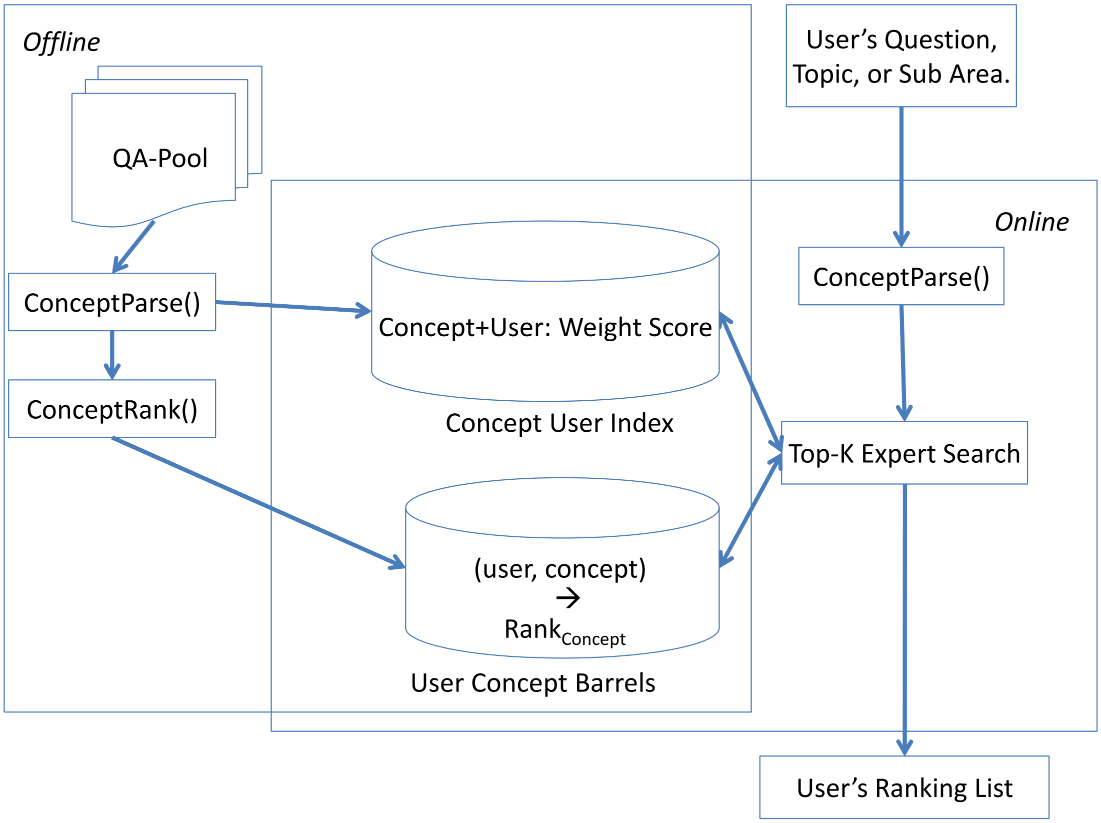


Figure 5.3: Proposed System for Inferring User-Concept Rank

Identifying specialists in CQAs is important because a lot of Q/A services can benefit from this. For example, Q/A communities such as Stack Overflow, Yahoo! Answers and Blurtit (`http://www.blurtit.com`) can benefit from this automated ranking approach to improve their answer quality by forwarding questions to appropriate specialists instead of allowing any answerer to answer these questions.

Contributions: The contributions of this chapter are:

- We proposed a framework to identify specialists for a concept in general and a specialist for a particular question. We automatically extracted meaningful and relevant concepts from questions and infer answerers' expertise score for each concept. We analyzed domain-specific information from several data sets and indicate how they can be used to analyze the answerer's expertise score. We also analyzed the importance of each concept and associated a weight for each concept. Considering both weight and the answerer's expertise score for each concept, a specialist for a particular question can be inferred by the Top-K search model.
- Extensive experimental analysis was performed on real-world data sets to show how the proposed framework provided more accurate results than the traditional global ranking approaches.

Chapter Organization:

Section 5.2 motivates and defines the problem of computing the answerers' expertise score. Section 5.3 introduces the framework of Concept-Rank. Section 5.4 shows extensive experimental results on real-world data set and their analysis. Section 5.5 offers conclusions.

5.2 Problem Statement

This chapter focuses on the general problem of mining the expertise level of users according to the level of concepts *in CQAs and rank them. This chapter also evaluates the effectiveness and utility of this concept-based ranking approach to improve answer quality in CQAs.*

Given a CQA data set consisting of users u_1, u_2, u_3, ..., u_n along with questions, answers, and available domain information (e.g., vote information), our goal is to infer the answerers' expertise for a question q_i and rank them using the notion of concept rank, which ranks each user at the level of a concept. Ranking of answerer for a question will facilitate: (i) CQA services to optimize the number of experts for the different domain as need and (ii) direct/route questions to appropriate answerers. We also point out that the user's expertise rank order is subjective and may vary for different evaluators. Furthermore, no real expertise rank exists in the CQA data set. Thus, it is difficult to find a gold standard to when defining an evaluated rank order. As a result, manual evaluation has been commonly done by researchers on this topic [31, 59]. In this chapter, we use manually evaluated rank order as the standard for comparing our automated prediction of ranking results for a question. For each data set, two experts analyze the questions and answers of a small number of users (10 in our case) and manually rank the expertise of these users for a particular question/topic.

5.3 Concept-Rank Framework

In our approach to Concept-Rank (See Figure 5.3), during the off-line process, we first extract concepts from questions and use their answers (or extracted ask-answer subgraphs) to pre-compute the answerer's expertise score for a concept. Then, we calculate the weight for each concept based on their occurrences in questions. At runtime, these scores are combined based on the concepts extracted from that question to form a composite answerer's expertise score for that question. In the following sections, we discuss our approach for three aspects: (i) extracting concepts from questions, (ii) computing the answerer's expertise score for each concept, and finally (iii) setting a weighted score for each concept.

5.3.1 Extracted Concepts

We view the concept as a basic cognitive unit for the user to understand a question. Further, we use two important characteristics associated with a concept. First, a concept is an cohesive unit of cognitive understanding; It cannot be split. For example, "C++" is the basic concept as a computer language, hence "C++" cannot be split into "C" and "++". Second, concept is also indispensable for answering a question. If a answerer does not understand a concept in a question properly, this answerer may not give a correct answer for this question. For example, Example 5.1 shows a sample question about "transferring file by socket communication" extracted from Stack Overflow community and experts manually select concepts for this question. In this example, in order to answer this question, users need to understand/master at less six concepts (e.g., "transfer", "file", "block", "C", "Unix", "socket") in this question; otherwise, this question cannot be answered very well. Moreover, Example 5.1 also indicates that these six concepts are all nouns except "transfer", so that these words which are nouns can be used as the concepts for a question. In order to automatically extract the concepts from these questions, we used the tagger software[2] which is used to tag each word as one of 48 parts of speech labels [86]. Table 5.1 shows the tags extracted from the question in Example 5.1. In Table 5.1 five concepts tagged by three "noun" speech tags (e.g., "Proper noun, singular" (NNP), "Noun, plural" (NNS), "Noun, singular or mass" (NN)) match well with the manually marked six concepts. In order to select the better speech tags , we run a lot of experiments on these real Q/A data sets. Finally, we choose the following five speech labels, such as "Noun, singular or mass" (NN), "Proper noun, singular" (NNP), "Noun, plural" (NNS), "Proper noun, plural" (NNPS) and "Foreign word" (FW), from the tagger that corresponds to the manually marked concepts. (See Section 5.4)

[2] this software can be downloaded from `http://nlp.stanford.edu/software/tagger.shtml`

Table 5.1: Extracted Tags from a Sample Question (In Example 5.1)

Words	tag (Full Name)	tag (Abbreviation)
In	Preposition/subordinating conjunction	IN
C	Proper noun, singular	*NNP*
(	Left bracket character	LRB
UNIX	Proper noun, singular	*NNP*
)	Right bracket character	RRB
,	Comma	,
how	wh-adverb	WRB
can	Modal	MD
I	Personal pronoun	PRP
transfer	Verb, base form	VB
and	Coordinating conjunction	CC
receive	Verb, base form	VB
a	Determiner	DT
file	Noun, singular or mass	*NN*
in	Preposition/subordinating conjunction	IN
multiple	Adjective	JJ
blocks	Noun, plural	*NNS*
using	Verb, gerund/present IN Preposition/subordinating participle	VBG
a	Determiner	DT
socket	Noun, singular or mass	*NN*
?	?	?

Example 5.1 ***Question (extracted from Figure 5.1):*** *In* ***C*** *(****UNIX****), how can I* ***transfer*** *and receive a* ***file*** *in multiple* ***blocks*** *using a* ***socket****?*

In this example, experts manually marks six words, such as "transfer", "file", "block", "C", "Unix", "socket", as the concepts.

5.3.2 Expertise Score for a Concept

Different domain-specific information can be extracted from CQAs to identify answerer's expertise level for each concept. A user asking/answering a question, which contains a concept, is one piece of information that can be used to evaluate that user's expertise level for that concept. Some global rank approaches (e.g., PageRank [31], HITS [32] and #Answer [31]) can also be applied to evaluate user's expertise level for each concept. In addition, for one concept the voting information for the answerer's answer can also be used to imply answerer's quality (See Figure 5.1, 3 is the voted score for Len Holgate's answer and 0 is the voted score for Clifford's answer.) Similarly, other information available in that domain can also be used for this purpose.

To predict the user-concept rank, we propose the following approaches.

- Q/A_Score, CR (Q/A): The intuition behind Q/A_Score measure is that if a user understands a concept (or a set of concepts), s/he will answer more questions on that (those) concept(s). Otherwise s/he is likely to ask questions on that (those) concept(s). We calculate the *Q/A_Score* as follows.

  $$Q/A\_Score(u_i, c_j) = \frac{|A(u_i, c_j)| - |Q(u_i, c_j)|}{\sqrt{|A(u_i, c_j)|^2 + |Q(u_i, c_j)|^2}} \tag{5.1}$$

  where $A(u_i, c_j)$ and $Q(u_i, c_j)$ indicate, respectively, answers and questions by user u_i for concept c_j. *Q/A_Score* describes the level of user participation. A *Q/A_Score* of -1 indicates a questioner whereas a score of 1 indicates an answerer for the concept c_j.

- PageRank Score, CR (PR): *For each concept*, we extract the question-answer graph and calculate PageRank authority score. The main difference between expertise rank [31] and this method is that expertise rank calculates PageRank score for all the concepts, but our approach will calculate the PageRank score for each concept individually. Because our question and answer graph is only relevant to one concept, it effectively overcomes the unrelated transitivity issue that we discussed earlier in Chapter 4. Therefore, PageRank authority score for a concept receives better accuracy than the original rank score derived from all concepts.

- HITS Score, CR (HITS): In the same way, for each concept, we extract the question-answer graph and calculate HITS authority score. For the same reason for the CR (PR) algorithm, the HITS algorithm for each concept is likely to achieve a higher accuracy than the original HITS rank using all concepts.
- Answer Score, CR (#A): The intuition behind this approach is that if an answerer answers a lot of questions on this concept, this answerer is likely to have mastery over that concept. Therefore, the number of answered questions by this answerer which contains this concept can be used as this answerer's expertise score for this concept. Since this score varies significantly from concept to concept, it needs to be normalized. We normalize this score using maximum number of answered questions for each concept. $maxA\_Score(c_j)$ is the maximum number of questions answered by a answerer which contains concept c_j and is computed as follows:

$$maxA\_Score(c_j) = max\{A\_Score(u_i, c_j)|i = 1, ..., n\} \tag{5.2}$$

where $A\_Score(u_i, c_j)$ is the number of answered questions by answerer u_i which contains concept c_j, and n is the total number of answerers in the community. The normalized NA_Score is computed as

$$NA\_Score(u_i, c_j) = \frac{A\_Score(u_i, c_j)}{maxA\_Score(c_j)} \tag{5.3}$$

For the answer score, if answerer u_i does not answer any questions which contains the concept c_j, $NA\_Score(u_i, c_j)$ is equal to 0; if this answerer u_i answers the maximum number of questions which contains concept c_j, $NA\_Score(u_i, c_j)$

is equal to 1. In other words, if $NA\_Score(u_i, c_j)$ is equal to 1, this answerer u_i is a best answerer in the community to have mastery over that concept c_j.

- Voted Score, CR (V): The above-mentioned four methods (e.g., CR (Q/A), CR (PR), CR (HITS), CR (#A)) do not consider the quality of an answer. In Q/A communities, user vote-based methods are widely used to mark answer quality which can be used for deriving concept rank. In this method the higher the votes for an answer, the better quality of this answer. There are two kinds of voting principles. The first one is "Support Vote", which means voters are only allowed to give a positive vote for an answer (e.g., Yahoo! Answers). The second one is "Support and Oppose Vote," where a voter can give a positive vote to support an answer or a negative vote to oppose an answer (e.g., *Stack Overflow*). For the "Support Vote," we directly use the support vote as a voted score. However, for the "Support and Oppose Vote," we use the difference between the "Support and Oppose Vote" and the "Minimum Support and Oppose Vote" in this question as a voted score for this answer. For example, Table 5.2 shows a small sample of questions and answers from the Stack Overflow community. All these four posts in this sample are "Socket" related questions. Since the $minV\_Score(u_3, \text{``}Socket\text{''}, q_2)$ is equal to -1, the $V\_Score(u_3, \text{``}Socket\text{''}, q_2) = 3$ - (-1) = 4.

  Since this voted score varies significantly from question to question, it needs to be normalized. We normalize the voted score using the minimum and maximum values of the scores for *each* questions. $minV\_Score(u_i, c_j, q_k)$ ($maxV\_Score(u_i, c_j, q_k)$) as the minimum (maximum) voted score of all answers to user u_i's question q_k about concept c_j are computed as follows:

$$minV\_Score(u_i, c_n, q_k) = min\{V\_Score(u_j, c_n, q_k)|j = 1, ..., n\} \quad (5.4)$$

$$maxV\_Score(u_i, c_n, q_k) = max\{V\_Score(u_j, c_n, q_k)|j = 1, ..., n\} \quad (5.5)$$

and the normalized V_Score is computed as

$$NV\_Score(u_i, c_n, q_k) = \frac{\frac{V\_Score(u_i,c_n,q_k) - minV\_Score(u_i,c_n,q_k)}{maxV\_Score(u_i,c_n,q_k) - minV\_Score(u_i,c_n,q_k)} + \epsilon}{1 + \epsilon} \quad (5.6)$$

which is in the range $[\frac{\epsilon}{1+\epsilon}, 1]$. An ϵ is used to adjust the lower bound of normalized values. We do not want to set the normalized value for the answer receiving the lowest voted score to 0 because these answers have been assessed for quality as opposed to answers that have not been voted upon (which receive a normalized score of 0). Thus, if an answer receives the lowest voted score, its normalized quality score is equal to $\frac{\epsilon}{1+\epsilon}$. In our experiments, ϵ is set to 0.1. ϵ value greater than 0.1 or 0.15 does not make sense as it is a compensatory value. In Table 5.2, since in Post2 (Q2) $maxV\_Score(u_3, \text{“}Socket\text{”}, q_2)$ is equal to 5 (4-(-1)) and $minV\_Score(u_3, \text{“}Socket\text{”}, q_2)$ is equal to 0 (-1 - (-1)), $NV\_Score(u_3, \text{“}Socket\text{”}, q_2)$ is 0.82.

