NAIST-IS-DD1261028

Doctoral Dissertation

Supporting Effective Knowledge Sharing through Extracting Search Activities in a Community of Interest

Papon Yongpisanpop

February 5, 2015

Department of Information Science
Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Papon Yongpisanpop

Supporting Effective Knowledge Sharing through Extracting Search Activities in a Community of Interest[*]

Papon Yongpisanpop

Abstract

A Community of Interest (CoI) is a group of people who share a common interest or passion. The purpose of the CoI is to provide a place where people who share a common interest can go and exchange information, ask questions, and express their opinions about the topic. With todays technology people can access information online by using web search engine, which allow people to discover and share knowledge within a CoI. High turnover in CoI is one of the important causes that make it hard to capture and share knowledge. The question is whether we can create a system that will capture community-wide knowledge in a real time and make it widely available to all its members or not. This dissertation focuses on the issue of reducing knowledge-sharing efforts while people search and reuse users past search results to improve future search results to be more relevant for a community of interest. It proposes Search Activity Knowledge Extraction (SAKE) model, which extract the knowledge from search behavior and share it through the CoI. A framework called Adaptive Search Framework (ASF) based on the SAKE model can collect ten most used keywords and evaluate Top-5 and Top-10 search results using standard Topic-Sensitive PageRank (OSim and KSim) in the CoI environment. A user experience study to evaluate the user satisfaction after two weeks of using the proposed search engine revealed that the search results were significantly improved compared with conventional search engines (Google and Bing). This study can reduce the effort of search results

[*]Doctoral Dissertation, Department of Information Science, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD1261028, February 5, 2015.

sharing in a CoI and get everyone to focus on what they are searching. ASF returns more relevant results to the searchers.

Keywords:

knowledge sharing, collaborative search, search engine

Acknowledgements

This dissertation would not have been completed without the people who had supported me wholeheartedly. The same people who are still guiding me in pursuing my endeavors. Please allow me to take this opportunity to show my appreciation to everyone: First and foremost, to Professor Kenichi Matsumoto, who always cares about me during my stay in Japan, respectively. He taught me to always look at the big picture, as well as give importance to the littlest details. Everyone I have come across with in NAIST knows him as a kind and great person. I am happy to be part of his laboratory. To Professor Keiichi Yasumoto, His advise during my presentations inspired me a lot of new idea and drive on my research. I really appreciated that. To Associate Professor Akito Monden, my adviser, whom I am thankful for his endless support of my work and ideas. His vision greatly impacted my work and his invaluable comments helped me to improve on my research quality. My special thanks goes to Associate Professor Masao Ohira, his advices and ideas gave me a fresh perspective on my research. His kindness and patience has put me at ease and made me more confident in presenting my work. I would never forget his unexpected words that always bring a smile to my face. To Assistance Akinori Ihara, He is like my big brother here in Japan. He never gets tired helping me and being my role model of a hard working researcher. To Assistance Hideaki Hata, your idea has always light me up in the dark. To Associate Professor Arnon Rungsawang and Dr. Pattara Leelaprute from Kasetsart University. They always come to visit me here at NAIST, I have learn a lot from their advise. Also I would like to thank Ministry of Education, Culture, Sports, Science and Technology (MEXT) for granting me the greatest scholarship, which supported my allowance through my student life here in Japan. Last but not least, I would like to thank all members of SE Lab and Thai students in NAIST for their friendship and support. Special friendship thank goes to my little brother, Kong-kun, We spent a great time during these 5 years. Many thanks goes to Hamada-san, Nagano-San, Ohta-san, Yurie-san from International student section support. Finally, I dedicate this dissertation to my family and my girlfriend, their encouragement, understanding and trust in me reminded me that for every chapter of life. I love you all.

To my family,
for their love, endless support and encouragement

List of publications

Peer review journal paper

1. Papon Yongpisanpop, Masao Ohira, Akinori Ihara, Kenichi Matsumoto, “Adaptive Search Framework: Better Search Result for Community”, Journal of The Infosocionomics Society, Vol.8, No.2, 2014 *(related to Chapter 3, 5, 7, 8)*

Peer review international conference

1. Papon Yongpisanpop, Passakorn Phannachitta, Masao Ohira, and Kenichi Matsumoto, “An Adaptive Search Framework for Supporting cooperative work in organization,” In Proceedings of the 10th Asia Pacific Conference on Computer Human Interaction (APCHI2012), August 2012. (Matsue, Japan) *(related to Chapter 5, 6)*

2. Papon Yongpisanpop, Masao Ohira, and Kenichi Matsumoto, “Adaptive Search Engine for Organization Using Crowdsourcing Techniques,” In Proceedings of The 2011 International Computer Science and Engineering Conference (ICSEC2011) September 2011. (Bangkok, Thailand) *(related to Chapter 4)*

3. Papon Yongpisanpop, Masao Ohira, and Kenichi Matsumoto, “Community Search: A Collaborative Web Searching with A User Ranking System,” In Proceedings of the 14th International Conference on Human-Computer Interaction (HCII2011), July 2011. (Orlando, Florida, USA) *(related to Chapter 1, 2)*

National Conference Paper and Workshop

1. Papon Yongpisanpop, Passakorn Phannachitta, Masao Ohira, and Kenichi Matsumoto, “Adapting Search Engine for Organization using Adaptive Search Framework”, IPSJ Groupware and Network Services (SIGGN2011), (Aichi, Japan) *(related to Chapter 6)*

Contents

List of Figures

List of Tables

Chapter 1

Introduction

We are often involved with more than one organization in these days [2], for example, education, employment, family, or recreation community. With today's technology, people are able to access large amount of information from electronic resources such as websites, social networks, and webboards [3, 4]. Since the information in WWW is growing, it has become increasingly difficult for users to find information related to the topics in the context of their community. For example a user in the software company is new to a one of the java component called "JavaBeans". He decide to google for the information using "java bean" as a keyword. The results that he/she expects is the context of java programming language. Unfortunately the results are mixed up with java programming and java coffee. People hit the search engines every day. According to the statistics [5], there are more than five thousands billion searches per day using Google in 2013. In many communities and organizations, employees use search engines to learn how to accomplish tasks, solve problems and gain information. There are many conventional search engines available, such as Google, Bing, or Yahoo. These major search engines return search results based on relevance scores reflecting the popularity of the results with the majority of people in the world. However, using only conventional search engines inside a specific type of community will not always give people the results relevant to the topics inside communities [6]. People are able to exchange information and ideas virtually [7] when performing search tasks inside a community. The results obtained can be valuable and reusable to others in the organization, but the organization lacks proper tools to facilitate the ability for people to share their results. Moreover, individual search results can extract insight and knowledge and reuse them to return more relevant search results to other people in the organization.

In 2006 Google launched the service called “Custom Search Engine” (formerly known as Google Custom Search[1]). It is a platform that allows web developers to feature specialized information in web searches, refine and categorize queries and create customized search engines, boxed on Google Search. This allows the user to tailor the search engine to the interests of specific users, taking into account the context and purpose of the search. For example, when a car salesman searches for “lotus” on Google search, there are many results about “lotus flowers” and “IBM lotus software”. The generic Google search does not limit the context of “lotus” to a brand of a car. Google Custom Search, on the other hand, could search only preselected websites about cars, providing more relevant results to the car salesman. However, Google Custom Search does not provide a collaborative search for users, nor are the results adapted to the interests of specific users in community and organization. The results are still based on popularity measures produced by the majority of people in the world.

In 2004, Digg[2] launched a news aggregator service, which is a website that allows people to discover and share contents from anywhere, with members of the community “voting” for materials. The website provides tools for the members of the community to discover contents, discuss topics, and connect with people with similar interests. Digg builds lists of popular contents from across the web. However, as with Google custom search, Digg uses the score from people’s votes to rank the contents which mean that searcher will obtain search results based on a majority score of people in the world.

On the other hand, Bing[3], which is a web search engine developed by Microsoft, announced its newest feature called “Adaptive Search” in September 2011. As explained by Adrian Cook[4], the concept of Adaptive Search is that “Every time you search on Bing, the information provided helps Bing understand what you are trying to do. The more you search, the more Bing can learn and

[1]https://www.google.jp/cse/

[2]http://digg.com/

[3]http://www.bing.com/

[4]http://blogs.bing.com/search/2011/09/14/adapting-search-to-you/

use that information to adapt the experience so that you can spend less time searching and accomplish what you set out to do." With Bing, search results for each individual user are personalized based on data collected during previous uses of the search engine. This data is used to determine the individual context of search queries and provide more personalized results. Bing still caters to individual. It learns from the data collected from users, predicts the interest of each user from that information, and returns search results related to the topics of interest to that user. Similar to Google Custom Search, Bing does not provide a collaborative web search feature for users right now, but Bing is a little ahead of Google. It uses collaborative search behavior to rank results, which means that Bing makes use of your click behavior while delivering search results for other users searching on the same keyword.

Another search engine tool called "Eurekster"[5] allows users to custom search portal and harness the knowledge, passion and behavior of online communities to improve the search experiences, while creating online assets for web publishers and enterprisers. Eurekster launched a personalized social search in 2004 before Google and Yahoo, which is an important next step for increasing relevancy. Users can build and customize their search portal on any topics, and share and distribute the social search to grow a community of interested. What makes Google, Bing and Yahoo different from Eurekster is Eurekster twisted concept of personalization to social, which is to provide personalized results based not on who you are but who you know.

1.1. Motivation

Many researches created algorithms and tools to solve irrelevant search results, but most of them only focused on personal information-seeking tasks rather than the community. For example, Teevan et al. [8] has been proposed using groups to improve personalised search, but they just focused on using group profiles to improve search rather than using people-search activities.

There are also many tools such as SearchTogether [9], Coagmento [10], CollabSearch [11], and Results Space [12], which allow people to find, save and share

[5]http://www.eurekster.com/

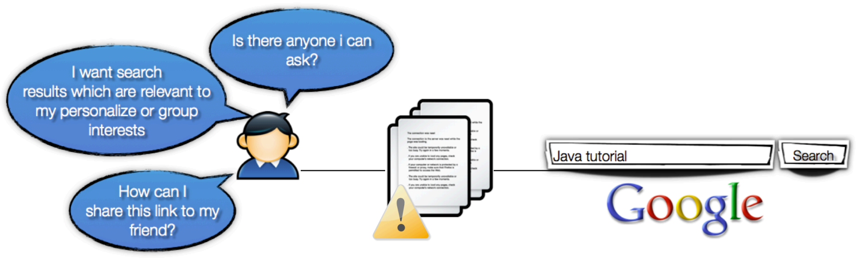

Is there anyone i can ask?
I want search results which are relevant to my personalize or group interests
How can I share this link to my friend?
Java tutorial
Search
Google

Figure 1.1. The problems found when user performs a search task.

documents and see the activities of others in the collaboration group.

To share search results among the members is sometime that will require change in people's behaviour. It means that people need to put some effort into it. Two researches [7, 13] have been studied to encourage people to share their knowledge in the communities. Cabrerab et al. [14] shows that sharing knowledge among community members requires a lot of efforts. A deep understanding of the organization culture is also a major challenge if you want to get some of that knowledge. It is very important to have a full understanding of individual and organizational culture. Another problem remains: who should introduce and drive knowledge flow management. This led us to consider what if there was an additional search engine layer that could learn from what users have searched before and adapt itself to return the top-ranked results that are more relevant to topics in the organization without modifying the conventional search engine itself.

Three hypothesizes of the research problem with using search engines in the community of interest are:

- Community does not have a proper tool to convert individual search results to community knowledge.
- Results from conventional search engines do not match with the topics inside the community.
- Sharing search results while collaborating requires a lot of effort.

To help address these problems, search engines could use information collected when users perform searches to choose and return results related to topics within a user's organization, which should allow users to spend less time searching. To do this, I decided to develop a framework implemented on top of normal search engines.

1.2. Contributions

This research has revealed what is missing in collaborative search and how to improve the quality of the search results in order to make the return results more relevant to Col. One of the most important problem is that people. We are filling the gap of what is missing in collaborative search by using knowledge management and information retrieval research areas together. We have indicated the gap of knowledge and the limitation of exiting collaboration search tools. The Search Activity Knowledge Extraction (SAKE) model is proposed. The SAKE model is derived from Knowledge Asset Management model (KAM), which is widely used to deal with the knowledge asset inside the organization. The concept of SAKE model is "People Process and Technology make Knowledge Flow". Web 2.0 and Information retrieval technique are used as a technology to fulfill the goal of SAKE model. To prove the effectiveness of the proposed model, The tool which applied SAKE model called Adaptive Search Framework (ASF) is developed. The ASF with SAKE model has been proved that it can help people in community to do the collaborative search and make search results sharing effortless. Applying knowledge sharing technique can help the community sustain the knowledge assets, which can be reuse in order to make the advantage for both community and users.

1.3. Thesis Layout

This dissertation is structured into four parts. The first part is knowledge management and sharing. This part consists of two chapters. In the first chapter, brief backgrounds of community-driven knowledge base and Knowledge sharing through search activity is introduced. Part two has two chapters and presents the evolution of information retreival of search engine and social ranking algorithm. Chapter 4 discusses the existing search engines and tools. In chapter 5, the investigation of related social ranking algorithm and the proposed algorithm called "SAKERank" are discussed and revealed. Part three is about design and implementation of Adaptive Search Framework (ASF). Chapter 6 introduces the architecture, user interface, data flow, and database. Chapter 7 shows the experimental design & procedure and evaluation measurements that are used to evaluate the ASF. Chapter 8 presents the results of performance and usability of the ASF. The performance results show the degree of overlapping search results from Google, Bing, ASF with ideal result from the experts using *OSim* and the ranking similarity by using *KSim*. For the usability, Participants' experience is observed and questioned. In the system usability scale (SUS) consists of three parts which are information seeking, knowledge sharing & management, and tool usage. The dissertation finally concludes with a discussion and summary of contributions and an outlook into the future.

Part I

Knowledge Management & Sharing

Chapter 2

Community-driven Knowledge Base

“Knowledge can be best understood as a social phenomenon, and efforts to work with it are better structured as some group effort than by individuals”

As mentioned in Enterprise Knowledge Management [15] in 1998, knowledge management entails formally managing knowledge resources in order to facilitate access and reuse of knowledge, typically by using advanced information technology. KM is formal in that knowledge is classified and categorized according to a pre-specified but evolving ontology into structured and semistructured data and knowledge bases. The overriding purpose of enterprise KM is to make knowledge accessible and reusable to the organization. Knowledge resources vary for particular industries and applications, but they generally include manuals, letters, summaries of responses to clients, news, customer information, and knowledge derived from work processes. A wide range of technologies are being used to implement KM systems: e-mail; databases and data warehouses; group support systems; browsers and search engines; intranets and internets; expert and knowledge-based systems; and intelligent agents.

As organizations store an increasing amount of information and knowledge in data and knowledge warehouses and in data and knowledge bases, they are attempting to manage that knowledge in more efficient ways. Historically, organizational knowledge has been stored on paper and in people's minds. Unfortunately, paper has limited accessibility and is difficult to update. And when people leave, they take most of their knowledge with them, so reuse is not always feasible. Thus, firms have moved to data and knowledge warehouses and to data and knowledge bases to improve accessibility, updatability, and archivability of data and knowledge.

2.1. Community of Interest (CoI)

A community of interest is a group of people who share a common interest or passion. These members exchange ideas and thoughts about the given passion, but may know little about each other outside of this area. Participation in a community of interest can be compelling, entertaining and create a sticky community where people return frequently and remain for extended periods.

In other words, Community of interest is a gathering of people assembled around a topic of common interest. Its members take part in the community to exchange information, to obtain answers to personal questions or problems, to improve their understanding of a subject.

In this research, I have developed a web search and result sharing environment, which will be described in Chapter 6 for community of interest based on these characteristics:

- A group of people interested in sharing information and discussing a particular topic that interests them.
- Members are not necessarily experts or practitioners of the topic around which the CoI has formed.
- The purpose of the CoI is to provide a place where people who share a common interest can go and exchange information, ask questions, and express their opinions about the topic.
- Membership in a CoI is not dependent upon expertise - one only needs to be interested in the subject.

2.2. The Evolution of Knowledge Mangement

According to Larry Prusak's foreword message in Mastering Organisational knowledge Flow [16] many organizations. Several organisations realized that it was actually possible to do something about knowledge in their organizations. The "something" that could be done was often discussed among knowledge practitioners. However the idea could be summarised in this way:

- Knowledge in organizations is most likely to be found in existing or emergent document.
- The key to managing these documents are better systems - either technology systems of cleverer taxonomies.
- Incentives can easily be developed to encourage the production and use of these document.
- All of these activities can be measured for their effectiveness within the organization and their costs can be justified this way.
- Knowledge was the result of individual action and thinking and the individual is the most efficient unit of analysis for work with knowledge in the firm.
- Knowledge management projects had a very strong technological component.

These general idea were termed *Knowledge Management (KM)* by many people, and by 1995 these ideas had taken hold and much effort and expenditures were being burned up in putting them into practice. Ideas have consequences and these surely did as knowledge practitioners, consultants, and technology vendors all jumped on the KM world to implement these systems.

Unfortunately, the ideas were flawed. They were not so much wrong as misguided in their approach. Since almost all new movements build on the skeletons of earlier movements, KM looked very much like information management, and, not surprisingly, the results produced by these new KM projects diappointing the knowledge advocates and especially the users and clients who were expecting great thing from the more effective use of knowledge within the organization.

However, rather than admitting defeat and leaving the field, practitioners rethought many of their assumptions and came up, again with the help of consultants and academics, with some new working hypotheses that seemed much closer to the reality of how knowledge actually works in organizations.

Needless to say, this was not the case for all KM projects. Many stuck with the old models and some still do. But it can be fairly said that these retro efforts almost all became absorbed into more traditional information technology projects and lost their focus and user enthusiasm. They are still fading from sight.

Here are the new assumptions of KM

- Knowledge can be best understood as a social phenomenon, and efforts to work with it are better structured as some group effort than by individuals.
- Working with knowledge needs some mixture or combination of technology, strategy, human capital, and social capita approaches.
- It is almost impossible to effectively measure knowledge, and it is not worth the effort to do so.
- A holistic approach is therefore called for difficult as this may be to formulate and implement.

2.3. What is missing in Collaborative Search?

My previous research [17] indicated the gap of knowledge and possible limitation of existing custom search engines (e.g., Google, Digg, Bing, and Eurekster), which most likely to be used for collaboratively while searchers perform search tasks together. In this section will be explained about the limitation of the tools and missing features for community search.

Usually, there are a lot of information flowing around when people are performing search tasks, but at the same time many of them are "information rich and knowledge poor" [18]. Morris et al. [19] discussed why people don't share and use collaborate tools. One of the reasons is that people just don't seem willing to move from a standard search tool to another tool for a collaborative search. Tools dedicated to collaborative information-seeking have not enjoyed widespread success to date [19, 20].

However if people want to share their search results among community members, they need to put some efforts toward that. The study by Michale et al. [21] revealed that when people are searching in pairs and in larger groups, they are doing it without the benefit of specialized search tools. Instead, they are using out-of-channel-of-communication tools, such as email, texting, phone calls, and social media communication [19, 20, 22].

The question is what's missing from a collaborative search? The first feature in the next generation of collaborative search should be focused on offering individuals the ability to define their personal constraints or preferences as they search. The research by Fisher et al. [23] has shown that people can build on simple representations that already created by others, so if a search tool can recognize that, it could make more intelligent suggestions.

In order to collaborate, the second feature is the search tool should be able to recognize what has been viewed by anyone in the group of searchers. It should rerank based on this information, for example Querium [24] has made searchers' explicit ratings visible, but the evaluations of this tool are done in laboratories in which the environment and variable are fixed.

The last key feature, and the most important one, is to help searchers discover new and unknown information. The tool should be able to compare what has already been added to a community and deliver new results which are relevant to the searchers' and communities' topics.

As far as I know, there is no simple, high-usability tools, which concern about these key features available for community and the part that automated knowledge extraction from search results has not been studied yet. So these questions remain open.

Chapter 3

Knowledge Sharing Through Search Activity

"KM involves individuals making knowledge requests of experts associated with a particular subject on an ad hoc basis"

3.1. Knowledge Asset Management model (KAM)

The purpose of this research [25] is to suggest a model for knowledge asset management as a foundation for correlating human capital management to organizational performance. This model suggests moving people to the center of an asset management model to correlate their value to an organization's performance. A systems engineering approach is suggested as a model where inputs develop the knowledge asset, mechanisms inventory the asset and outputs utilize the asset In the knowledge economy, people are an organization's most valuable asset, yet a methodology that correlates their value to organizational performance. The model is lacking The goal is to propose a knowledge asset model for human capital management. Future research will be performed to substantiate the model and develop a measurement instrument to investigate the correlation between knowledge asset management and organizational performance.

3.2. Search Activity Knowledge Extraction Model (SAKE)

To extract the knowledge from search behavior and share it through the community is one of the goal in this research. I created a model called Search Activity Knowledge Extraction (SAKE), which is shown in Figure 3.1.

SAKE is a mixture of individual capital, technology, strategy, and social capital approaches [25], which were derived from the KAM model ,which mentioned in the Section 3.1, SAKE model starts with the individual, which represents searcher inside CoI, a searcher searches for the information from outside resources using web search engine. After the searcher had obtained the results, the results and searcher interaction are sent to extract knowledge in technology and strategy parts. In technology and strategy part, search results are clustered to the topics ,which are related to the community of interest and reranks based on the popularity inside the community. Searher's search behavior are also considered as a varible to to rerank the search results Once the information has been processed, it is ready to be shared throughout the community.

I designed the process of the usage implement SAKE model as shown in Figure 3.2. In the community, a searcher usually uses a search engine to search information. The searcher passes the keyword to the search engine to obtain the results. After the searcher clicks, looks at the results or maybe bookmarks them, those search results remain as individual information. The aim of the process is to change the nature of the individual search results into public knowledge. In the technology part, it analyzes search results using search-result clustering to cluster them to our pre-defined topics. Pre-defined topics are the specific topics, which relate to the community. I have to defines the topics ahead of time in order to make it easy to classify our search results. Then the results is ranked using our re-ranking algorithm to enrich search results. When other searchers inside the community perform a search, relating to the previous search and topic, the process will be able to return a more relevant search. This gets more direct results than conventional search engines.

