UTILIZING FORUM META-INFORMATION TO IMPROVE RELEVANCE IN FORUM DISCOVERY

BY

HAN-WEN YEH

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2013

Urbana, Illinois

Adviser:

Associate Professor Chengxiang Zhai

ABSTRACT

The World Wide Web is abundant with discussion forums about all sorts of topics, and it is becoming increasingly difficult for people to find quality forums about a certain topic in which to join. While many modern search engines crawl through discussion forums and return threads in various forums based on each thread's relevance to query terms, there are not a lot of search engines that can tell a user the best forum to join to discuss about a topic of interest. We have observed that users looking for forums to join may be more interested in joining one with a high posting activity from a diverse community. However, current search engines do not fully utilize information from forums to provide the best search results to these users.

We propose a specialized retrieval system that looks for discussion forums and returns a list of forums based on a user's search query. When ranking search results, this system not only takes relevance into account, but also the posting activity and number of members from each forum. We evaluate our system over a manually-selected set of 150 forums that cover ten general topics, and show that our system retrieves forums that are relevant to the search query and are more appealing to users than forums retrieved from conventional search engines.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude and appreciation to my advisor, Professor Chengxiang Zhai, without whom this thesis would not have been possible. Many of the topics introduced in this thesis were the results of his attention to detail and willingness to explore new ideas, and thanks to his utmost dedication to his students I was able to complete this thesis.

I would also like to extend my thanks to my seniors at the University of Illinois whom I have had the pleasure of becoming acquainted with. Their mentoring and words of advice have helped me settle in as a first-year graduate student. I especially want to thank Hyun Duk Cho for his willingness to help me day and night, no matter how busy he is with his own work.

Finally, I want to thank all my friends and family members living all over the world who have supported me throughout this journey. Thank you for always encouraging me and cheering me on, and please continue to look after me from here on out.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Background

Ever since the dawn of the Internet, people around the globe have relied on it as a place for the discussion of ideas with others that have similar interests. In this day and age, there exist a multitude of discussion forums about any topic one may wish to discuss. Moreover, there are hundreds upon hundreds of different forums that exist for the discussion of each topic. Given the number of forums in existence, then, it is not easy for one to find the best forums about topics one is interested in without specialized tools. Search engines such as Google and Bing can, to an extent, help users find forums about topics they are interested in. However, conventional search engines currently treat forums as regular websites and do not utilize the valuable information hidden within the forums themselves when ranking forums as search results. In addition, many search engines produce multiple search results from the same forum, which may be useful if a user is searching for a particular thread in a forum, but not really useful for users who are looking for forums to visit and join.

1.2 Related Works

As the number of websites on the Internet increased, numerous search engines had been developed for specialized searches on specific topics. We surveyed many self-proclaimed forum search engines to see how useful they are to people looking for forums to join. Unfortunately, the majority of these search engines, including Omgili [1], do not return

forums as search results, but return forum posts or threads as results, and thus may not be as useful for users looking for forums to join. We did, however, find a few websites that provide a search functionality that produced forums as search results and are more geared towards people looking for forums to visit and join. These websites are Forums Search [2], boardreader [3], and Big Boards [4].

Forums Search is a forum search engine that tries to return forum home pages instead of discussion threads. However, we analyzed the strengths and weaknesses of Forums Search, and we found that its search results did not only include forum home pages, but also discussion threads from within forums, and additional pages from non-forum sites such as Wikipedia and Answers.com. Additionally, we found that although a basic search system was implemented, the algorithm that was used did not necessarily take the popularity and activity of the sites into account.