We use the average voted score of all answers as the answer quality score for that answerer.

$$A\_Quality(u_i, c_j) = \frac{\sum_{k=1}^{|A(u_i,c_j)|} NV\_score(u_i, c_j, q_k)}{|A(u_i, c_j)|} \quad (5.7)$$

where $Vote(u_i, c_j, q_k)$ describes the voted score for u_i's answer in question q_k which contains concept c_j. $|A(u_i, c_j)|$ describes the number of questions answered by u_i which contains c_j. A higher value indicates this answerer has mastery over this concept. For example, in Table 5.2, $A\_Quality(u_3, \text{“}Socket\text{”})$ is equal to 0.33.

Table 5.2: A Small Sample Question Answer Community about Concept “socket”

Post 1			Post 2			Post 3			Post 4		
Q_1	U_4	V_S	Q_2	U_5	V_S	Q_3	U_5	V_S	Q_4	U_3	V_S
A_1	U_1	8	A_1	U_2	4	A_1	U_1	6	A_1	U_1	1
A_2	U_2	2	A_2	U_3	3	A_2	U_3	2	A_2	U_2	1
A_3	U_3	1	A_3	U_4	-1	A_3	U_2	2	A_3	U_4	0

[1] In this example, V_Score (V_S) is extracted from Stack Overflow community. Since Stack Overflow community supports the “Support and Oppose Vote” system, u_4’s voting score is “-1”. We use the the difference between “Support and Oppose Vote” and “minimum Support and Oppose Vote” as the voted score.

In addition to answer quality, the number of questions this answerer answered on concept c_j also captured whether or not answerer u_i was familiar with that concept c_j. Hence, to address this problem, we first define the concept’s broad value $C\_Broad(c_i)$, which is used to describe the breadth of a concept. If the concept is narrow, that means this concept has been asked in very few questions. Therefore, if a concept is narrow, it also means that users who answer these few questions related to concept are likely to be experts in this concept and vice versa. For example, in Example 5.1, concept “C” is a broad concept in *Stack Overflow* community; users asked 12,144 questions about the concept “C”. Compared with concept “C,” concept “socket” is a narrow concept; users only asked 749 questions about the concept “socket”. Therefore, compared with concept “socket”, concept “C” is difficult for a user to understand/master since the denotation of concept “C” is broad (C is more of a topic or an area than a concept); in order to master the “C” language, users need to learn a lot of other concepts (e.g., “array,” “point,” “network communication,” etc). However, concept “socket” is easier for a user to learn than concept “C” since the denotation of concept "socket" is relative narrow; to master the concept "socket," users only need to learn the network communication. Thus, we define the $C\_Broad(c_i)$ equation as follows.

$$C\_Broad(c_i) = \alpha|Q(c_i)| \tag{5.8}$$

where $|Q(c_i)|$ describes the number of questions which contains concept c_i, and α is the parameter to adjust the notion of breadth. In our experiments, α is set to 1. Therefore, we use the $C\_Broad(c_i)$ to evaluate the difficulty of concept c_i to be learned and also use the number of answered questions for this answerer to evaluate this answerer as a potential master of this concept. Therefore, we have the following equation for the question factor.

$$Q\_Quality(u_i, c_j) = (1 - \frac{1}{(1 + |A(u_i, c_j)|)})^{log_2(C\_Broad(c_j))} \tag{5.9}$$

where $|A(u_i, c_j)|$ is the number of questions answered by u_i which contains concept c_j. This equation indicates that if this concept c_j is a narrow concept ($C\_Broad(c_j)$ is small) and this answerer u_i, meanwhile, answers a lot of questions about this concept ($|A(u_i, c_j)|$ is large), u_i should have a high expertise score for this concept c_j. In Table 5.2, since $C\_Broad(c_j)$ is equal to 4 (4 questions about concept "Socket") and $|A(u_i, c_j)|$ is equal to 3 (answerer u_3 answers 3 questions around these 4 questions), $Q\_Quality(u_3, \text{“}Socket\text{”})$ is equal to 0.56. Since both of these parts are equally important to decide user quality, we multiply these two parts together and have the following equation to calculate the vote score. If any one of the two has a low value, the answerer's expertise score will also be low.

$$VR\_Score(u_i, c_j) = Q\_Quality(u_i, c_j) \times A\_Quality(u_i, c_j) \tag{5.10}$$

5.3.3 Weight of a Concept

The main function of concept-based expert ranking system is the enhancement of retrieval effectiveness. When a new question is posed, the system are able to do two things: (i) find the experts who can answer this question better using the concepts occurring in the question, and (ii) find experts based on the importance of the concepts to this specific question. In other words, it may not be appropriate to assign an "importance" weight to a concept that is global in nature. As each concept in a question has a different importance, we need to assign a weight for each concept for this question. For example, in Example 5.1, "socket," "UNIX" should set the higher weight than the other four concepts (e.g., "C," "file," "transfer," "blocks" etc.) since these two concepts, "socket" and "UNIX", are much more exact to identify whether or not the answerer understand this question. In this section, we mainly discuss approaches to set a weight for each concept.

Based on our observations, three factors should be considered to set the weight for each concept. First, concepts that are frequently mentioned in a question appear to be useful. This suggests that a *concept frequency* factor can be used as part of the weighting system measuring the concept's frequency of occurrence in a question. To be same as the term-frequency (tf) [85] which is widely used in the document retrieval system, we define the concept frequency for a question (See qc in Table 5.3). In addition, we also define a binary weight for a concept in a question to make a comparison between two weighting functions (See qb in Table 5.3). However, since most questions in Q/A community consist of a short text (the average character length

Table 5.3: Weighting Alternatives for Concepts

Name	Formula	Description
Question-Concept Frequency Component		
qb	1.0	binary weight is equal to 1 for concept t_1, iff this question contains concept c_i. Otherwise, 0.
qc	$\frac{m}{M}$	concept frequency for a question (m is the number of times of this concept c_i used in this question, and M is the number of times these concepts used in this question)
User-Concept Frequency Component		
ub	1.0	binary weight is equal to 1 for concept t_1, iff this user answers a question which contains this concept c_i.
uc	$\frac{m}{M}$	concept frequency for a user (m is the number of questions this user answered which contains this concept c_i, and M is the number of questions this user answered)
User-Concept Collection Frequency Component		
ux	1.0	binary weight is equal to 1 for concept t_1, iff this user answers a question which contains this concept c_i.
uf	$\frac{n}{N}$	an collection frequency factor (N is the number of users, and n is the number of users who answer a question contains concept c_i)
ui	$log\frac{N}{n}$	multiply original tf factor by an inverse collection frequency factor (N and n are the same as uf).

of a question is 614.3 in SO data set), this concept frequency does not distinguish the importance of that concepts. We will discuss this further in Section 5.4.

Second, we also consider the frequency of concepts addressed by the answerer. Noting preferences for specific kinds of questions and their ability to handle the concepts occurring in these questions. The *concept frequency* factor can be used as part of the weighting system measuring the frequency of occurrence of the concepts in the potential expert's answered question (See uc in Table 5.3). Meanwhile, we also define a binary weight for the answerer as the comparison (See ub in Table 5.3). However, concept frequency for this answerer only means this answerer knows this concept better than the other concepts but this does not mean that this person has grasped this concept better than other answerer. For example, answerer A totally answers 10 questions and in these ten questions two questions are about "socket." User B answers 50 questions and among them four questions are about "socket". In this example, uc (user A) (2/10 = 0.2) is higher than uc (user B) (4/50 = 0.08). However, clearly answerer B is the proper answerer for "socket" question because B answers more "socket" questions than A. In other words, answerer's bias for the concept should not be used to evaluate answerer's expertise for that concept. Therefore, the concept frequency of an answerer should not be used in the weighting system. The experimental results indicate this clearly (See Section 5.4).

Third, we also obverse that the concepts which are widely used should be useful to evaluate the answerer's expertise. We define three types of weighting functions to set different weights for the concepts. First, ui (See ui in Table 5.3) is similar to the inverse document index idf [85], which sets a high weight for these narrow concepts. Second, uf (See uf in Table 5.3) sets a high weight for broad concepts. Third, we also define a binary weight as the baseline for comparison (See ux in Table 5.3). Since more users are likely to ask questions and provide answers on broad concepts, we have more statistical information to rank these users; therefore, the accuracy of the uf approach should be better than the other two approaches (ui, ux). Related experimental results will be shown in Section 5.4.

In conclusion, we only use uf to set the weight for each concept.

When a new question comes, we extract concepts from this question and calculate the top-k expertise score of answerers using the following.

$$Rank(q_i, u_j) = \sum_{k=1}^{|C(q_i)|} uf(c_k) \times rank(u_j, c_k) \tag{5.11}$$

where $rank(u_j, c_k)$ describes the expertise rank score for user u_j for concept c_k[3], $uf(c_k)$ is the weight of concept c_k and $|C(q_i)|$ is the number of concepts in question q_i.

5.4 Experimental Analysis

We used different data sets to test our approaches. We also performed a number of experiments to validate various aspects of our approach ranging from accuracy of concept extraction to accuracy of our approaches, etc.

Stack Overflow (SO) data set: SO data set focuses on computer programming topics. Unlike other traditional Q/A services, SO allows a answerer to modify other answerer's answers. In other words, when an answerer wants to answer a question, s/he has two choices: modify an existing answer or provide a new answer. As a result, the average number of answers for each question is only 2.36. We only consider the first answerer who posts the answer as the answerer, because, in most cases, the first answerer is likely to provide a larger contribution than other answerers. Each question in this community is marked with a topic tag (e.g., "C" or "Oracle"). We used questions tagged as "C" ("Oracle") as SO-C (SO-O) data set. Broader statistical characteristics of these two data sets are shown in Table 5.4.

Turbo Tax (TT) Data set: TT service discusses tax-related issues. This community enrolls many experts to answer questions, so most of the users are mainly questioners. Thus, the average number of answers for each question is only about 1.11. Table 5.4 also shows TT data set characteristics.

[3] According to the different domain information, $rank(u_j, c_k)$ is calculated by each abovementioned approach (e.g., Q/A_Score, PageRank score, HITS score, Answer score, Voted score) respectively.

Table 5.4: Complete Data set Characteristics

Data set	#Questions	#Answers	#Users
SO-C	25,942	91,615	17,085
SO-O	8,644	21,879	5,722
TT	501,978	567,515	486,176

5.4.1 Accuracy of Concept Extraction

In this experiment, we used two independent experts from the computer science department and the business department who are familiar, respectively, with "C," "Oracle" and "Tax-related issues" to extract concepts (or words which are useful to understand this question) from 100 randomly chosen questions in these three data sets. We used the Jaccard similarity coefficient [87] to compare two experts' evaluation results. The Jaccard similarity coefficient between two experts' results was 0.891. In order to maintain consistency of evaluation, we only kept the concepts which are selected as meaningful concepts by both experts as the manual standard and then compared this manual standard with automatically extracted results.

We used tagger software[4] to automatically extract concepts and compare the terms by each speech labels with this manual standard. In this experiment, we used the precision, recall and F-measure [88] . Precision score described the fraction of relevant marked terms of that speech label; recall score was the fraction of relevant marked terms that was retrieved, and F-measure was the harmonic mean of precision and recall. Figure 5.4 showed the precision, recall and F-measure score of each speech label for SO-C data set[5]. Our experimental results clearly indicated that "NNPS,"

[4] For tagger software, although the accuracy of left3words model is sightly lower than bidirectional model, the speed of left3words are much faster than bidirectional model. Therefore, since we need to extract the concepts from a large number of questions (See Table 5.4) in SO-C data set, we use left3words model to mark the label for each concept.

[5] The other two data sets (e.g., SO-O data set and TT data set) have similar distributions.

“NNP,” “NNS,” “NN,” “FW” and “Adjective” (JJ) was six highest accuracy labels (e.g., 1 for “NNPS,” 0.86 for “NNP,” 0.84 for “NNS,” 0.75 for “NN,” 0.59 for “FW” and 0.45 for “JJ”). The labels “NNPS”, “NNS”, “NNP”, “NN” was used to describe nouns; the label “FW” indicated a foreign origin term, and the label “JJ” indicated adjective. In addition, in Table 5.5 as each feature was added one at a time in *accuracy rank order* as shown in Figure 5.4, one was able to observe consistent decrease of accuracy but improvement in F-measure score for all three data sets (from 0.006 to 0.82 for the SO-C data set, from 0.008 to 0.83 for the SO-O data set, and from 0.006 to 0.85 for the TT data set). Table 5.5 also showed that only five labels (“NNPS,” “NNP,” “NNS,” “NN,” “FW”) were used in accuracy experiment, the F-measure reached a high score (0.81 for SO-C data set, 0.83 for SO-O data set, and 0.85 for TT data set), and the other “JJ” label did not improve the F-measure score well (1.2% improved for SO-C data set, 0.23% improved for SO-O data set, and -1.16% improved for TT data set). Therefore, we only used five labels (e.g, “NNPS,” “NNP,” “NNS,” “NN,” “FW”) to extract the concepts from these three data sets.

