การปรับโฟรไฟล์สืบค้นให้เป็นส่วนบุคคลด้วยวิธีเสาะหาแบบมด

นางสาว ภัททิรา พินิจค้า

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์และสารสนเทศ ภาควิชาคณิตศาสตร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2552

PERSONALIZATION OF SEARCH PROFILE USING ANT FORAGING APPROACH

Ms. Pattira Phinitkar

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science Program in Computer Science and Information

Department of Mathematics

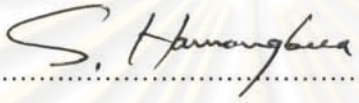
Faculty of Science

Chulalongkorn University

Academic Year 2009

Thesis Title	PERSONALIZATION OF SEARCH PROFILE USING ANT FORAGING APPROACH
By	Ms. Pattira Phinitkar
Field of Study	Computer Science
Thesis Advisor	Associate Professor Peraphon Sophatsathit, Ph.D.

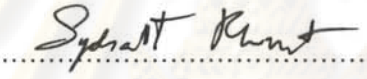
Accepted by the Faculty of Science, Chulalongkorn University in Partial Fulfillment of the Requirements for the Master's Degree



..... Dean of the Faculty of Science

(Professor Supot Hannongbua, Dr.rer.nat.)

THESIS COMMITTEE



..... Chairman

(Suphakant Phimoltares, Ph.D.)

..... Thesis Advisor

(Associate Professor Peraphon Sophatsathit, Ph.D.)

..... External Examiner

(Assistant Professor Kriengkrai Porkaew, Ph.D.)

ภัททิรา พินิจค้า : การปรับโฟรไฟล์สืบค้นให้เป็นส่วนบุคคลด้วยวิธีเสาะหาแบบมด. (PERSONALIZATION OF SEARCH PROFILE USING ANT FORAGING APPROACH) อ. ที่ปรึกษาวิทยานิพนธ์หลัก : รศ. ดร. พีระพนธ์ โสพัศสถิตย์, 73 หน้า.

ในปัจจุบันนี้ข้อมูลข่าวสารและผู้ใช้งานบนอินเตอร์เน็ตเพิ่มจำนวนขึ้นอย่างรวดเร็ว ด้วยเหตุนี้วิธีการในการนำข้อมูลข่าวสารมาเสนอแก่ผู้ใช้งานจึงมีเพิ่มมากขึ้น แต่อย่างไรก็ตาม ยังคงเป็นเรื่องยากที่ผู้ใช้งานจะสามารถสืบค้นข้อมูลข่าวสารที่ตรงกับความต้องการของผู้ใช้งานได้อย่างรวดเร็ว ทั้งนี้เนื่องมาจากผู้ใช้งานแต่ละคนมีความชอบและความสนใจที่แตกต่างกัน ถึงแม้ว่าผู้ใช้งานแต่ละคนจะใช้คำในการค้นหาคำเดียวกัน สาเหตุสำคัญของปัญหานี้คือผลลัพธ์ที่ได้จากการสืบค้นข้อมูลข่าวสารของผู้ใช้งานแต่ละคนที่ใช้คำในการค้นหาคำเดียวกันเหมือนกัน ด้วยเหตุนี้ผู้ใช้งานจึงต้องเสียเวลาในการพิจารณาผลลัพธ์แต่ละอันว่าผลลัพธ์อันไหนที่ตรงกับความต้องการของผู้ใช้งานมากที่สุด

วิทยานิพนธ์ฉบับนี้นำเสนอการปรับโฟรไฟล์สืบค้นให้เป็นส่วนบุคคล โดยผลลัพธ์ที่ได้จากการสืบค้นจะตรงกับความต้องการของผู้ใช้มากที่สุด