Big Boards is a website boasting of database of 2337 forums as of the writing of this thesis. Forums are sorted by two levels of categorization and users looking for a forum to join can go to the forum directory provided by the website and narrow down on the category of forum they are searching for. We found this to be a very similar service to what Yahoo! Directory provides, except for forums. In addition, for each of the forums in its database, Big Boards also stores various pieces of information about the forum (including the number of posts and members each forum has) in the database, and for each forum category in the forum directory, forums can be sorted by each of the attributes stored in the database. Finally, Big Boards also provides a search functionality for its users, where users can search for forums about topics they are interested in. However, it is not possible to sort search results by any of the attributes available in the forum directory, and upon further

investigation, we found that forums only appeared in search results if the URL, name, or description of the forum contained parts of the search query. For example, in the hopes of finding a forum about the Honda Accord, we searched for the term “Accord”. Neither of the two forums that were returned were about Honda Accords – they only had the word “according” in their descriptions. In addition, a search for “Honda Accord” returned zero results, while a search for “Honda” returned many forums, most of which contain discussions about the Honda Accord.

Boardreader is a search engine that allows users to search for forum posts, forum threads, and subforums within a discussion forum. While it does not have a forum search functionality, we found that the subforum search feature to be helpful in finding forums that are not necessarily about a certain topic, but has a non-trivial amount of posts about that topic. However, Boardreader’s subforum search only ranks results by relevance, and does not use information like the number of posts and number of members in each forum to rank search results. Despite its shortcomings, however, we found Boardreader to have the most useful forum search functionality out of all the websites we surveyed.

In summary, we found that there is currently no forum search engine that is able to return forums as search results while also being aware of hidden subtopics within the forum and aware of the amount of activity inany of the forums.

1.3 Contributions

In this thesis, we propose a forum search engine that is designed to return forums that produce results that are more useful to users than results returned by the forum search engines mentioned in the previous section. Specifically, we propose two novel features that

we incorporate into our system that allow our search engine to be more useful to users than conventional forum search engines.

Firstly, in the absence of forums directly about the search query, the system we propose can return forums that are not about the search query, but have a non-trivial amount of threads and posts about the search query. As mentioned in the previous section, the Big Board forum search engine does not retrieve any forums when we searched for "accord", but retrieved many Honda-related forums where much of the discussion is about Honda Accords. Search engines that return forum threads or posts, as well as Boardreader's subforum search, automatically incorporate this feature due to the nature of the webpages that are returned, but these search engines do not return forums as search results. Our system aims to be just as effective in this respect as these search engines, but only return forum homepages as search results.

Secondly, the system we propose supports the novel feature of activity awareness. That is, our system gives increased importance to forum sites that we concluded are more active or have more information and would thus be more likely to be useful to the user. We made the observation that people who want to join forums would be more interested in joining forums with a large number of posts from a large community, and we believe that users of a forum search engine would be more satisfied if the top results were forums with high activity. With this additional feature, our search engine would facilitate forum search based on the query topics but would serve a greater purpose and return search results of higher quality than forum search engines in existence. This is the key novelty feature in our system.

The rest of the thesis is organized as follows. In Chapter 2, we outline the list of forums that we use for our prototype system and describe the method in which we extract data from these forums. In Chapter 3, we outline the method in which the index for our prototype system is built and describe in detail the algorithm we use for our prototype system and in Chapter 4, we conduct both automated and human evaluation on this system in three different ways and analyze whether our system is more useful to people looking for forums or not. Finally, in Chapter 5, we conclude and discuss future works.

CHAPTER 2

DATA

In this chapter, we talk about the text and numeric forum data used by our forum search engine system. We first talk about the list of forums we will use to evaluate our prototype system and the way in which they are categorize. We then talk about how we extract the data that we want from each forum in our list.

2.1 Data Categorization

Due to time constraints and the level of complexity required in building a web crawler that retrieves an unbiased list of forums, we opted to manually create the list of forums we would be using for our prototype system. In addition, manually choosing the forums we use allows us to label each forum by topic, which allows us to evaluate the relevance of our search results more easily. The system can be further extended to include more forums by developing a more automatic crawler of forums, which we leave as future work. For our prototype system, we collected data from 150 discussion forums. The forums spanned were classified into ten general topics and each topic had 15 forums. The topics are: soccer, movies and TV, music, video games, computer programming, food, travel, health, cars, and fashion.