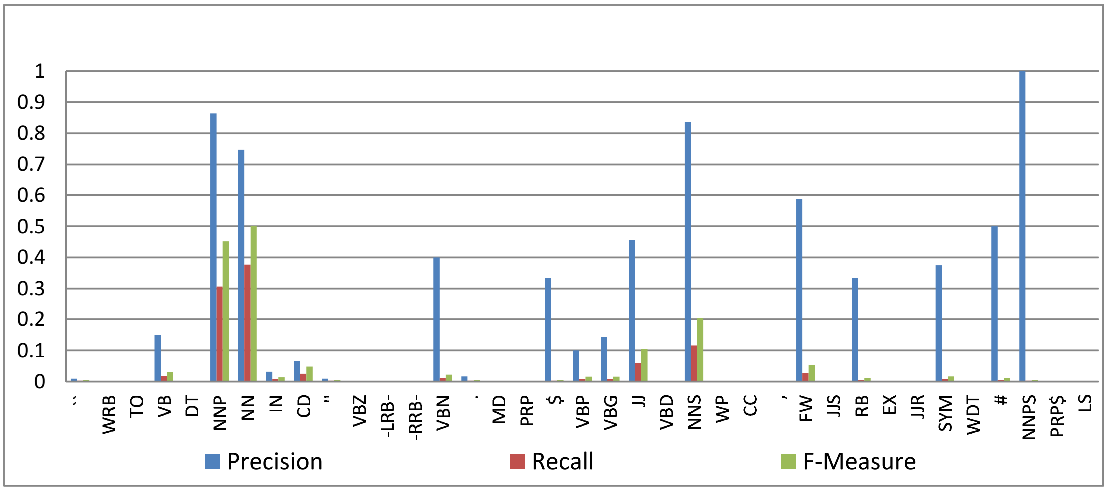


Figure 5.4: The accuracy of each speech label marked by tagger software for SO-C data set. The description of each tag shows in [86]

Table 5.5: Speech Label Analysis of Three Data Sets

Data Sets	SO-C			SO-O			TT		
Label	Precision	Recall	F-Measure	Precision	Recall	F-Measure	Precision	Recall	F-Measure
“NNPS”	1	0.002	0.006	0.98	0.004	0.008	0.96	0.003	0.006
+“NNS”	0.86	0.12	0.21	0.82	0.14	0.24	0.88	0.17	0.28
+“NNP”	0.86	0.42	0.57	0.82	0.47	0.60	0.82	0.56	0.67
+“NN”	0.80	0.80	0.80	0.81	0.85	0.83	0.81	0.88	0.84
+“FW”	0.79	0.83	0.81	0.80	0.87	0.83	0.80	0.90	0.86
+“JJ”	0.75	0.90	0.82	0.77	0.90	0.83	0.79	0.91	0.85

5.4.2 Accuracy of Concept Rank

In this section, we analyze the accuracy of ranking approaches proposed in the previous section. We used five of the sample questions given in Table 5.6 for each data set. For these studies, a standard with which to compare results is extremely important. Our problem for finding a standard was exacerbated by the fact that the notion of expertise itself can be quite subjective. For each question, these is no user expertise rank information nor can it be derived from the data sets. Hence, as has been done by other researchers (e.g., [31]), we used human experts to manually evaluate the user's expertise for all these three data sets. Due to the large number of users (See Table 5.4), it is impossible to manually rate all users in each data set. Hence, for each question, we randomly chose 10 users who have answered at least 10 questions to provide enough content for manual evaluation. Five levels of expertise as shown in Table 5.8 were used. The evaluation questions and user expertise ranking list (evaluated by two experts) can be downloaded from our web site (`http://itlab.uta.edu/yzcai/evaluation.zip`), and readers can check our evaluation results in detail. We have used two independent experts who are very familiar with the C language and the Oracle database from the computer science department to evaluate the two SO data sets. We have used two experts from the business department to evaluate the Turbo Tax data set. *None of these experts take part or are associated with this research.* After each expert evaluated the data sets independently, for sanity

Table 5.6: Test Questions Used for three data sets

Id	Question
5 Questions for SO-C data set	
1	I am a beginner for C language. Can you recommend some books to me?
2	What is a typedef enum in Objective C?
3	Are memory leaks ever ok?
4	How do you pass a function as a parameter in C?
5	What is the difference between a definition and a declaration?
5 Questions for SO-O data set	
6	Get list of all tables in Oracle?
7	I want query which will take less time to fetch the data as my database size is huge. Can you give me some advise?
8	What is the difference between FETCH/FOR to loop a CURSOR in PL/SQL?
9	I have a column in my oracle database which is a string (e.g., Item a,Item b,Item c,Item d)? I want to replace that string to Item c,Item b,Item d,Item d. How can I do it?
10	How to update column with null value?
5 Questions for TT data set	
11	How to import my w-2 information?
12	Why is my 1099-R early withdrawal being taxed more than 10% even though I 20% was already withheld?
13	How do I file an extension for my 2012 taxes?
14	When will I get my tax refund?
15	How much additional tax for converting from a Traditional IRA to a Roth?

check, we used Kendall's score [89] to compare these two experts' rank lists for each questions. The average Kendall's of these five questions in each data set distance between two experts is 0.693 for SO-C, 0.721 for SO-Oracle, and 0.781 for TT. Since for these two experts gave the different scores for these 10 users in each question. In order to maintain consistency of evaluation, we removed users from our evaluation whose score differs by more than one level. The final users used in the experiments shows in Table 5.7. After this, the average Kendall's score is improved to 0.773 for SO-C, 0.802 for SO-O, and 0.843 for TT. Thus, since some questions only has 7 users,

Table 5.7: Users Used in Accuracy Experiment

Data set	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
SO-C	9	10	7	8	8	9	9	10	9	7
SO-O	7	9	10	9	9	9	7	9	9	8
TT	10	10	10	9	9	10	10	10	10	9

Table 5.8: Five Level Expertise Rating for a Question

Level	Meaning	Description
5	Top Expert	Expert of concepts in this question and gave a perfect answer.
4	Professional Expert	Can answer this question well.
3	General Expert	Can give an answer for this question.
2	Learner	Knows some basic concepts for this question and gave a poor answer.
1	New Recruit	Just starting to learn and cannot give an answer.

only the average NDCG@1 to NDCG@7 are shown in our experiments. As we added the rating of these two raters, there were a total of 10 categories.

The metric used for comparison/evaluation is also important. In the information retrieval area, researchers used a number of measures to evaluate the rank list's accuracy; one of them was the DCG (Discounted Cumulative Gain) score [84]. Intuitively, the DCG score evaluation method penalized experts with a higher rank if they appear lower in the list. Hence, this evaluation metric matched well with our application requirement.[6] Since the DCG score was not between 0 and 1, we used the Normalized DCG (or NDCG) [84] to evaluate the ranked list. If NDCG@n was large, this algorithm's rank order matched well with the manual standard; If NDCG@n was small, this algorithm's rank order did not match well with the standard. In our ex-

[6] Kendall's Tau is a measure for the entire list where as NDCG can be calculated for various positions.

periment, we used the NDCG score to compare the results of these algorithms with two experts' evaluation results.

5.4.3 Evaluation of Approaches

In the literature, four methods [31, 32] was used for predicting the generalists in the CQAs (also See Chapter 4). In Section 5.3, we proposed five concept-based approaches - CR (Q/A), CR (PR), CR (HITS), CR (#A), CR (V) – for predicting specialists for a specific question. Our analysis compared these nine methods for accuracy to predict experts for these specific questions in Table 5.6.

- HITS: Jurczyk et al. [32] used the HITS authority score as the expertise score to identify users' expertise.
- PageRank (PR): Zhang et al. [31] used the PageRank score as the expertise score. In our experiments, the parameter d (the damping factor) of PageRank is set to 0.85.
- #Answers (#A): Zhang et al. [31] used the number of questions answered (or number of answers) as users' expertise score.
- Z_Score: Considering both the number of questions and answers, Zhang et al. [31] used Z_Score to identify users' expertise.
- Five approaches addressed in Section 5.3.2: CR (Q/A), CR (PR), CR (HITS), CR (#A) and CR (V).

5.4.4 Experimental Results

In this section, we first evaluate with differently weighted functions. Then, we compare the accuracy of different approaches to predict the expertise for various questions. In the end, we also do case study to analyze the experimental results.

5.4.4.1 Analysis of weighted formula

A number of concept-weighting experiments was described in the remainder of this chapter in which combination of concept frequency component, user-concept frequency component, and user-concept collection frequency component was used to analyze the accuracy to predict the expertise to a specific question (See Table 5.3). Since we considered both question and user factors to set the weight for a concept, we used six typical concept-weighting formulas to set different weights for a concept and reported the result of different weighted formula for the SO-C data set. Since these five approaches had similar regularities for these weighted formula (See Table 5.9, 5.10, 5.11, 5.12, 5.13), we mainly discuss the experimental results for the CR (V) approach. The experimental results indicated that (i) The weighted formulas *qb*.ub.ux and *qc*.ub.ux had similar values (0.852, NDCG@1 for qb.ub.ux and 0.853, NDCG@1 for qc.ub.ux). This seemed logical as the length of each question is very short (76% concepts appear only 1 or 2 times for each question in SO data set); hence, the importance of a concept in a question was not identified by the term frequency, (ii) as can be seen from Table 5.9, weighted formula qb.*ub*.ux (qb.*ub*.uf) did not improve the accuracy of formula qb.*uc*.ux (qb.*uc*.uf): from 0.811, NDCG@1 (0.873, NDCG@1) to 0.813, NDCG@1 (0.861, NDCG@1). Concept frequency of a user indicated that a user knew this concept better than the other concepts but did not mean that this user mastered this concept better than other users. Simply meant that this user did not necessarily an expert for this concept. (iii) Surprisingly, weight formula qb.ub.uf (0.873, NDCG@1) performed much better than qc.ub.ux (0.853, NDCG@1) and qb.ub.ui (0.813, NDCG@1). Although in traditional text mining, inverse document frequency (idf) was effective and useful to calculate similarity, in our experiment qb.ub.*ui* (qb.ub.*ux*), which was set a low weight for the broad concept,

Table 5.9: NDCG scores for concept-weighting formulas for CR (V) approach in SO-C data set

Weighted formulas	N@1	N@3	N@5	N@7
qb.ub.ux	0.852	0.883	0.902	0.919
qc.ub.ux	0.853	0.883	0.908	0.921
qb.uc.ux	0.853	0.881	0.906	0.915
qb.ub.uf	0.873	0.901	0.927	0.941
qb.ub.ui	0.811	0.845	0.864	0.889
qb.uc.uf	0.861	0.881	0.903	0.922
qb.uc.ui	0.813	0.843	0.861	0.886

[1] qb.ub.ux means qb, ub, and ux are used for the question-concept frequency component, user-concept frequency component, and user-concept collection frequency component.

Table 5.10: NDCG scores for concept-weighting formulas for CR (Q/A) approach in SO-C data set

Weighted formulas	N@1	N@3	N@5	N@7
qb.ub.ux	0.849	0.874	0.901	0.912
qc.ub.ux	0.845	0.873	0.887	0.905
qb.uc.ux	0.829	0.834	0.871	0.899
qb.ub.uf	0.852	0.883	0.903	0.931
qb.ub.ui	0.816	0.843	0.874	0.913
qb.uc.uf	0.832	0.863	0.893	0.911
qb.uc.ui	0.793	0.822	0.844	0.862

performed worse than qb.ub.*uf*, which was set a high weight for a broader concept. This was also explained for the broad concept where more users asked and answered the questions on these concepts wherein more statistic information was collected to rank these users. Therefore, in our experiment, we used uf (qb.ub.uf) as our weight formula to set the weight for each concept.

Table 5.11: NDCG scores for concept-weighting formulas for CR (PR) approach in SO-C data set

Weighted formulas	N@1	N@3	N@5	N@7
qb.ub.ux	0.846	0.863	0.899	0.901
qc.ub.ux	0.843	0.859	0.895	0.898
qb.uc.ux	0.826	0.843	0.885	0.891
qb.ub.uf	0.863	0.893	0.911	0.934
qb.ub.ui	0.797	0.829	0.852	0.869
qb.uc.uf	0.839	0.872	0.894	0.922
qb.uc.ui	0.796	0.827	0.851	0.864

Table 5.12: NDCG scores for concept-weighting formulas for CR (HITS) approach in SO-C data set

Weighted formulas	N@1	N@3	N@5	N@7
qb.ub.ux	0.840	0.862	0.894	0.914
qc.ub.ux	0.841	0.862	0.894	0.914
qb.uc.ux	0.831	0.852	0.874	0.904
qb.ub.uf	0.863	0.893	0.911	0.934
qb.ub.ui	0.781	0.803	0.824	0.842
qb.uc.uf	0.839	0.871	0.892	0.901
qb.uc.ui	0.781	0.803	0.824	0.842

Table 5.13: NDCG scores for concept-weighting formulas for CR (#A) approach in SO-C data set

Weighted formulas	N@1	N@3	N@5	N@7
qb.ub.ux	0.843	0.849	0.901	0.911
qc.ub.ux	0.837	0.841	0.891	0.917
qb.uc.ux	0.821	0.872	0.883	0.915
qb.ub.uf	0.862	0.894	0.918	0.933
qb.ub.ui	0.811	0.843	0.856	0.878
qb.uc.uf	0.832	0.854	0.991	0.913
qb.uc.ui	0.81	0.841	0.861	0.881

5.4.4.2 Accuracy analysis

Figure 5.5, 5.6 and 5.7 show the comparison of the nine approaches for all three data sets. Our experimental results clearly indicated: (i) in general, our approaches (CR (HITS), CR (PR), CR (V), CR (Q/A), CR (#A)) produced much better NDCG scores (for more n) compared to the other four global ranking methods (HITS, PR, Z_Score, Answers) proposed in the literature. The reason lied in the fact that the experts in this Q/A community (global experts) was not suitable to answer these special questions; (ii) #Answers and Z_Score methods had better NDCG curves than PageRank and HITS because the transitivity relationship in Q/A community was much weaker than web page graphs; (iii) PageRank and HITS had similar results because both of these two algorithms considered the transitivity property as we discussed in Section 5.3; (iv) Similarly, CR (#A) and CR (Q/A) had better NDCG curves than CR (PR) and CR (HITS), CR (V) and CR (PR) and CR (HITS) have much similar results; (v) The accuracy of these nine approaches in SO-C and SO-O data sets was much lower than the accuracy in the TT data set. Because these tax questions were mainly answered by these minority experts which were enrolled in the TT community (501,978 questions are answered by only 45,516 answerers), it was much easier to identify these experts from the TT data set. However, in the SO community, these questions were answered by various users (25,942 questions are answered by 17,085 users) so that it was much difficult to identify these experts from this data set, (vi) CR (Q/A) received the lowest accuracy compared with the other four concept rank approaches, namely CR (HITS), CR (PR), CR (V), CR (Q/A), CR (#A), since the Q/A_Score only classified each user as: questioner only, answerer only, or a combination thereof, but was not used to identify the user's expertise, and finally, (vii) as expected, CR (V) reached a highest accuracy in *all* three data sets because only this approach considered the answer quality.