จุดมุ่งหมายของวิทยานิพนธ์ฉบับนี้คือเสนอกระบวนการวิเคราะห์และวิธีการที่ทำให้ผลลัพธ์ที่ได้จากการสืบค้นตรงกับความต้องการของผู้ใช้งานมากที่สุดโดยได้รับแรงบัลดาลใจมาจากพฤติกรรมการหาอาหารของมด ขั้นตอนแรกคือการสร้างโฟรไฟล์ของผู้ใช้งานโดยเก็บข้อมูลมาจากกิจกรรมของผู้ใช้งานและนำมาวิเคราะห์ว่ามีใช้งานมีความสนใจในหัวข้ออะไรมากที่สุดโดยเลียนแบบพฤติกรรมการหาอาหารของมด ขั้นตอนที่สองคือการจัดหมวดหมู่ของข้อมูลที่ได้รับมาจากขั้นตอนแรก และขั้นตอนสุดท้ายคือนำข้อมูลที่ได้รับการวิเคราะห์และจัดหมวดหมู่แล้วมาช่วยในการสืบค้นของผู้ใช้งาน ในส่วนของการทดลองนั้นจะพิจารณาผลลัพธ์ที่ได้ว่าตรงกับความต้องการของผู้ใช้งานมากเท่าไร จากผลลการทดลองพบว่ากระบวนการข้างต้นช่วยเพิ่มผลลัพธ์ที่ตรงกับความต้องการของผู้ใช้งาน

ภาควิชา คณิตศาสตร์ ลายมือชื่อนิสิต ภัททิรา พินิจค้า

สาขาวิชา วิทยาการคอมพิวเตอร์และสารสนเทศ ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์ Pirapon Sotthivirat

ปีการศึกษา 2552

5173608123 : MAJOR COMPUTER SCIENCE AND INFORMATION

KEYWORDS : SEARCH PROFILE / SEARCH PERSONALIZATION / WORD SIMILARITY / ANT COLONY FORAGING

PATTIRA PHINITKAR : PERSONALIZATION OF SEARCH PROFILE USING ANT FORAGING APPROACH. THESIS ADVISOR : ASSOCIATE PROFESSOR PERAPHON SOPHATSATHIT, Ph.D., 73 pp.

As the volume of information and users grows rapidly on the Internet, it increases popularity of search as a method for retrieving relevant information. However, it is difficult for users to find relevant documents to their current needs. When users submit a query words to a search engine, users must look through huge of results which most of them are irrelevant to find the relevant ones. The main problem is the search results are selected and presented in the same way for every user. However, each user has his own interests and preferences.

This thesis is devoted to personalization search. The approach provides relevant search results based on the satisfaction of a user's needs.

This approach proposes a three-stage analysis of web navigation that yields search results being relevant to the user's interests and preferences. The approach is inspired by ant foraging behavior. The first stage is to build a user's profile based on user brewing histories and activities at the search sites to be proportional with the amount of pheromone deposited by the ants. The second stage classifies the user's profile data to manage information into concepts in a reference concept hierarchy. The final stage personalizes the search results based on the user's profile. The experiments mainly consider the search results with reference to the user's profile in presenting the most relevant results to the user. The study found that the approach improved the rank order of the relevant search results.

Department : Mathematics Student's Signature ภัทธีรา พินิจคำ

Field of Study : Computer Science and Information Advisor's Signature

Academic Year : 2009

ACKNOWLEDGEMENTS

Developing this thesis has been a challenging and remarkable experience. I would like to express my gratitude to my advisor, Associate Professor Peraphon Sophatsathit, who helped me discover my research interests as well as to find and shape my ideas. I appreciated the discussions with him, his feedback, his friendly attitude, and his assistance in writing reports throughout my thesis. I also appreciate Assistant Professor Dr. Saranya Maneeroj, Dr. Suphakant Phimoltares, and Assistant Professor Dr. Kriengkrai Porkaew for their supportive suggestions. The experience and knowledge that they provided were particularly enlightening and helpful for my research.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I
INTRODUCTION

As the amount of information available on web pages causes information overload problem, it is becoming difficult for users to find the relevant documents for their need. Accordingly the demand for personalized approaches for information access increases. Personalized approaches address the overload problem by gathering, storing, and analyzing users' information and representing the relevant information for individual users. The traditional search engines return the same list of search results for every user. In fact, different users usually have different needs. The search personalization must study the users' behavior as they interact with information sources.