For each of the general topics, some of the forums are about that general topic while others are about subtopics of that general topic. For example, out of the 15 forums about cars, only 6 of them are about cars in general. 3 of the forums are only about Honda cars

Soccer
• Soccer (7)
  ○ Liverpool FC (3)
  ○ Manchester United FC (1)
  ○ Australian A-League (1)
  ▪ Brisbane Roar (2)
  ○ SoCal soccer (1)
	Movies and TV
	• Movies (5)
	• TV (6)
	  ○ Asian TV (6)

Music
• Music (3)
  ○ Classical music (2)
  ▪ W. A. Mozart (1)
  ○ Pop music (1)
  ▪ Japanese pop (1)
  ▪ Korean pop (2)
  ○ Techno music (1)
  ○ Country music (1)
  ○ Rock music (1)
  ○ Jazz (2)
	Video Games
	• Video games (5)
	  ○ SimCity series (4)
	  ○ FIFA series (2)
	  ○ Flash games (2)
	  ○ Pokémon games (2)

Programming
• Programming (5)
  ○ Java programming (2)
  ○ Python programming (1)
  ○ C programming (1)
  ○ Web development (2)
  ○ Game development (4)
	Food
	• Food (5)
	  ○ Food in Portland, OR (1)
	  ○ Food in Chicago, IL (1)
	  ○ Cooking (6)
	  ▪ Recipes (2)

Travel
• Travel (5)
  ○ Backpacking (4)
  ○ Visa information (3)
  ○ Travel in Australia (3)
	Health
	• Health (5)
	  ○ Dieting (4)
	  ○ ADD/ADHD (2)
	  ○ Allergies (2)
	  ○ Teeth (2)

Cars
• Cars (6)
  ○ Sports cars (2)
  ○ Honda cars (3)
  ▪ Honda Civic cars (4)
	Fashion
	• Fashion (12)
	  ○ Men's fashion (3)

Figure 2.1: Breakdown of forums by topic

and a further 3 are strictly about the Honda Civic. A detailed breakdown of the subtopics in each general topic can be found in Figure 2.1.

We chose to keep track of subtopic information for evaluation purposes. We would like our system to return the forum that is most relevant to the search query and to do this, we

take the number of members and number of posts in each forum into account. However, a forum about a general topic is more likely to have more members and more posts than a forum about a subtopic due to the broader scope encompassed by the former. If a user searches for forums about a subtopic, our system should still try to rank forums about the subtopic with decent numbers of members and posts higher than larger forums about the general topic the query belongs to. To ensure that our system does indeed do this, we keep track of whether each of our 150 forums is a forum about a general topic or a forum about a subtopic and during evaluation, we will evaluate whether searching for a subtopic will make the subtopic forums be ranked higher than general topic forums or not.

An alternative way to evaluate this without keeping track of subtopic forums is by asking human evaluators to search for a subtopic of one of the general topics and see if the forums about the subtopic have higher rankings than forums about the general topic or not. However, this method of evaluation is not very controlled and there is no guarantee that our limited database of forums has forums about the subtopic that human evaluators choose.

2.2 Data Extraction

In order to obtain the data we need for our system, we wrote a custom crawler to crawl our list of 150 discussion forums. We used the crawler4j library for Java [5] to crawl up to 30MB of text data from each forum and stripped all HTML tags before writing to disk. Because our system is designed to return whole discussion forums as search results as opposed to forum threads, we also merged all the crawled pages from the same forum into one document. This allows our system to index each forum more easily.

In addition to query term relevance, our system also takes other forum data into account when ranking forums – namely, the number of members each forum has and the number of threads and posts each forum has. The number of members in a forum is relevant in determining how useful it is, as having more members lead to a higher forum activity as well as more viewpoints being expressed for each topic. The number of threads and posts each forum has is also relevant in determining how useful it is, as they measure the amount of information users can find from the forum.