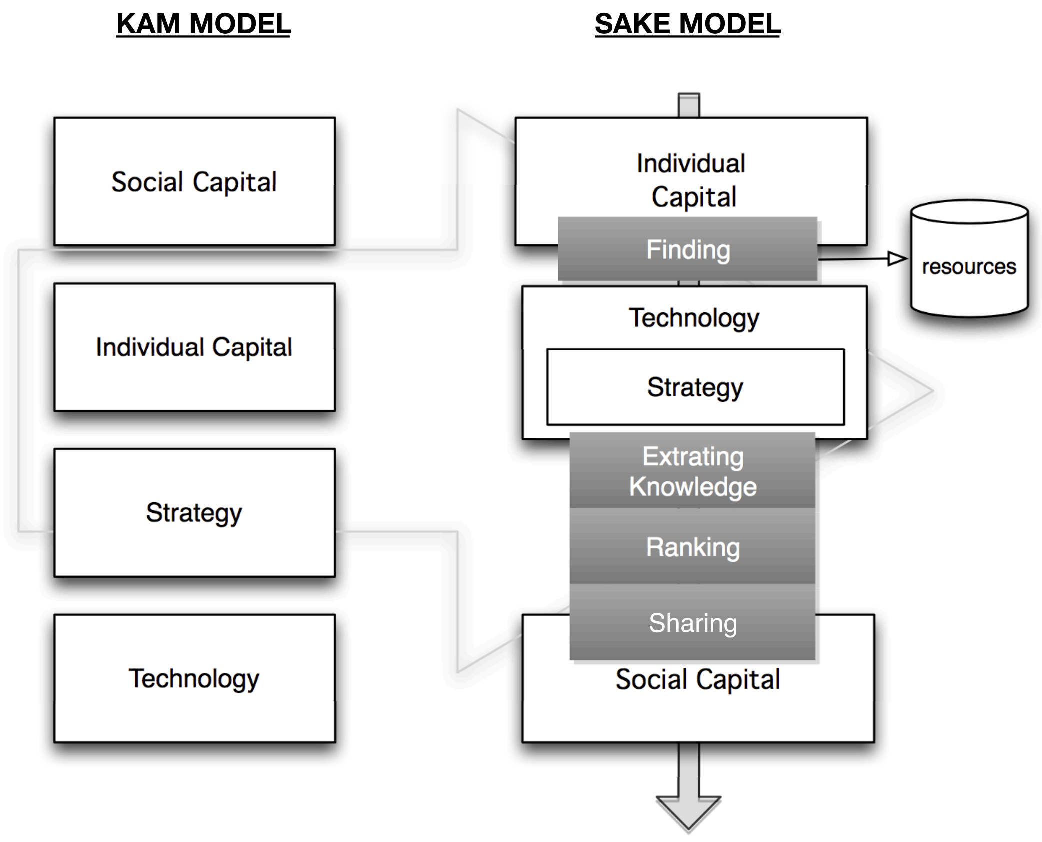


Figure 3.1. On the left is Knowledge Asset Management model (KAM). To manage the knowledge asset four attributes, which are Social capital, Individual capital, Strategy, and Technology are involved. On the right is Search Activity Knowledge Extraction model (SAKE). SAKE model derived the attributes from KAM and added actions between each attribute in order to extract the knowledge from search behavior.

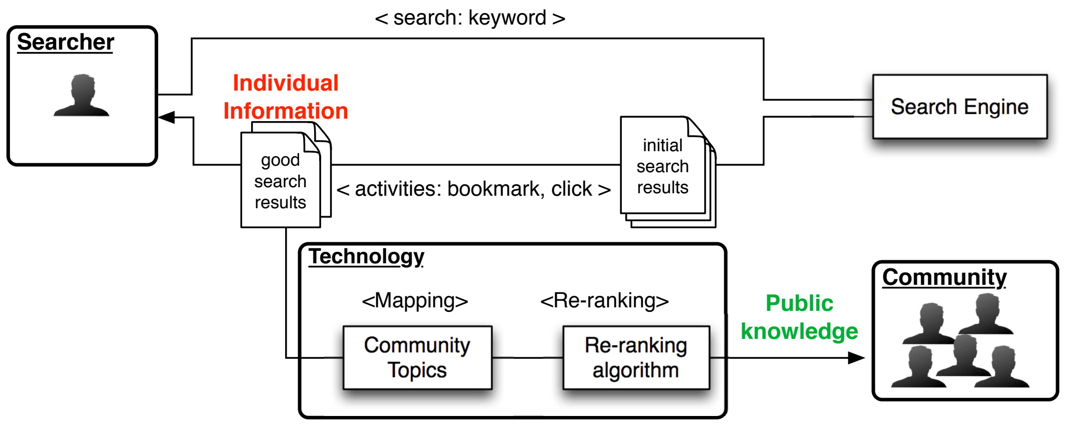


Figure 3.2. The process of the SAKE model. The process converts individual information to public knowledge in community.

Part II

Search Engine & Ranking Algorithm Evolution

Chapter 4

Evolution Information Retreival of Search Engine

“Every search engine wants to be more relevant. However, what is relevant to you might not be relevant to someone else.”

4.1. Web Search Engine

A web search engine is designed to search for information in World Wide Web. The search results that come from a search engine are generally presented in a list of results often referred to as SERPS, or “search engine results pages”. The information may consist of web pages, images, information and other types of files. Some search engines also mine data available in databases or open directories. Unlike web directories, which are maintained only by human editors, search engines also maintain real-time information by running an algorithm on a web crawler. In late 1990, Search engines were known as some of the brightest stars in the internet investing frenzy. A lot of companies entered the market spectacularly, Some of them have taken down their public search engine. Around 2000, Google’s search engine innovated PageRank and achieved better results for many searches. This iterative algorithm ranks web pages based on the number and PageRank of other web sites and pages that link there, on the premise that good or desirable pages are linked to more than others. Google also maintained a minimalist interface to its search engine. In contrast, many of its competitors embedded a search engine in a web portal.

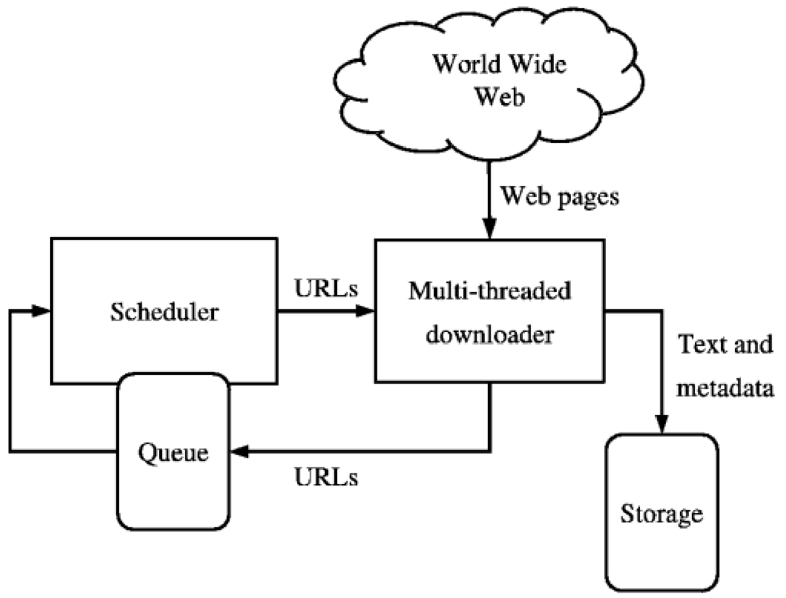


Figure 4.1. High-level architecture of a standard web crawler. based on image from “Effective Web Crawling” PhD. Thesis of Carlos Castillo [1]

A web search engine operates as follows:

- Web crawling
- Indexing
- Searching

Web search engines work by storing webpages’ information, which they retrieve from the HTML markup of the pages. These pages are retrieved by a web crawler (sometimes also known as a spider) an automated web crawler which follows every link on the site. Web crawling and indexing are performed alternately in a cycle as shown in Figure 4.1. At the beginning of a cycle, a web crawler retrieves all the webpages contents and stores them as files in a proper format (i.e. Stanford WebBase format). Next, each webpage is parsed into a plain text format and sent to an indexer to be analyzed. The web indexer then extracts each term in the page and adds the information to an index database. For example, the indexer extracts terms from the titles, headings, or special fields called meta tags. The purpose of indexing is to allow information to be looked up as quickly as possible. The cycle ends here, with the index database serving as a snapshot of the whole webpage set for the users queries. The web crawler then starts the operation again for the next cycle, and the index database will be updated again at the end of the cycle. The search engine then analyzes the contents of each page to determine how it should be indexed (for example, words can be extracted from the titles, page content, headings, or special fields called meta tags). Data about webpages are stored in an index database for use in later queries. A query from a user can be a single word. The index helps finding information relating to the query as quickly as possible [26]. When a user enters a query into a search engine (typically by using keywords), the engine examines its index and provides a listing of best-matching webpages according to its criteria, usually with a short summary containing the document's title and sometimes parts of the text. The index is built from the information stored with the data and the method by which the information is indexed [26].

The usefulness of a search engine depends on the relevance of the result set it gives back. While there may be millions of webpages that include a particular word or phrase, some pages may be more relevant, popular, or authoritative than others. Most search engines employ methods to rank the results to provide the "best" results first. How a search engine decides which pages are the best matches, and what order the results should be shown in, varies widely from one engine to another [26]. The methods also change over time as Internet usage changes and new techniques evolve. There are two main types of search engine that have evolved: one is a system of predefined and hierarchically ordered keywords that humans have programmed extensively. The other is a system that generates an "inverted index" by analyzing texts it locates. This first form relies much more heavily on the computer itself to do the bulk of the work.

Most Web search engines are commercial ventures supported by advertising revenue and thus some of them allow advertisers to have their listings ranked higher in search results for a fee. Search engines that do not accept money for their search results make money by running search related ads alongside the regular search engine results. The search engines make money every time someone clicks on one of these ads.

4.2. Taxonomy vs Folksonomy

With the need for managing and organization information in a structured manner was need of the hour in the information age. Taxonomy came in as one of the most common way to organize and structure content. Taxonomy is the practice and science of classification. The word is also used as a count noun: a taxonomy, or taxonomic scheme, is a particular classification. Originally taxonomy referred only to the classifying of organisms or a particular classification of organisms. [27] In a wider, more general sense, it may refer to a classification of things or concepts, as well as to the principles underlying such a classification. Taxonomy was used by web designers to organize the content in their web sites. And with information constantly getting appended to their site, taxonomy seemed to be a good approach for content management. But all was not well with this approach, soon the problems surfaced and web architects were looking for other alternatives. The primary reasons were:

- Taxonomy worked well with classification of living organisms, as it was about dealing with the same kind of information. However the case of websites was different. Every web site had different content so there was no particular classification scheme possible for the web sites and it called for unique classification schemes for different websites.
- It is difficult for the web architects to implement and maintain taxonomy.
- With existing sites which already have huge data, it was more difficult for the designers to implement taxonomy.
- The language or code used for taxonomy was not as user friendly as a system like this needs to be. This led to user dissatisfaction and led to user being averse of using taxonomy.

Folksonomy is the evolution of taxonomy. It is a way of describing crowd-driven categorisation of information. This practice is also known as collaborative tagging [28], social classification, social indexing, and social tagging. A broad folksonomy is the one in which multiple users tag particular content with a variety of terms from a variety of vocabularies, thus creating a greater amount of metadata for that content. A narrow folksonomy, on the other hand, occurs when a few users, primarily the content creator, tag an object with a limited number of terms. While both broad and narrow folksonomies enable the searchability of content by adding textual description - or access points - to an object, a narrow folksonomy does not have the same benefits as a broad folksonomy, which allows for the tracking of emerging trends in tag usage and developing vocabularies [29]. Folksonomies became popular on the Web around 2004 as part of social software applications such as social bookmarking and photograph annotation. Tagging, which is one of the defining characteristics of Web 2.0 services, allows users to collectively classify and find information. Some websites include tag clouds as a way to visualize tags in a folksonomy. However, tag clouds visualize only the vocabulary but not the structure of folksonomies, as do tag graphs.

Areas where Folksonomy is better with respect to Taxonomy are:

- It reduces the support and maintenance effort. This is because the folksonomies are organized and maintained by the user. So the burden on the web architects is reduced.

- Users can tag the information in a way which is easy for him to recollect and use later. So it leads to a lot of satisfaction. That is you can say that there is no confusion, the words used by the users are the words in the site.

4.3. Search results clustering

Annotation and clustering of search results are key parts of the solution proposed here. Clustering of search results groups webpages from the search results into categories. Some keywords return highly varied results. For example, the keyword “Apache” can return a set of links to the Apache tribe, Apache helicopter, Apache software foundation, and other types of Apache. Grouping these results into categories makes it easier for users to find the webpages they desire. There are several search result clustering tools, such as Apache Carrot^2, Vivisimo, and IBM Mapuccino. In this research, Apache Carrot^2, which is an open source library augmented with a set of supporting applications is used to build a search results clustering engine simply, without any limitation on the number of uses. Such clustering engines can organize a set of search results into topics, completely automatically and without external information such as taxonomies or pre-classified content. Since Apache Carrot^2 is a clustering engine designed for online use, clustering search results needed only URL, title, and snippet fields. However, this same simplicity may indicate a lack of in-depth contents, which may not achieve outstanding accuracy in the clustering result.

Carrot^2 is an Open Source Search Results Clustering Engine that I use in our framework. It can automatically organize small collections of documents (search results but not only) into thematic categories. Apart from two specialized document clustering algorithms, Carrot^2 offers ready-to-use components for fetching search results from various sources including GoogleAPI, Bing API, eTools Meta Search, Lucene, SOLR, and more. Carrot^2 is implemented in Java. Other non-Java platforms, such as PHP or Ruby, can call Carrot^2 clustering through its REST interface. Carrot^2 has 3 different algorithm (Lingo, ETC, Lingo3G) provided to use. Each of them has their own unique significant. I decided to use the default Carrot^2's algorithm which is Lingo because it cluster and give standard result sets. The other two algorithm are also great but one of them is a commercial and the other one give a different type of result set.

4.4. Social Bookmarking

A social bookmarking service is a centralized online service, which enables users to add, annotate, edit, and share bookmarks of web documents [30]. In a social bookmarking system, users save links to webpages that they want to remember and/or share. These bookmarks are usually public, and can be saved privately, shared only with specified people or groups, shared only inside certain networks, or another combination of public and private domains. The allowed people can usually view these bookmarks chronologically, by category or tags, or via a search engine. Most social bookmark services encourage users to organize their bookmarks with informal tags instead of the traditional browser-based system of folders, although some services feature categories/folders or a combination of folders and tags. They also enable viewing bookmarks associated with a chosen tag, and include information about the number of users who have bookmarked them. Some social bookmarking services also draw inferences from the relationship of tags to create clusters of tags or bookmarks. Many social bookmarking services provide web feeds for their lists of bookmarks, including lists organized by tags. This allows subscribers to become aware of new bookmarks as they are saved, shared, and tagged by other users. It also helps to promote your sites by networking with other social book markers and collaborating with each other. As these services have matured and grown more popular, they have added extra features such as ratings and comments on bookmarks, the ability to import and export bookmarks from browsers, emailing of bookmarks, web annotation, and groups or other social network features.

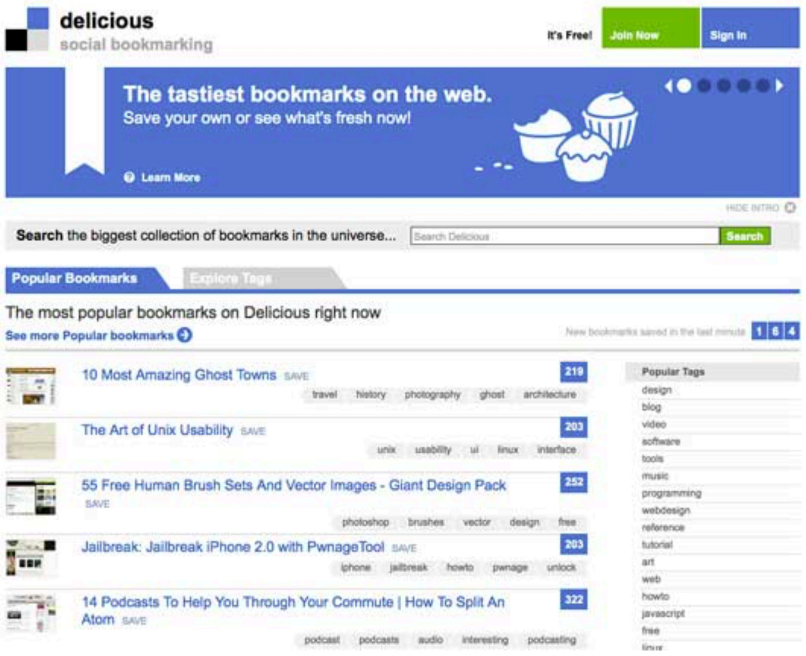
In this section three popular social bookmarkings are picked and will be described in details and their features in order to make the reader has a clear understand about what is social bookmarking. The social bookmarks have a very similar idea to this research in order to solve the problem of irrelevant results from conventional search engines and would like to optimize search results for a specific group.

4.4.1 Del.icio.us

Del.icio.us is a social bookmarking web service for storing, sharing, and discovering web bookmarks. Del.icio.us uses a non-hierarchical classification system in which users can tag each of their bookmarks with freely chosen index terms (generating a kind of folksonomy). Del.icio.us allowed users to group links with similar topics together to form a “Stack”, and include title and descriptions for the Stack page. Stacks could be worked on collaboratively with other users, and could be followed and shared with other users. Stacks were added in September 2011 and removed in July 2012. Del.icio.us has a“hotlist” on its home page and “recent” pages, which help to make the website a conveyor of Internet memes and trends. Users can also explore stacks on the home page by navigating categories.

Table 4.1. List of popular social bookmarking websites that are still active.

Delicious	The site was bought by Avos Systems on April 27, 2011 though was operated by Yahoo! until July 2011.	delicious.com
StumbleUpon	StumbleUpon is a discovery engine that finds and recommends web content to its users.	stumbleupon.com
Digg	Digg is a news aggregator with an editorially driven front page	digg.com
BibSonomy	BibSonomy is a system for sharing bookmarks and lists of literature.	bibsonomy.org
CiteULike	CiteULike is a web service which allows users to save and share citations to academic papers.	citeulike.org
Diigo	Diigo is a social bookmarking website which allows signed-up users to bookmark and tag web-pages. Additionally, it allows users to highlight any part of a webpage and attach sticky notes to specific highlights or to a whole page	
Pearltrees	Collaborative bookmark exploration and curation tool organized and presented like a mind map.	pearltrees.com
Pinterest	Pinterest is a web and mobile application company that offers a visual discovery, collection, sharing, and storage tool.	pinterest.com
Reddit	Users submit content in the form of either a link or a text ("self") post. Links and content can be voted on.	reddit.com
We Heart It	We Heart It is an image-based social network for inspiring images	weheartit.com
ZEEF	A curated directory, they filters the world's information with human intelligence, multiple curators can curate their own subject.	zeef.com


delicious
social bookmarking
It's Free! Join Now Sign In
The tastiest bookmarks on the web.
Save your own or see what's fresh now!
Learn More
HIDE INTRO
Search the biggest collection of bookmarks in the universe... Search Delicious Search
Popular Bookmarks | Explore Tags
The most popular bookmarks on Delicious right now
See more Popular bookmarks
New bookmarks saved in the last minute 1 6 4

10 Most Amazing Ghost Towns SAVE 219
travel history photography ghost architecture

The Art of Unix Usability SAVE 203
unix usability ui linux interface

55 Free Human Brush Sets And Vector Images - Giant Design Pack SAVE 252
photoshop brushes vector design free

Jailbreak: Jailbreak iPhone 2.0 with PwnageTool SAVE 203
iphone jailbreak howto pwnage unlock

14 Podcasts To Help You Through Your Commute | How To Split An Atom SAVE 322
podcast podcasts audio interesting podcasting

Popular Tags
design
blog
video
software
tools
music
programming
webdesign
reference
tutorial
art
web
howto
javascript
free
linux

Figure 4.2. The delicious user interface

To facilitate newcomers, Delicious provides an option to import bookmarks from the web browsers to its site so that new users can quickly get started with the site. Del.icio.us is one of the most popular social bookmarking services. Many features have contributed to this, including the website's simple interface, human-readable URL scheme, a novel domain name, a simple REST-like API, and RSS feeds for web syndication. Use of Del.icio.us is free. The source code of the site is not available, but a user can download his or her own data through the site's API in an XML or JSON format, or export it to a standard Netscape bookmarks format. All bookmarks posted to Del.icio.us are publicly viewable by default, although users can mark specific bookmarks as private, and imported bookmarks are private by default. The public aspect is emphasized; the site is not focused on storing private bookmark collections. Del.icio.us linkrolls, tagrolls, network badges, RSS feeds, and the site's daily blog posting feature can be used to display bookmarks on weblogs.

4.4.2 StumbleUpon

StumbleUpon is a discovery engine (a form of web search engine) that finds and recommends web content to its users. Its features allow users to discover and rate Web pages, photos, and videos that are personalized to their tastes and interests using peer-sourcing and social-networking principles.

StumbleUpon uses collaborative filtering (an automated process combining human opinions with machine learning of personal preference) to create virtual communities of like-minded Web surfers. Rating Web sites update a personal profile and generate peer networks of Web surfers linked by common interest. These social networks coordinate the distribution of Web content, so that users "stumble upon" pages explicitly recommended by friends and peers. Giving a site a thumbs up results in the site being placed under the user's "favorites". Furthermore, users have the ability to stumble their personal interests like "History" or "Games".

Users rate a site by giving it a thumbs up, thumbs down selection on the StumbleUpon toolbar, and can optionally leave additional commentary on the site's review page, which also appears on the user's blog. This social content discovery approach automates the "word-of-mouth" referral of peer-approved Web sites and simplifies Web navigation.

In the settings section of Stumbleupon you can further filter the types of webpages you may come across. There are interest filters which allow you to include only content for all ages, R rated content, or X rated content. There are also content filters in which you can choose to allow stumbles with audio, video, flash, and images.

4.4.3 Digg.com

Digg is a social news website that allows people to discover and share content from anywhere, with members of the community "voting" for material. The site's main function is to let users discover, share and recommend web content. Members of the community can submit a webpage for general consideration. Other members can vote that page up ("digg") or down ("bury"). Although voting takes place on digg.com, many websites add "digg" buttons to their pages, allowing users to

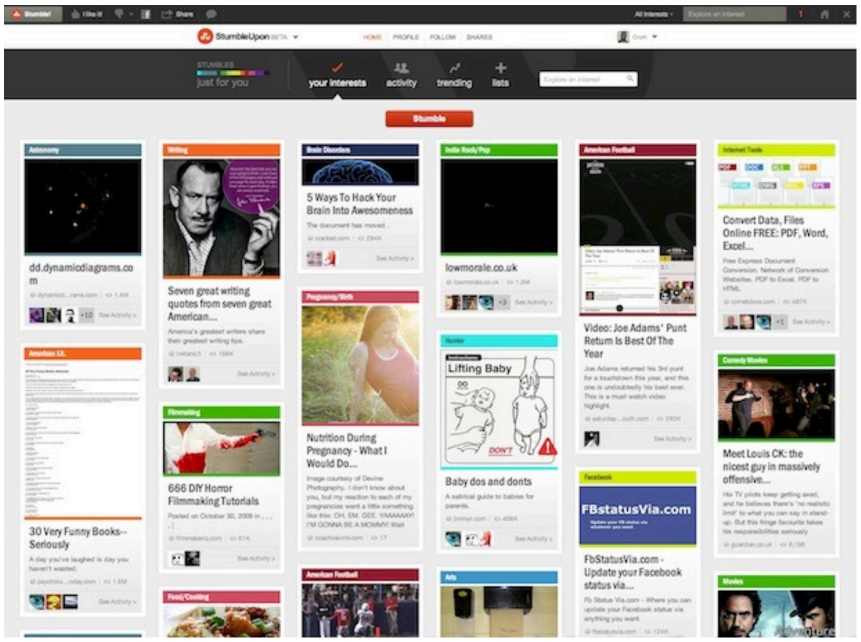


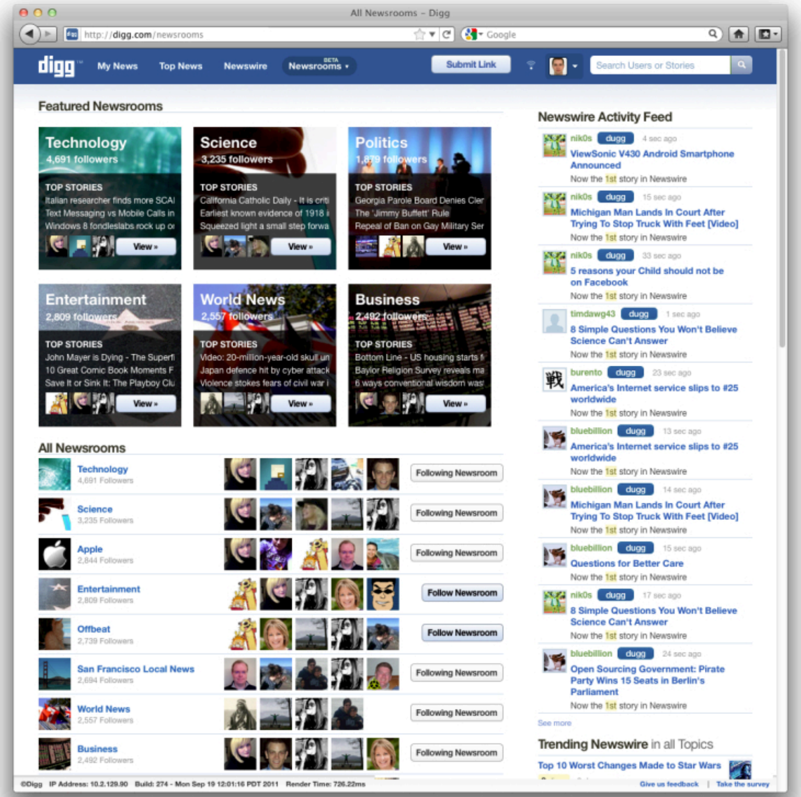
Figure 4.3. The stumbleUpon homepage

vote as they browse the web. The end product is a series of wide-ranging, constantly updated lists of popular and trending content from around the Internet, aggregated by a social network. The website, which shown in Figure 4.4 provides tools for members of the community to discover content, discuss topics, and connect with people with similar interests. Digg builds lists of popular content from across the web.