1.1 Motivation

The learning problem of search results has gained the irrelevant search results. As the volume of information grows rapidly on the Internet, more investment on web search engines follows suit. However, search engine returns the search results based on user's query keyword by PageRank algorithm, rather than user's interests. This PageRank algorithm is based on the frequency of keywords, link popularity, and the frequency of query negotiation. When users enter a query, the search engine must look through hundreds of search results to find the relevant search results. The core problem is that search results are presented in the same way for every user. For example, two people searching for "palm", the first person is looking for information on the Personal Digital Assistant (PDA) and the other person is looking for a palm tree, but they will get exactly the same search results. Determining the relevance of search results mostly relies on the user's own background, interests and preferences.

To address this issue, a solution that can personalize the information selection and presentation is needed. User's profile is an important component of personalization system. A user profile represents the interests and preferences of a specific user that can be used to supplement information about the search. This information could be used to narrow down the number of topics considered when retrieving the results, increasing the likelihood of including the most interesting results from the user's perspective.

Since user's interest and preferences always change over time, the user's profile must be updated to keep it up-to-date in the same manner as ants that are always looking

for a new food source and updating the new path to the colony. Thus, the fundamental principle of user's profile creation is inspired by the nature of ant colony foraging behavior. An ant leaves pheromone chemical as a communication means on the quality and quantity of food found at a source when foraging for food. The amount of pheromone deposit can be used as the weight of interesting items. Therefore, pheromone update keeps a user's profile up to date at all times.

The approach provides a straightforward methodology to build a user's profile based on interest scores which are derived from pheromone deposited by the ants. This profile reflects the user's behavior as pheromone being accumulated or evaporated. In the mean time, the content keywords of user's profile are classified in a reference concept hierarchy. The content is systematically processed and verified with the help of a set of experiments to carry out personalized search according to the approach.

1.2 Current Technologies and Problems

Major limitations of search engines nowadays fall into some categories as below.

1. Lack of personalization in accessing the information. Many sources that are available on the web, offer different kinds of users. However, these resources increase the information overload situation. Every user is commonly treated in the same way, while they have different interests and preferences. Hence, this problem needs a method that can help filter out irrelevant information.
2. Lack of timeliness in being informed about new available information. Users often do not have time to undergo a time consuming of manual search to keep themselves informed and up-to-date.

The above limitations are example of search problems that modern search engines are trying to enhance their search results so that they can better anticipate the intensions of the users.

The search engines have come a long way since their modest beginning. They have already developed through two major stages. The first stage was based simply on matching keywords in documents which the same search results were shown to all users. Therefore, there was a limit to the effectiveness of keyword matching. When two users typed the same search words, they might be looking for something completely different. The second stage examined how users interact with the search engine to predict their intent. Several options surfaced, the first one was creating some types of user's profile to collect information about

their interests and preferences, either by having them complete a form, recording their activities and search histories. Unfortunately, users often hesitated to share their personal information with search engines. Besides, the search histories might not be helpful in predicting user's future intention. Thus, the aggregated information collected from a large number of interactions between users and search engines were results.

Personalization will have a big impact on the way users search. According to the evolving stages of search engines, the focus of search engines has changed from matching keywords to providing the user's interests and preferences.

In general, personalization can be applied to searching in two different ways:

1. By providing tools that help users organize their own past searches, preferences, and visited URLs.
2. By creating and maintaining sets of user's interests, stored in profiles, which can be used by retrieval process of a search engine to yield better results.

The first approach is applied by many new toolbars and browser add-ons. The Filangy Search History Tool is an example of tool that tries to help users organize their search histories and web pages visited. It provides automatic full text caching of every web page viewed in user's browser and integrates selected search results from user's cache into resulting clustered pages. However, privacy becomes a serious issue since https pages are not cached. Fortunately, A9, Ask, Google, and Yahoo Search Engines also provide history search which users can perform further search within the previous search context. For example, if a user is looking for something related to "cars" that he has searched previously but didn't recognize exactly how it was done, he could search for "cars" and find all the queries containing that word.