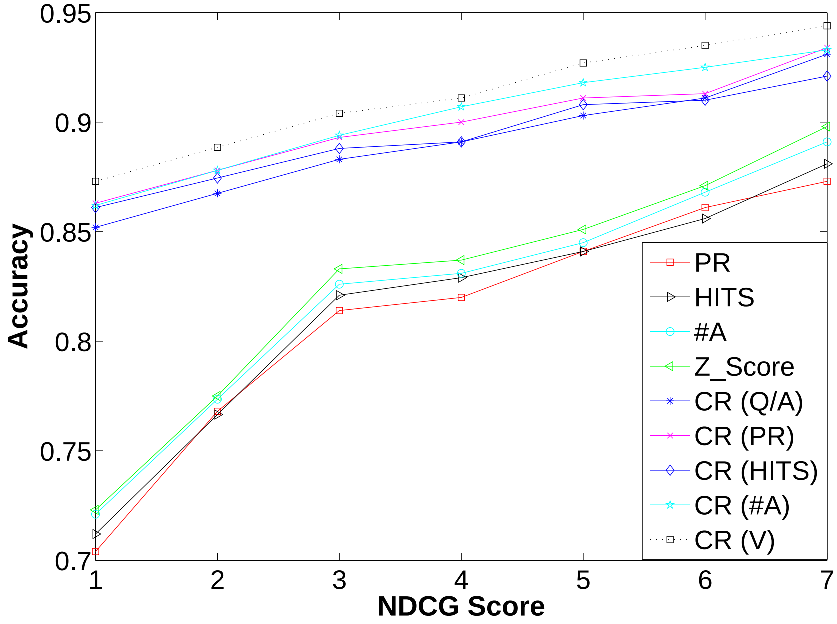


Figure 5.5: Average NDCG score for five questions for SO-C data set

Two Case Study: However, we did not conclude that for any question, a specialist will be *much* better than generalist. Figures 5.8 and 5.9 show the NDCG curve for question 1 and question 3 in SO-C data set (See Table 5.6). Our experimental results indicated that for question 1 in SO-C data set our approaches (CR (HITS), CR (PR), CR (V), CR (Q/A), CR (#A)) had a very similar NDCG curve as compared to the other four global ranking methods (HITS, PR, Z_Score, Answers). Since question 1 was a general question, the experts who were the generalists in this CQAs also gave a good answer for this question. Thus, these global ranking methods also gave good results. Meanwhile, since these general questions contained broad concepts (e.g., the broad concept “C”; 14,085 users used concept “C” around all 17,085 users) was

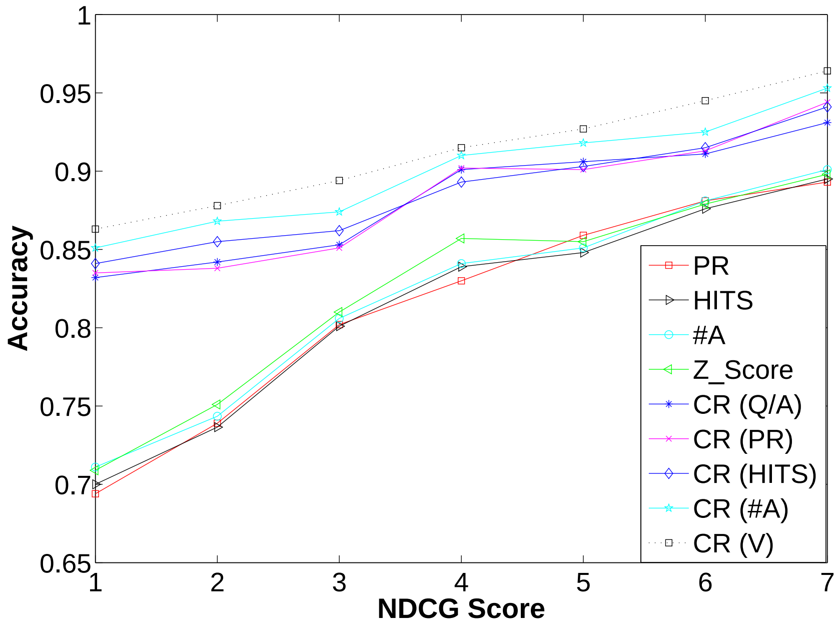


Figure 5.6: Average NDCG score for five questions for SO-O data set

contained in question 1, these concept rank approaches had very similar results to those found in the global approaches. However, question 3 was a specific question which discussed "memory leak" in C language. For this question, the experts, who was only familiar with "memory" problem, were able to give good answers. Therefore, our concept based approaches had much better NDCG curve as compared to the other four global ranking methods (See Figure 5.8). In conclusion, for these general questions (e.g., question 1) concept rank approaches had similar results as the global rank approaches, but for the specific questions (e.g., question 3) concept rank approaches performed much better than the global rank approaches. However, in the real CQAs there were a large number of specific questions. Therefore, our concept rank approach proved to be more much useful in helping the community find the proper users to answer these questions.

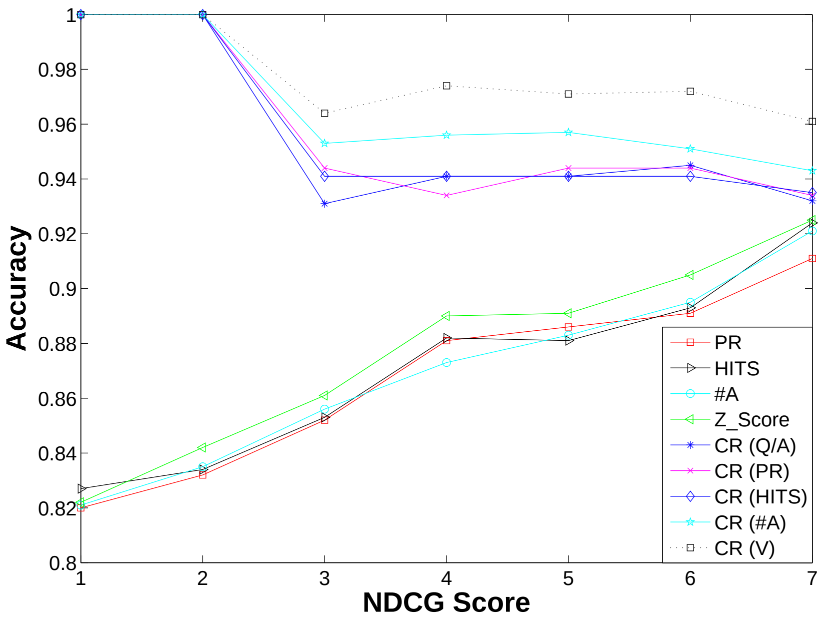


Figure 5.7: Average NDCG score for five questions for TT data set

5.5 Conclusion

The former research work mainly focused identifying the global experts (termed generalists) in CQA services, but for some more specific questions these global experts was not always able to respond as completely as needed and sometimes simply did not have an answer. Based on this observation, we proposed the Concept-Rank framework and several approaches to identify the specialist for a particular question by considering answer quality and graph structure. We demonstrated the effectiveness of our approach by comparing them with traditional global rank approaches and were pleased with the positive results.

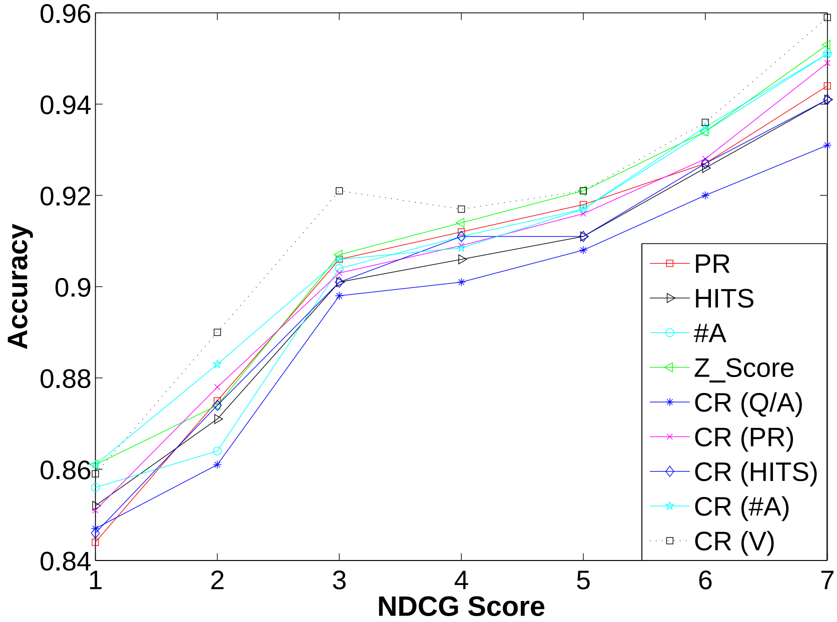


Figure 5.8: NDCG score for Question 1 in SO-C data set

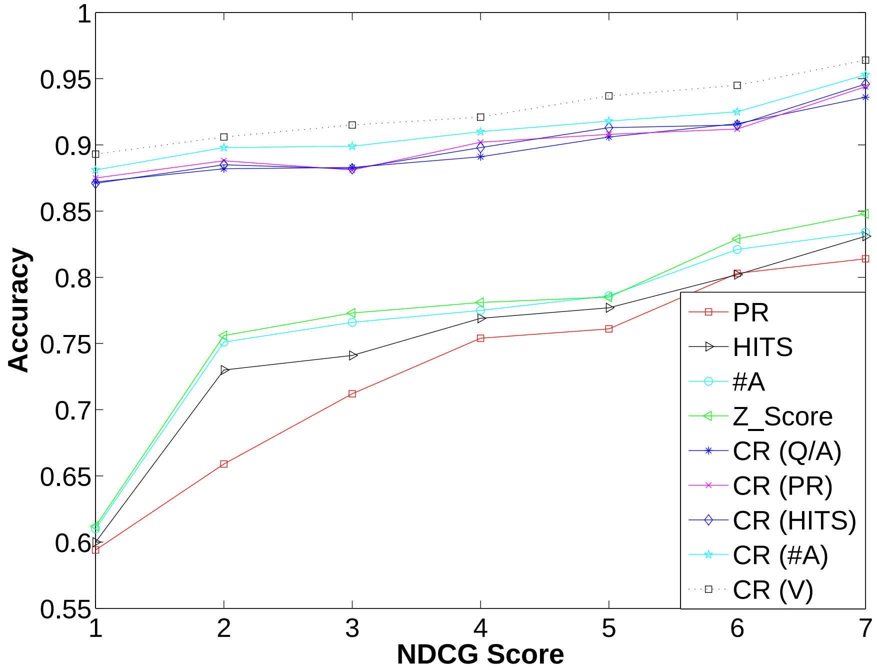


Figure 5.9: NDCG score for Question 3 in SO-C data set

CHAPTER 6

Non-negative Matrix Decomposition Problem

Ranking algorithms have been widely used for web and other networks to infer quality/popularity. Both PageRank and HITS were developed for ranking web pages from a web reference graph. Nevertheless, these algorithms have also been applied extensively for a variety of other applications such as question-answer services, author-paper graphs, and others where a graph can be deduced from the data set. The intuition behind HITS has been explained in terms of hubs and spokes as two values are inferred for each node. HITS has also been used extensively for ranking in other applications although it is not clear whether the same intuition carries over. It would be beneficial if we can understand these algorithms mathematically in a general manner so that the results can be interpreted and understood better for different applications. This chapter provides such as understanding for applying HITS algorithm to other applications.

In this chapter, we generalize the graph semantics in terms two underlying concepts: in-link probability (ILP) and out-link probability (OLP). Using these two, the rank scores of nodes in a graph are computed. We propose the standard non-negative matrix factorization (NMF) approach to calculate ILP and OLP vectors. We also establish a relationship between HITS vectors and ILP/OLP vectors which enables us to better understand HITS vectors associated with any graph in terms of these two probabilities. Finally, we illustrate the versatility of our approach using different graph types (representing different application areas) and validate the results. This work provides an alternative way of understanding HITS algorithm for a variety of applications.

6.1 Introduction

Empirical studies and theoretical modeling of networks have been the subject of a large body of research work in statistical mathematics and computer science [90, 91, 92]. Network ideas have been widely applied with success to the topics as diverse as the world wild web [58], scientific citation [62], email communication [59], community question answer services (CQAs) [80, 31], epidemiology [93], ecosystems [94, 95] and bioinformatics [96], to name but a few. Since a number of applications need to identify an order (or ranking) of nodes from graphs, several ranking algorithms have been proposed to bring order to these graphs. In 1999, Kleinberg [33] proposed the HITS algorithm to calculate the hub and authority score in a web reference graph. Around the same time, Page et al. [58] proposed the PageRank algorithm to identify web page authority in a web reference graph. Since these two algorithms are parameter-free and easy to compute, they have been widely used in numerous real-world applications. Meanwhile, these two algorithms have also been extended to different types of graphs to analyze the order of nodes, such as paper citation graph [62], email graph [59], bioinformatics graph [61], ask-answer graph [31] and so on. Although the intuition behind HITS has been explained in the context of web reference graphs, their intuition for other applications is not so clear. This chapter revisits the HITS approach to provide an alternative way to compute it and provide a graph property-based intuition.

As an example, consider a web writer, Steve, who is creating his own personal web page. Because he works at Oracle company, he wants to use a reference to Oracle company in his personal web page. Steve searches on key word "Oracle" and

Table 6.1: Possible URLs to Introduce the Oracle Company

Description	URL
Oracle Main Page	`www.oracle.com/`
Oracle Wikipedia	`http://en.wikipedia.org/wiki/Oracle`
Oracle On Twitter	`http://twitter.com/#!/oracle`
...	...

finds a lot of related web pages (See Table 6.1). Steve can use any of these web page to introduce his company. However, Steve chooses Oracle's main page for his personal web page because Steve believes that Oracle main page is a better web page to introduce Oracle company than others. In other words, in the web design process a user typically chooses the best web page to link to.

We observe two characteristics from the web reference graph. First, links in the reference graph describe the explanation relationship. A web writer creates a URL to a web page because this web writer wants to use this web page to explain an anchor text in his/her web page. In Example 1, web writer, Steve, finds a web page to introduce/explain Oracle company (anchor text). Second, a writer in these graphs always chooses a higher quality web page (in his/her opinion) to create a link. In Example 1, Steve chooses Oracle main page as the target web page because he believes that this web page is a better web page to introduce Oracle company than others. In summary, we have one important observation for these web reference graphs: *In the web reference graph writers/web page developers typically choose higher quality web page to explain anchor an text*[1].

For retrieval, users input a few key words and the search engine returns web pages ranked by their relevance. Thus, the global rank (ordered by web page's quality) can be defined as the probability of the web pages to be used to explain the input key

[1] We do not consider spam user's behavior because it is more of an exception.

words. Our observation is that in a web reference graph, the web page writer uses a reference (or link) in his/her page based on the quality of the referenced web page (in his/her opinion). Therefore, the global rank can also be deemed as the probability of this web page to be chosen as the quality page to explain anchor texts. In other words, this rank score can also be understood as the probability of this web page to receive a link from other web pages.