4.5. Enterprise bookmarking

Enterprise bookmarking is derived from Social bookmarking that got its modern start with the launch of the web site del.icio.us in 2003. The idea is to encourage Enterprise 2.0 users to tag, organize, store, and search bookmarks of both webpages on the Internet and data resources stored in a distributed database or fileserver. This is done collectively and collaboratively in a process by which users add tag (metadata) and knowledge tags [31].

Social bookmarking vs Enterprise bookmarking In a social bookmark-



Featured Newsrooms

Technology
4,691 followers
TOP STORIES
Italian researcher finds more SCAI
Text Messaging vs Mobile Calls in
Windows 8 fondleslabs rock up o

Science
3,235 followers
TOP STORIES
California Catholic Daily - It is crit
Earliest known evidence of 1918 i
Squeezed light a small step forwa

Politics
1,879 followers
TOP STORIES
Georgia Parole Board Denies Cler
The 'Jimmy Buffett' Rule
Repeal of Ban on Gay Military Ser

Entertainment
2,809 followers
TOP STORIES
John Mayer is Dying - The Superfi
10 Great Comic Book Moments F
Save It or Sink It: The Playboy Clu

World News
2,557 followers
TOP STORIES
Video: 20-million-year-old skull un
Japan defence hit by cyber attack
Violence stokes fears of civil war i

Business
2,492 followers
TOP STORIES
Bottom Line - US housing starts f
Baylor Religion Survey reveals ma
6 ways conventional wisdom was

All Newsrooms

Technology	4,691 Followers	Following Newsroom
Science	3,235 Followers	Following Newsroom
Apple	2,844 Followers	Following Newsroom
Entertainment	2,809 Followers	Follow Newsroom
Offbeat	2,739 Followers	Follow Newsroom
San Francisco Local News	2,694 Followers	Following Newsroom
World News	2,557 Followers	Following Newsroom
Business	2,492 Followers	Following Newsroom

Newswire Activity Feed

nik0s dugg 4 sec ago
ViewSonic V430 Android Smartphone Announced
Now the 1st story in Newswire

nik0s dugg 15 sec ago
Michigan Man Lands In Court After Trying To Stop Truck With Feet [Video]
Now the 1st story in Newswire

nik0s dugg 33 sec ago
5 reasons your Child should not be on Facebook
Now the 1st story in Newswire

timdawg43 dugg 1 sec ago
8 Simple Questions You Won't Believe Science Can't Answer
Now the 1st story in Newswire

burento dugg 23 sec ago
America's Internet service slips to #25 worldwide
Now the 1st story in Newswire

bluebillion dugg 13 sec ago
America's Internet service slips to #25 worldwide
Now the 1st story in Newswire

bluebillion dugg 14 sec ago
Michigan Man Lands In Court After Trying To Stop Truck With Feet [Video]
Now the 1st story in Newswire

bluebillion dugg 15 sec ago
Questions for Better Care
Now the 1st story in Newswire

nik0s dugg 17 sec ago
8 Simple Questions You Won't Believe Science Can't Answer
Now the 1st story in Newswire

bluebillion dugg 24 sec ago
Open Sourcing Government: Pirate Party Wins 15 Seats in Berlin's Parliament
Now the 1st story in Newswire

Trending Newswire in all Topics
Top 10 Worst Changes Made to Star Wars

©Digg IP Address: 10.2.129.90 Build: 274 - Mon Sep 19 12:01:16 PDT 2011 Render Time: 726.22ms

Figure 4.4. The digg user interface

ing system, individuals create personal collections of bookmarks and share their bookmarks with others. These centrally stored collections of Internet resources can be accessed by other users to find useful resources. Often these lists are publicly accessible, so that other people with similar interests can view the links by category or by the tags themselves. Most social bookmarking sites allow users to search for bookmarks which are associated with given "tags", and rank the resources by the number of users which have bookmarked them.

Enterprise bookmarking is a method of tagging and linking any information using an expanded set of tags to capture knowledge about data. It collects and indexes these tags in a web-infrastructure knowledge base server residing behind the firewall. Users can share knowledge tags with specified people or groups, shared only inside specific networks, typically within an organization. Enterprise bookmarking is a knowledge management discipline that embraces Enterprise 2.0 methodologies to capture specific knowledge and information that organizations consider proprietary and are not shared on the public Internet.

Enterprise bookmarking provides Tag management which is the combination of uphill abilities (e.g. faceted classification, predefined tags, etc.) and downhill gardening abilities (e.g. tag renaming, moving, merging) to better manage the bottom-up folksonomy generated from user tagging.

The most significant new release was the Jumper 2.0 platform[1], which is an open source web application script for collaborative search and knowledge management powered by a shared enterprise bookmarking engine that is a fork of KnowledgebasePublisher[2]. Jumper empowers users to compile and share collaborative bookmarks by crowdsourcing their knowledge, experience and insights using knowledge tags. Users tag, link, and rate structured data and unstructured data sources, including relational databases, flat file databases, medical imaging, content management systems, and any network file system. It is an interactive, user-submitted recommendation engine which uses peer and social-networking principles to reference any information located in distributed storage devices, either inside or outside the firewall, and capture the collective knowledge about that content, media, or data. Jumper 2.0 lets you search and share high-value content, media, or data across remote locations using knowledge tags to capture knowledge about the information in distributed storage. It collects these tags in a tag profile. The tag profiles are stored in an interactive knowledge base and search engine.

4.6. Adaptive Search Engine

Most of the conventional search engines today give back search results based on the popularity of the webpages, however the concept of an Adaptive Search Engine is that of a search engine that will be able to learn from the data collected from each individual user, predict the interests of each user from that information,

[1]http://www.trilexnet.com/labs/jumper/

[2]http://sourceforge.net/projects/kbpublisher/

and return search results related to the topics of interest to that user. In other words, a user is more likely to be interested in search results related to things that the user usually searches for. An adaptive search engine puts today's search in the context of the user history of searches.

4.6.1 Google Custom Search

Google custom search allows users to create a search engine that searches only the contents of a specific website or that focuses on a particular topic. With Google custom search, users can select, prioritize, or ignore specific websites. This allows the user to tailor the search engine to the interests of specific users, taking into account the context and purpose of the search. For example, when a car salesman searches for lotus on Google search, there are many results about lotus flowers and IBM lotus software. The generic Google search does not limit the context to that of the lotus which is a brand of car. A Google custom search, on the other hand, could search only preselected websites about cars, providing more relevant results to the car salesman. However, the Google custom search engine does not provide any way for users to collaborate to perform searches, nor are the results adapted to the interests of specific users in an organization. The results are still based on popularity measures produced by the majority of users in the world.

4.6.2 Bing: Adaptive Search

Bing is a web search engine developed by Microsoft. In September 2011, Bing announced its newest feature which was called "Adaptive Search". As explained by Adrian Cook, the concept of Adaptive Search is that "Every time you search on Bing, the information provided helps Bing understand what you're trying to do. The more you search, the more Bing can learn and use that information to adapt the experience so you can spend less time searching and accomplish what you set out to do." With Bing, search results for each individual user are personalized based on data collected during previous uses of the search engine. This data is used to determine the individual context of search queries and deliver more personalized results.

Chapter 5

Social Ranking Algorithms

5.1. Backgroud

5.1.1 HITS

Hyperlink-Induced Topic Search [32] (HITS; also know as hubs and authorities) is a link analysis algorithm that rates Web pages. The idea behind Hubs and Authorities stemmed from a particular insight into the creation of web pages when the Internet was originally forming; that is, certain web pages, known as hubs, served as large directories that were not actually authoritative in the information that it held, but were used as compilations of a broad catalog of information that led users directly to other authoritative pages. In other words, a good hub represented a page that pointed to many other pages, and a good authority represented a page that was linked by many different hubs

In the HITS algorithm, the first step is to retrieve the most relevant pages to the search query. This set is called the root set and can be obtained by taking the top n pages returned by a text-based search algorithm. A base set is generated by augmenting the root set with all the web pages that are linked from it and some of the pages that link to it. The web pages in the base set and all hyperlinks among those pages form a focused subgraph. The HITS computation is performed only on this focused subgraph. According to Kleinberg the reason for constructing a base set is to ensure that most (or many) of the strongest authorities are included. Authority and hub values are defined in terms of one another in a mutual recursion. An authority value is computed as the sum of the scaled hub values that point to that page. A hub value is the sum of the scaled authority values of the pages it points to. Some implementations also consider the relevance of the linked pages.

The algorithm performs a series of iterations, each consisting of two basic steps:

Authority Update: Update each node's Authority score to be equal to the sum of the Hub Scores of each node that points to it. That is, a node is given a high authority score by being linked from pages that are recognized as Hubs for information.

Hub Update: Update each node's Hub Score to be equal to the sum of the Authority Scores of each node that it points to. That is, a node is given a high hub score by linking to nodes that are considered to be authorities on the subject.

The Hub score and Authority score for a node is calculated with the following algorithm:

- Start with each node having a hub score and authority score of 1.
- Run the Authority Update Rule
- Run the Hub Update Rule
- Normalize the values by dividing each Hub score by square root of the sum of the squares of all Hub scores, and dividing each Authority score by square root of the sum of the squares of all Authority scores.
- Repeat from the second step as necessary.

To begin the ranking, $\forall p$, $auth(p) = 1$ and $hub(p) = 1$. Two types of updates are considered: Authority Update Rule and Hub Update Rule. In order to calculate the hub/authority scores of each node, repeated iterations of the Authority Update Rule and the Hub Update Rule are applied. A k-step application of the Hub-Authority algorithm entails applying for k times first the Authority Update

Rule and then the Hub Update Rule.
Authority Update Rule $\forall p$, $auth(p)$ is updated to be the summation:

$$auth(p) = \sum_{i=1}^{n} hub(i)$$

where n is the total number of pages connected to p and i is a page connected to p. That is, the Authority score of a page is the sum of all the Hub scores of pages that point to it.
Hub Update Rule $\forall p$, $hub(p)$ is updated to be the summation:

$$hub(p) = \sum_{i=1}^{n} auth(i)$$

where n is the total number of pages p connects to and i is a page which p connects to. Thus a page's Hub score is the sum of the Authority scores of all its linking pages
Normalization The final hub-authority scores of nodes are determined after infinite repetitions of the algorithm. As directly and iteratively applying the Hub Update Rule and Authority Update Rule leads to diverging values, it is necessary to normalize the matrix after every iteration. Thus the values obtained from this process will eventually converge.

5.1.2 PageRank

PageRank is a link analysis algorithm and it assigns a numerical weighting to each element of a hyperlinked set of documents, such as the World Wide Web, with the purpose of "measuring" its relative importance within the set. The algorithm may be applied to any collection of entities with reciprocal quotations and references. The numerical weight that it assigns to any given element E is referred to as the PageRank of E and denoted by $PR(E)$. Other factors like Author Rank can contribute to the importance of an entity.

A PageRank results from a mathematical algorithm based on the webgraph, created by all World Wide Web pages as nodes and hyperlinks as edges, taking into consideration authority hubs such as cnn.com or usa.gov. The rank value indicates an importance of a particular page. A hyperlink to a page counts as a vote of support. The PageRank of a page is defined recursively and depends on the number and PageRank metric of all pages that link to it ("incoming links"). A page that is linked to by many pages with high PageRank receives a high rank itself.

Numerous academic papers concerning PageRank have been published since Page and Brin's original paper [33]. In practice, the PageRank concept may be vulnerable to manipulation. Research has been conducted into identifying falsely influenced PageRank rankings. The goal is to find an effective means of ignoring links from documents with falsely influenced PageRank [34].

Other link-based ranking algorithms for Web pages include the HITS algorithm invented by Jon Kleinberg (used by Teoma and now Ask.com) [32], the IBM CLEVER project, the TrustRank[1] algorithm and the hummingbird algorithm[2].

5.2. Related Algorithms

5.2.1 FolkRank

The FolkRank algorithm [13] operates on the folksonomy model context as tag clouds specified in folksanomy [35] which is a quadruple $F := (U, T, R, Y)$, where U , T , R are finite sets of instances of *users*, *tags*, and *resources*. Y defines a relation, *the tag assignment*, between these sets,that is,$Y \subseteq UTR$. A tag cloud can be computed for users, tags, and resources, e.g. the tag cloud for a user u can be defined as $TC_U(u) = \{\{t, w(u,t)\}|(u,t,r)Y, w(u,t) = |\{r \in R : (u,t,r) \in Y\}|\}$. Hence, the weight assigned to a tag is simply corresponds to the usage frequency of the tag. We normalize the weights so that the sum of the weights assigned to the tags in the tag cloud is equal to 1. Furthermore, we use $TC_Uk(u)$, $TC_Tk(t)$, and $TC_Rk(r)$ respectively to refer to the tag cloud that contains only the top k tags, which have the highest weight. FolkRank transforms the hypergraph that is spanned by the tag assignments into a weighted tripartite graph G_F, where an edge connects two entities (user, tag, or resource)

[1]http://en.wikipedia.org/wiki/TrustRank

[2]http://en.wikipedia.org/wiki/Google_corollaryHummingbird

if both entities occur together at a tag assignment within the folksonomy, and the weight of an edge corresponds to the amount of their co-occurrences. For example, the weight of an edge connecting a tag t and a resource r is defined as $w(t, r) = |\{u \in U : (u, t, r) \in Y\}|$ and thus corresponds to the number of users, who have annotated r with t. The constructed graph GF serves as input for an adaption of the Personalized PageRank [33]: $\overrightarrow{w} \leftarrow dAG_F\overrightarrow{w} + (1d)\overrightarrow{p}$, where the adjacency matrix AG_F models the folksonomy graph G_F, $\overrightarrow{p}$ allows to specify preferences (e.g. for a tag) and d enables to adjust the influence of the preference vector. FolkRank applies the adapted PageRank twice, first with $d = 1$ and second with $d < 1$ (in our evaluation we set $d = 0.7$ as done in [36]). The final vector, $\overrightarrow{w} = \overrightarrow{w}d < 1\overrightarrow{w}d = 1$, contains the FolkRank of each folksonomy entity.

5.2.2 GFolkRank

GFolkRank [37] is a context-sensitive ranking algorithm that is based on FolkRank. It expects a group context folksonomy which is a 5 tuple $F = (U, T, R, G, Y)$ as input and adapts the process of transforming the hypergraph spanned by the folksonomy into the weighted folksonomy graph G_F. It interprets groups as artificial tags and creates new tags $t_g \in T_G, T_G \cap T = \phi$, for each group g. These artificial tags are assigned to all resources contained in g, whereby the user who added a resource to the group, is declared as the tagger. The set of nodes is thus extended by $T_G : V_B = V_A \cup T_G$. The edges added to V_F by the GFolkRank algorithm are: $E_B = E_A[\{\{u, t_g\}, \{t_g, r\}, \{u, r\}|u \in U, t_g \in T_G, r \in \check{R}, u$ has added r to group g. We use a constant value w_c to weight these edges because a resource is usually added only once to a certain group.

5.2.3 GRank

GRank [38] is a group-sensitive ranking algorithm as well as GFolkRank. However, GRank is not based on FolkRank, but exploits group context folksonomy in a straightforward way. Given a query tag t_q, the GRank algorithm detects a set of tag assignments $(u, t, r, g) \in \check{Y}_q$, where the resource $r \in \check{(R)}$ is (a) directly annotated with t_q, (b) contained in a group that is tagged with t_q, (c) grouped together with a resource directly annotated with t_q, or (d) a group which contains a resource directly annotated with t_q. The entities $(users, tags, andresources)$ are then weighted according to their occurrence frequency within the tag assignments of $\check{Y}_q$. For more details on GRank we refer the reader to [37, 38].

5.2.4 Ranking user's relevance to a topic

This iterative ranking algorithm [39] approach is to calculate web-user's relevance through link analysis under a unified framework where the importance of web-pages and web-users mutually reinforce each other in an iterative way.

This algorithm similar with HITS algorithm [32], There are two assumptions in calculating the importance of users and webpages. The first assumption is that the importance of a webpage is influenced not only by the link structure of the webpages but also by the frequency and expertise of user visited them.

Expert detection using link analysis

Expert finding system can identify experts for a given query topic and differentiate their level of expertise. For example, for the query such as find expert in image retrieval, our system may return a list of individuals ranked by their level of expertise in the field of image retrieval. For clarity, we first describe the general model and the link analysis algorithm and then we describe the procedure of how to get the ranked experts for the given query topic.

$$\begin{aligned}
a(p) &= \beta \textstyle\sum_{q\to p} h(q) + (1-\beta) \textstyle\sum_{r\to p} u(r) \\
h(p) &= \beta \textstyle\sum_{p\to q} a(q) + (1-\beta) \textstyle\sum_{r\to p} u(r) \\
u(r) &= (1-\beta) \left(\textstyle\sum_{r\to p} a(p) + \textstyle\sum_{r\to q} h(q) \right)
\end{aligned}$$

5.3. Proposed Algorithm: SAKERank

In this section I first recapitulate ranking algorithms developed in previous work before I introduce a new algorithm: SAKERank. Our proposed SAKERank algorithm derived from Kleinberg's HITS algorithm [32] and Ranking user's relevance to a topic [39].

The underlying assumptions are: 1) the more high quality webpages the user has visited, the more experienced the user is. 2) the more frequently a webpage

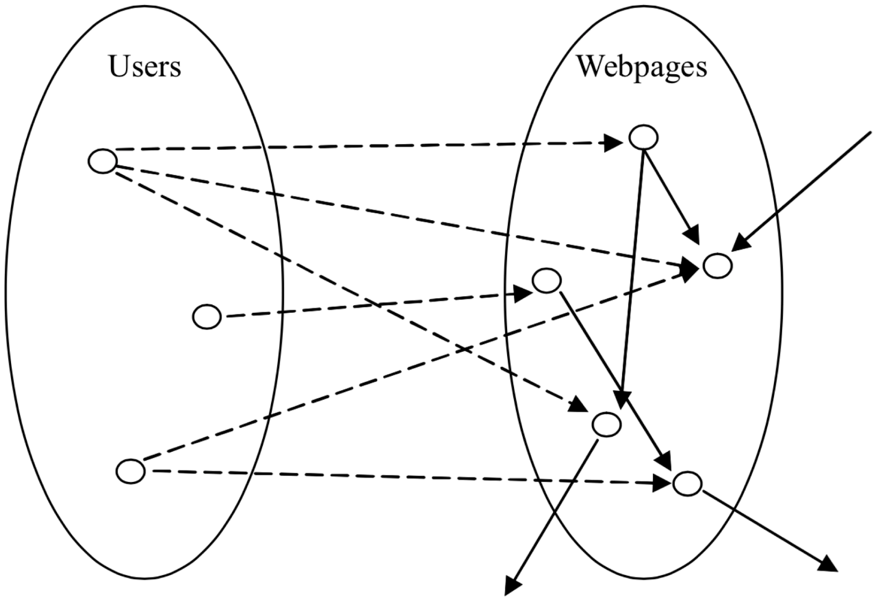


Figure 5.1. The directed graph model of expert finding system

is cited by other pages and visited by experienced users, the higher quality the webpage is. 3) the importance/quality of the users and pages may reinforce each other in an iterative way.

SAKERank calculates score for webpages and users based on hubs, authorities and user's level of expertise inside the community of interest. As shown in Equation 5.1, in this algorithm, the authority weighting of a webpage *a(p)* calculated by combining the sum of the hub values of all pages *h(q)* pointing to p and the sum of the weights of all people *u(r)* visited (weight by ω_2), individual bookmarks (weight by ω_3), and group bookmarks p. This combination forms the final authority weight of p. The hub weight is similarly calculated. The weight of a user *u(r)* is calculated by summing up the authority and hub weights of all pages he has visited (weight by ω_2), or bookmarked by himself (weight by ω_3) or group (weight by ω_4). Another term is indirectly influenced by the user r's weight which comes from his group participation (weight by 1-ω_4) The score in this term comes from webpages that all people in a group have bookmarked as a group.

$$
\begin{aligned}
a(p) &= \omega_1 \sum_{q \to p} h(q) + (1-\omega_1)\left(\omega_2 \sum_{r \to p} u(r) + (1-\omega_2)\left((\omega_3 \sum_{s \to p} u(s) + (1-\omega_3)\sum_{t \to p} u(t))\right)\right) \\
h(p) &= \omega_1 \sum_{p \to q} a(q) + (1-\omega_1)\left(\omega_2 \sum_{r \to p} u(r) + (1-\omega_2)\left((\omega_3 \sum_{s \to p} u(s) + (1-\omega_3)\sum_{t \to p} u(t))\right)\right) \\
u(r) &= \omega_2(\sum_{r \to i} a(i) + \sum_{r \to j} h(j)) + (1-\omega_2)\left(\omega_3(\sum_{r \to k} a(k) + \sum_{r \to l} h(l)) + (1-\omega_3)\right. \\
&\quad \left. + \left(\omega_4(\sum_{r \to m} a(m) + \sum_{r \to n} h(n)) + (1-\omega_4)(\sum_{r \to o} a(o) + \sum_{r \to p} h(p))\right)\right) \qquad (5.1)
\end{aligned}
$$

One of the most important things we must pay attention to is the convergence of our algorithm. Due to the difficulty of mathematical proof, we can only perform a series of experiments to test it. The difference of the page authority and hub scores and user weight is plotted for each iteration, which ranged from iteration 1 to iteration 20. The difference di is defined as $d_i = \sum(w_i - w_{i-1})^2$, where w_i represents the user importance and authority/hub values at iteration i. The value of a_i, h_i, u_i after each iteration is plotted in Figure 5.3. From Figure 5.3 we found that the difference of the webpages authority/hub values and users importance weight between adjacent iterations will drop significantly after 5 iterations and show a strong tendency to zero. This proves the convergence of our algorithm in a practical way.