Recently, search engines have been improved with personalization features according to the above second approach. For instance, Google My Search History provides history feature which automatically keeps track of all web searches and every page that the user has viewed from search results. My Search History differs from automatic caching feature in Google. My Search Results does not save the web pages but it saves user's search behavior and provides easy way to rediscover both user's past queries and the search result pages. Moreover, Yahoo introduces Personal Search with personalization features including search history, the ability to save pages to a personal web, and block URLs from appearing in search results.

All these systems have interesting features that can guide users to find better information but they represent the user's search requirements with overall profile rather than

trying to identify specific topics of interest.

The thesis approach focuses on personalization in search based on implicit feedback. Many implicit feedback systems capture browsing histories through proxy servers or desktop activities through installation of bots on a personal computer. These technologies require direct participation of the user in order to install the proxy server or the bot. Desktop bots can capture all activities whereas proxy servers can capture all Web activities. The approach demonstrates that profiles created from this information can be used to identify, and promote relevant results for individual users. However, the thesis approach automatically captures user's information from user's web page visited and selects search results by means of installed software on a personal computer.

1.3 Research Objectives

The main objective of this thesis is to offer the most relevant search results to a user by personalization search according to user's profile.

To achieve this objective, the thesis addresses three separate goals:

1. Building a user's profile to represent the user's interests and preferences.
2. Classifying the user's profile data to keep track of the objects of similar properties.
3. Personalizing search results to provide the most relevant search results to the user.

1.4 Scope

The thesis work will confine to the following areas:

1. Support only the web pages which contain information of URL, tag<title>, and tag<meta name="description"> relating to search contents.
2. Employ word relations based solely on WordNet.
3. Encompass four sets of word categories, namely, Technology, Zoology, Botany, and Finance.
4. Test the experiments on Yahoo search engine and Yahoo Motif.
5. Support only keyword style query.
6. Support only English language.

1.5 Research Methodology

Research approach of this thesis adopts ant colony foraging algorithm to perform both gathering the user's interests and updating the user's profile. The process employs cosine similarity to rearrange the order of search results which the first one denotes the most relevant to the user in order to reduce time consumption.

The approach consists of three phases:

Phase 1: Building the user's profile.

The user's profile represents the user's interests, preferences, and background. It consists of a set of categories, each of which encompasses a set of elements and its corresponding weight. The fundamental principle of user's profile creation is inspired by the nature of ant colony foraging behavior.

Phase 2: Classifying the user's profile data.

After collecting user's information to be archived in the user's profile, some of this information may be similar by category, others may be different. Organizing this information is a necessary requirement to keep track of the objects of similarity properties. The approach utilizes WordNet and Leacock-Chodorow measure in the classification process.

Phase 3: Personalizing the search results by means of the user's profile.

A typical search query often ends up with plentiful results that contain few relevant ones. Search personalization mechanism can reduce unwanted search results by reordering search results with their relevance to suit the user's profile.

1.6 Benefits

The benefits of this research work are increasing effectiveness and efficiency of search engines, while appreciably decreasing the time it takes to find the desired information.

1.7 Thesis Overview

This thesis is organized as follows. Chapter 2 describes a general background of related studies. A detailed discussion of underlying technologies and existing approaches are included in Chapter 3. Chapter 4 describes the thesis methodology. The results from

the thesis methodology are reported in Chapter 5. Chapter 6 summarizes the overall of this thesis and some possible future work is also proposed.

CHAPTER II
THEORETICAL BACKGROUND

Before the World Wide Web (www) was first introduced back in the 1990's, the Web was first started as a networked information project. The underlying principle of the project was simply to provide a convenient way for users to browse and contribute their information over the digital world. Eventually, all users around the world could share their information freely over the Web. However, it was difficult that the freedom of sharing information suit the users' different needs. Frequently, when users performed a simple search, the users would find numerous search results which were usually irrelevant to their expectation.

As the web evolves, so does user's demand increase. Searching becomes an important activity in daily lives. During the interaction with search engines, each user obviously generates valuable personal search history data. Logging of personal search history offers a means to learn about the user's interests and preferences, thus improving the search results for the user.