In general, for any node in a graph we define out-link probability (OLP) and in-link probability (ILP) to describe the rank of that node. The OLP (ILP) of node a describes the probability of node a to create an outgoing (incoming) link to other nodes, respectively. Both ILP and OLP represent different semantics of a graph. A node with high ILP (as determined by the number of in-edges) represents a node with high *quality or popularity* whether it is web pages or citations or friends etc. In contrast, a node with high OLP represents a node that *independently* indicates that the node has high *connectivity* which can be interpreted as a hub for a web reference graph, as a paper with large number of citations, or a person with large number of friends. It is also possible to interpret, in general, high in-degree (ILP) as indicating depth while high out-degree (OLP) as indicating breadth. Of course, it is also possible for a node to be both. Hence, we need to calculate both of these values for each node. We use two vectors u and v, respectively, to represent OLP and ILP values of each node; Thus, uv^T describes the probability of creating a link between any two nodes. Let e be the number of edges. Since we can easily calculate the probability of creating a link between any two nodes as $\frac{1}{e}L$ (where L is the adjacency matrix), we obtain the vector u and v^T by solving a cost function $min||\frac{1}{e}L - uv^T||_{l_2}^2$. Since we calculate ILP vector v^T and OLP vector u by $d$ecomposing the adjacency matrix L, we term this approach as *ILOD* (pronounced *Illiad*) approach.

We also establish a relationship between HITS and ILOD. The hub vector of HITS has the same rank order of OLP vector u and the authority vector of HITS has the same rank order of ILP vector v^T. In Kleinberg's description [33], we only know the calculation of HITS but do not know the meaning of HITS scores for each node. Since we establish a relationship between HITS and ILOD, we can use OLP and ILP score to explain HITS vectors. We also stress that this understanding is very important because we can clearly know the meaning of these scores when we extend HITS approach to different types of social graphs.

Furthermore, we can apply the ILOD approach to diverse graph types that represent different applications. In the first application (directed graph), we apply this approach to identify experts in Community Question Answering services (CQAs). Here, an edge is drawn from the user who asked a question to the user who answered it. The user's expertise in the CQAs can also be described as the probability of this user answering many questions so that we use ILP score as the user's expertise score (See Section 6.5). However, OLP does not represent anything in this graph (as it only means that the individual has asked a large number of questions that gets translated to incoming edges when those questions are answered by someone)[2]. In the second application (bipartite graph), we apply this approach to identify experts (based on the number of accepted papers in a conference) in an author-paper graph. The user's expertise score is described as the probability of this user's paper to be accepted in an ICDM conference (this author-paper graph) and hence we use OLP score as the user's expertise score (See Section 6.5) in this application. Again, ILP does not have any significance here. These two applications clearly demonstrate the relevance of

[2] Note that this formulation does not account for the *quality* of answers but only the *number* of answers! However, it is possible to include quality information if weights representing quality are assigned to edges.

ILP and OLP based on application semantics and which one needs to be used and why!

Contributions:

- This chapter analyzes the ranking problem in a graph. Rather than random traversal (or hubs and spokes), we provide an alternative intuition as to why links in graphs represent qualitative information.
- We propose the concept of ILP and OLP in a graph and use the non-negative matrix factorization (NMF) to calculate ILP and OLP vectors for a graph using its connectivity information. We prove that hub and authority vectors of HITS have the same rank order, respectively, as ILP and OLP vectors. We also argue that HITS vector is the rank-1 approximation of adjacency matrix L.
- We demonstrate how the concept of ILP and OLP can be applied to diverse real-world applications (graphs of different characteristics). The experimental results validate the relationship between HITS and ILOD.

Road Map: Section 6.2 defines the graph models and Section 6.2.1 describes problem statement. Section 6.3 defines the probability model on the graph. Section 6.4 represents our contributions along with the ILOD algorithm and the relationship between ILOD and HITS. Section 6.5 shows experimental results for different applications and their analysis. Section 6.6 has conclusions.

6.2 Graph Model and Problem Statement

This research focuses on two kinds of graphs: directed and bipartite. The edges of a graph can also have weights to reflect preferences, quality etc. For the current discussion, we will not be considering them.

Directed Graph: In many applications, objects and relationships are modeled as a directed graph $G = (V, E)$ where each vertex in V represents an object in a particular

Table 6.2: Adjacency Matrix L for Fig. 6.1

	a	b	c	d	e	f	g
a	0	1	1	1	1	0	1
b	1	0	0	0	0	0	0
c	1	1	0	0	0	0	0
d	0	1	1	0	1	0	0
e	1	0	1	1	0	1	0
f	1	0	0	0	1	0	0
g	0	0	0	0	1	0	0

domain and an edge in E describes the relationship between objects. For example, Figure 6.1 shows a directed graph that describes the web reference graph extracted from Stanford web site (`http://www.stanford.edu/`). Each node is a web page and a directed edge is drawn from web page a to b if page b's URL is used by page a. In addition, we use the adjacency matrix L to store the graph connectivity. The adjacency matrix L of Figure 6.1 is shown in Table 6.2.

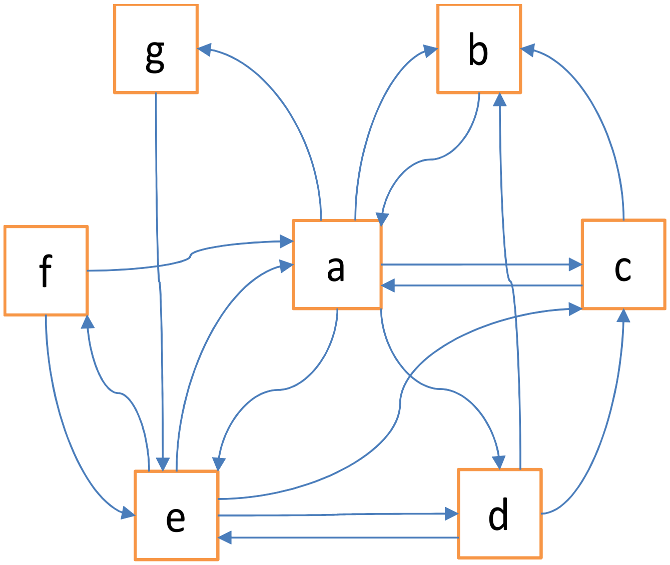


Figure 6.1: A Sample Web Reference Graph

Table 6.3: Adjacency Matrix L for Fig. 6.2

	p_1	p_2	p_3	p_4
u_1	1	0	1	1
u_2	0	1	0	0
u_3	0	1	1	0
u_4	0	1	0	0
u_5	0	0	1	0

Bipartite Graph: In some applications, objects and relationships are modeled as a bipartite graph $G = (U \cup V, E)$ where nodes can be divided into two disjoint groups U and V such that no edge connects the vertices in the same group and an edge in E describes the relationship between objects in different groups. For example, Figure 6.2 is a bipartite graph extracted from the ICDM conferences papers (`http://www.informatik.uni-trier.de/~ley/db/conf/icdm/`). U is a set of authors and V is a set of ICDM papers. An edge is drawn from author u_1 to paper p_1 if author u_1 publishes a paper p_1 in ICDM. We use the biadjacency matrix L to store the graph structure. The biadjacency matrix L of Figure 6.2 is shown in Table 6.3.

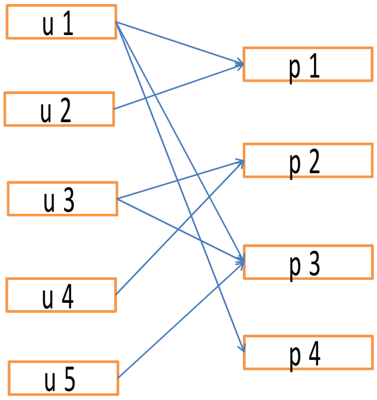


Figure 6.2: An Author-Paper Bipartite Graph

6.2.1 Problem Statement

The global rank of a web page can be deemed as the probability of this web page to be chosen as the quality web page to explain anchor text. In other words, the global rank of a reference graph can also be considered as the probability of this web page to have a link from other web pages. Moreover, in the author-paper graph, researchers are interested in identifying the expert among all authors. User's expertise score can also be deemed as the probability of this user's paper to be accepted in ICDM conference (is selected as the author node). In other words, the user's expertise rank can also be considered as the probability of this user to create a link to the papers. Therefore, these two probabilities (out-link denoted as OLP and in-link denoted as ILP) can be used to describe the semantics of the graphs of diverse real world-applications.

Given a graph G with nodes and edges, our research focuses on computing the in-link and out-link probabilities of each node that represents the connectivity of that graph. This allows one to understand the characteristics of the graph as well as choose appropriate algorithms for analyzing the graph.

6.3 Graph Characteristics

In this section, we first define three types of probability that cab be associated with a graph and then discuss the relationship among them.

6.3.1 Types of Edges

Consider that we have a bag[3] which contains all the edges of graph G (See Figure 6.3). We have three types of edges in this bag for this sample space: (i) *Out-*

[3] You need a bag for representing multiples edges between nodes. They can be differentiated, for example, by assigning numbers to each edge.

Table 6.4: The link probability matrix *LP* for Figure 6.1

	a	b	c	d	e	f	g
a	0	0.06	0.06	0.06	0.06	0	0.06
b	0.06	0	0	0	0	0	0
c	0.06	0.06	0	0	0	0	0
d	0	0.06	0.06	0	0.06	0	0
e	0.06	0	0.06	0.06	0	0.06	0
f	0.06	0	0	0	0.06	0	0
g	0	0	0	0	0.06	0	0

Link(a) Edges: is an edge in which a is the start node, (ii) *In-Link(b)* Edges: is an edge in which node b is the end node, and (iii) *Link(a,b)* edge: is an edge $< a, b >$ which is from node a to node b. Each type is a bag in the most general sense.

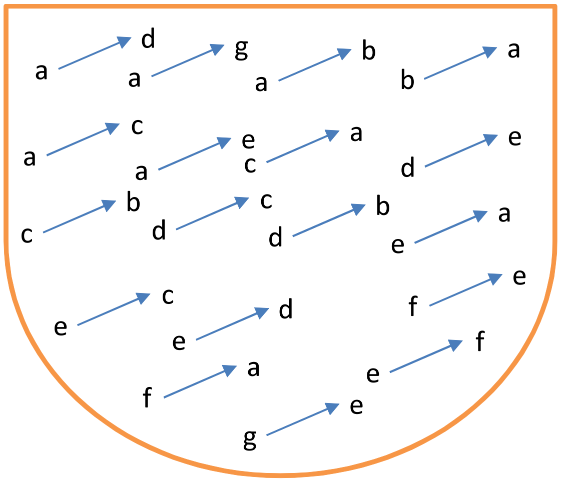


Figure 6.3: A Sample Space of Figure 6.1 (18 edges)

We can now define probabilities associated with edges as follows.

Definition P(Out-Link(a)), also called as the Out-Link Probability (OLP) of node a, is the probability of node a to be connected to other nodes with an outgoing edge.

Definition P(In-Link(a)), also called In-Link Probability (ILP) of node a, is the probability of node a to be connected from other nodes with an incoming edge.

Definition P(Link(a,b)) is the probability of edge $< a, b >$ is the probability of an edge from node a to node b. It is computed as the ratio of the number of edges that start with a and end with b to the total number of edges.

All of the above probabilities can be easily obtained from graph G. For example, let us select Link(a,b) from that bag. Let e be the total number of edges in graph G. Therefore, if there is *one* edge in Link(a,b), $P(Link(a, b)) = \frac{1}{e}$; if there is no link in Link(a,b), $P(Link(a, b)) = \frac{0}{e} = 0$. Thus, we have the following equation to describe this probability:

$$P(Link(a, b)) = \left\{ \begin{array}{cc} 0, & \text{(no link between node } a \text{ and node } b\text{)} \\ \dfrac{1}{e}, & \text{(1 edge in the bag Link(a,b)} \end{array} \right\} \tag{6.1}$$

Table 6.4 shows the probability of links between two nodes for Figure 6.1. We term this matrix as Link Probability matrix LP.

6.3.2 Relationship among Three probabilities

Out-Link(a), *In-Link(a)*, and *Link(a,b)* are bags. It is also evident that if and only if both *Out-Link(a)* and *In-Link(b)* is non-empty, then *Link(a,b)* may be non-empty. Therefore, we have the following:

$$Link(a, b) = Out - Link(a) \cap In - Link(b) \quad and \quad P(Link(a, b)) = \frac{|Link(a, b)|}{e} \tag{6.2}$$

The relationship between *Out-Link(a)* and *In-Link(b)* from node a (or the edge *Link(a,b)* has an associated semantics. The presence of an edge (i.e., *Link(a,b))* indicates an an explicit association between the two nodes either as an answer (in an ask-answer graph of example (i) a road relationship (in a road map graph of example,

(ii) an authorship (in the bipartite graph of example, (iii) etc. This explicitly indicates the reasoning for the presence of this link from a specific node a to a specific node b.

We ignore the sematic meaning of these links[4] as is done in the literature. However, these two events, *Out-Link(a)* and *In-Link(b)*, are not independent. The relationship between the two can be captured by vectors whose (matrix) multiplication results in the results in the link probability.

6.4 ILOD Approach

6.4.1 Motivation

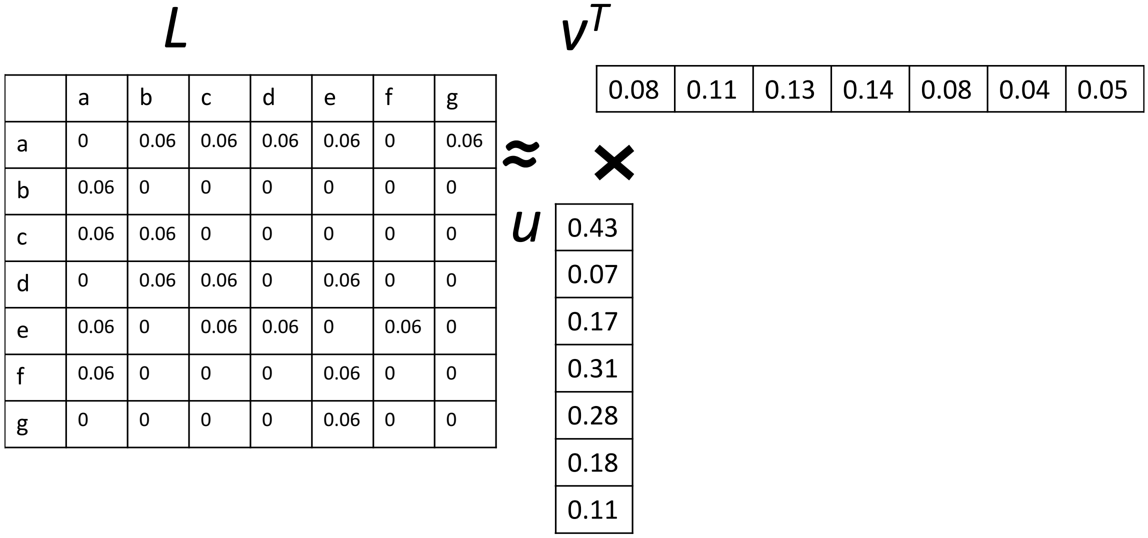
Figure 6.4 highlights the motivation to calculate *P(Out-Link(a))* and *P(In-Link(b))*. Vector u (also called OLP vector) and v^T (also called ILP vector) are two vectors to store *P(Out-Link(*))* and *P(In-Link(*))* score for each node. We expect that uv^T can match well with link probability matrix LP ($LP = \frac{1}{e}L$, see Figure 6.4). We also assume that the noise data in a graph follows the normal distribution [63], the non-negative vectors u and v^T can be obtained by solving the following optimization problem.