SAKERank has been created in order to calculate the score for each user and each webpage related to the specific topic. The score allows us to rank the users inside the CoI and also re-rank the webpages related to the topic inside CoI.

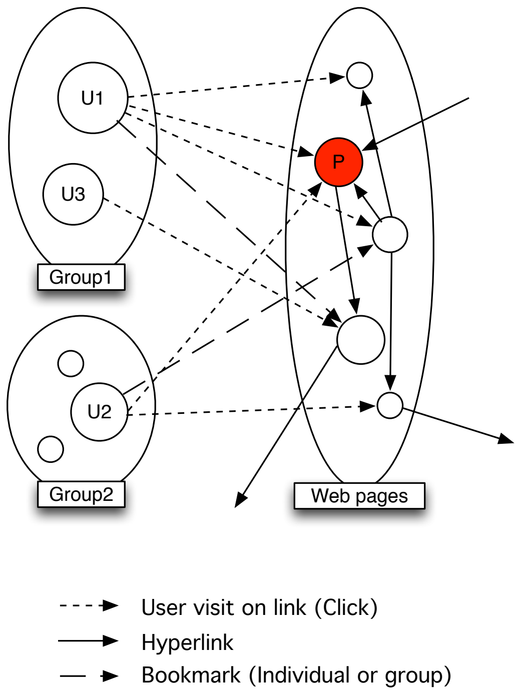


Figure 5.2. This directed graph model shows the interaction of users and webpages. It is an example how to calculate webpage P's score using SAKERank.

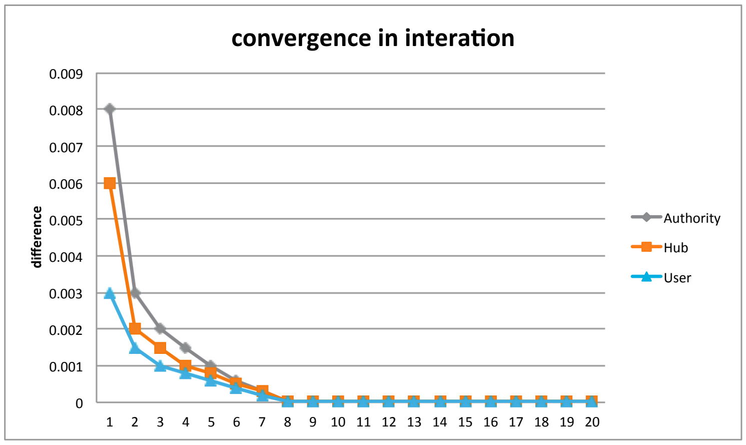


Figure 5.3. The convergence of our proposed algorithm (β=0.8)

Part III

Design & Implementation

Chapter 6

Adaptive Search Framework

To prove that the SAKE model can help communities solve the problems I mentioned in Section 3.2, I developed a framework called Adaptive Search Framework (ASF). The framework completes the technology part as shown in Figure 3.2 and applies the idea of the SAKE model.

6.1. Architecture

Figure 6.1 presents an overview of the ASF architecture. The framework is composed of three layers: interface, server, and job. Each layer serves different purposes and works independently from each other.

The top layer is the user interface layer, where people perform searching and obtain results the same as using normal search engines. I also developed a browser extension which allows people to organize search results.

The middle layer is the server layer, which returns search results related to the keywords and topics focused in the community. This layer communicates with search engines, which in this case are Google and Bing, by using their APIs. Then I submit a user's query and obtain search results. The framework uses Apache $Carrot^2$ to cluster the results into groups. Tags will be added to those groups to make them more descriptive and meaningful. Those data will be cumulatively stored in the ASF database. Then the server layer returns the combined search results, which contains both results from the conventional search engines and those retrived from ASF. After the user interacts with these results, the system

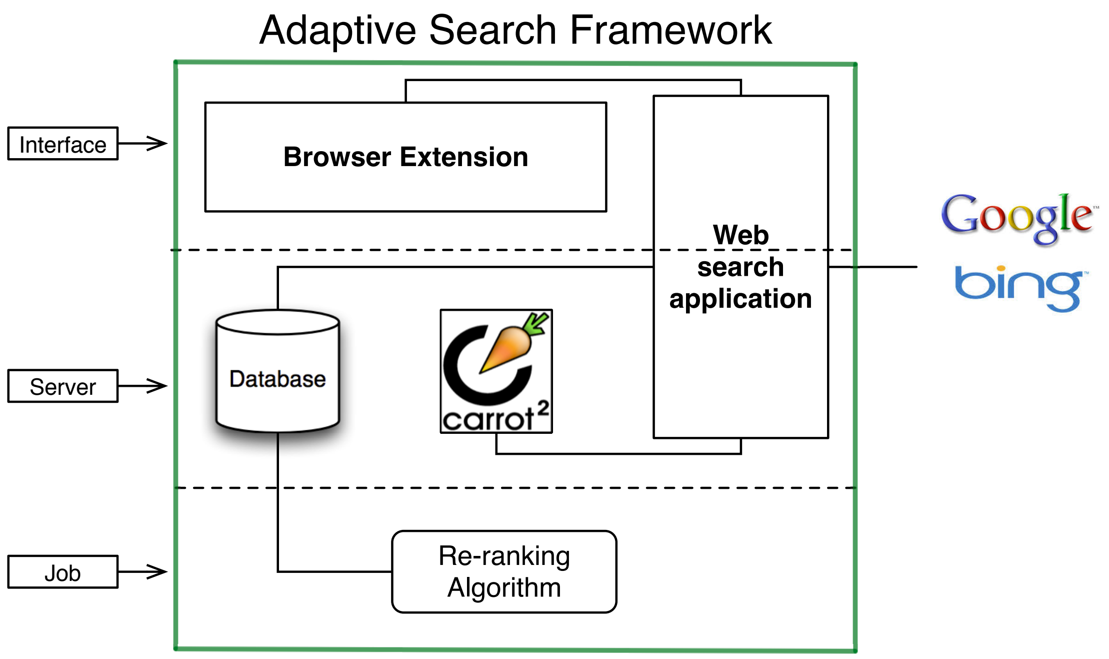


Figure 6.1. Architecture of Adaptive Search Framework.

stores information about the query, web pages, and tags that the user interacts with.

The bottom layer, called job layer, is scheduled on a regular basis. It calculates scores for people and web pages using data stored in the database.

6.2. User Interface Design

Adaptive Search Framework has two separate interfaces which are Internet browser extension and web application interface. The design in Figure 6.2 shows the user interface of the result page after the user has submitted the search keyword to ASF. The Section 1 in Figure 6.2 2 called a suggestion area. This section shows only the top *10* search results that have been analyzed by people inside the community and are already stored inside our database. Also, the results are already ranked by iterative re-ranking algorithm, which will be described in Section 6.3.

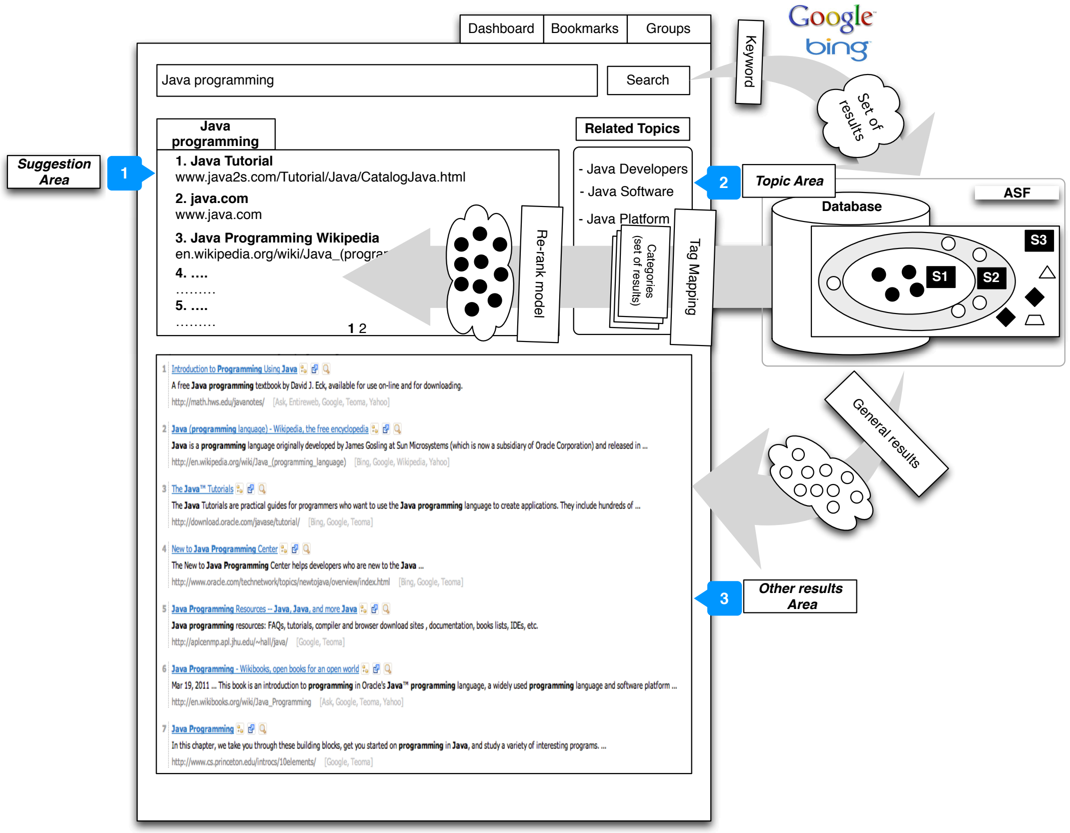
Section 2 in Figure 6.2 is related topics section, which lists all the topics that related to the community keywords that have been predefined. The last section shows general results retrieved from Google and Bing in each category. After the user hits the search button to submit the keyword, ASF uses Google and Bing APIs to obtain the general results for the keywords. After I got the general result set, ASF uses Apache $Carrot^2$ to cluster results into categories. AFS will select categories related to the topics in the community by using Tag Mapping method. Irrelevant results obtained from a major web search engine will be in section 3. Section 1 only contains results related to the topics inside the community. Section 1 contains the top-10. Section 2 is the topics, not yet reached in the top-10. I will describe the technique of our data flow in the next section.

The purpose of Internet browser Extension as shown in Figure 6.3 is to help users easily interact with websites. For example, bookmark, create groups and categories, etc. The web browser interface gives users a fully control on their account inside the framework.

6.3. Data Flow

In ASF, people perform two main activities, searching and bookmarking. In this section, I explain an implementation of our data flow by using town example illustrated in Figure 6.4.

To begin with searching, as shown in figure 6.2 a user has submitted query q, it will be sent to conventional search engine API. Top n ranked web pages will be returned as a search result set of W. Each w_i also contains 3 components, those are"url", "title", and "snippet". For a further process, I send the whole set W to Carrot^2 API where each component of w_i's will be treated as search result clustering source that return clustered labels of each w_i. I call these clustered label as tags (t), and I denote a set T_i as a set of w_i's corresponding tags. At this step, I process two sets, W and T. Next, I check each member of W if it has already been stored in our framework database. Note that I will explain the store's condition soon later in this section. In case w_i has already been in the database, I will update T_i to the stored w_i record called w_i'. That is the replacement of w_i's tags will be $T_i' \cup T_i$. For each tag t_i, it will later be assigned to several categories C_x or group categories G_X. I also have to update the linkage



1 [Introduction to **Programming** Using **Java**]
A free **Java programming** textbook by David J. Eck, available for use on-line and for downloading.
http://math.hws.edu/javanotes/ [Ask, Entireweb, Google, Teoma, Yahoo]

2 [**Java** (**programming** language) - Wikipedia, the free encyclopedia]
Java is a **programming** language originally developed by James Gosling at Sun Microsystems (which is now a subsidiary of Oracle Corporation) and released in ...
http://en.wikipedia.org/wiki/Java_(programming_language) [Bing, Google, Wikipedia, Yahoo]

3 The **Java**™ Tutorials
The **Java** Tutorials are practical guides for programmers who want to use the **Java programming** language to create applications. They include hundreds of ...
http://download.oracle.com/javase/tutorial/ [Bing, Google, Teoma]

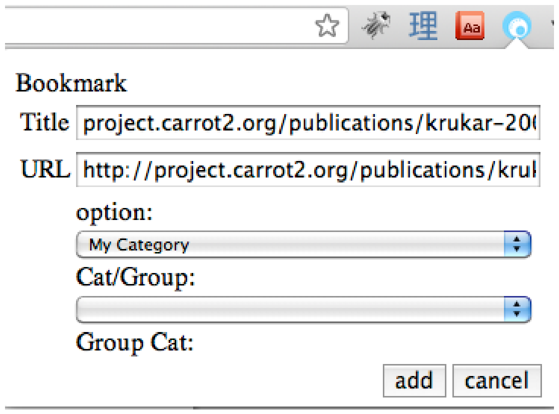
4 New to **Java Programming** Center
The New to **Java Programming** Center helps developers who are new to the **Java** ...
http://www.oracle.com/technetwork/topics/newtojava/overview/index.html [Bing, Google, Teoma]

5 **Java Programming** Resources -- **Java**, **Java**, and more **Java**
Java programming resources: FAQs, tutorials, compiler and browser download sites , documentation, books lists, IDEs, etc.
http://aplcenmp.apl.jhu.edu/~hall/java/ [Google, Teoma]

6 **Java Programming** - Wikibooks, open books for an open world
Mar 19, 2011 ... This book is an introduction to **programming** in Oracle's **Java**™ **programming** language, a widely used **programming** language and software platform ...
http://en.wikibooks.org/wiki/Java_Programming [Ask, Google, Teoma, Yahoo]

7 **Java Programming**
In this chapter, we take you through these building blocks, get you started on **programming** in **Java**, and study a variety of interesting programs. ...
http://www.cs.princeton.edu/introcs/10elements/ [Google, Teoma]

Figure 6.2. The UI design of Adaptive Search Framework is devided into 3 sections (Suggestion, topic, Other results). Suggestion area shows the results top-10 restuls that have been analyzed by users inside the CoI. Topic area shows the related topics involve with the search keyword and CoI. Other results area shows the normal results which are retrived from Google and Bing.


Bookmark

Title project.carrot2.org/publications/krukar-20

URL http://project.carrot2.org/publications/kru

option: My Category

Cat/Group:

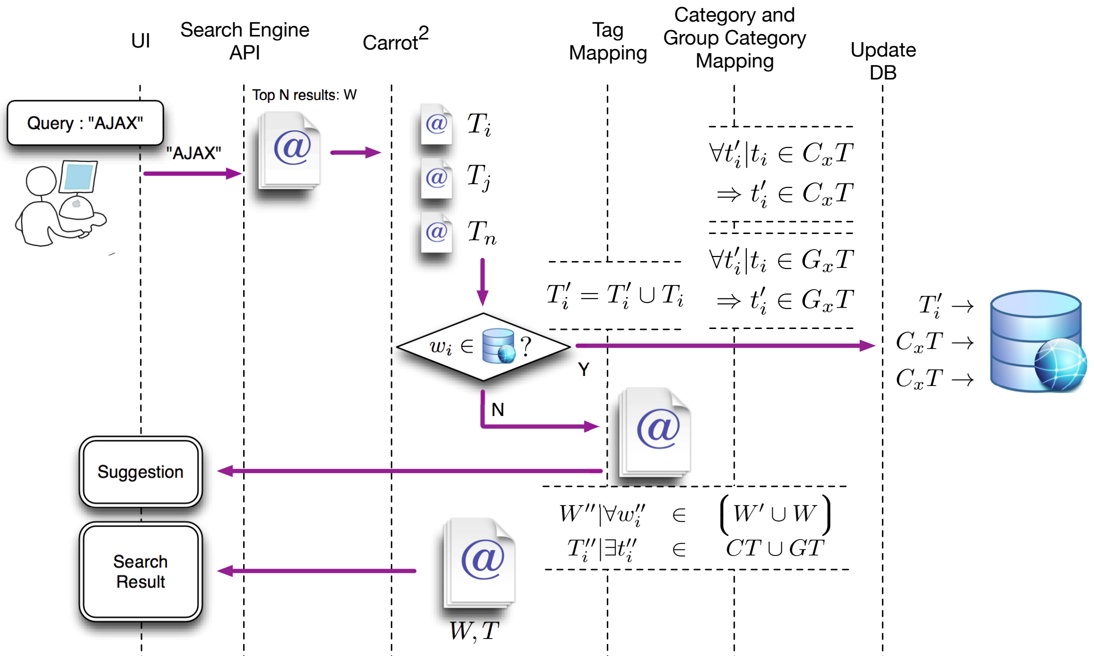
Group Cat:

Figure 6.3. Example of web browser extension

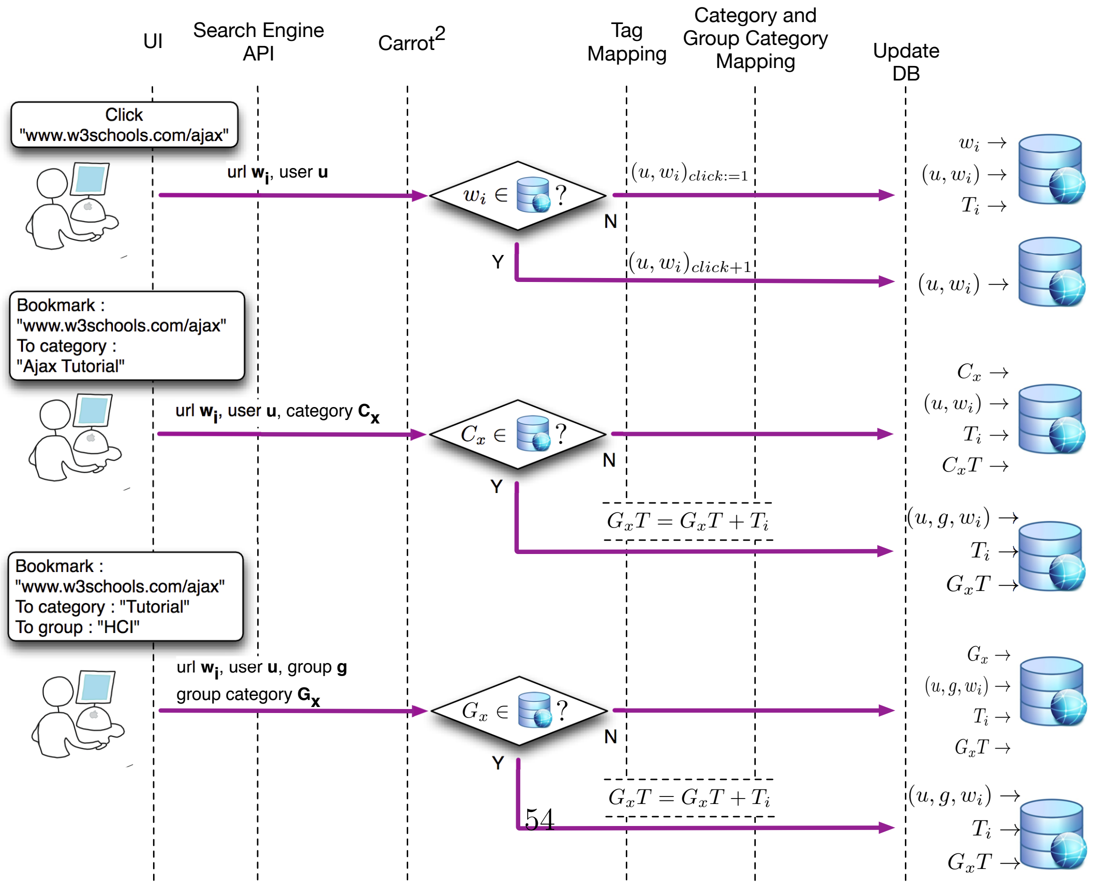
between new tags and the stored categories and group categories (C_xT and G_xT). That is $\forall t | t \in T_i$ will be update to each category C_x in C_xT linkage where C_x is its corresponded category to t_i.

On the other hand, in case that w_i is nonexistence in the database, it will only be processed for our suggestion feature. The web pages suggestion list consists of web pages the database, which has at least one category corresponded to at least one tag of a web page returned from the top n ranking search result set. To achieve that, first of all I merge all T_i into T ($T = \bigcup_1^n T_i$) and list all C_x that have a link to t_i into CT ($CT = \bigcup_1^n C_xT | \exists t_i \in C_xT$) and GT ($GT = \bigcup_1^n G_xT | \exists t_i \in G_xT$). I then do a reverse mapping from CT and GT to obtain T' that consist of all t'_i in $CT \cup GT$, and do a reverse mapping again from T' and get a set W' that satisfied the aforementioned condition. I also need to combine some records from the current search result set W in to the suggested web page set W'. Since I have precessed T', I can map if a t' has a link with a w_i in W and obtain the suggestible web page list from W. At last web pages suggestion list W'' come from W' which are web pages those are existed in the database, merging with W that satisfied the tag condition. So that it will become $W'' | \forall w''_i \in (W' \cup W) \to w''_i \in W''$ pairs with $T''_i | \exists t''_i \in (CT \cup GT) \to t''_i \in T''_i$ To show the original result set form the conventional search engine, I render only W in that area.

[Searching data flow]


UI
Search Engine API
Carrot^2
Tag Mapping
Category and Group Category Mapping
Update DB
Query : "AJAX"
"AJAX"
Top N results: W
T_i
T_j
T_n
$\forall t'_i | t_i \in C_x T \Rightarrow t'_i \in C_x T$
$\forall t'_i | t_i \in G_x T \Rightarrow t'_i \in G_x T$
$T'_i = T'_i \cup T_i$
$w_i \in$?
Y
N
$T'_i \rightarrow$
$C_x T \rightarrow$
$C_x T \rightarrow$
Suggestion
Search Result
$W'' | \forall w''_i \in (W' \cup W)$
$T''_i | \exists t''_i \in CT \cup GT$
W, T

[Clicking through the search result, individual bookmarking and group bookmarking data flow]


UI
Search Engine API
Carrot^2
Tag Mapping
Category and Group Category Mapping
Update DB
Click "www.w3schools.com/ajax"
url $\mathbf{w_i}$, user $\mathbf{u}$
$w_i \in$?
N
Y
$(u, w_i)_{click:=1}$
$(u, w_i)_{click+1}$
$w_i \rightarrow$
$(u, w_i) \rightarrow$
$T_i \rightarrow$
$(u, w_i) \rightarrow$
Bookmark : "www.w3schools.com/ajax" To category : "Ajax Tutorial"
url $\mathbf{w_i}$, user $\mathbf{u}$, category $\mathbf{C_x}$
$C_x \in$?
N
Y
$G_x T = G_x T + T_i$
$C_x \rightarrow$
$(u, w_i) \rightarrow$
$T_i \rightarrow$
$C_x T \rightarrow$
$(u, g, w_i) \rightarrow$
$T_i \rightarrow$
$G_x T \rightarrow$
Bookmark : "www.w3schools.com/ajax" To category : "Tutorial" To group : "HCI"
url $\mathbf{w_i}$, user $\mathbf{u}$, group $\mathbf{g}$ group category $\mathbf{G_x}$
$G_x \in$?
N
Y
$G_x T = G_x T + T_i$
$G_x \rightarrow$
$(u, g, w_i) \rightarrow$
$T_i \rightarrow$
$G_x T \rightarrow$
$(u, g, w_i) \rightarrow$
$T_i \rightarrow$
$G_x T \rightarrow$

Clicking a Result Link

As shown in Figure 6.2, After a search result has been clicked in the user interface, the URL of the clicked link w_i and the user identity u will be passed to the framework. If wi has never been clicked by any user, its identity will not be stored in the database. So I need to create it with its initialized click counter by one. I then store user-click with w_i record as a tuple $(u,\ w_i)$, and the w_i's corresponding tags T_i in the database. If w_i is already existed, I just update tuple $(u,\ w_i)$ by increased the click counter by one.