To extract these pieces of information, we ran a simple script that parsed the home page of each of the forums and looked for certain regular expressions that correspond to each piece of information. For forums where the script was not able to find the relevant information, we manually extracted the information from the forum. Processes that can potentially further automate the data extraction process will be discussed in chapter 5. It must also be mentioned that even though all 150 of the forums we crawled displayed the number of members, threads, and posts, we found that not all discussion forums display these figures publically. Thus, this thesis makes the simplifying assumption that all forums publically display the number of members, threads, and posts, and further discussion on the possibility of estimating these numbers for forums that hide them is done in chapter 5.

CHAPTER 3

FORUM SEARCH

In the previous chapter, we talked about the data to be used by our prototype system. In this chapter, we talk about how this data will be indexed and retrieved by our system. First, we will talk briefly about how our system indexes the forum text and numeric data. Then, we explore several measures that can be used to determine whether a forum is useful to a user or not. These measures incorporate both relevance to a search query as well as the amount of activity in each forum. Finally, we will combine these measures to form an experimental scoring algorithm for our prototype system.

3.1 Indexing

To build our index, we opted to use Lucene [6], a popular information retrieval toolkit for Java. When Lucene builds an index, each file is treated as a single document. This had posed a potential problem for us as we wanted collections of documents – namely, those from the same forum – to be treated a single document. However, the multitude of separate webpages for our forums had already been merged into a single file per forum by the crawler, and thus we did not have to do any extra work to make Lucene recognize each forum as a single document. In addition to indexing the forum text data, we also stored the number of members, threads and posts for each forum into a separate database, as these numbers will eventually be used by Lucene during the document scoring process.

3.2 Scoring and Ranking

As mentioned in previous chapters, the forum search engines that are currently available do not have very sophisticated retrieval algorithms. In this section we propose a scoring algorithm for forum search engines that, given a search query, is designed to give a forum that is more useful a higher rank than one that is not as useful. We will first introduce several measures that can help determine how useful a forum is to a user based on their search query, and then we will explore how to use each of these measures in our scoring algorithm.

3.2.1 Relevance

To assess the relevance of a forum to a user's query, our general idea is to leverage an existing standard retrieval model. Many general retrieval models have been proposed, including vector space models [7] and probabilistic retrieval models [8]. Specifically, we use the TF-IDF weighting method [9], is a numerical measure that determines the relevance of query terms to a document in the context of a collection of documents. This measure is important to our system because, for a search query term, a high TF-IDF score for a forum indicates that the forum may be relevant to the search query. In addition, we believe that TF-IDF will not only help our system discover forums discussing the search query terms, but also forums not entirely about the search query terms but have a large number of threads about the search query terms. For example, a forum about Honda cars will have a high TF-IDF score for the query term “Honda”, and because the forum also contains many threads about the Honda Accord, it will probably also have a high TF-IDF score for the query term “accord”. This is good because if our database of forums, like Big

Boards [4], does not contain a forum solely discussing Honda Accords, our system should return forums about Honda cars in general as being the most relevant.

In our system, we are using a modified version of the TF-IDF algorithm incorporated into Lucene [6] known as the Lucene Practical Scoring Function [10], which, given a query q and a document d, scores the relevance of d to q as follows:

$$\text{lpsf}(q,d) = \text{coord}(q,d)\ \text{queryNorm}(q) \sum_{\substack{\forall t,\\ t \in q}} \left(\text{tf}(t,d)\ \text{idf}(t)^2\ \text{getBoost}(t)\ \text{norm}(t,d)\right) \quad (3.1)$$

where:

- $\text{coord}(q,d)$ is a scoring factor based on the number of terms in q that exist in d
- $\text{queryNorm}(q)$ is a normalizing factor used by Lucene to make scores between queries comparable
- $\text{tf}(t,d)$ is the term frequency factor, which measures the frequency in which term t appears in d; this is computed as:

  $$\text{tf}(t,d) = \text{freq}(t,d)^{\frac{1}{2}} \quad (3.2)$$

  where $\text{freq}(t,d)$ is the number of times t appears in d
- $\text{idf}(t)$ is the inverse document frequency, which measures the rarity in which term t occurs in the collection of documents; this is computed as:

  $$\text{idf}(t) = 1 + \log \frac{|D|}{\text{docFreq}(t) + 1} \quad (3.3)$$

  where $|D|$ is the total number of documents in the collection and $\text{docFreq}(t)$ is $|\{d \in D : t \in d\}|$, the number of documents containing t.
- $\text{getBoost}(t)$ is a Lucene function that allows for a customizable list of terms to be weighted differently; this function returns 1.0 for all terms in our system

- $\text{norm}(t, d)$ is the document length normalization factor, which prevents longer documents from gaining a term-frequency advantage over shorter documents

3.2.2 Activity

In addition to the Lucene Practical Scoring Function [10], which only measures the relevance of a query to a document, our scoring algorithm should also take the amount of activity in each forum into account. This can be measured using the number of members, threads and posts in each forum, which we have stored in a database. Our system has three factors that measure forum activity: total posts, member activity, and thread activity.

Total posts is a factor based on the total number of posts in the forum. This is an indicator of the activity of the forum, as forums with high post counts are the direct result of the active contributions of its members. For a forum d, this is computed as:

$$\log_{\alpha}(\text{p}(d) + \alpha) \tag{3.4}$$

where $\text{p}(d)$ is the number of posts in d and α is a constant to control the amount this measure affects the total score. The logarithm prevents forums with high post counts from gaining too much of an advantage over forums with lower posts counts and also reduces the differences in scores between forums whose post counts are very high. This makes sense because for two forums with sufficiently high post counts, we should be more interested in other measures than the number of extra posts one forum has over another. The addition by the constant α guarantees that this measure is greater than or equal to 1 and prevents the over-penalizing of forums with post counts less than α.

Member activity is a factor based on the average number of posts each member makes. This is a direct indicator of the activity of the forum, as the more posts each member makes, the more active the forum is. For forum d, this is computed as:

$$\log\left(\frac{\mathrm{p}(d)}{\mathrm{m}(d)+1}+1\right) \tag{3.5}$$

where $\mathrm{m}(d)$ is the number of members in d. The addition by 1 to the denominator prevents division by 0 and the addition by 1 to the fraction prevents the measure from being a negative number. Similarly to the total posts factor, a logarithm is used to prevent forums with a high post-member ratio from gaining too much leverage over forums with a lower ratio and also reduces the differences in scores between forums with a high ratio and forums with a lower ratio.

Thread activity is a factor similar to member activity, but based on the average number of posts in each thread. This is an important factor because a forum with a low number of posts per thread could be an indication that very few members reply to posts in the forum. For forum d, this is computed as:

$$\log\left(\frac{\mathrm{p}(d)}{\mathrm{t}(d)+1}+1\right) \tag{3.6}$$

where $\mathrm{t}(d)$ is the number of threads in d.

The activity of a forum d, then, can be measured by combining the factors mentioned above in the following function:

$$\mathrm{activity}(d) = \log_{\alpha}(\mathrm{p}(d)+\alpha)\ \log\left(\frac{\mathrm{p}(d)}{\mathrm{m}(d)+1}+1\right)\log\left(\frac{\mathrm{p}(d)}{\mathrm{t}(d)+1}+1\right) \tag{3.7}$$

It must be noted that while there are potentially many other ways to combine them, as a first work in studying this ranking problem, we simply explored this form of combination,

leaving the exploration of other forms as a future work. In addition, it should also be noted that, because the forum activity does not need a query to be computed, it can be pre-computed for each forum and stored into database and does not need to be computed each time a search is made on the system.