Unfortunately, user's interests and preferences change periodically to keep them up to date at all time. This is akin to ant colony behavior. When ants find a plentiful food source, many ants will go there to retrieve food and bring back to their nest. An route on their transport, the ants lay pheromone which results in pheromone deposit. If they find a new food source which is more abundant than the old one, they will leave the old food source and head for the new food source. Therefore, the pheromone deposit on the new path soon becomes stronger than the old path. This process calls for constant update of pheromone value as one want to mimicking ant foraging behavior.

Ant colony algorithms were initially used to solve combinatorial problems, such as the well known Traveling Salesman Problem, to arrive at an optimal solution, and hence the Ant Colony Optimization (ACO). However, the usefulness of the ACO algorithms is expanded in other scientific areas like data mining and web search. Accordingly, this chapter is dedicated to give an account on background which relates to several research fields, e.g., ant colony optimization, information retrieval, evidence for document relevance, and personalization.

2.1 Ant Colony Optimization

Ant colony optimization is a technique for optimization which was introduced in the

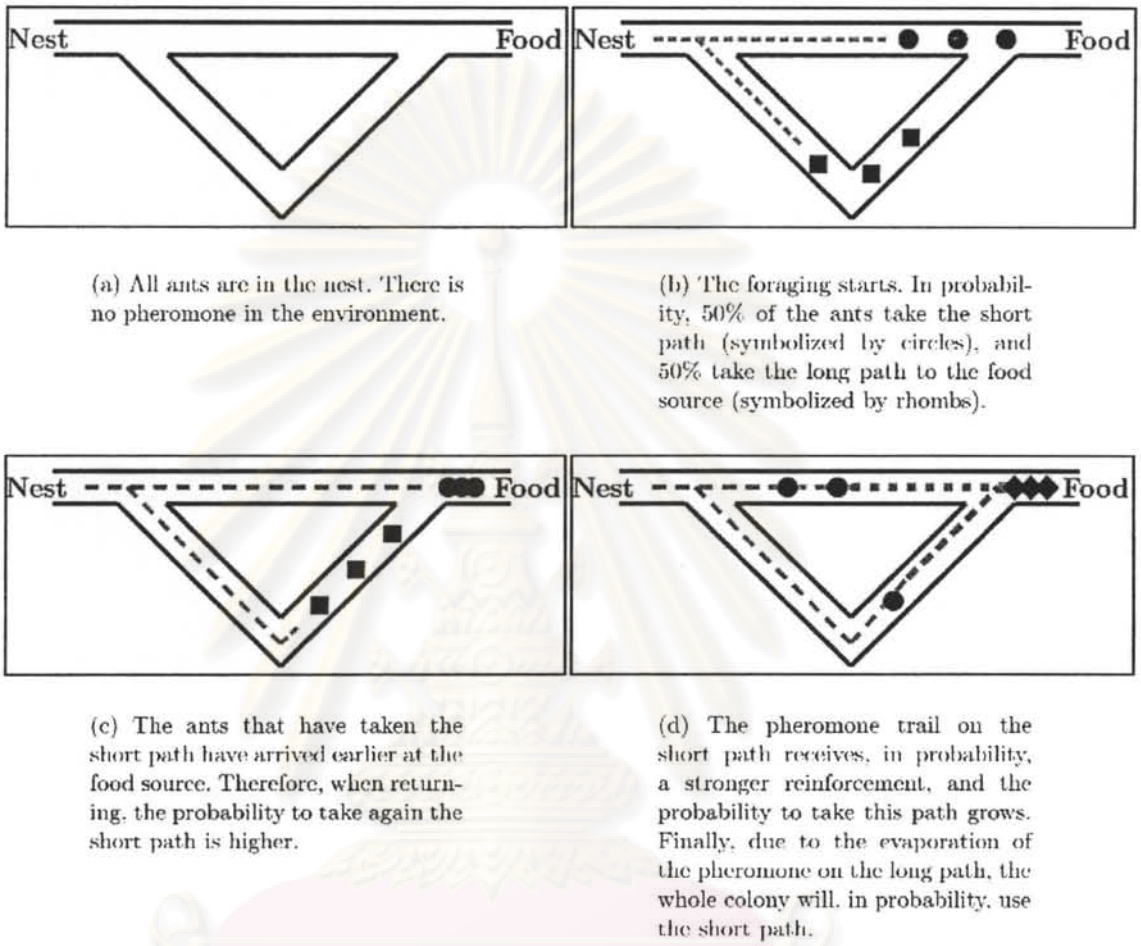
1990's. Ant colony optimization is inspired by the foraging behavior of real ant colonies. Ant Colony Optimization (ACO) is a branch of a newly developed form of artificial intelligence called swarm intelligence. Swarm intelligence is a field which studies the emergent collective intelligence of groups of simple agents. In groups of insects, which live in colonies, such as ants and bees, individual agent can only do simple tasks on its own, while the colony's cooperative work is the main reason determining the intelligent behavior it shows. The core of ant behavior is the indirect communication between the ants by means of chemical pheromone trails, which enables them to find short paths between their nest and food sources.