Bookmarking a Web Page

In ASF there are two bookmarking types , bookmarking for oneself and for group. The difference between both of them is the feedback scope of suggesting a new web page to a user. Individual bookmarking only influences altogether group of the user.

In case when bookmarked web page has been clicked through from the user interface, the identity of that web page w_i must be stored in the database. Then, the required parameters in this bookmark case are w_i , u and the bookmark category C_x. At first I bind u with w_i and store $(w_i\ ,u)$ as a bookmark record and store it in the database as a bookmark identity. If C_x has just been created right before a user bookmarked it, I have to store it as a record in the database at first. I then map all T_i, which correspond to the bookmark page w_i to C_x as C_xT, and store all of them in the database.

However, if a user chooses to bookmark any web pages without searching from the framework, I need to process its corresponding tags at first. I choose to pass that web page's basic components such as title and URL through the search interface as query that allows tags to be processed as well as the ordinary routine.

Bookmarking a Web Page into a group

In-group bookmarking, tags and categories will be similarly processed to individual bookmarking. A bookmark page w_i and its corresponding tags T_i are bound with a group category G_x as G_xT and all of them will be stored in the database. The difference between the bookmarked records are bound from user, group, and web together as a tuple (u, g, w_i) instead of (u, w_i) in an individual bookmarking. I also do the same if a bookmarked page did not come from the search result by passing a query of the web page's basic components to the framework.

6.4. Database

For the database, ASF uses MYSQL[1] as a relational database management system. In figure 6.5 shows the database schema in my Adaptive Search Framework. It is quite easy to understand because there are only five domains that I need to focus on: User table stores the personal information of each users. (e.g. ID, email, password, etc.) User also links with click and bookmark tables which store the interaction of the users and webpages. User can belong to any group and be able to create the category which can store the webpages. The details of the website such as title, snippet, and score are stored in Web and WebContent table. Tag store keywords or terms assigned to category, website, and group. Tag describe and categorize an item and allows it to be found again by browsing or searching.

6.5. Category Classification

Adaptive Search Framework identifies experts for a given query topic and rank users' level of expertise by collecting the keywords that user search and then categorize webpages that users interact with (clicks and bookmarks). Then the SAKERank algorithm is used to calculate the score for each user and webpage. The result from the algorithm will be able to tell which topic users have been experienced on. To do this, I need to create a program to collect keywords and be able to categorize them into their own categories that I already prepared beforehand in the CoI. This program divided into two parts. The preparation process part as show in the Figure 6.6 is to create a model that will be able to predict which category the keywords should be. The model that will be used to predict which category the keywords should belong to is needed to be train first. In the preparation process, first, I have to prepare the categories which has the potential to be able to relate to the target CoI. For example, If the target

[1]http://en.wikipedia.org/wiki/MySQL

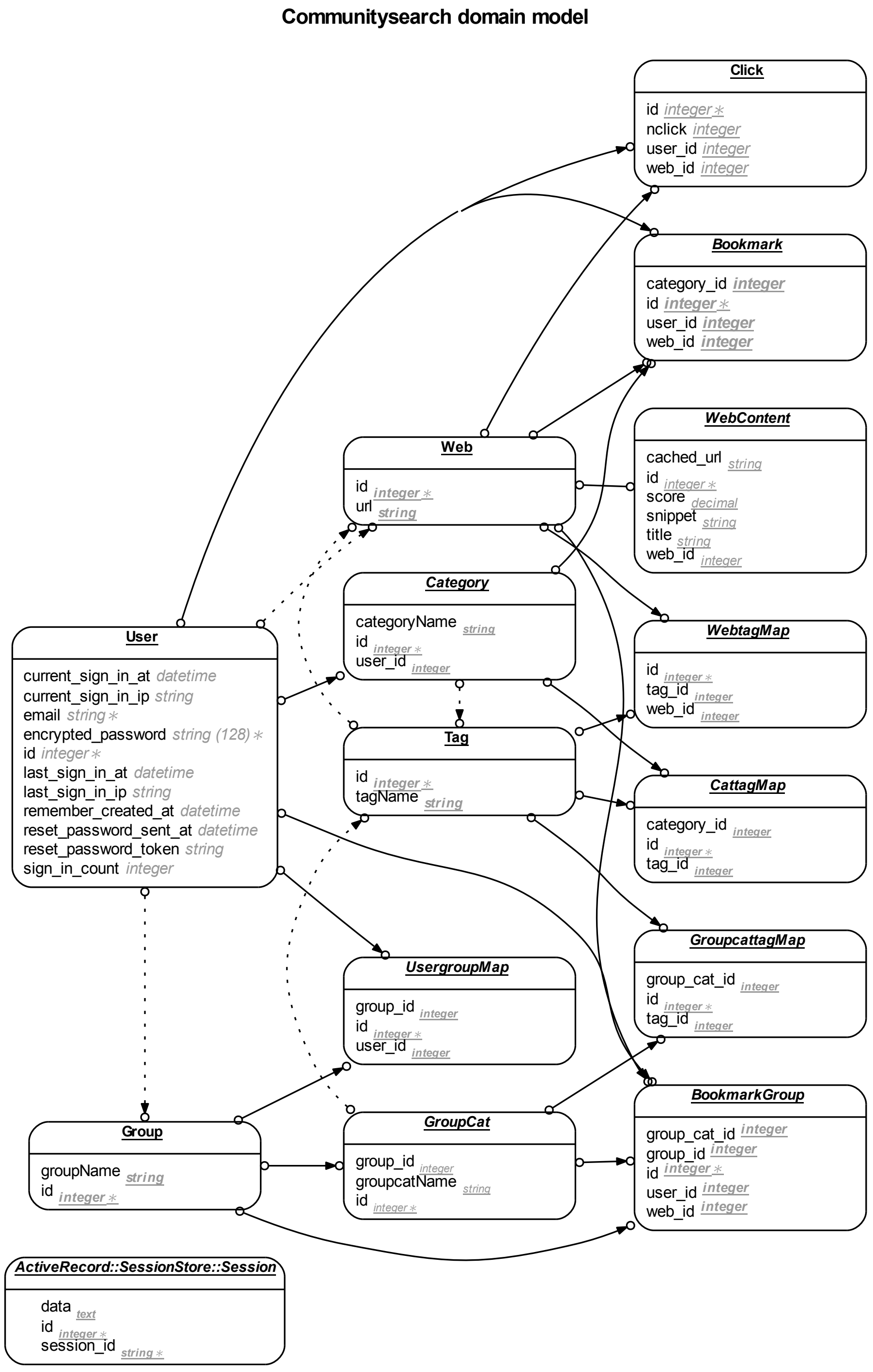


Figure 6.5. Database schema of Adaptive Search Framework

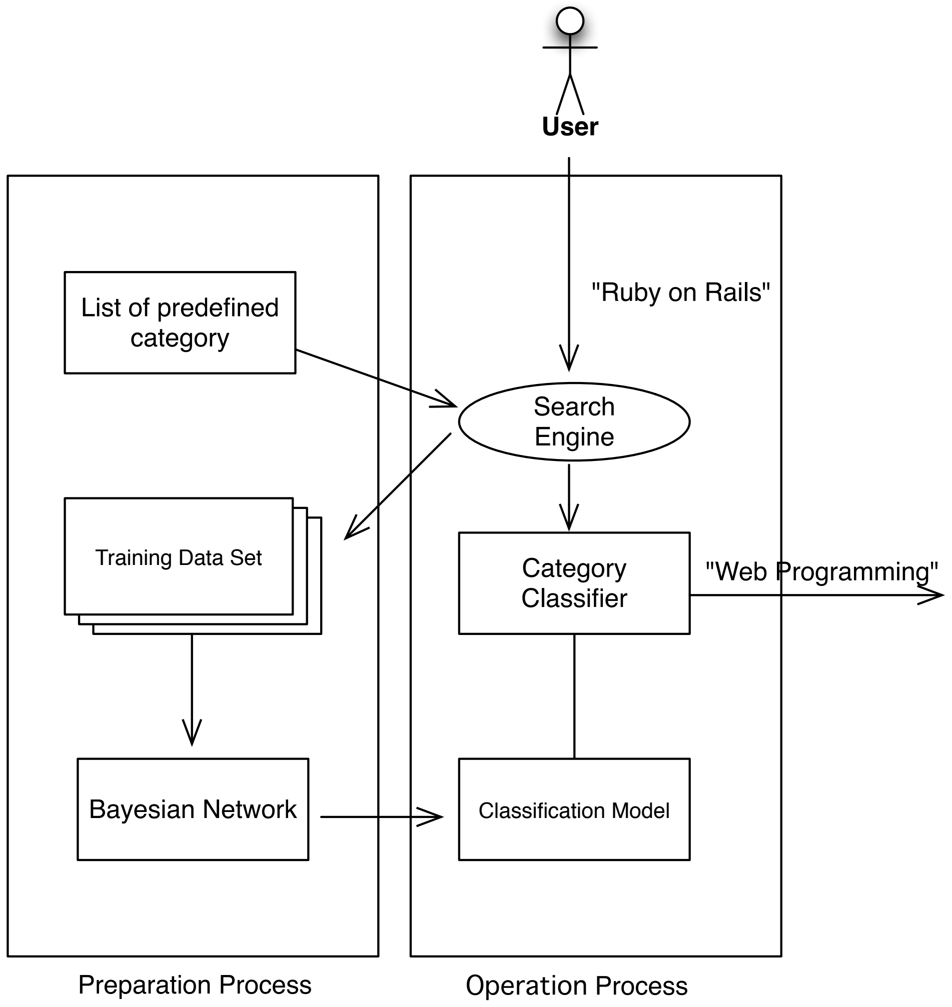


Figure 6.6. Show the steps of how to build a classification model

CoI is a software company, a lot of your categories should relate to the software and programming languages. After predefined the category, the model based is built up based on the hypothesis that if the documents have a lot of the same keywords of the category; it means that this document should be in the category. For example, the documents in java Programming category should have a lot of "java" keywords inside the document. Then search engine return what keywords that the web pages contain and the keywords are sent to he Bayesian network[2] to build the classification model.

The operation process part in the Figure 6.6 is the process to predict which category the keyword should be by using the trained model from the Bayesian Network process. After we got the classification model, when user performs a search task, the keyword will be sent to search engine again and obtain the return result. The results will be map to the model that has already been trained. Finally ASF will be able to tell which keyword should goes to which category.

[2]http://en.wikipedia.org/wiki/Bayesian_network

Chapter 7

Experiment

7.1. Experimental Design and Procedure

The performance of ASF consists of two things: 1. The ability to return more relevant search results to users compared to search results from Google and Bing, 2. ASF reduces knowledge-sharing effort with Search Activity Knowledge Extraction (SAKE). In order to investigate the performance ASF , our experiment will follow these procedures.

1. **Select a community**: Our target is small to medium size community where people have a variety of knowledge skills and level and frequently use search engines for solving tasks or searching for information.

2. **Define search target area and task**: We analyzed previous web search logs of a selected community and interviewed members in order to define the top-*10* topics inside the community. We created a task which contains three simple questions in each topic for participants to complete during the experiment.

3. **Set up the framework**: ASF is deployed inside the community and is implemented on top of Google and Bing to get initial search results.

4. **Gather participants**: I gathered as many participants as I can from the community. We analyzed each participant and separate them into three

groups: *expert*, *experienced*, and *newcomer* based on their profile, knowledge and skill.

5. **Create ideal result sets**: Since we do not have standard result sets with which to measure, we need to create them. An ideal result set is a set of results, which we assume is the best results which are ranked from 1 to 10 based on the topic. We asked participants in an expert group to create a set of ideal results for each topic. First, I asked participants to select topics in which he/she has expertise. Then I asked them to rank 10 websites that have the information or solution to solve the given task on their topics. We will use the ideal result set as a standard for measuring results from ASF, Google, and Bing later on.

6. **Search, share, and complete tasks**: The duration of our experiment is two weeks, in order to gain enough data to measure the performance of ASF. For the first week I let participants from the experienced group use ASF to complete the given task. Then, participants from the newcomer group will use ASF in the decode week to complete the same task.

7. **Evaluate the performance**: To bring clarity to our results for both ASF's performance and participants' experience, I compare top-*5* and top-*10* results for overlapping and ranking similarity from ASF Google and Bing in each topic based on the ideal ranking set. After that, each participant in experienced and newcomer groups do a post-test survey to rate ASF's overall usage.

7.2. Evaluation measurements

Overlapping and ranking similarity

I use two standard measurement methods, which are *OSim* and *KSim* [40], when comparing search result rankings. *OSim* indicates the degree of overlap between top k search results of two ranking between t_1 and t_2 (t_1 = Ideal results, t_2 = Retrieved results).

$$OSim_k(t_1, t_2) = \frac{|R_1 \cup R_2|}{k} \tag{7.1}$$

In the comparisons, to compare top *5* and top *10* I will use $k = 5$ and 10. *OSim* score has [0,1] range. The higher *OSim* score means that both search results sets have more overlap results. *OSim* score lets us know the extent to which search result sets are similar to each other. The overlap measure *OSim* gives an incomplete picture of the similarity of two rankings. It does not indicate the degree to which the relative orderings of the top k results of two rankings are in agreement.

To show the similarity of two rankings more accurately, *KSim* is used to determine the degree of agreements in which pairwise distinct documents u and v within top-k rank has the same relative order in both ranking t_1 and t_2.

$$KSim_k(t_1,t_2)=\frac{|(u,v):t'_1,t'_2 \text{ agree on order of } (u,v),u\neq v|}{(|\cup|)(|\cup|-1)} \tag{7.2}$$

KSim has [0,1] range. The higher KSim means that both result sets have similar rankings. However measurement method that based on Kendall*, It will have a disagreement on the long-range comparison between t_1 and t_2. In this research, I aim to use *KSim* to compare only top *5* and top *10* search results.

Together, *OSim* and *KSim* are suited to measure the quality of a ranking with respect to an optimal ranking. Our evaluation are based on ... Given 10 keywords, asking experts to create ideal ranking for each of keywords, which represented from their perspective the most precise top 10 rankings.

7.3. Post-test survey (SUS)

To clarify participants' experience of using ASF after two weeks, I constructed a system usability scale (SUS) post-test survey [41] as shown in Appendix 10.4 for participants. The survey contains 10 questions, which is clustered into 3 categories: *Information Seeking*, *Knowledge sharing and management*, and *Tool usage*. Each category contains three questions. Participants can give the score for each question from 0 to 4(*Strongly disagree*, *disagree*, *neutral*, *agree*, and *Strongly agree*). Also, Participants are able to leave any comment for each question. The

Table 7.1. Selected topics in software engineering field.

ajax tutorial
text mining method
software metrics for agile software development
java programming tutorial
web service
object oriented design
empirical study in software engineering
software development effort estimation
software maintenance technique
software product line

post-test survey for both experienced and newcomer groups are given by the end of first and second week.

7.4. Participants

Software Engineering Laboratory (SE Lab) in Nara Institute of Science and Technology was selected as an experiment community. I analyzed previous search keyword and information related to the community to defined ten topics as shown in Table 7.1. ASF was deployed in an Ubuntu server in the laboratory and it used MYSQL to store the data. The framework was implemented on top of two conventional search engines, which are Google and Bing, in order to obtain initial search results.

Twenty participants, including three professors and seventeen students with various levels of knowledge and skill, were recruited for this experiment. First, we divided participants based on their profile into three groups: *expert*, *experienced*, and *newcomer*. Three professors and two Ph.D. students were in the expert group. Seven master students from second year were in the experienced group, and the rest of eight participants, being first-year master students, were in the newcomer group. The participants in the expert group were asked to create an ideal result set (shown in Figure 10.1) based on the task as shown in Figure 10.2 and 10.3.

After the framework has launced, I asked seven participants from the experienced group used ASF for one week to complete all the given tasks. Next, I asked eight participants from the newcomer group to use our ASF to find information for the second week. Finally, I collected participants' activities and search results from each topic by each week since ASF startup until the end of the second week. I compared the results by using standard measurement, which are *OSim* and *KSim* [40], to determine the overlap and similarity of the results. We also administered the post-test surveys and interviewed participants for the satisfaction of our ASF usage.

Chapter 8

Results

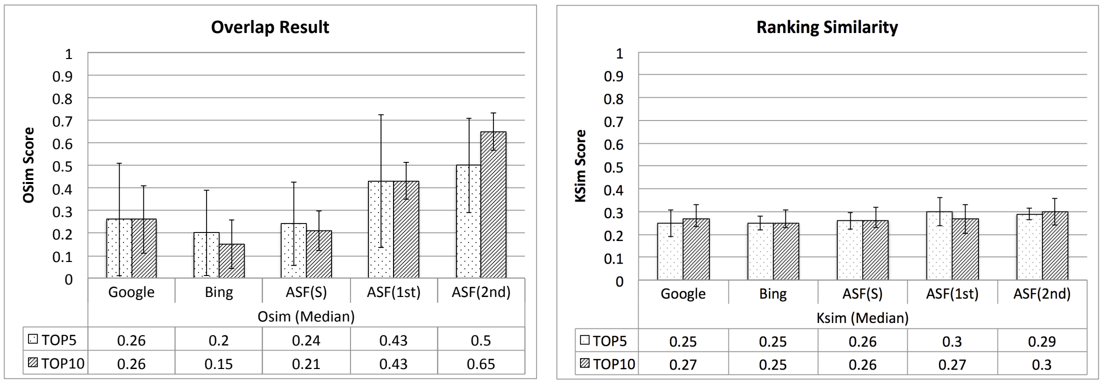
8.1. Overlapping and Similarity Search Results

First, I compared the degree of overlapping results, including results from Google, Bing and ASF with ideal result sets by using *OSim* measurement.

The left side of Figure 8.1 shows the average *OSim* score of top *5* and top *10* search results which came from the ten selected topics as shown in Table 7.1 from Google, Bing and ASF from SE Lab. After ASF was launched, the average *OSim* score of ASF was 0.24 for top *5* and 0.21 for top *10*. This indicates that five links from ASF have around one link similar to the ideal result set of 5, and 10 links from ASF have around two links appear in ideal result set of 10. In the beginning, ASF score was a little bit lower than Google but higher than Bing.

After one week, I calculated ASF *OSim* score again. *OSim* score of ASF for top *5* increased from 0.24 to 0.43 and top *10* increased from 0.21 to 0.43, which are now higher than both Google and Bing. After I finished the experiment in the second week, I calculated *OSim* score for ASF again and I found the score was even higher. With the result, I now can assume that ASF is getting better by when participants use it to search for the same target.

Then I compare the ranking similarity of the results for each search engine with the ideal result sets. The result shows that the ranking from Google, Bing, and ASF compared to the ideal result set are quite the same which is between 0.23 and 0.3. The *KSim* score of ASF in the second are 0.29 for top *5* and 0.3 for top *10*, which are slightly better than Google and Bing. The reason that the



	Google	Bing	ASF(S)	ASF(1st)	ASF(2nd)
□ TOP5	0.26	0.2	0.24	0.43	0.5
▨ TOP10	0.26	0.15	0.21	0.43	0.65

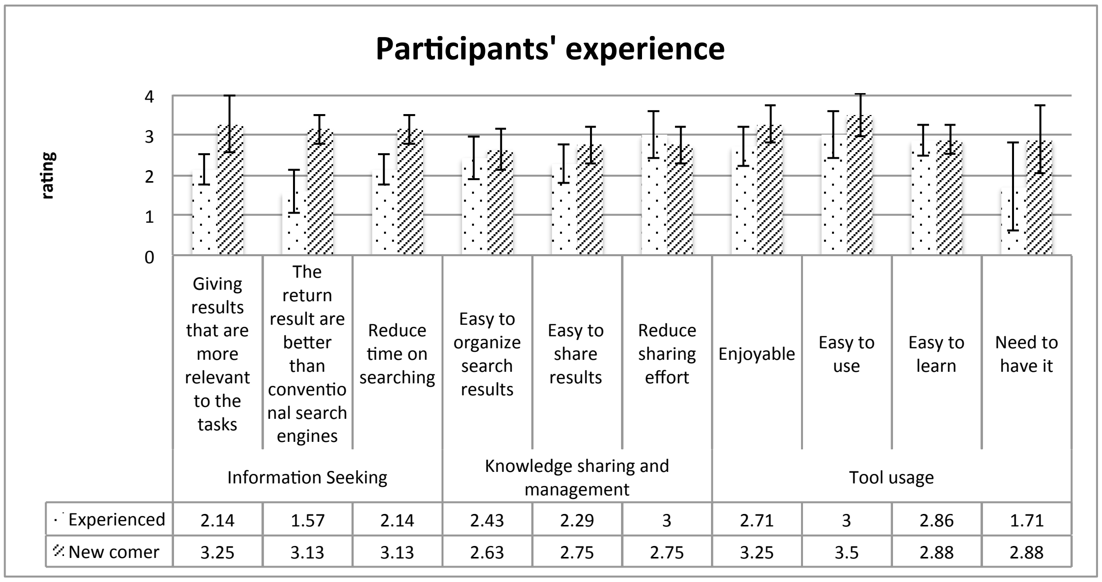
	Google	Bing	ASF(S)	ASF(1st)	ASF(2nd)
□ TOP5	0.25	0.25	0.26	0.3	0.29
▨ TOP10	0.27	0.25	0.26	0.27	0.3

Figure 8.1. The comparison of search results from Google, Bing, ASF (startup), ASF (1st week), and ASF (2nd week) with ideal search result based on the topics inside SE Lab. The diagram on the left shows *OSim* score for the degree of overlap results in top *5* and top *10* search results. The diagram on the right shows *KSim* score for the similarity of each raking compare with ideal search result.

KSim score of ASF are quite similar is because I might not give more enough time for ASF to collect data from participants which can affect the performance of iterative re-ranking algorithm to re-rank the results.

8.2. Participants' Experience

To evaluate the usability of ASF, I use system usability scale which was originally developed by John Brooke [41] to survey the participants after 2 weeks. The survey consists of 10 statements to which participants rate their level of agreement. It contains *Information Seeking*, *Knowledge sharing and management*, and *Tool usage* parts. We will explain the results separately by the following. The result as shown in Figure 8.2 is what I collected from 15 participants. 7 participants are experienced users and 7 participants are newcomer users. Each item's score contribution will rage from 0 to 4.



	Giving results that are more relevant to the tasks	The return result are better than conventional search engines	Reduce time on searching	Easy to organize search results	Easy to share results	Reduce sharing effort	Enjoyable	Easy to use	Easy to learn	Need to have it
	Information Seeking			Knowledge sharing and management			Tool usage			
⋰ Experienced	2.14	1.57	2.14	2.43	2.29	3	2.71	3	2.86	1.71
▨ New comer	3.25	3.13	3.13	2.63	2.75	2.75	3.25	3.5	2.88	2.88

Figure 8.2. Post-test usage survey results from participants.