3.2.3 Combined Scoring Function

Our system treats TF-IDF and activity as two orthogonal dimensions of the score and combines the measures we mentioned in the previous sections into the following scoring function:

$$\text{score}(q, d) = \text{lpsf}(q, d)\ \text{activity}(d) \tag{3.8}$$

where q is a query, d is a forum, and $\text{score}(q, d)$ is a score representing how useful forum d is to a user, given that the user is search for a forum about q. Lucene computes this score for every forum in the database and returns the forums with the highest scores with the forum with the highest score ranked as number 1. Once again, it must be noted that even though there are many other ways to combine the two measures for the scoring function, we only explored this form of combination and we leave the exploration of other methods to future work.

CHAPTER 4

EVALUATION

In the previous chapters, we introduced a forum search engine system that, given a search query, tries to return the most useful forum for that search query. In this chapter, we evaluate the performance of our prototype of the system. We run three different forms of tests. First, we evaluate the precision of search results returned by the system. Then, we ask human evaluators to evaluate the order in which our system ranks results. Finally, we ask human evaluators to evaluate our system against the forum search engines listed in section 1.2. Through these tests, we look to prove that our system, in addition to being able to return forums relevant to search queries, also incorporates the two novel features we listed in section 1.3.

4.1 Evaluation Based on Assumed Relevance

Our system incorporates TF-IDF [9] into the scoring function to raise the rank of relevant documents. However, it is possible that the other measures used in our scoring function may overpower the TF-IDF measure and give irrelevant forums a high score. To ensure that our system does return relevant forums as search results, we evaluate the precision of our system against one that uses Lucene's default TF-IDF scoring function, the Lucene Practical Scoring Function [10], without any modifications. In addition, in order to ensure that our idea of computing the logarithms of each of the activity measures is a method worth pursuing, we also evaluate the precision of our system against the precision of a system

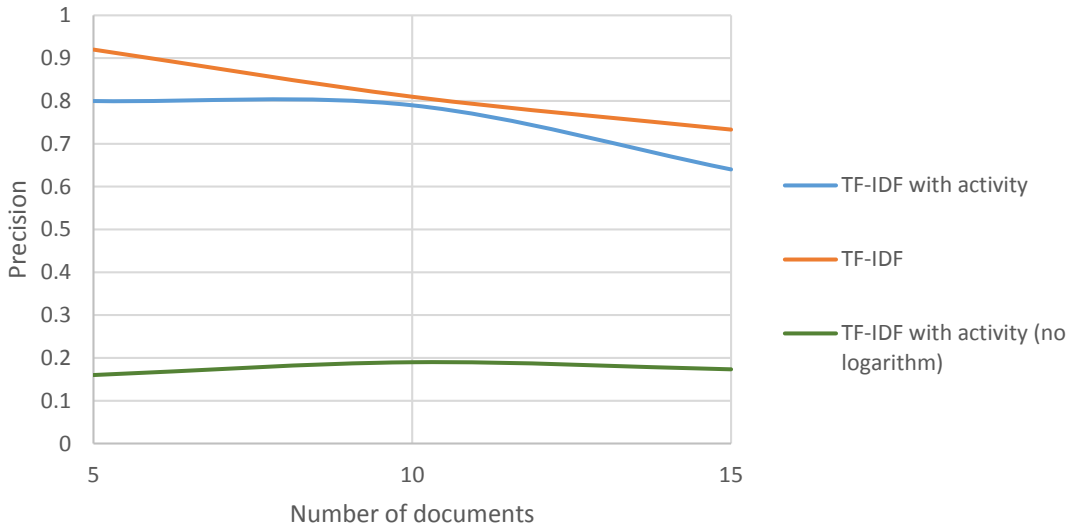

Figure 4.1: Precision at 5, 10, and 15 documents for the three evaluated models

that uses the same combination of measures for the scoring function but does not take the logarithm of each factor of the activity measure. Precision is defined as follows:

$$\text{precision} = \frac{\text{documents retrieved and relevant}}{\text{documents retrieved}}$$

We run 10 queries on each of the systems and find the precision of each system at 5, 10, and 15 documents. All three systems have the same index of 150 forums outlined in section 2.1 and each of the queries represents one of the 10 general topics each forum falls under. A result is deemed relevant if it belongs in the general topic the search query was from. In addition, for our system, we use a value of 100 for our constant α. The results of our evaluation are likely to change if the constant is changed, but we leave further experimentation with different values of α as future work.