OLP and ILP problem: *Given a link probability matrix LP ($\frac{1}{e}L$), find non-negative vectors u and v^T to minimize the function*

$$\frac{1}{2}||LP - uv^T||_{l_2}^2 \tag{6.3}$$

which is the typically used mean square error. Other metrics have also been used for this purpose[63]. The product uv^T is called a non-negative vector factorization of $\frac{1}{e}L$, although the product uv^T is not necessarily equal to the link probability matrix

[4] Recalled that these link-based algorithms, such as PageRank and HITS ignore the sematic meaning of these links.



	a	b	c	d	e	f	g
a	0	0.06	0.06	0.06	0.06	0	0.06
b	0.06	0	0	0	0	0	0
c	0.06	0.06	0	0	0	0	0
d	0	0.06	0.06	0	0.06	0	0
e	0.06	0	0.06	0.06	0	0.06	0
f	0.06	0	0	0	0.06	0	0
g	0	0	0	0	0.06	0	0

u
0.43
0.07
0.17
0.31
0.28
0.18
0.11

0.08	0.11	0.13	0.14	0.08	0.04	0.05

Figure 6.4: Calculation of vector u and v^T

LP (equal to $\frac{1}{e}L$). Clearly, the product uv^T is an approximate factorization of rank at one.

Figure 6.4 also shows the results of matrix decomposition of Figure 6.1. In this example, since node a has the highest number of out links (equal to 5), web page a has a highest probability to create a link to the other nodes ($u(a) = 0.43$); since node b has the lowest number of in-links (equal to 1), web page b has the lowest probability to create a link to the other nodes ($u(b) = 0.07$). Similarly, ILP vector v^T can be explained in the same way.

6.4.2 *I*LP and *O*LP *D*ecomposition for Adjacency Matrix L (ILOD)

The multiplicative update rules for u and v^T in the gradient descent mechanism of Lee and Seung [63] change when the cost function 6.3 is minimized. u^k gives a vector of OLP for all nodes on the k^{th} iteration and u_i^k describes the i^{th} element of vector u^k. Similarly, $(v^T)^k$ gives a vector of ILP for all nodes on the k^{th} iteration and $(v^T)_i^k$ describes the i^{th} element of vector $(v^T)^k$. We successively compute u^{k+1} and $(v^T)^{k+1}$

based on $u^{(k)}$ and $(v^T)^k$. Notice that Lee et al. [97, 63] indicate that Nonnegative Matrix Factorization does not have a global solution (a unique solution). Since solving the standard NMF objective function $(min||LP - v^T u||^2_{l_2}$, s.t. $LP, v^T, u \geq 0)$ is NP-hard problem, Lee et al. [97, 63] use the gradient descent approach to approximately calculate the matrix v^T and u. Therefore, with a differently initialized v^T and u, algorithm may converge to a different solution. Usually, researchers typically initialize the matrix with random values, run the algorithm several times and choose the highest accuracy result. However, OLP and ILP decomposition problem, being a rank-1 NMF problem, has a global minimizer (a unique solution, See Lemma 1). Thus, u and v^T will converge to the same vectors with any initialized value (excluding $u^0 = \mathbf{0}$ and $(v^T)^0 = \mathbf{0}$). We start with u^0 and $(v^T)^0$ where u^0 and $(v^T)^0$ are two vectors which contain all ones:

$$u^0 = \mathbf{1} \quad \textit{and} \quad (v^T)^0 = \mathbf{1} \tag{6.4}$$

To compute u^{k+1} and $(v^T)^{k+1}$ from u^k, $(v^T)^k$, we, respectively, use the following equations:

$$u_i^{k+1} = u_i^k \frac{(LPv^k)_i}{(u^k(v^T)^k v^k)_i} \quad \textit{and} \quad (v^T)_i^{k+1} = (v^T)_i^k \frac{((u^T)^k LP)_i}{((u^T)^k u^k (v^T)^k)_i} \tag{6.5}$$

Algorithm 1 outlines the process to calculate the OLP and ILP vectors. It takes in 1 argument link probability matrix LP. In line 1-2, algorithm first initializes variables and sets u and v^T vectors as unit vectors. Line 3 is used to stop this iterative algorithm. Although the convergence of iterative non-negative matrix decomposition can be guaranteed in theory (See [63]), practical computation uses a maximum number of iterations (say, K). In all of our experiments we have seen rapid conver-

Algorithm 2 *ILOD*

```
Require:
    Link Probability Matrix $LP$;
Ensure:
    ILP vector $u$; OLP vector $v^T$;
$u = \mathbf{1}$;
$v^T = \mathbf{1}$;
For i = 1: Max-Iteration $K$
$u = u.*(LPv)./(uv^Tv + 10^{-7})$; % To avoid dividing by 0, we add $10^{-7}$ (very small value).
$v^T = v^T.*(u^TLP)./(u^Tuv^T + 10^{-7})$;
End For
return $u$ and $v^T$;
```

gence, which relative rank score stabilizing in 200 iterations. Hence, we have fixed the number of iterations (K) to 200.

We also analyze the time and space requirements needed for this approach. Because the graph extracted from the real applications is very sparse, we only store the edges; therefore, the space required is only $O(e)$ where e is the number of edges in this graph. Let n be the number of nodes in this graph. The time complexity of ILOD is $O(Kn^2)$ because in each iteration this algorithm calculates $LP \times v$ or $u^T \times LP$ which needs $O(n^2)$. In fact, ILOD algorithm has the same time and space complexity as the HITS algorithm.

6.4.3 Relationship between HITS and ILOD

In this section, we provide some theoretical analysis of HITS and ILOD.

Lemma 6.4.1 *LP being the link probability matrix of graph G, the pair u and v^T are local minimizers of cost equation $\frac{1}{2}||LP - uv^T||_{l_2}^2$ if and only if u and v^T are the principle eigenvectors of LL^T and L^TL respectively.*

Proof Since $LP = \frac{1}{e}L$, LP and L have the same non-negative matrix decomposition vector u and v^T. We just need to prove the pair of vectors u and v^T are local minimizer of cost equation $\frac{1}{2}||L-uv^T||_{l_2}^2$ if and only if u and v^T are the non-negative eigenvectors of LL^T and L^TL.

(1) The necessary condition.

Without loss the generality, the vectors u and v^T are respectively partitioned as $(u \ \ 0)$ and $(0 \ \ v^T)$ and the adjacency matrix L is partitioned as follows:

$$L = \begin{pmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{pmatrix} \tag{6.6}$$

To keep gradient descent, Lee et al. [63] indicate that:

$$uv^Tv - Lv \geq 0, vuu^T - L^Tu \geq 0 \tag{6.7}$$

Then, we have

$$\begin{pmatrix} uv^T & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} v^T \\ 0 \end{pmatrix} - \begin{pmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{pmatrix} \begin{pmatrix} v^T \\ 0 \end{pmatrix} \geq 0 \tag{6.8}$$

and

$$\begin{pmatrix} vu^T & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} u \\ 0 \end{pmatrix} - \begin{pmatrix} L_{11}^T & L_{21}^T \\ L_{12}^T & L_{22}^T \end{pmatrix} \begin{pmatrix} u \\ 0 \end{pmatrix} \geq 0 \tag{6.9}$$

Then, we have $L_{21}v^T \leq 0$, $L_{12}^T \geq 0$, $u(||v||^2u - L_{11}v) = 0$ and $v^T(||v||^2u - L_{11}u) = 0$.

Since L_{21}, L_{12}, u, $v^T \geq 0$, we have:

$$||u||^2||v||^2u = L_{11}L_{11}^Tu \tag{6.10}$$

and

$$||u||^2||v^T||^2v^T = L_{11}^TL_{11}v \tag{6.11}$$

Thus, u and v^T are, respectively, the eigenvector of LL^T and L^TL.

(2) The sufficient condition.

Let $R_+^{m\times n}$ be the set of $m \times n$ non-negative matrices and $R^{m\times n}$ be the set of $m \times n$ real matrices. $L \in R_+^{m\times n}(m \geq n)$ has a singular value decomposition:

$$L = U\Sigma V^T \tag{6.12}$$

where $U \in R^{m\times m}$ and $V \in R^{n\times n}$ are orthogonal matrices and $\Sigma \in R^{m\times n}$ is an rectangular diagonal matrix with λ_1, λ_2, ..., λ_n on the diagonal where $\lambda_1 \geq \lambda_2 \geq$ ... $\geq \lambda_n \geq 0$ are the singular values of L. u and v^T are the non-negative eigenvector of LL^T and L^TL.

Then, for matrix rank $r = 1$, the matrix $L_1 = \lambda_1 uv^T$ is a global minimizer of the problem.

$$min_{uv^T\in R^{n\times m}}\frac{1}{2}||L - uv^T||_{l_2}^2 \tag{6.13}$$

and its error is $\frac{1}{2}||L - uv^T||_{l_2}^2 = \frac{1}{2}\lambda_1^2$.

Based on the above, we establish the relationship between ILOD and HITS algorithm.

Theorem 6.4.2 *Given a probability matrix LP, the OLP vector u obtained by the NMF algorithm corresponds to the hub vector of the HITS algorithm and the ILP* *vector v^T obtained by the NMF algorithm corresponds to the authority vector of the HITS algorithm.*

Proof Kleinberg [33] has indicated that hub rank vector of HITS algorithm is the principle eigenvector of LL^T and the authority rank vector of HITS algorithm is the principle eigenvector of L^TL.

According to Lemma 6.4.1, OLP vector u is the hub vector of HITS algorithm and ILP vector v^T is the authority vector of HITS algorithm.

6.5 Experimental Validation

In this section, we apply NMF approach to a number of graph types corresponding to two different applications (corresponding to two types of graphs), illustrating and validating the relationship between HITS and ILOD.

6.5.1 Directed Graphs

In this application, we first focus on identifying experts from Community Question Answering services (or CQAs). Identifying expertise from a CQA data set is useful in many ways: (i) allows one to intrinsically rank (or group) users in the community, (ii) this can be beneficially used for identifying good answers, and (iii) the CQA service can keep them by providing incentives and route questions to these experts for delivering better answers. Other approaches have been used for this purpose. For example, in order to calculate user's expertise score, Zhang et al. [31] and Jurczyk et al. [32] extracts the ask-answer graph from these CQAs. Nodes represent users in the Q/A community and a directed edge is drawn from user u_1 to user u_2 if user u_2 answers one or more questions asked by u_1. Table 4.1 shows a few questions and some of their answers from the Stack Overflow service for the “C” language. Figure 6.5 shows the ask-answer graph for Table 6.5 using the ask-answer paradigm.

Table 6.5: A Sample Ask-Answer Graph

User	Votes	Content
Questioner A		In C arrays why is this true? a[5] == 5[a]
Answerer B	330	Because a[5] will evaluate to: *(a + 5) and 5[a] will evaluate to: *(5 + a)
Answerer D	-6	You can search the result on the Google.
Questioner B		What is the best tool for creating an Excel Spreadsheet with C#?
Answerer C	144	You can use a library called Excel Library. It's a free, open source library posted on Google Code.
Questioner E		How can Inheritance be modeled using C?
Answerer A	0	See also: http://stackoverflow.com/questions/351733/can-you-write-object-oriented-code-in-c

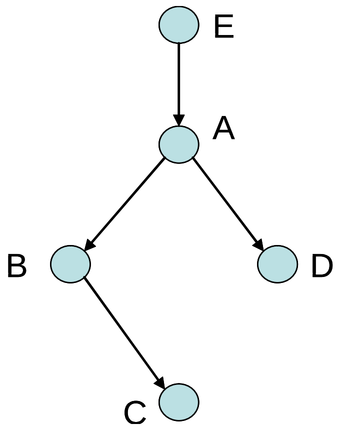


Figure 6.5: A Sample Ask-Answer Graph

The user's expertise score can be described as the probability of this user to answer a question in this CQAs since these CQAs requires us to identify the appropriate user to answer posed questions. Since a directed edge links an asker (a user who asks a question) with an answerer (a user who answers a question), the ILP of user u_1 describes a probability of this user u_1 to answer the other users' questions. Therefore, we use ILP score as the user's expertise score for CQAs. In our experiment, we use Stack Overflow (SO) data set to identify user's expertise score.

Table 6.6: Complete Data set Characteristic

Data set	#Ques	#Answers	#Answerers	#Questioners
SO-O	8,644	21,879	4,279	5,722

Stack Overflow (SO) data set (`http://stackoverflow.com/`): This CQAs focuses on computer programming topics. SO allows a user to modify other users answers. In other words, when an answerer wants to answer a question, s/he has two choices: modify an existing answer or provide a new answer. As a result, the average number of answers for each question is only 2.36. In our experiments, we only consider the first user who posts the answer as the answerer, because, in most cases, the first user is likely to provide a significant contribution than other users. Each question in this community is marked with a topic tag (e.g., “Oracle”). We use questions which are marked as “Oracle” as SO-O data set and broader statistical characteristics of this data sets are shown in Table 6.6.

We compare the HITS (Auth) and ILOD (I) for this ask-answer graph to identify the user’s expertise score and Table 6.7 shows the top 10 users and their ranks score for ILOD and HITS approaches. In Table 6.7, if we normalize HITS (Authority) vector and ILOD (ILP) vector, these two vectors match completely. Thus, HITS (Authority) score of a user in this ask-answer graph can be explained as the probability of this user to answer questions.