Most real ants are blind. However, each ant deposits a chemical substance on the ground called pheromone while it is walking. Pheromone encourages the following ants to stay close to previous moves. Dorigo's experiments [1] demonstrated the complex behavior of ant colonies. For instance, a set of ants built a path to some food. A barrier with two ends was placed in their way such that one end of the barrier was more distant than the other. In the beginning, equal numbers of ants spread around the two ends of the barrier. Since all ants had almost the same speed, the ants going around the nearer end of the barrier returned before the ants going around the farther end. With time, the amount of pheromone the ants deposit increased more quickly on the shorter path, and so more ants preferred this path. This positive effect is called autocatalysis. The difference between the two paths is called the preferential path effect. It is the result of the differential deposition of pheromone between the two sides of the obstacle since the ants following the shorter path will make more visits to the source than those following the longer path. Because of pheromone evaporation, pheromone on the longer path fades away with time.

2.1.1 The Origins of Ant Colony Optimization

Marco Dorigo [1] introduced the first ACO algorithms. The development of these algorithms was inspired by the ant colonies. Ants are social insects. They live in colonies and their behavior is governed by the goal of colony survival rather than being focused on the survival of individuals. When searching for food, ants initially explore the area surrounding their nest in a random. While moving, ants leave a chemical pheromone trail on the ground. Ants can smell pheromone. When choosing their way, they tend to choose paths marked by strong pheromone. As soon as an ant finds a food source, it evaluates the quantity and the quality of the food and carries some of it back to the nest. During the return

trip, the quantity of pheromone that an ant leaves on the ground may depend on the quantity and quality of the food. The pheromone trails will guide other ants to the food source. This is shown in Figure 2.1.



(a) All ants are in the nest. There is no pheromone in the environment.

(b) The foraging starts. In probability, 50% of the ants take the short path (symbolized by circles), and 50% take the long path to the food source (symbolized by rhombs).

(c) The ants that have taken the short path have arrived earlier at the food source. Therefore, when returning, the probability to take again the short path is higher.

(d) The pheromone trail on the short path receives, in probability, a stronger reinforcement, and the probability to take this path grows. Finally, due to the evaporation of the pheromone on the long path, the whole colony will, in probability, use the short path.

Figure 2.1: An experimental setting between the ants' nest and the food source exist two paths of different lengths that demonstrates the shortest path finding capability of ant colonies

2.1.1.1 The Cataglyphis and Ocymyrmex Species

It was mentioned briefly that ACO is based on the foraging behavior of ants, such as how the ants locate and collect food. However, various species of ant exist, each of which exhibits a distinctly different foraging behavior. It is more accurate to describe ACO as being inspired by the recruitment strategy of ants which use chemical markers to mark the

location of a plentiful food source. This section describes two different species of ant, the cataglyphis and ocymyrmex ants.