8.2.1 Information Seeking

Question 1: Which search engine gives better results that related to the tasks given inside the community?

After one week participants are satisfied with the results from ASF more. The average score of participants in the *experienced* group is 3.14, which is just a little bit above the average. The average score of participants in the newcomer group is 3.25/4 which is very high. The reason is because ASF can return the results that have been already analyzed by participants in the experienced group in the first week to the participants in the newcomer group which are using it in the second week. One of the participants in the newcomer group reported that

> *"Most of the results in the suggestion area are really useful. They are what I expected to answer the questions."*

Question 2: Are the results from ASF better than conventional search engines (Google and Bing).

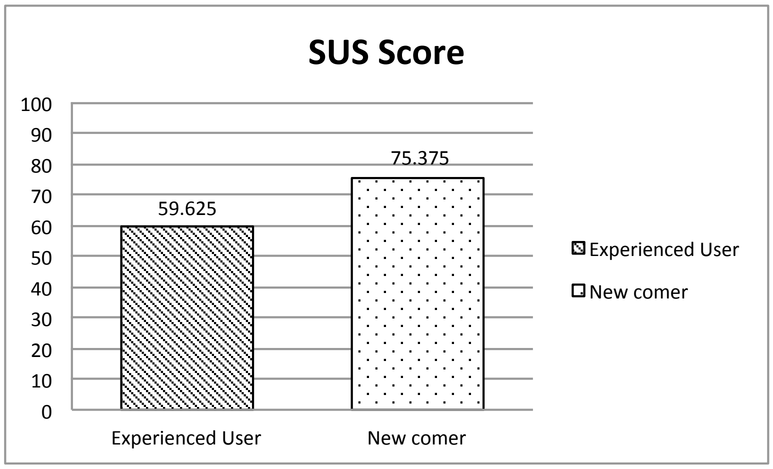


Figure 8.3. System Usability Scale (SUS) of ASF in percentage. <50: Not acceptable, 50-70: Marginal, >70: Acceptable

The score is 1.57/4 from the experienced group, which is quite low. One of the participants in the experienced group said

> " *The results from the suggestion area are quite the same as the results from both Google and Bing. It made me confused which link should I click on.*"

The results from the newcomer group are 3.13/4, which is significantly high. Most of the participants from the newcomer group have said that the results from the suggestion area are what they were looking for.

Question 3: Can AFS help user reduce time on searching?

Participants from the newcomer group rated quite a high score, which is 3.13/4, for this one. This results indicates that ASF can save time for them while searching for the information inside the community.

8.2.2 Knowledge sharing and management

In this part, I aim to discover the performance of our ASF on Knowledge sharing and management inside the SE Lab.

Question 4: How easy is it to organize search results?

The scores, which are 2.43/4 and 2.5/4, from both the experienced and the newcomer group are quite the same. One of the participants in the experienced group has reported that

> " *It is very simple to organize my favorite websites just like when I add the webpages using my browser. ASF allowed me to share my webpages to others in the community.*"

Question 5 and 6: How easy participants can share the results via ASF?

Both scores from the experienced and the newcomer groups are quite acceptable. Participants benefited from the knowledge-sharing and the management features of ASF inside the community while doing the search collaboration with other members.

> " *ASF allows me to bookmark the webpages that I like into the topics which are already created by other members inside the community. Also I can look through all the webpages that have been shared from others inside the topics.*"

> " *When I found a really cool webpage, I don't need to waste my time copying the URL and send it via Facebook or Line anymore. I can just add the link into the topic and my friend will be able to see it.*"

8.2.3 Tool usage

Question 7, 8, 9, and 10: Are participants satisfied by the overall usage of ASF?

In this part participants are quite satisfied with the overall usage based on the scores, which are all above average. Many participants said

> "*There are many useful features that Google and Bing do not provide us.*"

Due to the many features inside our ASF, participants felt that ASF helped them to get useful information. One participant said that

> "*This framework provides our research community to have their own search engine.*"

One person said

> "*It gave me better search results compared to major search engines.*"

To calculate a sus score first sum the socre contributions from each type of participant (Experienced and New comer). Each item's score contribution will rage from 0 to 4. Multiply the sum of the scores by 2.5 to obtain the overall SUS Score. Consider the data in Figure 8.3. The sum of the score of experienced users is 59.62 which is in marginal rage and the overall score for new comer is 75.37, which is in acceptable rage.

We do note that the tasks that I asked participants to do were limited in a small scale, but they were enough to evaluate the framework and prove our basic assumption of our ASF that we committed at the early section.

Part IV

Synopsis

Chapter 9

Discussion

9.1. How can ASF return Better-suited Results to CoI?

To provide additional insight into the results related to the first question, we extracted data from the database to create a relation graph showing the users and web pages from Software engineering laboratory. In this relation graph, users are also separated into their own special groups. Figure 9.1 inside the square blue rectangle area, also shows website ids 8, 4, 3, 117, and 116, the top 5 results shown in Figure 9.1 for Ajax tutorial query. ASF selected these five websites for the suggestion box because they have a high ranking as shown in Figure 9 and are related to be Ajax tutorial keyword.

9.1.1 How does framework relate apparently unrelated URLs?

To help address the second question, I examined the ability of the ASF to suggest websites that have not been viewed by users, but are related to keywords and topics that the users are interested in. As an example, I arranged a scenario to test whether the ASF would relate Superman and Clark Kent. As shown in figure 9.2 in the search on the back, at first when a user searches for "Superman movie" there was nothing in the suggestion box because this keyword was new to the framework. However, after a user clicked or bookmark some of the websites about the Superman movie, as shown in the middle search picture, another search

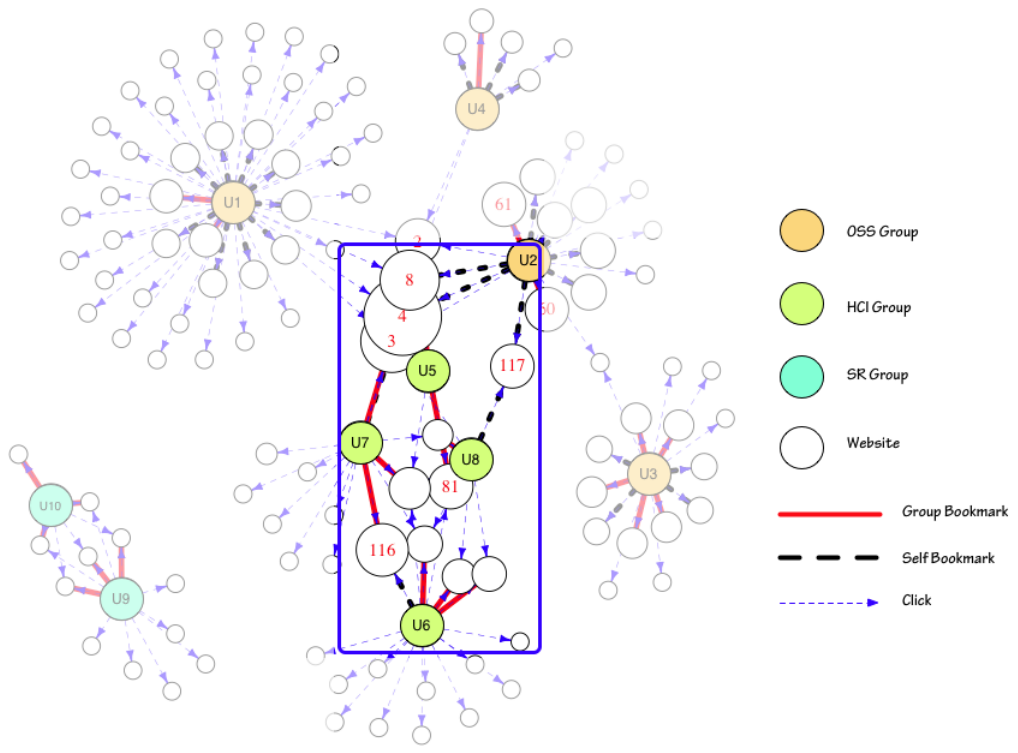


Figure 9.1. Relation graph between users and websites inside Software Engineering Laboratory.

for “Superman” resulted in several websites in the suggestion box related to the Superman movie because the framework knows that this user is interested in the Superman movie. So when this user searches for Superman, the ASF provides information about movies based on the data from the previous search. Finally, the front picture in figure 9.2 shows that when a user searches for “Clark Kent,” Superman’s secret identity, the ASF can suggest that the user should also look for Superman. However, a similar search for “Clark Kent” on the major search engines provides top ranked results only related to Clark Kent. This indicates that the Adaptive Search Framework can suggest results better related to a user’s interests and topics.

In the research paper called “Learn from Web Search Logs to Organize Search

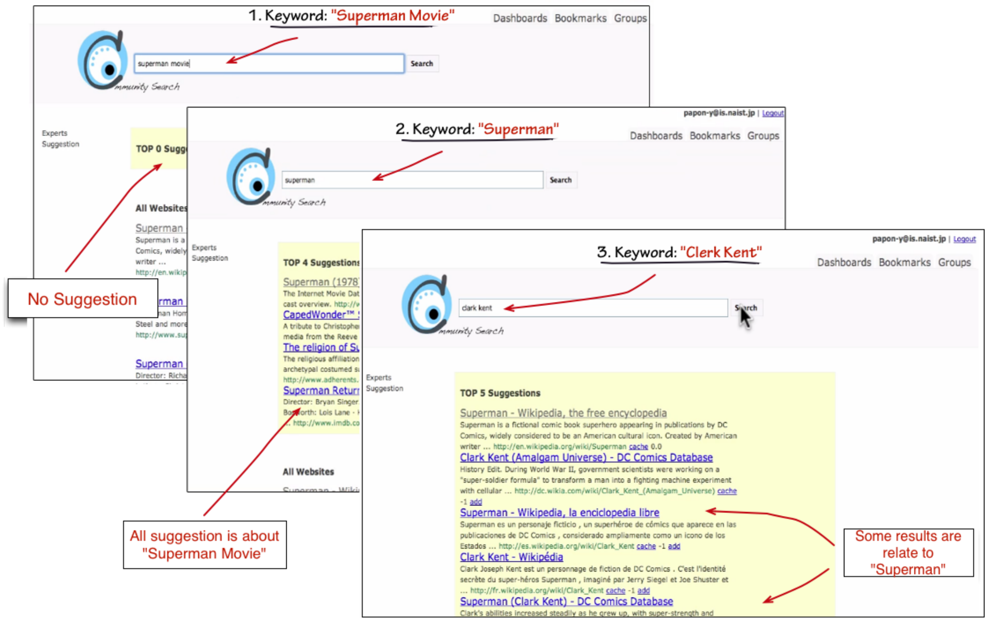
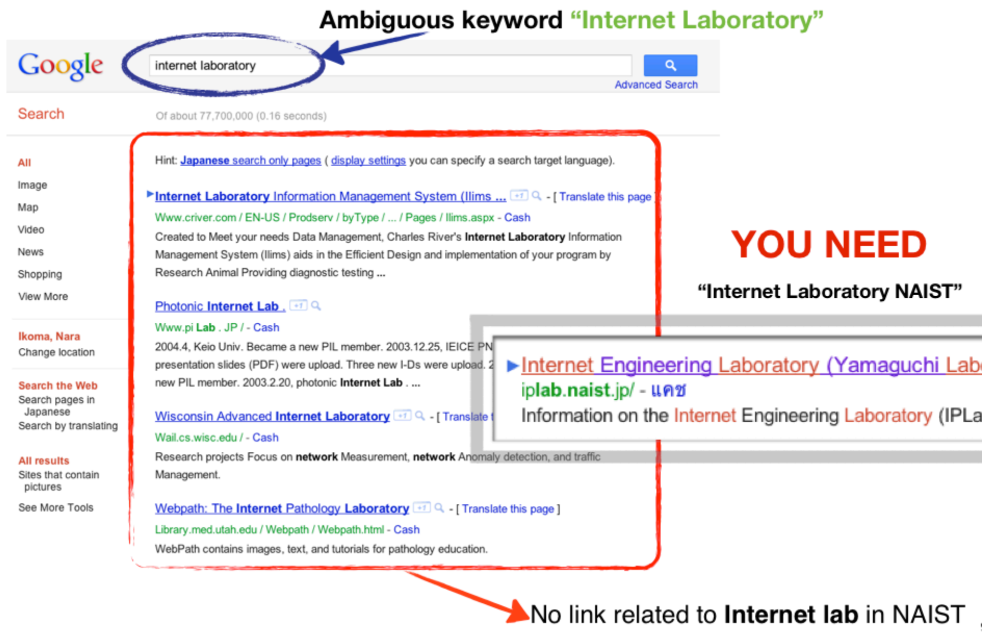


Figure 9.2. 1. At first, the framework cannot suggest anything when a user searches for "Superman Movie" because it is a new topic for the framework. 2. After a user has interacted with some of the websites that involve Superman, the framework can suggest some websites when a user searches for "Superman" based on the previous search data. 3. Also, the framework can suggest Superman when a user searches for Clark Kent.

results" [42] has shown the comparison between the cluster-based method and log-based method. The cluster-based method has to rely on the keywords extracted from the snippets to construct the tag for each cluster. The log-based method use the center of each star cluster [42] as the label for the corresponding cluster.

From the table 9.1, the log-based method gives more readable and more specific labels because it generates labels based on users' queries. But in general, when I apply our framework into the organization, I cannot gather all the members search log data to create a structure information for the framework to learn from. The framework will have to learn a bit by bit from the users and when I gain enough web search logs from the members, I will also cluster our links in the organization by using that web search log together with the clustering engine



Of about 77,700,000 (0.16 seconds)

Hint: Japanese search only pages (display settings you can specify a search target language).

Internet Laboratory Information Management System (Ilims ... - [Translate this page]
Www.criver.com / EN-US / Prodserv / byType / ... / Pages / Ilims.aspx - Cash
Created to Meet your needs Data Management, Charles River's **Internet Laboratory** Information Management System (Ilims) aids in the Efficient Design and implementation of your program by Research Animal Providing diagnostic testing ...

Photonic Internet Lab .
Www.pi **Lab** . JP / - Cash
2004.4, Keio Univ. Became a new PIL member. 2003.12.25, IEICE PN presentation slides (PDF) were upload. Three new I-Ds were upload. 2 new PIL member. 2003.2.20, photonic **Internet Lab** . ...

Wisconsin Advanced Internet Laboratory - [Translate t
Wail.cs.wisc.edu / - Cash
Research projects Focus on **network** Measurement, **network** Anomaly detection, and traffic Management.

Webpath: The Internet Pathology Laboratory - [Translate this page]
Library.med.utah.edu / Webpath / Webpath.html - Cash
WebPath contains images, text, and tutorials for pathology education.

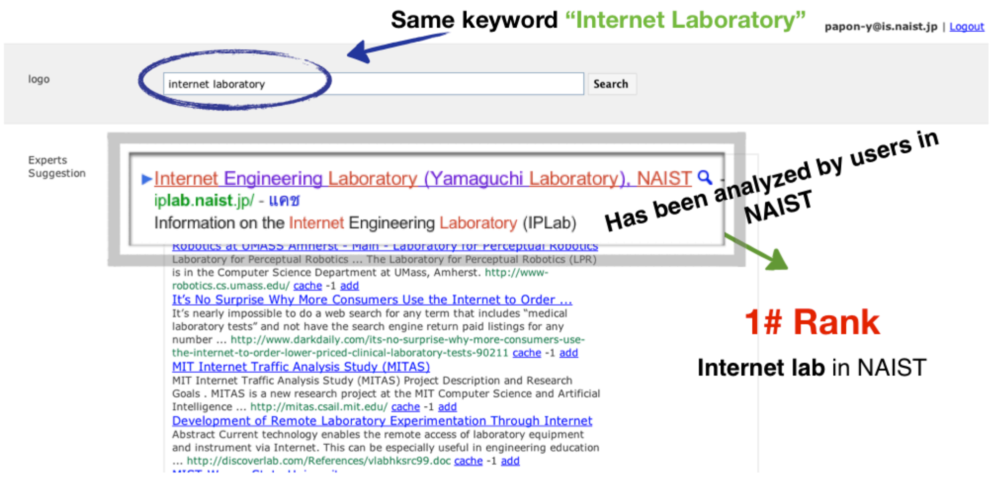
Internet Engineering Laboratory (Yamaguchi Lab
iplab.naist.jp/ - แคช
Information on the Internet Engineering Laboratory (IPLa

Figure 9.3. Returned results from Google shows that Google doesn't give the results that are related topics inside CoI.

In figure 9.3 shows you the results from Google when you searched for "internet laboratory". As the results what you expect is "iplab.naist.jp" which is the internet laboratory in Nara Institute of Science and Technology but what Google returned at the very first top ranks are internet laboratories from other universities because those internet laboratories have higher ranking score based on the popularity.

Log-based method	**Cluster-based method**
jaguar animal	jaguar, auto, accessories
jaguar auto accessories	jaguar, type, prices
jaguar cats	jaguar, panthera, cats
jaguar repair	jaguar, services, boston

Table 9.1. Label comparison between Log-based method and Cluster-based method


Same keyword "Internet Laboratory"
papon-y@is.naist.jp | Logout
logo
Search
Experts Suggestion
Has been analyzed by users in NAIST
1# Rank
Internet lab in NAIST

internet laboratory

▶Internet Engineering Laboratory (Yamaguchi Laboratory), NAIST
iplab.naist.jp/ - แคช
Information on the Internet Engineering Laboratory (IPLab)

Robotics at UMASS Amherst - Main - Laboratory for Perceptual Robotics
Laboratory for Perceptual Robotics ... The Laboratory for Perceptual Robotics (LPR) is in the Computer Science Department at UMass, Amherst. http://www-robotics.cs.umass.edu/ cache -1 add

It's No Surprise Why More Consumers Use the Internet to Order ...
It's nearly impossible to do a web search for any term that includes "medical laboratory tests" and not have the search engine return paid listings for any number ... http://www.darkdaily.com/its-no-surprise-why-more-consumers-use-the-internet-to-order-lower-priced-clinical-laboratory-tests-90211 cache -1 add

MIT Internet Traffic Analysis Study (MITAS)
MIT Internet Traffic Analysis Study (MITAS) Project Description and Research Goals . MITAS is a new research project at the MIT Computer Science and Artificial Intelligence ... http://mitas.csail.mit.edu/ cache -1 add

Development of Remote Laboratory Experimentation Through Internet
Abstract Current technology enables the remote access of laboratory equipment and instrument via Internet. This can be especially useful in engineering education ... http://discoverlab.com/References/vlabhksrc99.doc cache -1 add

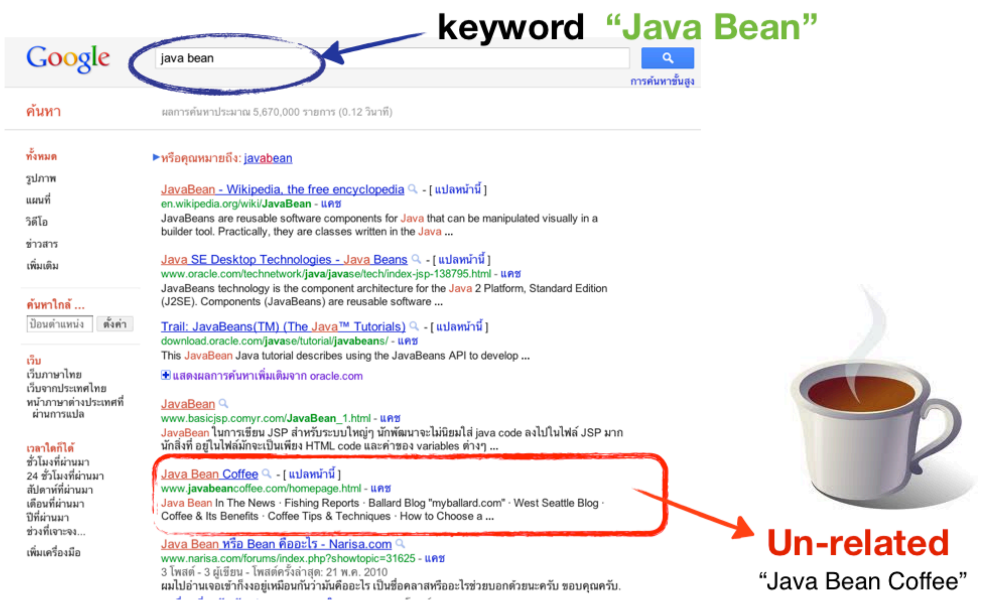
Figure 9.4. The results from Adaptive Search Framework using the same keyword to search as in figure 9.3. The results can prove that Adaptive Search Framework knows what result is related to the topic in organization.

By using the same keyword "internet laboratory" to search for the results in Adaptive Search Framework. The Framework is able to suggest the internet laboratory in our university back to the user because the framework knows exactly that this link is related to the organization and might be the one that user who search for the internet laboratory in this organization might be looking for.

which we believe it will be more accurate to give us the label of webpages.

9.1.2 What about more general topics?

The results from the experiment shows that ASF yield better results when it is in a community that focuses on some specific area but it may not be optimal for the general topics. The information based on the majority of people in the world could return the best results for the general topics.



ผลการค้นหาประมาณ 5,670,000 รายการ (0.12 วินาที)

▸หรือคุณหมายถึง: javabean

JavaBean - Wikipedia, the free encyclopedia - [แปลหน้านี้]
en.wikipedia.org/wiki/JavaBean - แคช
JavaBeans are reusable software components for Java that can be manipulated visually in a builder tool. Practically, they are classes written in the Java ...

Java SE Desktop Technologies - Java Beans - [แปลหน้านี้]
www.oracle.com/technetwork/java/javase/tech/index-jsp-138795.html - แคช
JavaBeans technology is the component architecture for the Java 2 Platform, Standard Edition (J2SE). Components (JavaBeans) are reusable software ...

Trail: JavaBeans(TM) (The Java™ Tutorials) - [แปลหน้านี้]
download.oracle.com/javase/tutorial/javabeans/ - แคช
This JavaBean Java tutorial describes using the JavaBeans API to develop ...
แสดงผลการค้นหาเพิ่มเติมจาก oracle.com

JavaBean
www.basicjsp.comyr.com/JavaBean_1.html - แคช
JavaBean ในการเขียน JSP สำหรับระบบใหญ่ๆ นักพัฒนาจะไม่นิยมใส่ java code ลงไปในไฟล์ JSP มาก นักสิ่งที่ อยู่ในไฟล์มักจะเป็นเพียง HTML code และค่าของ variables ต่างๆ ...

Java Bean Coffee - [แปลหน้านี้]
www.javabeancoffee.com/homepage.html - แคช
Java Bean In The News · Fishing Reports · Ballard Blog "myballard.com" · West Seattle Blog · Coffee & Its Benefits · Coffee Tips & Techniques · How to Choose a ...