The precisions at 5, 10, and 15 documents are shown in Figure 4.1. As can be seen, our system (labeled "TF-IDF with activity" in the figure) matches up fairly well with the system that is ranking forums purely based on TF-IDF relevance scores, coming within 0.02 difference in precision at 10 documents. The reason that our system does not perform

Model	MAP
TF-IDF with activity	0.70
TF-IDF	0.81
TF-IDF with activity (no logarithm)	0.19

Table 4.1: Mean average precision of the three evaluated models

quite as well as a pure relevance-scoring system in terms of precision is because our list of forums only contains 15 forums per general topic, which is not a lot, and many of them may not be forums that people would like to join. For example, many of these forums have more members than posts, which shows that not a lot of people join this forum to contribute their own ideas, and many of these forums also have less than 2 posts per thread on average, which shows that not a lot of people respond to threads in these forums.

In addition to finding the precision at 5, 10, and 15 documents, we also calculated the mean average precision (MAP) of each of our systems. MAP, a common way of evaluating ranking accuracy, is defined as follows:

$$\text{MAP} = \frac{\sum \text{precision at the rank of each relevant document retrieved}}{\text{number of relevant documents}}$$

The MAP values are shown in Table 4.1, and just like the precision at 5, 10, and 15 values, our system's MAP is comparable to but slightly lower than that of a pure relevance-scoring system.

One topic that did not score too well in the TF-IDF system (0.8 precision at 5 documents, 0.4 precision at 10 documents, 0.4 precision at 15 documents, and a MAP of 0.44) and scored even worse in our system (0.6 precision at 5 documents, 0.4 precision at 10 documents, 0.27 precision at 15 documents, and a MAP of 0.35) was "video games". We discovered that this was because people in video game forums usually refer to video games

as simply “games”, as video games are the only kinds of games being discussed in these forums. In addition, many sports, movie, and television forums also contain many instances of the term “video”, where people ask for or upload video clips to these forums.

Finally, it can also be seen in both Figure 4.1 and Table 4.1 that the system which did not use logarithms in the activity metrics scored significantly lower than our system that did use logarithms. This shows that logarithms are indeed needed when calculating each activity metric to prevent any of the activity metrics from affecting a document’s score too much.

4.2 Evaluation Against Past Works

We have now shown that our system, as a whole, does produce results with a precision comparable with that of a system using only TF-IDF. We would also like to evaluate our system against the existing forum search engines that we explored in Chapter 1 to determine if our system is able to produce more relevant results than existing forum search engines. The search engines we evaluate our system against are Big Boards [4] and boardreader [3]. Because we cannot easily automate this evaluation, we have asked five human evaluators to help us with the evaluation.

In the evaluation, we give a human evaluator a search query and the top 5 results from each of the search engines we are evaluating, including ours, for that query without telling the evaluator which set of results are from which search engine. We then ask the evaluator to choose the set of results that is the most useful if they want to join a forum to discuss the search query. Each evaluator is given 20 queries to evaluate. 10 of them represent the general topics our forums fall under and the other 10 are queries made up by the evaluator

Search Engine	Score
Big Boards	0.14
Boardreader	0.13
Our prototype system	0.73

Table 4.2: Scores for each of the forum search engines evaluated

where each of the queries is a subtopic of each of these general topics. For each query, the search engine for which the evaluator selects as having the most useful forum results is given a score of 1.0 and the other two search engines are given a score of 0.0. The scores for each of the search engines are averaged over all 20 queries for all five evaluators to find the overall average score for each engine. This score denotes how often an evaluator thinks that a forum search engine has better results than its two competitors and is an indicator of how often a forum search engine produces more relevant results than the two other forum search engines.

Table 4.2 contains the scores of each of the search engines evaluated, and it can be seen that our prototype system has a much higher score than the two other forum search engines. This, then, shows that our prototype system produces the most relevant results in general out of the three forum search engines evaluated.