6.5.2 Undirected Graph

In this application, We focus on identifying the hub city from a road map graph (the hub city is a transportation centrality of this area) because this approach can be applied to find a good logistics hub location in a map. Figure 6.6 shows a road map in Texas (24 cities and 8 main roads and 6 auxiliary roads). We extract a city-road

Table 6.7: Top 10 Experts in SO-O

ILOD (I)	Score	HITS (Auth)	Score
Community	0.017	Community	0.82
RenderIn	0.004	RenderIn	0.19
Steven	0.0028	Steven	0.14
Jason Baker	0.0022	Jason Baker	0.11
Mark Harrison	0.0020	Mark Harrison	0.10
rima	0.0019	rima	0.09
Frustrated	0.0018	Frustrated	0.09
Peter Lang	0.0017	Peter Lang	0.08
Omnipresent	0.0015	Omnipresent	0.07
Tom	0.0014	Tom	0.07

graph from this road map. The node in this graph is a city and a undirected edge is drawn from city c_1 to c_2 if there is a road between city c_1 and city c_2 . Since this city-road graph is undirected graph, the probability matrix LP is a symmetrical matrix; therefore, ILP vector u and OLP vector v should have the same rank order. In this road map, the link probability of a city is explained as the probability of a traveler to reach to this city from any city in this map so that this vector can be used to identify the hub city for this road map.

We compare the HITS (hub) and ILOD (O) for this undirected graph to identify the hub city and Table 6.8 shows the top 10 cities and their ranks score for ILOD and HITS approaches. In Table 6.8, if we normalize HITS (Authority) vector and ILOD (OLP) vector, these two vectors match completely. Thus, HITS (hub) score of a city in this road graph can be explained as the probability of this city to have a road to the other city.

6.5.3 Bipartite Graphs

In this application, we focus on identifying experts from an author-paper graph. This application is similar to some retrieval systems, such as Arnetminer system

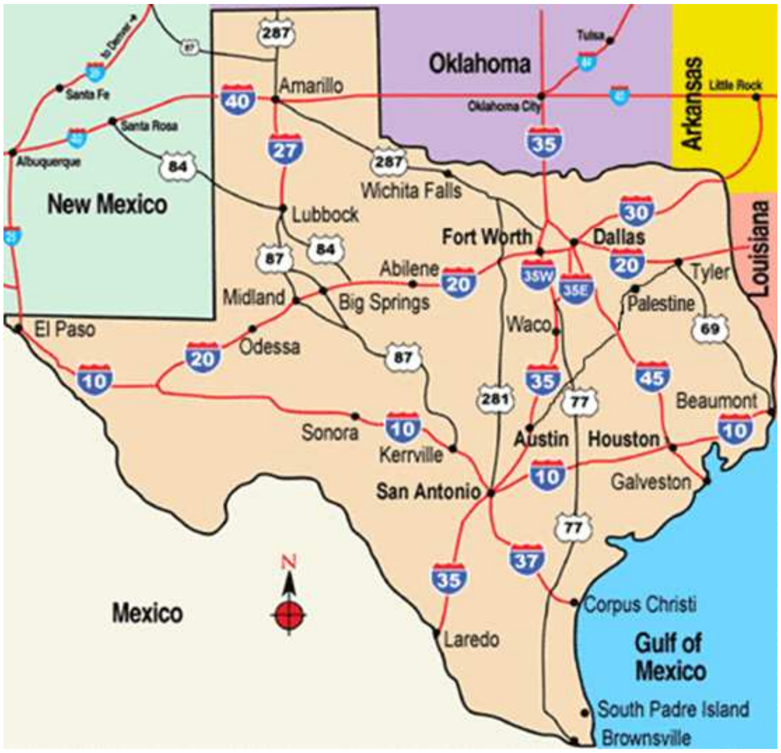


Figure 6.6: Road Map in Texas

(`http://arnetminer.org`) and LinkedIn system (`http://www.linkedin.com/`). Our ranking model is easy to extend to the bipartite graph.

We use the ICDM data set to represent this type of application. The ICDM data set (`http://www.informatik.uni-trier.de/~ley/db/ICDM`) contains all the regular and short papers that appeared in the ICDM conference from 2001 to 2011. We extract author-paper graph from these papers which includes 806 papers, 1820 authors and 2625 relationships between papers and authors. The short and regular papers are set as the same weight in this author-paper graph. The detail information shows in Table 6.9.

For Figure 6.2, The u vector describes the probability of each user to write a paper and the v^T vector describes the probability of this paper to be written by a user. The probability of a user to write a paper can be used to measure the user's

Table 6.8: Top 10 Hub City in Texas

ILOD (O)	Score	HITS (hub)	Score
Dallas	0.068	Dallas	0.075
Big Spring	0.058	Big Spring	0.064
Fort Worth	0.058	Fort Worth	0.064
San Antonio	0.055	San Antonio	0.061
Kerrville	0.054	Kerrville	0.060
Palestine	0.054	Palestine	0.060
Midland	0.053	Midland	0.059
Lubbock	0.053	Lubbock	0.058
Waco	0.048	Waco	0.053
Abilene	0.047	Abilene	0.052

Table 6.9: Complete ICDM Data set Characteristic

Data set	#Authors	#Paper	#Relationship
ICDM	806	1,820	2,625

expertise score because a good expert always publishes research papers in the good conference. In other words, we can intuitively understand that an expert will have authored more papers in the author-paper graph. hence, we use the OLP vector u to identify an author as an expert in this graph. Similarly, HITS (Hub) can be used to identify an author as an expert in this graph.

We compare HITS (hub) and OLP vector in our experiments. Table 6.10 shows, respectively, the top 10 experts and ranks score for ILOD and HITS approaches. In Table 6.10, if we normalize HITS (Hub) vector and ILOD (OLP) vector, these two vectors are the same. Thus, HITS score of author-paper graph can be explained as the probability of an author getting his paper accepted in this conference so that this score can be used as the expertise score.

Table 6.10: Top 10 Experts from the ICDM Conference (2001 to 2011)

ILOD (O)	Score	HITS (Hub)	Score
Zheng Chen	0.025	Zheng Chen	0.57
Jun Yan	0.021	Jun Yan	0.47
Lei Ji	0.016	Lei Ji	0.36
Shuicheng Yan	0.015	Shuicheng Yan	0.35
Junshi Huang	0.014	Junshi Huang	0.30
Ning Liu	0.010	Ning Liu	0.23
Ying Chen	0.004	Ying Chen	0.10
Zheng Chen	0.003	Zheng Chen	0.06
Siyu Gu	0.002	Siyu Gu	0.05
Qiang Yang	0.002	Qiang Yang	0.05

6.6 Conclusions

In this chapter, we have discussed the ranking problem in web and social networks. We have proposed ILP and OLP of a graph to help understand HITS approach in contexts other than web graphs. We have established that the two probabilities identified in this chapter correspond to the hub and authority vectors of the HITS approach. Then, we have used the standard non-negative matrix decomposition approach to calculate these two probabilities for each node. Then, we have proved the relationship between HITS vectors and ILP/OLP vectors. In addition, we have applied ILOD to different types of graphs representing applications other than the web graph. Experimental results validate the relationship between ILOD approach and HITS algorithm. This chapter provides an alternative intuition for the use of HITS (or NMF) approach which explains the relevance of either the hub vector or the authority vector or both.

CHAPTER 7

Conclusions

The principal contribution of this work is to provide a new insight to the paradigms of answer quality and expertise identification in CQAs. In particular, we made the following novel contributions.

Predict answer quality in CQAs: Based on the analysis of Q/A data sets and the dynamic nature of CQA services, we, for the first time, identified the temporal role of the participant and its impact on answer quality. Based on that observation, we identified a set of temporal features for predicting answer quality in CQA services. For these services, user characteristics can be better captured with temporal features than the traditional ones proposed in the literature (both textual and non-textual). Further, we demonstrated the effectiveness and superiority of temporal features by comparing our features with the features and the classification approach used in the literature on multiple diverse data sets. Meanwhile, we also argued for the use of learning to rank approaches as a more appropriate model for predicting accuracy of answer quality as it pertains to CQA services. Using the RankSVM learning to rank approach, we performed extensive experimental analysis on diverse data sets to demonstrate that the proposed features work well for predicting the *best* answer as well as *non-best* answers.

ExpertRank framework to rank a user using both domain information and graph structure: Based on the differences between the ask-answer graphs extracted from a CQAs and the conventional web graphs, we chose domain information useful for mining expertise rank from Q/A data sets. We proposed the ExpertRank framework and several approaches to predict the global expertise level (or a generalist) of users by considering both answer quality and graph structure. We argued and demonstrated why structure information alone was not sufficient (typically used in PageRank and other algorithms) for these applications and why domain specific quality information was important. We demonstrated the effectiveness of our approach using several large diverse data sets by comparing with traditional link-based approaches.

ConceptRank framework for identifying specialists in CQAs: The ranking research primarily focuses on identifying global experts (those with breadth) in CQA services, but these global experts may not be best-suited to answer a specific question. Based on this observation, we proposed the ConceptRank framework to identify expertise at the level of a concept as a basic building block. Using the concept-rank, other higher levels of expertise for answering a specific question or for answering a sub-topic can be accomplished. We proposed several approaches to identify specialists for a particular question by considering answer quality and graph structure. We demonstrated the effectiveness of our approach by comparing them with traditional global rank approaches.

Relationship between HITS and Non-negative Matrix Factorization: we discussed the ranking problem in web and social networks and proposed ILP and OLP of a graph to help understand HITS approach in contexts other than web graphs. We established that the two probabilities identified correspond to the hub and authority vectors of the HITS approach and have used the standard non-negative matrix decomposition approach to calculate these two probabilities for each node. Then, we proved HITS vectors had the same rank order as that of ILP/OLP vectors. In addition, we applied ILOD to different types of graphs representing applications other than the web graph. Experimental results validated the relationship between ILOD approach and HITS algorithm.

To summarize, this thesis addressed the two most important problems in the CQAs – (i) analysis of answer quality, and (ii) analysis of answerer quality. We proposed novel solutions to these two problems that used machine learning and link-based approaches. The results presented in this thesis made these data archived much more useful from a search perspective. They was able to be used for routing questions in real-time to appropriate persons to get the best possible answer. As the scale and spread of question/answer communities continues to rise, this dissertation could act as a stepping stone for spawning new threads of research in these areas, which in turn, would lead to the development of the next generation of sophisticated frameworks for information retrieval using question/answer archives as yet another information source.

References

[1] C. Shah and J. Pomerantz, "Evaluating and predicting answer quality in community q/a," in *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval.* ACM, 2010, pp. 411–418.

[2] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web." *Stanford InfoLab Technical Report*, 1999.

[3] M. Sahami, V. Mittal, S. Baluja, and H. Rowley, "The happy searcher: Challenges in web information retrieval," *PRICAI 2004: Trends in Artificial Intelligence*, pp. 3–12, 2004.

[4] W. Meng, C. Yu, and K. Liu, "Building efficient and effective metasearch engines," *ACM Computing Surveys (CSUR)*, vol. 34, no. 1, pp. 48–89, 2002.

[5] M. Bergman, "White paper: the deep web: surfacing hidden value," *Journal of Electronic Publishing*, vol. 7, no. 1, 2001.

[6] A. Telang, C. Li, and S. Chakravarthy, "One size does not fit all: Towards user-and query-dependent ranking for web databases," *Knowledge and Data Engineering, IEEE Transactions on*, no. 99, pp. 1–1, 2009.

[7] S. Eissen and B. Stein, "Analysis of clustering algorithms for web-based search," *Practical Aspects of Knowledge Management*, pp. 168–178, 2002.

[8] E. Chu, A. Baid, X. Chai, A. Doan, and J. Naughton, "Combining keyword search and forms for ad hoc querying of databases," in *Proceedings of the 35th SIGMOD international conference on Management of data.* ACM, 2009, pp. 349–360.

[9] J. English, M. Hearst, R. Sinha, K. Swearingen, and K. Yee, "Hierarchical faceted metadata in site search interfaces," in *CHI'02 extended abstracts on Human factors in computing systems.* ACM, 2002, pp. 628–639.

[10] O. Ben-Yitzhak, N. Golbandi, N. Har'El, R. Lempel, A. Neumann, S. Ofek-Koifman, D. Sheinwald, E. Shekita, B. Sznajder, and S. Yogev, "Beyond basic faceted search," in *Proceedings of the international conference on Web search and web data mining.* ACM, 2008, pp. 33–44.

[11] J. Madhavan, S. Jeffery, S. Cohen, X. Dong, D. Ko, C. Yu, and A. Halevy, "Web-scale data integration: You can only afford to pay as you go," in *Proceedings of CIDR*, 2007, pp. 342–350.

[12] S. Kambhampati, U. Nambiar, Z. Nie, and S. Vaddi, "Havasu: A multi-objective, adaptive query processing framework for web data integration," in *ASU CSE.* Citeseer, 2002.

[13] K. Chang, B. He, and Z. Zhang, "Toward large scale integration: Building a metaquerier over databases on the web," in *Proceedings of CIDR*, 2005, pp. 44–55.

[14] H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, and J. Widom, "Integrating and accessing heterogeneous information sources in tsimmis," in *Proceedings of the AAAI Symposium on Information Gathering*, vol. 3, 1995, pp. 61–64.

[15] W. Cody, L. Haas, W. Niblack, M. Arya, M. Carey, R. Fagin, M. Flickner, D. Lee, D. Petkovic, P. Schwarz, *et al.*, "Querying multimedia data from multiple repositories by content: the garlic project," in *Proceedings of the third IFIP WG2*, vol. 6, 1995, pp. 17–35.

[16] O. Duschka and M. Genesereth, "Infomaster-an information integration tool," in *Proceedings of the International Workshop on Intelligent Information Integration*

during the 21st German Annual Conference on Artificial Intelligence, KI-97, Freiburg, Germany, 1997.

[17] A. Levy, "The information manifold approach to data integration," *IEEE Intelligent Systems*, vol. 13, no. 5, pp. 12–16, 1998.

[18] W. Cohen, "Data integration using similarity joins and a word-based information representation language," *ACM Transactions on Information Systems (TOIS)*, vol. 18, no. 3, pp. 288–321, 2000.

[19] Z. Ives, D. Florescu, M. Friedman, A. Levy, and D. Weld, "An adaptive query execution system for data integration," *ACM SIGMOD Record*, vol. 28, no. 2, pp. 299–310, 1999.

[20] M. Lenzerini, "Data integration: A theoretical perspective," in *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems.* ACM, 2002, pp. 233–246.

[21] J. Castillo, A. Silvescu, D. Caragea, J. Pathak, and V. Honavar, "Information extraction and integration from heterogeneous, distributed, autonomous information sources-a federated ontology-driven query-centric approach," in *Information Reuse and Integration, 2003. IRI 2003. IEEE International Conference on.* IEEE, 2003, pp. 183–191.