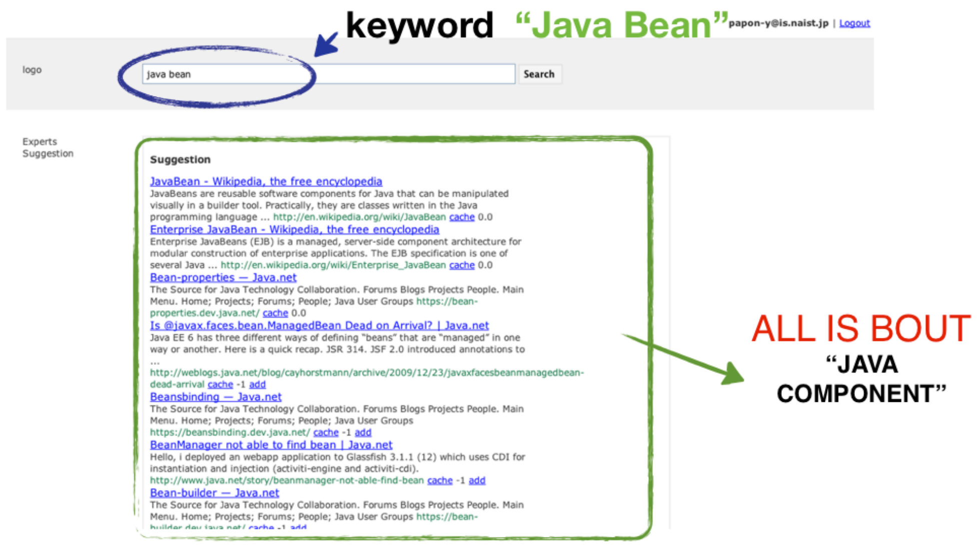
Java Bean หรือ Bean คืออะไร - Narisa.com
www.narisa.com/forums/index.php?showtopic=31625 - แคช
3 โพสต์ - 3 ผู้เขียน - โพสต์ครั้งล่าสุด: 21 พ.ค. 2010
ผมไปอ่านเจอเข้าก็งงอยู่เหมือนกันว่ามันคืออะไร เป็นชื่อคลาสหรืออะไรช่วยบอกด้วยนะครับ ขอบคุณครับ.

Figure 9.5. The results from Google, which have the same keyword but different in meaning.

Next I tried to test if the ASF has the ability to clustering the results to be returned to users by learning from the interested topics of each user. For example, a student inside Software Engineering laboratory search for “java bean”, What student expect should be related to the programming language component. Using just a conventional search engine, The results which related to the same keyword but different meaning can also come up.

9.2. ASF’s Effectiveness?

The experiments only show that ASF works effectively on a small-sized community. For a larger sized community, which may have, more than hundreds of member, we assume that people will produce noises inside the database. We plan to do more experiments on a larger sized community in the future. One of the ASF key purpose is to get rid of irrelevant results that do not related to the topics inside the CoI, which can help member reduce time locating the useful results. There are several aspects on effectiveness that can serve both high level of knowledge member and novice member. For example, high-level members can



java bean

Suggestion

JavaBean - Wikipedia, the free encyclopedia
JavaBeans are reusable software components for Java that can be manipulated visually in a builder tool. Practically, they are classes written in the Java programming language ... http://en.wikipedia.org/wiki/JavaBean cache 0.0

Enterprise JavaBean - Wikipedia, the free encyclopedia
Enterprise JavaBeans (EJB) is a managed, server-side component architecture for modular construction of enterprise applications. The EJB specification is one of several Java ... http://en.wikipedia.org/wiki/Enterprise_JavaBean cache 0.0

Bean-properties — Java.net
The Source for Java Technology Collaboration. Forums Blogs Projects People. Main Menu. Home; Projects; Forums; People; Java User Groups https://bean-properties.dev.java.net/ cache 0.0

Is @javax.faces.bean.ManagedBean Dead on Arrival? | Java.net
Java EE 6 has three different ways of defining "beans" that are "managed" in one way or another. Here is a quick recap. JSR 314. JSF 2.0 introduced annotations to ...
http://weblogs.java.net/blog/cayhorstmann/archive/2009/12/23/javaxfacesbeanmanagedbean-dead-arrival cache -1 add

Beansbinding — Java.net
The Source for Java Technology Collaboration. Forums Blogs Projects People. Main Menu. Home; Projects; Forums; People; Java User Groups
https://beansbinding.dev.java.net/ cache -1 add

BeanManager not able to find bean | Java.net
Hello, i deployed an webapp application to Glassfish 3.1.1 (12) which uses CDI for instantiation and injection (activiti-engine and activiti-cdi).
http://www.java.net/story/beanmanager-not-able-find-bean cache -1 add

Bean-builder — Java.net
The Source for Java Technology Collaboration. Forums Blogs Projects People. Main Menu. Home; Projects; Forums; People; Java User Groups https://bean-

Figure 9.6. The ASF solving problem by using webpages that have been analyzed by users inside organization to map with the tags and return them back when other users search for that keyword.

To solve the problem that web pages have the same keyword but different meaning, ASF learns from the keywords and maps them to the topics inside our software engineering laboratory and suggests the results that have been analyzed by users inside the laboratory which related to the topics inside to the user.

earn the benefits from using ASF to get rid of irrelevant results. They are also more likely to use ASF to contribute useful results to the community. For novice members, ASF helps them reducing time locating the results that are useful and relevant to the topics in the community. We are not trying to replace major web search engine. What we are trying to do is improving the search results selection method for a community and reduce knowledge sharing effort inside CoI. We suggest that the most effectiveness method for information seeking tasks is to using both major web search engine to search for general topics and ASF to search for specific results that related to the community.

Chapter 10

Conclusion

This dissertation focuses on how to support effective knowledge sharing through search activities. The motivation came from long time ago when we performed a collaborate task which we need to use search engines to search for the information, which is specific to the topics among the group. What we have found is there were many irrelevant search results that returned from conventional search engine. When we want to share the very first idea is how to improve search results which are more relevant to the community of interest based on a collaborative search. The SAKE model has been proposed and ASF was developed in order to help people inside a community of interest reduce knowledge-sharing effort while searching and to obtain search results that are more relevant to topics of interest to the community. ASF uses data gathered from users performing ordinary searches, clicking on links, and bookmarking to enrich the data and increase effectiveness. SAKERank algorithm calculate score for webpages and users iteratively. We performed an experiment in a real community which is in software engineering lab at Nara Institute of Science and Technology. The goal is to test the performance of the SAKERank algorithm, which is able to return more relavant results to the CoI's topics and the users satisfaction of the usage of ASF. The standard measurement,*OSim* and *KSim* matrics, have been used in order to evaluate the performance of SAKERank algorithm. And the SUS's post-test survey was conducted to get the feedback from the participants after the use of ASF. The results suggest the framework which applied the SAKE model could benefit both people and community. We strongly believe that over time our model and framework can help people get better search results and help the community increase productivity.

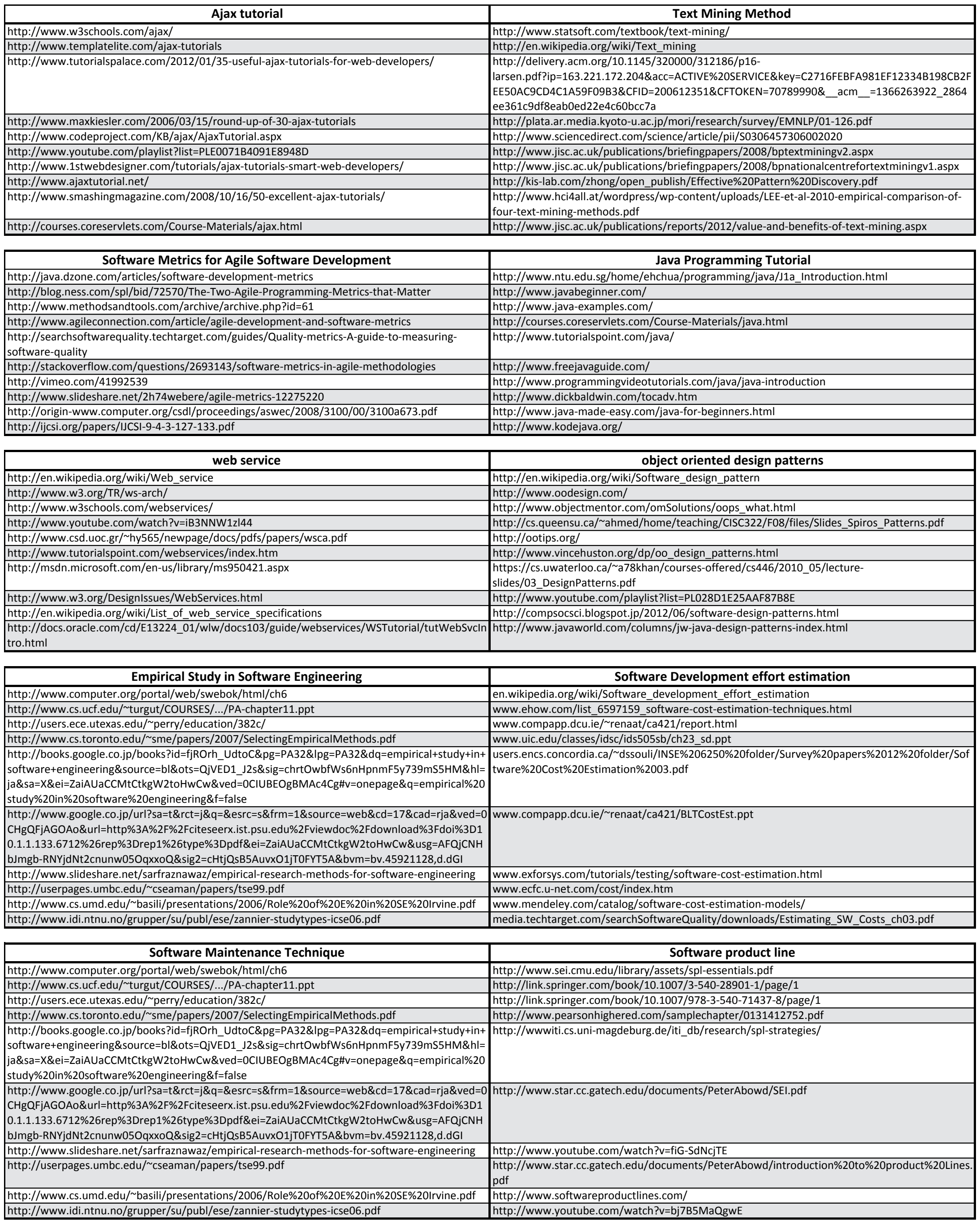
For the future work we will investigate more on:

- To test ASF in a variety type of community, ASF will be deployed in different communities to determine whether our proposed solution should target what size and topic of community.
- How to classify the expert from the community using search behavior. To do this, it may be necessary to predefine a set of experts related to the community, or to develop a method to let the ASF identify users and their expertise from their searches.
- Personalization will be integrated into ASF in order to fine-grained the results and make them more relevant to each user.

Finally, we believe that this research could contribute in one way or another to the people and community which are seeking the way to retrieve the knowledge from users' search behavior and encourage them to share their knowledge with no time.

Part V

Appendix



Ajax tutorial	Text Mining Method
http://www.w3schools.com/ajax/	http://www.statsoft.com/textbook/text-mining/
http://www.templatelite.com/ajax-tutorials	http://en.wikipedia.org/wiki/Text_mining
http://www.tutorialspalace.com/2012/01/35-useful-ajax-tutorials-for-web-developers/	http://delivery.acm.org/10.1145/320000/312186/p16-larsen.pdf?ip=163.221.172.204&acc=ACTIVE%20SERVICE&key=C2716FEBFA981EF12334B198CB2FEE50AC9CD4C1A59F09B3&CFID=200612351&CFTOKEN=70789990&__acm__=1366263922_2864ee361c9df8eab0ed22e4c60bcc7a
http://www.maxkiesler.com/2006/03/15/round-up-of-30-ajax-tutorials	http://plata.ar.media.kyoto-u.ac.jp/mori/research/survey/EMNLP/01-126.pdf
http://www.codeproject.com/KB/ajax/AjaxTutorial.aspx	http://www.sciencedirect.com/science/article/pii/S0306457306002020
http://www.youtube.com/playlist?list=PLE0071B4091E8948D	http://www.jisc.ac.uk/publications/briefingpapers/2008/bptextminingv2.aspx
http://www.1stwebdesigner.com/tutorials/ajax-tutorials-smart-web-developers/	http://www.jisc.ac.uk/publications/briefingpapers/2008/bpnationalcentrefortextminingv1.aspx
http://www.ajaxtutorial.net/	http://kis-lab.com/zhong/open_publish/Effective%20Pattern%20Discovery.pdf
http://www.smashingmagazine.com/2008/10/16/50-excellent-ajax-tutorials/	http://www.hci4all.at/wordpress/wp-content/uploads/LEE-et-al-2010-empirical-comparison-of-four-text-mining-methods.pdf
http://courses.coreservlets.com/Course-Materials/ajax.html	http://www.jisc.ac.uk/publications/reports/2012/value-and-benefits-of-text-mining.aspx

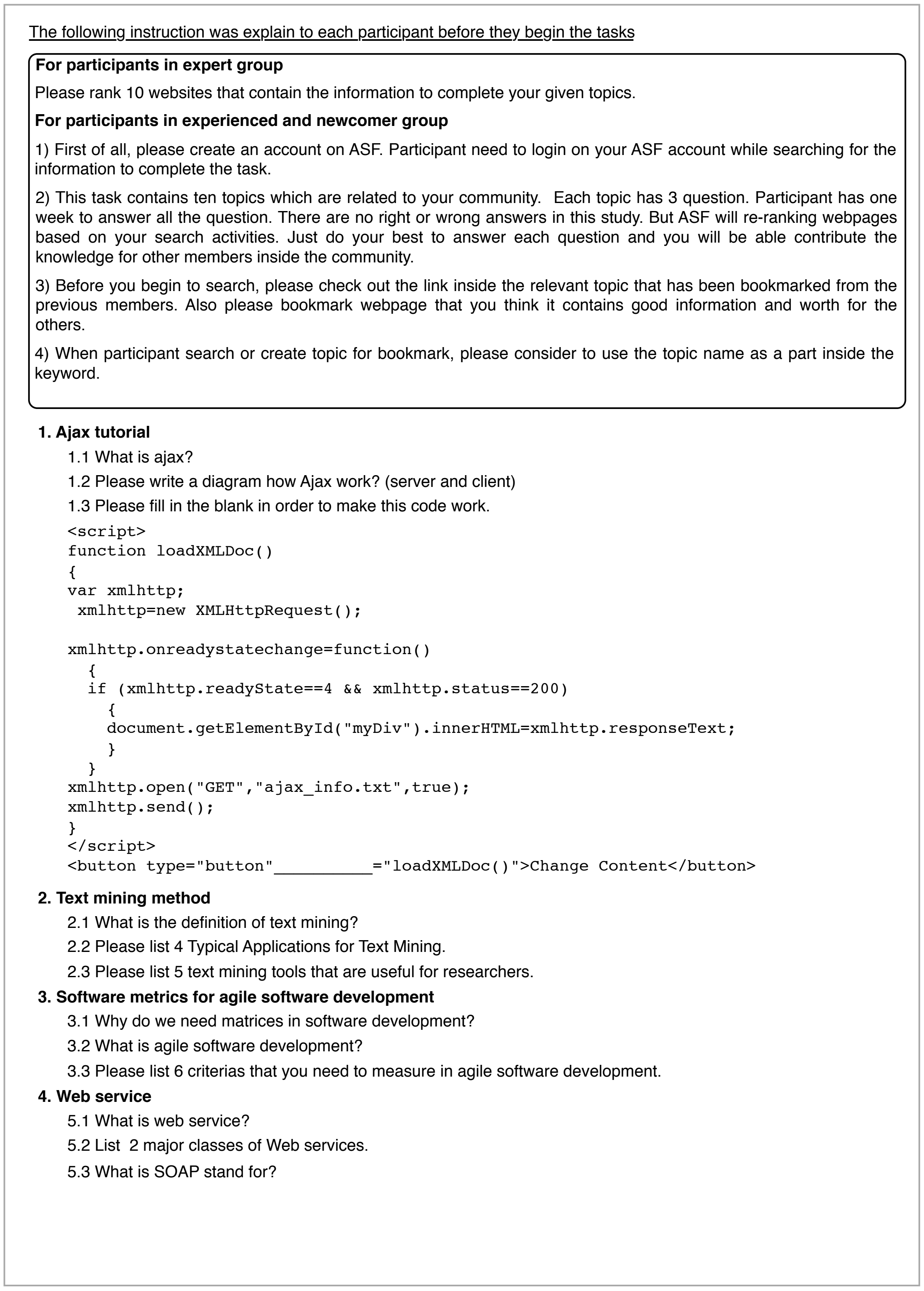
Software Metrics for Agile Software Development	Java Programming Tutorial
http://java.dzone.com/articles/software-development-metrics	http://www.ntu.edu.sg/home/ehchua/programming/java/J1a_Introduction.html
http://blog.ness.com/spl/bid/72570/The-Two-Agile-Programming-Metrics-that-Matter	http://www.javabeginner.com/
http://www.methodsandtools.com/archive/archive.php?id=61	http://www.java-examples.com/
http://www.agileconnection.com/article/agile-development-and-software-metrics	http://courses.coreservlets.com/Course-Materials/java.html
http://searchsoftwarequality.techtarget.com/guides/Quality-metrics-A-guide-to-measuring-software-quality	http://www.tutorialspoint.com/java/
http://stackoverflow.com/questions/2693143/software-metrics-in-agile-methodologies	http://www.freejavaguide.com/
http://vimeo.com/41992539	http://www.programmingvideotutorials.com/java/java-introduction
http://www.slideshare.net/2h74webere/agile-metrics-12275220	http://www.dickbaldwin.com/tocadv.htm
http://origin-www.computer.org/csdl/proceedings/aswec/2008/3100/00/3100a673.pdf	http://www.java-made-easy.com/java-for-beginners.html
http://ijcsi.org/papers/IJCSI-9-4-3-127-133.pdf	http://www.kodejava.org/

web service	object oriented design patterns
http://en.wikipedia.org/wiki/Web_service	http://en.wikipedia.org/wiki/Software_design_pattern
http://www.w3.org/TR/ws-arch/	http://www.oodesign.com/
http://www.w3schools.com/webservices/	http://www.objectmentor.com/omSolutions/oops_what.html
http://www.youtube.com/watch?v=iB3NNW1zl44	http://cs.queensu.ca/~ahmed/home/teaching/CISC322/F08/files/Slides_Spiros_Patterns.pdf
http://www.csd.uoc.gr/~hy565/newpage/docs/pdfs/papers/wsca.pdf	http://ootips.org/
http://www.tutorialspoint.com/webservices/index.htm	http://www.vincehuston.org/dp/oo_design_patterns.html
http://msdn.microsoft.com/en-us/library/ms950421.aspx	https://cs.uwaterloo.ca/~a78khan/courses-offered/cs446/2010_05/lecture-slides/03_DesignPatterns.pdf
http://www.w3.org/DesignIssues/WebServices.html	http://www.youtube.com/playlist?list=PL028D1E25AAF87B8E
http://en.wikipedia.org/wiki/List_of_web_service_specifications	http://compsocsci.blogspot.jp/2012/06/software-design-patterns.html
http://docs.oracle.com/cd/E13224_01/wlw/docs103/guide/webservices/WSTutorial/tutWebSvcIntro.html	http://www.javaworld.com/columns/jw-java-design-patterns-index.html

Empirical Study in Software Engineering	Software Development effort estimation
http://www.computer.org/portal/web/swebok/html/ch6	en.wikipedia.org/wiki/Software_development_effort_estimation
http://www.cs.ucf.edu/~turgut/COURSES/.../PA-chapter11.ppt	www.ehow.com/list_6597159_software-cost-estimation-techniques.html
http://users.ece.utexas.edu/~perry/education/382c/	www.compapp.dcu.ie/~renaat/ca421/report.html
http://www.cs.toronto.edu/~sme/papers/2007/SelectingEmpiricalMethods.pdf	www.uic.edu/classes/idsc/ids505sb/ch23_sd.ppt
http://books.google.co.jp/books?id=fjROrh_UdtoC&pg=PA32&lpg=PA32&dq=empirical+study+in+software+engineering&source=bl&ots=QjVED1_J2s&sig=chrtOwbfWs6nHpnmF5y739mS5HM&hl=ja&sa=X&ei=ZaiAUaCCMtCtkgW2toHwCw&ved=0CIUBEOgBMAc4Cg#v=onepage&q=empirical%20study%20in%20software%20engineering&f=false	users.encs.concordia.ca/~dssouli/INSE%206250%20folder/Survey%20papers%2012%20folder/Software%20Cost%20Estimation%2003.pdf
http://www.google.co.jp/url?sa=t&rct=j&q=&esrc=s&frm=1&source=web&cd=17&cad=rja&ved=0CHgQFjAGOAo&url=http%3A%2F%2Fciteseerx.ist.psu.edu%2Fviewdoc%2Fdownload%3Fdoi%3D10.1.1.133.6712%26rep%3Drep1%26type%3Dpdf&ei=ZaiAUaCCMtCtkgW2toHwCw&usg=AFQjCNHbJmgb-RNYjdNt2cnunw05OqxxoQ&sig2=cHtjQsB5AuvxO1jT0FYT5A&bvm=bv.45921128,d.dGI	www.compapp.dcu.ie/~renaat/ca421/BLTCostEst.ppt
http://www.slideshare.net/sarfraznawaz/empirical-research-methods-for-software-engineering	www.exforsys.com/tutorials/testing/software-cost-estimation.html
http://userpages.umbc.edu/~cseaman/papers/tse99.pdf	www.ecfc.u-net.com/cost/index.htm
http://www.cs.umd.edu/~basili/presentations/2006/Role%20of%20E%20in%20SE%20Irvine.pdf	www.mendeley.com/catalog/software-cost-estimation-models/
http://www.idi.ntnu.no/grupper/su/publ/ese/zannier-studytypes-icse06.pdf	media.techtarget.com/searchSoftwareQuality/downloads/Estimating_SW_Costs_ch03.pdf

Software Maintenance Technique	Software product line
http://www.computer.org/portal/web/swebok/html/ch6	http://www.sei.cmu.edu/library/assets/spl-essentials.pdf
http://www.cs.ucf.edu/~turgut/COURSES/.../PA-chapter11.ppt	http://link.springer.com/book/10.1007/3-540-28901-1/page/1
http://users.ece.utexas.edu/~perry/education/382c/	http://link.springer.com/book/10.1007/978-3-540-71437-8/page/1
http://www.cs.toronto.edu/~sme/papers/2007/SelectingEmpiricalMethods.pdf	http://www.pearsonhighered.com/samplechapter/0131412752.pdf
http://books.google.co.jp/books?id=fjROrh_UdtoC&pg=PA32&lpg=PA32&dq=empirical+study+in+software+engineering&source=bl&ots=QjVED1_J2s&sig=chrtOwbfWs6nHpnmF5y739mS5HM&hl=ja&sa=X&ei=ZaiAUaCCMtCtkgW2toHwCw&ved=0CIUBEOgBMAc4Cg#v=onepage&q=empirical%20study%20in%20software%20engineering&f=false	http://wwwiti.cs.uni-magdeburg.de/iti_db/research/spl-strategies/
http://www.google.co.jp/url?sa=t&rct=j&q=&esrc=s&frm=1&source=web&cd=17&cad=rja&ved=0CHgQFjAGOAo&url=http%3A%2F%2Fciteseerx.ist.psu.edu%2Fviewdoc%2Fdownload%3Fdoi%3D10.1.1.133.6712%26rep%3Drep1%26type%3Dpdf&ei=ZaiAUaCCMtCtkgW2toHwCw&usg=AFQjCNHbJmgb-RNYjdNt2cnunw05OqxxoQ&sig2=cHtjQsB5AuvxO1jT0FYT5A&bvm=bv.45921128,d.dGI	http://www.star.cc.gatech.edu/documents/PeterAbowd/SEI.pdf
http://www.slideshare.net/sarfraznawaz/empirical-research-methods-for-software-engineering	http://www.youtube.com/watch?v=fiG-SdNcjTE
http://userpages.umbc.edu/~cseaman/papers/tse99.pdf	http://www.star.cc.gatech.edu/documents/PeterAbowd/introduction%20to%20product%20Lines.pdf
http://www.cs.umd.edu/~basili/presentations/2006/Role%20of%20E%20in%20SE%20Irvine.pdf	http://www.softwareproductlines.com/
http://www.idi.ntnu.no/grupper/su/publ/ese/zannier-studytypes-icse06.pdf	http://www.youtube.com/watch?v=bj7B5MaQgwE

Figure 10.1. Ideal Result sets for ten topics inside Software Engineering Laboratory. Each topic contains 10 websites which are ranked by participants in expert group. The highlight urls are the overlap results of ASF and Ideal results.