Comparing the results we obtained from automatic evaluation and human evaluation, we see that although standard retrieval models can return more topically-relevant forums, users of forum search engines prefer forums that are not only relevant in topic, but also have many users and are active in discussions, supporting our hypothesis that we need to optimize ranking by considering other factors such as activity.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

We have introduced a document scoring function for forum retrieval. This function is novel in that both relevance to search query and forum activity are considered when a document is scored, whereas in current forum search engines, only relevance to the search query is considered in document scoring, and forum information such as the number of posts, members, and threads are not utilized. We first described several measures that we use in our scoring function, each of which helps determine whether a forum is relevant to a user's interests or not. We then evaluated the performance of our prototype system against a system using TF-IDF, a commonly used information retrieval weighting method and showed that the accuracy of our system's results are comparable with that of the TF-IDF system. Finally, we evaluated our system against two existing forum search engines and showed that our prototype system outperformed both search engines.

We believe that the system we proposed in this thesis has the potential to impact the way people search for forums to join. We are also aware that the system we proposed makes a lot of simplifying assumptions and believe that it can be even more powerful by adding in more sophisticated features. This chapter will outline a couple ways we think we can work on in the future to improve our system.

Of course, one thing that our system will benefit much from is if our system had more forums in the index. Our prototype system only had 150 websites from 10 general categories, and that is a very insignificant fraction of all the discussion forums on the

Internet. In order to obtain more forum data, we need to have a better crawler that can automatically crawl through websites on the Internet, look for forums, and download web pages belonging to forums. In addition, our crawler also needs to be able to automatically extract the number of members, posts, and threads from each forum. For our prototype, we ran a simple script to look for certain regular expressions, but not every forum displayed their numbers in the format we were looking for. Instead, we would like to have a system that looks for entities representing the number of members, posts, and threads, and extracts these entities from each forum.

Another feature we would like to include in our system is a way to measure the number of members, posts, and threads for forums that do not display this information. As mentioned in chapter 2, not all discussion forums publically display some information that our system needs. In order to properly rank these forums, we would like our system to somehow estimate these pieces of information for forums that do not display them. In addition, for a better gauge of popular topics talked about in each forum, we would like the textual forum data to be cleaned more before building the index. Our system currently strips HTML tags from all the downloaded forum data. However, there still exists much noise in our data. Some of the things that we would like to strip out from our forum data include the headers and footers from each page, the member names and information beside each post, and spam posts in general.

Finally, one more thing that we are interested in investigating is the method of combining the different measures for the scoring function. In Chapter 3, we only explored one way to combine these measures, but there are many other possible ways of combining these measures, and we would like to experiment with some other methods to look for the

best way to design the scoring function. In addition, we would also like to experiment with different values of the constant α used in the activity measure to see how performance is affected.

REFERENCES

[1] “Omgili Forum Search,” 2011; http://omgili.com/

[2] “Forums Search,” 2011; http://forumssearch.com/

[3] “boardreader,” 2013; http://boardreader.com/

[4] “Big Boards,” 2013; http://big-boards.com/

[5] “crawler4j,” 2013; http://code.google.com/p/crawler4j/

[6] “Apache Lucene Core,” 2013; http://lucene.apache.org/core/

[7] G. Salton, A. Wong, and C. S. Yang. “A Vector Space Model for Automatic Indexing,” in *Communications of the ACM*, vol. 18, nr. 11, pages 613–620, 1975.

[8] ChengXiang Zhai. “Statistical Language Models for Information Retrieval. Synthesis Lectures on Human Language Technologies,” Morgan & Claypool Publishers, 2008.

[9] G. Salton and M. McGill, editors. “Introduction to Modern Information Retrieval,” McGraw-Hill, 1983.

[10] “Class TFIDFSimilarity,” 2013; http://lucene.apache.org/core/4_2_0/core/org/apache/lucene/search/similarities/TFIDFSimilarity.html