[22] M. Doerr, J. Hunter, and C. Lagoze, "Towards a core ontology for information integration," *Journal of Digital information*, vol. 4, no. 1, 2011.

[23] R. Domenig and K. Dittrich, "A query based approach for integrating heterogeneous data sources," in *Proceedings of the ninth international conference on Information and knowledge management.* ACM, 2000, pp. 453–460.

[24] D. Florescu, A. Levy, and A. Mendelzon, "Database techniques for the world-wide web: A survey," *SIGMOD record*, vol. 27, no. 3, pp. 59–74, 1998.

[25] M. Friedman, A. Levy, and T. Millstein, “Navigational plans for data integration,” in *Proceedings of the National Conference on Artificial Intelligence.* JOHN WILEY & SONS LTD, 1999, pp. 67–73.

[26] L. Liu and C. Pu, “An adaptive object-oriented approach to integration and access of heterogeneous information sources,” *Distributed and Parallel Databases*, vol. 5, no. 2, pp. 167–205, 1997.

[27] A. Bozzon, M. Brambilla, S. Ceri, and P. Fraternali, “Liquid query: multi-domain exploratory search on the web,” in *Proceedings of the 19th international conference on World wide web.* ACM, 2010, pp. 161–170.

[28] D. Braga, D. Calvanese, A. Campi, S. Ceri, R. Daniel, D. Martinenghi, P. Merialdo, and R. Torlone, “Ngs: a framework for multi-domain query answering,” in *Data Engineering Workshop, 2008. ICDEW 2008. IEEE 24th International Conference on.* IEEE, 2008, pp. 254–261.

[29] A. Telang, S. Chakravarthy, and C. Li, “Query-by-keywords (qbk): Query formulation using semantics and feedback,” *Conceptual Modeling-ER 2009*, pp. 191–204, 2009.

[30] G. Littlepage and A. Mueller, “Recognition and utilization of expertise in problem-solving groups: Expert characteristics and behavior.” *Group Dynamics: Theory, Research, and Practice*, vol. 1, no. 4, p. 324, 1997.

[31] J. Zhang, M. Ackerman, and L. Adamic, “Expertise networks in online communities: structure and algorithms,” in *Proceedings of the 16th international conference on World Wide Web.* ACM, 2007, pp. 221–230.

[32] P. Jurczyk and E. Agichtein, “Discovering authorities in question answer communities by using link analysis,” in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management.* ACM, 2007, pp. 919–922.

[33] J. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM (JACM)*, vol. 46, no. 5, pp. 604–632, 1999.

[34] D. M. Strong, Y. W. Lee, and R. Y. Wang, "Data Quality in Context," *Communications of the ACM*, vol. 40, no. 5, pp. 103–110, 1997.

[35] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web," University of Stanford, California, USA, Tech. Rep., 1999, http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf.

[36] J. M. Kleinberg, "Authoritative Sources in a Hyperlinked Environment," *Journal of the ACM*, vol. 46, no. 5, pp. 604–632, 1999.

[37] J. Cho and R. E. Adams, "Page Quality: In Search of an Unbiased Web Ranking," in *SIGMOD*. Baltimore, Maryland, USA: ACM, 2005, pp. 551–562.

[38] X. Zhu and S. Gauch, "Incorporating Quality Metrics in Centralized/distributed Information Retrieval on the World Wide Web," in *SIGIR*. Athens, Greece: ACM, 2000, pp. 288–295.

[39] J. Jeon, W. B. Croft, J. H. Lee, and S. Park, "A Framework to Predict the Quality of Answers with Non-textual Features," in *SIGIR*. Seattle, Washington, USA: ACM, 2006, pp. 228–235.

[40] C. Shah and J. Pomerantz, "Evaluating and Predicting Answer Quality in Community QA," in *SIGIR*. Geneva, Switzerland: ACM, 2010, pp. 411–418.

[41] F. M. Harper, D. Raban, S. Rafaeli, and J. A. Konstan, "Predictors of Answer Quality in Online Q&A Sites," in *SIGCHI*. Florence, Italy: ACM, 2008, pp. 865–874.

[42] J. Bian, Y. Liu, E. Agichtein, and H. Zha, "Finding the Right Facts in the Crowd: Factoid Question Answering over Social Media," in *WWW*. Madrid, Spain: ACM, 2008, pp. 467–476.

[43] M. Surdeanu, M. Ciaramita, and H. Zaragoza, "Learning to Rank Answers on Large Online QA Collections," in *ACL*. Columbus, Ohio, USA: The Association for Computer Linguistics, 2008, pp. 719–727.

[44] K. Balog, L. Azzopardi, and M. De Rijke, "Formal models for expert finding in enterprise corpora," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2006, pp. 43–50.

[45] J. Shetty and J. Adibi, "Discovering important nodes through graph entropy the case of enron email database," in *Proceedings of the 3rd international workshop on Link discovery*. ACM, 2005, pp. 74–81.

[46] C. Campbell, P. Maglio, A. Cozzi, and B. Dom, "Expertise identification using email communications," in *Proceedings of the twelfth international conference on Information and knowledge management*. ACM, 2003, pp. 528–531.

[47] B. Dom, I. Eiron, A. Cozzi, and Y. Zhang, "Graph-based ranking algorithms for e-mail expertise analysis," in *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*. ACM, 2003, pp. 42–48.

[48] J. Liu, Y. Song, and C. Lin, "Competition-based user expertise score estimation," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information*. ACM, 2011, pp. 425–434.

[49] R. Herbrich, T. Minka, and T. Graepel, "Trueskill: A bayesian skill rating system," *Advances in Neural Information Processing Systems*, vol. 19, p. 569, 2007.

[50] D. Mease, "A penalized maximum likelihood approach for the ranking of college football teams independent of victory margins," *The American Statistician*, vol. 57, no. 4, pp. 241–248, 2003.

[51] J. Bian, Y. Liu, D. Zhou, E. Agichtein, and H. Zha, "Learning to recognize reliable users and content in social media with coupled mutual reinforcement," in *Proceedings of the 18th international conference on World wide web*. ACM, 2009, pp. 51–60.

[52] A. Pal, R. Farzan, J. Konstan, and R. Kraut, "Early detection of potential experts in question answering communities," *User Modeling, Adaption and Personalization*, pp. 231–242, 2011.

[53] A. Pal, F. M. Harper, and J. A. Konstan, "Exploring question selection bias to identify experts and potential experts in community question answering," *ACM Trans. Inf. Syst.*, vol. 30, no. 2, p. 10, 2012.

[54] J. Ayanian, P. Hauptman, E. Guadagnoli, E. Antman, C. Pashos, and B. McNeil, "Knowledge and practices of generalist and specialist physicians regarding drug therapy for acute myocardial infarction," *New England Journal of Medicine*, vol. 331, no. 17, pp. 1136–1142, 1994.

[55] L. Harrold, T. Field, and J. Gurwitz, "Knowledge, patterns of care, and outcomes of care for generalists and specialists," *Journal of General Internal Medicine*, vol. 14, no. 8, pp. 499–511, 2001.

[56] J. Rose, E. O'Toole, N. Dawson, C. Thomas, A. Connors Jr, N. Wenger, R. Phillips, M. Hamel, D. Reding, H. Cohen, *et al.*, "Generalists and oncologists show similar care practices and outcomes for hospitalized late-stage cancer patients," *Medical care*, vol. 38, no. 11, p. 1103, 2000.

[57] B. Landon, I. Wilson, S. Cohn, C. Fichtenbaum, M. Wong, N. Wenger, S. Bozzette, M. Shapiro, and P. Cleary, "Physician specialization and antiretroviral therapy for hiv," *Journal of general internal medicine*, vol. 18, no. 4, pp. 233–241, 2003.

[58] L. Page, S. Brin, R. Motwani, and T. Winograd, “The PageRank Citation Ranking: Bringing Order to the Web,” Stanford Digital Library Technologies Project, Tech. Rep., 1998. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.31.1768

[59] C. Campbell, P. Maglio, A. Cozzi, and B. Dom, “Expertise identification using email communications,” in *Proceedings of the twelfth international conference on Information and knowledge management.* ACM, 2003, pp. 528–531.

[60] B. Dom, I. Eiron, A. Cozzi, and Y. Zhang, “Graph-based ranking algorithms for e-mail expertise analysis,” in *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery.* ACM, 2003, pp. 42–48.

[61] G. Iván and V. Grolmusz, “When the web meets the cell: using personalized pagerank for analyzing protein interaction networks,” *Bioinformatics*, vol. 27, no. 3, pp. 405–407, 2011.

[62] N. Ma, J. Guan, and Y. Zhao, “Bringing pagerank to the citation analysis,” *Information Processing & Management*, vol. 44, no. 2, pp. 800–810, 2008.

[63] D. Seung and L. Lee, “Algorithms for non-negative matrix factorization,” *Advances in neural information processing systems*, vol. 13, pp. 556–562, 2001.

[64] S. Li, X. Hou, H. Zhang, and Q. Cheng, “Learning spatially localized, parts-based representation,” in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1. IEEE, 2001, pp. I–207.

[65] M. Cooper and J. Foote, “Summarizing video using non-negative similarity matrix factorization,” in *Multimedia Signal Processing, 2002 IEEE Workshop on.* IEEE, 2002, pp. 25–28.

[66] V. Pauca, F. Shahnaz, M. Berry, and R. Plemmons, "Text mining using nonnegative matrix factorizations," in *Proc. SIAM Intl conf on Data Mining*, 2004, pp. 452–456.

[67] J. Brunet, P. Tamayo, T. Golub, and J. Mesirov, "Metagenes and molecular pattern discovery using matrix factorization," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 12, p. 4164, 2004.

[68] F. Sha, L. Saul, and D. Lee, "Multiplicative updates for nonnegative quadratic programming in support vector machines," *Departmental Papers (CIS)*, p. 29, 2002.

[69] N. Srebro, J. Rennie, and T. Jaakkola, "Maximum Margin Matrix Factorization," *Advances in neural information processing systems*, vol. 17, no. 5, pp. 1329–1336, 2005.

[70] F. Wang, T. Li, and C. Zhang, "Semi-supervised clustering via matrix factorization," in *Proceedings of The 8th SIAM Conference on Data Mining*, 2008.

[71] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine* 1," *Computer networks and ISDN systems*, vol. 30, no. 1-7, pp. 107–117, 1998.

[72] Y. Liu, J. Bian, and E. Agichtein, "Predicting information seeker satisfaction in community question answering," in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval.* ACM, 2008, pp. 483–490.

[73] J. Jeon, W. Croft, J. Lee, and S. Park, "A framework to predict the quality of answers with non-textual features," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval.* ACM, 2006, pp. 228–235.

[74] T. Joachims, "Optimizing search engines using clickthrough data," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2002, pp. 133–142.

[75] M. G. Kendall, "A new measure of rank correlation," *Biometrika*, vol. 30, no. 1/2, pp. 81–93, 1938.

[76] D. Radev, H. Qi, H. Wu, and W. Fan, "Evaluating web-based question answering systems," *Ann Arbor*, vol. 1001, p. 48109, 2002.

[77] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[78] F. Harper, D. Raban, S. Rafaeli, and J. Konstan, "Predictors of answer quality in online q&a sites," in *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems.* ACM, 2008, pp. 865–874.

[79] B. Everitt, A. Skrondal, and I. Books24x7, *The Cambridge dictionary of statistics.* Cambridge University Press Cambridge, 2002, vol. 4.

[80] L. Adamic, J. Zhang, E. Bakshy, and M. Ackerman, "Knowledge sharing and yahoo answers: everyone knows something," in *Proceeding of the 17th international conference on World Wide Web.* ACM, 2008, pp. 665–674.

[81] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39–43, Mar. 1953.

[82] S. Redner, "How popular is your paper? An empirical study of the citation distribution," *European Physical Journal B*, vol. 4, pp. 131–134, 1998.

[83] G. Salton, A. Wong, and C. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[84] J. Herlocker, J. Konstan, L. Terveen, and J. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 5–53, 2004.

[85] K. Jones, “A statistical interpretation of term specificity and its application in retrieval,” *Journal of documentation*, vol. 28, no. 1, pp. 11–21, 1972.

[86] M. Marcus, M. Marcinkiewicz, and B. Santorini, “Building a large annotated corpus of english: The penn treebank,” *Computational linguistics*, vol. 19, no. 2, pp. 313–330, 1993.

[87] Y. Yin and K. Yasuda, “Similarity coefficient methods applied to the cell formation problem: A taxonomy and review,” *International Journal of Production Economics*, vol. 101, no. 2, pp. 329–352, 2006.

[88] D. Powers, “Evaluation: From precision, recall and f-factor to roc, informedness, markedness & correlation,” *School of Informatics and Engineering, Flinders University, Adelaide, Australia, Tech. Rep. SIE-07-001*, 2007.

[89] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, “Evaluating Collaborative Filtering Recommender Systems,” *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 5–53, 2004.

[90] M. Newman, “The structure and function of complex networks,” *SIAM review*, pp. 167–256, 2003.

[91] R. Rao Nadakuditi and M. Newman, “Graph spectra and the detectability of community structure in networks,” 2012.

[92] M. Newman, *Networks: an introduction.* Oxford University Press, Inc., 2010.

[93] D. Welch, G. Nicholls, A. Rodrigo, and W. Solomon, “Integrating genealogy and epidemiology: the ancestral infection and selection graph as a model for reconstructing host virus histories,” *Theoretical population biology*, vol. 68, no. 1, pp. 65–75, 2005.

[94] S. Pimm, “The complexity and stability of ecosystems,” *Nature*, vol. 307, no. 5949, pp. 321–326, 1984.

[95] J. Lawton, “What do species do in ecosystems?” *Oikos*, pp. 367–374, 1994.

[96] T. Aittokallio and B. Schwikowski, “Graph-based methods for analysing networks in cell biology,” *Briefings in bioinformatics*, vol. 7, no. 3, pp. 243–255, 2006.

[97] D. Lee, H. Seung, *et al.*, “Learning the parts of objects by non-negative matrix factorization,” *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

Bibliography Statement

Yuanzhe Cai was born in Baoji, China in 1984. He received his Bachelor of Engineering degree in Computer Science and Engineering from Xidian university, China in July 2005. He received his Masters of Science degree in Computer Science from Renmin University, in July 2008. He received his Doctor of Philosophy in Computer Science and Engineering from The University of Texas at Arlington in May 2014.

He has served as a Graduate Teaching Assistant in the Department of Computer Science and Engineering at The University of Texas at Arlington from 2009 till 2014. His research interests are mainly focused on social network analysis, including expertise ranking in the community, recommendation system, and similarity calculation.