The following instruction was explain to each participant before they begin the tasks

For participants in expert group

Please rank 10 websites that contain the information to complete your given topics.

For participants in experienced and newcomer group

1) First of all, please create an account on ASF. Participant need to login on your ASF account while searching for the information to complete the task.

2) This task contains ten topics which are related to your community. Each topic has 3 question. Participant has one week to answer all the question. There are no right or wrong answers in this study. But ASF will re-ranking webpages based on your search activities. Just do your best to answer each question and you will be able contribute the knowledge for other members inside the community.

3) Before you begin to search, please check out the link inside the relevant topic that has been bookmarked from the previous members. Also please bookmark webpage that you think it contains good information and worth for the others.

4) When participant search or create topic for bookmark, please consider to use the topic name as a part inside the keyword.

1. Ajax tutorial

1.1 What is ajax?

1.2 Please write a diagram how Ajax work? (server and client)

1.3 Please fill in the blank in order to make this code work.

```
<script>
function loadXMLDoc()
{
var xmlhttp;
 xmlhttp=new XMLHttpRequest();

xmlhttp.onreadystatechange=function()
  {
  if (xmlhttp.readyState==4 && xmlhttp.status==200)
    {
    document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
    }
  }
xmlhttp.open("GET","ajax_info.txt",true);
xmlhttp.send();
}
</script>
<button type="button"__________="loadXMLDoc()">Change Content</button>
```

2. Text mining method

2.1 What is the definition of text mining?

2.2 Please list 4 Typical Applications for Text Mining.

2.3 Please list 5 text mining tools that are useful for researchers.

3. Software metrics for agile software development

3.1 Why do we need matrices in software development?

3.2 What is agile software development?

3.3 Please list 6 criterias that you need to measure in agile software development.

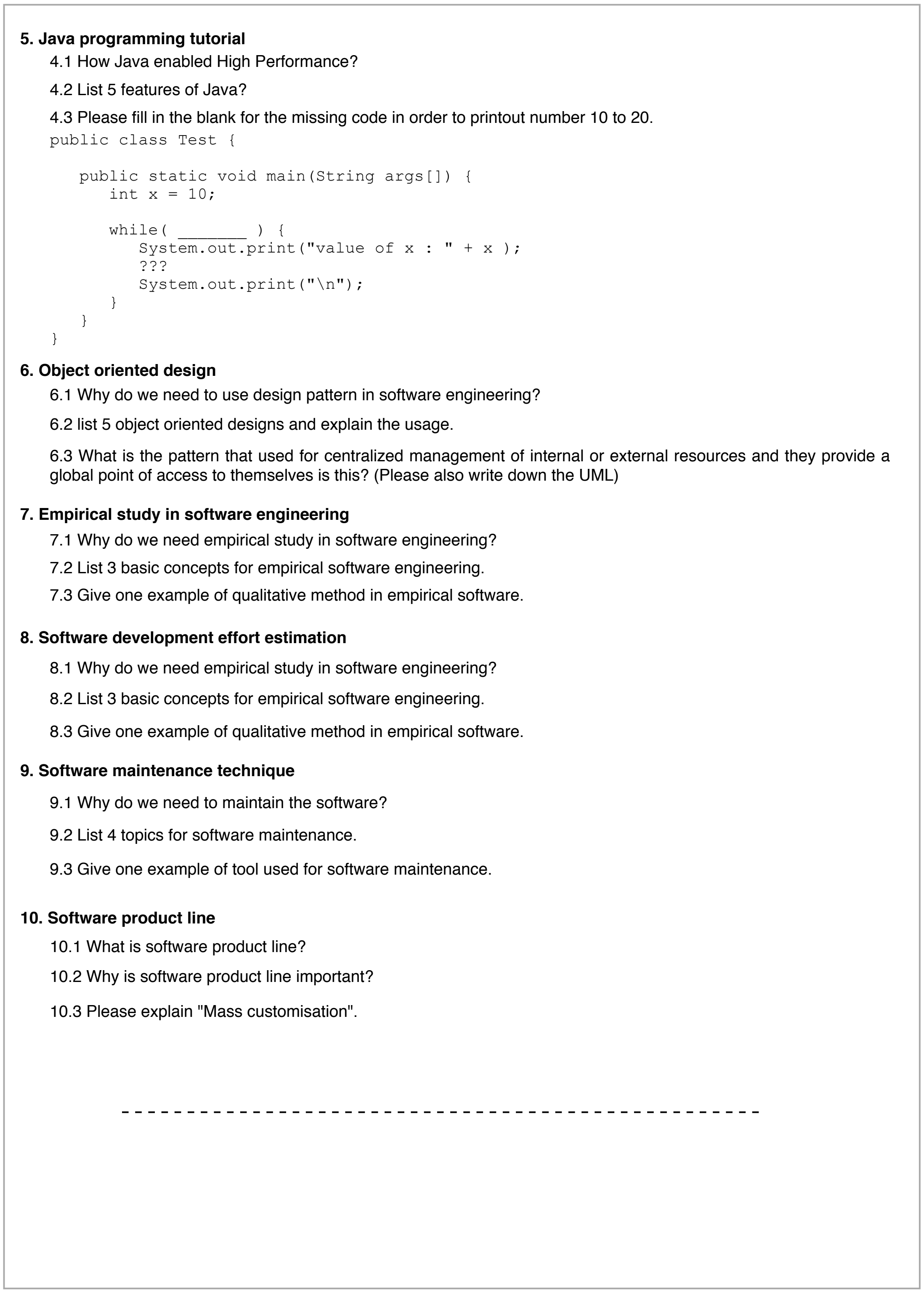
4. Web service

5.1 What is web service?

5.2 List 2 major classes of Web services.

5.3 What is SOAP stand for?

Figure 10.2. Sample of the given tasks. (Page 1)



5. Java programming tutorial

4.1 How Java enabled High Performance?

4.2 List 5 features of Java?

4.3 Please fill in the blank for the missing code in order to printout number 10 to 20.

```
public class Test {

   public static void main(String args[]) {
      int x = 10;

      while( _______ ) {
         System.out.print("value of x : " + x );
         ???
         System.out.print("\n");
      }
   }
}
```

6. Object oriented design

6.1 Why do we need to use design pattern in software engineering?

6.2 list 5 object oriented designs and explain the usage.

6.3 What is the pattern that used for centralized management of internal or external resources and they provide a global point of access to themselves is this? (Please also write down the UML)

7. Empirical study in software engineering

7.1 Why do we need empirical study in software engineering?

7.2 List 3 basic concepts for empirical software engineering.

7.3 Give one example of qualitative method in empirical software.

8. Software development effort estimation

8.1 Why do we need empirical study in software engineering?

8.2 List 3 basic concepts for empirical software engineering.

8.3 Give one example of qualitative method in empirical software.

9. Software maintenance technique

9.1 Why do we need to maintain the software?

9.2 List 4 topics for software maintenance.

9.3 Give one example of tool used for software maintenance.

10. Software product line

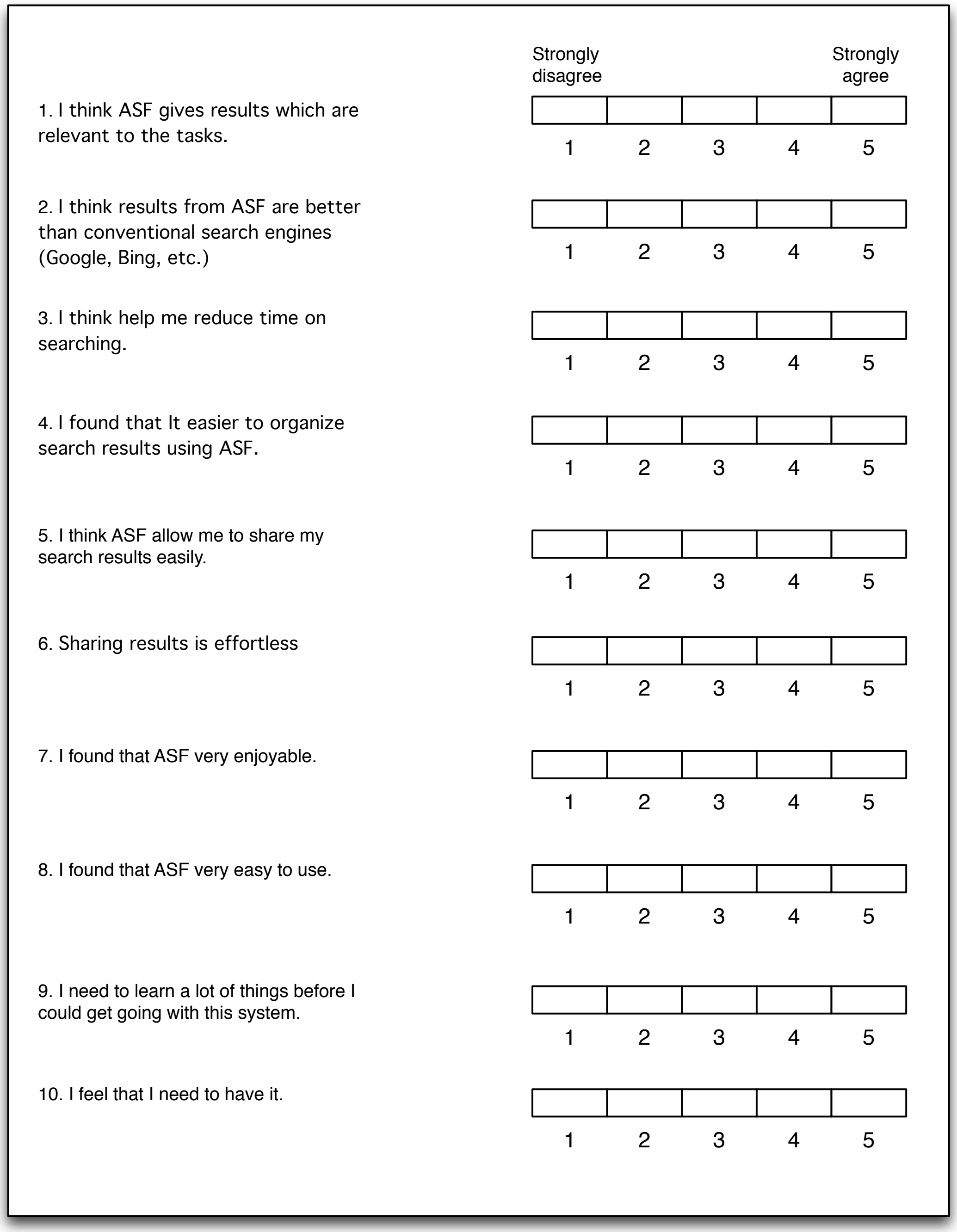
10.1 What is software product line?

10.2 Why is software product line important?

10.3 Please explain "Mass customisation".

- -

Figure 10.3. Sample of the given tasks. (Page 2)



	Strongly disagree				Strongly agree
1. I think ASF gives results which are relevant to the tasks.	1	2	3	4	5
2. I think results from ASF are better than conventional search engines (Google, Bing, etc.)	1	2	3	4	5
3. I think help me reduce time on searching.	1	2	3	4	5
4. I found that It easier to organize search results using ASF.	1	2	3	4	5
5. I think ASF allow me to share my search results easily.	1	2	3	4	5
6. Sharing results is effortless	1	2	3	4	5
7. I found that ASF very enjoyable.	1	2	3	4	5
8. I found that ASF very easy to use.	1	2	3	4	5
9. I need to learn a lot of things before I could get going with this system.	1	2	3	4	5
10. I feel that I need to have it.	1	2	3	4	5

Figure 10.4. System Usability Scale, it consists of 10 statements to which users rate their level of agreement. three of statements are about information seeking, another three are about knowledge sharing and management, and the last four statement are the ease of usage.

Bibliography

[1] C. Castillo, “Effective web crawling,” *SIGIR Forum*, vol. 39, pp. 55–56, June 2005.

[2] P. Anderson, “Understanding communities.” http://www.psawa.com/UnderstandingCommunities.html.

[3] K. Sherif, J. Hoffman, and B. Thomas, “Can technology build organizational social capital? the case of a global IT consulting firm,” *Information & Management*, vol. 43, no. 7, pp. 795–804, 2006.

[4] H. Kamarul, Faizal, *Understanding the determinants of continuous knowledge sharing intention within business online communities.* PhD thesis, Auckland University of Technology, nov 2012.

[5] S. Brain, “Google annual search statistics.” http://www.statisticbrain.com/google-searches/.

[6] S. Kashoob, J. Caverlee, and K. Kamath, “Community-based ranking of the social web,” in *Proceedings of the 21st ACM Conference on Hypertext and Hypermedia*, pp. 141–150, ACM, 2010.

[7] F. Cabrerab, Elizabeth and A. Cabrerab, “Fostering knowledge sharing through people management practices,” *The International Journal of Human Resource Management*, vol. 16, no. 5, pp. 720–735, 2005.

[8] J. Teevan, M. R. Morris, and S. Bush, “Discovering and using groups to improve personalized search,” in *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pp. 15–24, ACM, 2009.

[9] M. R. Morris and E. Horvitz, "Searchtogether: An interface for collaborative web search," in *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, UIST '07, pp. 3–12, ACM, 2007.

[10] C. Shah, "Effects of awareness on coordination in collaborative information seeking," *J. AM. Soc. Information Science and Technology*, vol. 64, pp. 1122–1143, 2013.

[11] Z. Yue, S. Han, and D. He, "An investigation of search processes in collaborative exploratory web search," *Proc. Am. Soc. Information Science and Technology*, vol. 49, pp. 1–4, 2012.

[12] C. Robert, C. Annie, T., H. Katie, A. Jaime, S. Le, and G. Marchionini, "Design and evaluation of a system to support collaborative search," *Proc. Am. Soc. Information Science and Technology*, vol. 49, pp. 1–10, 2012.

[13] A. Ardichvilli, "Motivation and barriers to participation in virtual knowledge-sharing communities of practice," *Journal of knowledge Management*, vol. 7, no. 1, pp. 64–77, 2003.

[14] F. Cabrerab, Elizabeth and A. Cabrerab, "Knowledge-sharing dilemmas," *Organization Studies*, vol. 23, no. 5, pp. 687–710, 2002.

[15] D. E. O'Leary, "Enterprise knowledge management," *Computer*, vol. 31, no. 3, pp. 54–61, 1998.

[16] F. Leistner, *Mastering Organizational Knowledge Flow: How to Make Knowledge Sharing Work.* John Wiley Sons, 2010, illustrated ed., 2010.

[17] P. YONGPISANPON, M. OHIRA, A. IHARA, and K. MATSUMOTO, "Adaptive search framework: Better search result for community," *Journal of The Infosocionomics Society*, vol. 8, pp. 29–43, 3 2014.

[18] J. Grundspenkis, "Agent based approach for organization and personal knowledge modelling: knowledge management perspective," *Journal of Intelligent Manufacturing*, vol. 18, no. 4, pp. 451–457, 2007.

[19] M. R. Morris, "A survey of collaborative web search practices," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1657–1660, ACM, 2008.

[20] M. R. Morris, "Collaborative search revisited," in *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, pp. 1181–1192, ACM, 2013.

[21] M. B. Twidale and D. M. Nichols, "Collaborative browsing and visualisation of the search process," in *IN PROCEEDINGS OF ELVIRA-96, MILTON KEYNES*, pp. 48–7, 1996.

[22] R. Capra, G. Marchionini, J. Velasco-Martin, and K. Muller, "Tools-at-hand and learning in multi-session, collaborative search," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 951–960, ACM, 2010.

[23] K. Fisher, S. Counts, and A. Kittur, "Distributed sensemaking: Improving sensemaking by leveraging the efforts of previous users," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pp. 247–256, ACM, 2012.

[24] G. Golovchinsky, A. Diriye, and T. Dunnigan, "The future is in the past: designing for exploratory search intention within business online communities," in *Proceedings of Information Interaction in Context Symp*, IIIX '12, pp. 52–61, ACM, 2012.

[25] P. Jackson, "A systems approach for human capital management," *Information and Knowledge Management*, vol. 37, no. 4, pp. 399–403, 2007.

[26] S. Jawadekar, Waman, *Knowledge Management: Text Cases*. Tata McGraw-Hill Education Private Ltd, 2011.

[27] "Taxonomy." https://en.wiktionary.org/wiki/taxonomy.

[28] R. Lambiotte and M. Ausloos, "Collaborative tagging as a tripartite network," pp. 1114–1117, 2006.

[29] T. Vander Wal, “Explaining and showing broad and narrow folksonomies.” http://www.vanderwal.net/random/entrysel.php?blog=1635.

[30] M. G. Noll and C. Meinel, “Web search personalization via social bookmarking and tagging,” in *Proceedings of the 6th International The Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference*, ISWC’07/ASWC’07, pp. 367–380, Springer-Verlag, 2007.

[31] G. Scott and H. Bernardo, “The structure of collaborative tagging systems,” *Journal of Information Science*, vol. 32, no. 2, pp. 198–208, 2006.

[32] J. M. Kleinberg, “Authoritative sources in a hyperlinked environment,” *JOURNAL OF THE ACM*, vol. 46, pp. 604–632, 1999.

[33] S. Brin and L. Page, “The anatomy of a large-scale hypertextual web search engine,” in *Proceedings of the Seventh International Conference on World Wide Web 7*, WWW7, pp. 107–117, Elsevier Science Publishers B. V., 1998.

[34] Z. Gyongyi, P. Berkhin, H. Garcia-Molina, and J. Pedersen, “Link spam detection based on mass estimation,” in *Proceedings of the 32Nd International Conference on Very Large Data Bases*, VLDB ’06, pp. 439–450, VLDB Endowment, 2006.

[35] H. Andreas, J. Robert, S. Christoph, and S. Gerd, “A social bookmark and publication sharing system,” in *In Proc. First Conceptual Structures Tool Interoperability Workshop*, pp. 87–102, 2006.

[36] H. Andreas, J. Robert, S. Christoph, and S. Gerd, “A ranking algorithm for folksonomies,” in *Proc. of Workshop on Information Retrieval*, 2006.

[37] F. Abel, N. Henze, and D. Krause, “On the effect of group structures on ranking strategies in folksonomies,” in *In R. Baeza-Yates and I. King, editors, Weaving Services, Location, and People on the WWW. Springer*, 2009.

[38] F. Abel, N. Henze, and D. Krause, “A novel approach to social tagging: Groupme! enhancing social tagging systems with groups,” in *In 4th Int. Conf. on Web Information Systems and Technologies*, 2008.

[39] J. Wang, Z. Chen, L. Tao, W.-Y. Ma, and L. Wenyin, "Ranking user's relevance to a topic through link analysis on web logs," in *Proc of the WIDM '02*, pp. 49–54, ACM, 2002.

[40] T. H. Haveliwala, "Topic-sensitive pagerank," in *Proc of the WWW '02*, WWW '02, pp. 517–526, ACM, 2002.

[41] B. John, "Sus-a quick and dirty usability scale," *Usability evaluation in industry*, vol. 187, p. 194, 1996.

[42] X. Wang and C. Zhai, "Learn from web search logs to organize search results," in *Proc of the SIGIR '07*, pp. 87–94, 2007.

[43] E. Agichtein, E. Brill, and S. Dumais, "Improving web search ranking by incorporating user behavior information," in *Proc of the SIGIR '06*, pp. 19–26, ACM, 2006.

[44] S. Dumais, E. Cutrell, and H. Chen, "Optimizing search by showing results in context," in *Proc of the CHI '01*, pp. 277–284, ACM, 2001.

[45] M. Hearst, "What's missing from collaborative search?," *Computer*, vol. 47, no. 3, pp. 58–61, 2014.

[46] M. R. Morris, J. Teevan, and S. Bush, "Enhancing collaborative web search with personalization: Groupization, smart splitting, and group hit-highlighting," in *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work*, pp. 481–484, ACM, 2008.

[47] K. Sugiyama, K. Hatano, and M. Yoshikawa, "Adaptive web search based on user profile constructed without any effort from users," in *Proc of the WWW '04*, pp. 675–684, ACM, 2004.

[48] J. Teevan, "How people recall, recognize, and reuse search results," *ACM Trans. Inf. Syst.*, vol. 26, no. 4, pp. 19:1–19:27, 2008.

[49] X. Wang and C. Zhai, "Learn from web search logs to organize search results," in *Proc of the SIGIR '07*, pp. 87–94, ACM, 2007.

[50] J. Lave and E. Wenger, *Situated Learning: Legitimate Peripheral Participation.* Cambridge: Cambridge University Press, institute for research on learning report ed., 1990.

[51] J. Jiang, D. He, and J. Allan, "Searching, browsing, and clicking in a search session: Changes in user behavior by task and over time," in *Proceedings of the 37th International ACM SIGIR Conference on Research &#38; Development in Information Retrieval*, SIGIR '14, pp. 607–616, ACM, 2014.

[52] S. Sreekumar and S. Ashish, "Integrating structured and unstructured data using text tagging and annotation," *Business Intelligence Journal*, 2009.

[53] R. Baeza-Yates and E. Davis, "Web page ranking using link attributes," in *Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers &Amp; Posters*, WWW Alt. '04, pp. 328–329, 2004.

[54] J. Schiller, "Modern meeting management and information retrieval: Automatic protocol generation and meeting work," in *Proceedings of the 24th ACM SIGPLAN Conference Companion on Object Oriented Programming Systems Languages and Applications*, OOPSLA '09, pp. 875–880, 2009.

[55] N. Eiron, K. S. McCurley, and J. A. Tomlin, "Ranking the web frontier," in *Proceedings of the 13th International Conference on World Wide Web*, WWW '04, pp. 309–318, 2004.

[56] M. Melucci, "On rank correlation in information retrieval evaluation," *SIGIR Forum*, vol. 41, no. 1, pp. 18–33, 2007.

[57] L. Nie, B. D. Davison, and X. Qi, "Topical link analysis for web search," in *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pp. 91–98, 2006.

[58] S. Park, Y. Kim, U. Lee, and M. Ackerman, "Understanding localness of knowledge sharing: A study of naver kin 'here'," in *Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI '14, pp. 13–22, 2014.

[59] M. Strohmaier, “Purpose tagging: Capturing user intent to assist goal-oriented social search,” in *Proceedings of the 2008 ACM Workshop on Search in Social Media*, SSM ’08, pp. 35–42, 2008.