The cataglyphis and ocymyrmex ants have developed to fill the unique ecological niche. Both of these species rely on the harsh environment to provide them with food, since they search their local area for insects which have succumbed to the extreme heat and stress of this harsh environment. The interesting about cataglyphis and ocymyrmex ants is that they don't use a chemical marker to recruit other ants to a food source. They rather use an internal memory to influence their own choice of direction to travel from the nest. The rules for movement of this two species are described below:

- Continue to forage in the direction of the preceding foraging trip whenever this trip has been successful in finding food.
- If foraging trip is unsuccessful then abandon this direction and randomly select a new direction, decreasing the probability of doing so as the number of previously successful runs increases.

This individual behavior leads to an efficient foraging pattern for this specific environment which in the absence of food will result in the ants searching the environment at random, or in the case of food being found will subtract foraging resources away from the global pool of resources to exploit this discovered food source until it is consumed. The colony will revert to its initial behavior with a small bias towards searching previously promising areas, and if food is found again in this biased area then this ant will die out reverting the colony back to its initial completely random state.

Interestingly, if one were to hypothesize about placing two different abundant food resources within close and far foraging range of the colony that the emergent effect would be that the members of the colony would split randomly between the resources and stick to one even though one resource is better. As there is no intra-colony communication mechanism, which would allow the colony to converge on the better resource. However, since these species have evolved to exhibit a decentralized control which argues well with their inherently unstable and dynamic environment.

2.1.1.2 The Tetramorium Caespitum Species

Tetramorium caespitum ants are quite distinct from the desert dwelling ants described above. These ants have developed to suit a different ecological niche where food sources are wealthy and the desired effect is to optimize the distribution of resources to maximize

the food collection activity. These ants also rely on randomness to influence their decision making behavior, however with the absence of a long term memory they rely more on intra-colony communication mechanisms to influence their foraging decisions.

This species of ant exhibits three distinct behaviors, group-recruitment, mass-recruitment and random exploration. Group recruitment occurs when an ant finds a new food source, returns to the nest and upon returning to the nest attempts to coerce other ants to follow it back to the food source laying pheromone along the trail as they move. This group recruitment will eventually lead to a mass-recruitment if the food source is large enough and the pheromone trail is reinforced enough so that ants can follow the pheromone trail. Random exploration can occur at any stage where an ant following a pheromone trail decides to leave the trail to search virgin territory in the hope of finding more food or a more efficient path to already discovered food. The probability of such an event occurring is inversely proportional to the amount of pheromone and directly proportional to the distance away from the nest.

2.1.1.3 The Iridomyrmex Humilus Species

The importance of randomness in the model above is to encourage exploration and avoid exploitation of one food source neglecting other possibly more rich food sources or shorter paths. This emergent effect was perhaps most profoundly demonstrated in the double bridge experiment performed by Denebourg [2]. In this experiment a single food source was placed away from a nest of Iridomyrmex humilus ants and two bridges of unequal length connected the nest to the food. Initially the ants were observed to use both bridges fairly equally to retrieve the food, however eventually the majority of the colony favored the shorter branch over the longer branch. The researchers explained this emergent effect by the fact that a shorter distance means that ants can forage on this path more quickly and over time this branch will be positively reinforced with more pheromone.

2.1.2 Ant Inspired Search Algorithms

The ant behavior as described above leads to complex emergent properties such as efficient resource allocation and shortest path finding. Therefore, these emergent properties exist without the requirement for centralized control of colony resources. This leads to the development of new subfield loosely referred to as ant inspired algorithms.

2.1.2.1 Ant Systems

The barrier with two ends experiment lead to the development of three algorithms by Dorigo, Ant-density, Ant-quantity and Ant-cycle for application to the traveling salesman problem (TSP). In these algorithms each ant iteratively constructs a solution to the TSP by probabilistically selecting an edge to include in the growing tour based on a nearest neighbor heuristic and an artificial pheromone which is adapted by the artificial ants as the search progresses, the specific way that this pheromone is added and adapted is what distinguishes these algorithms. After experimenting with the three simple models ant-cycle was shown to be the most effective at optimizing the TSP problems addressed. The Ant-cycle algorithm was later refined and reintroduced as Ant Systems (AS).

2.1.2.2 An Ant-inspired Vector-based Algorithm for Continuous Spaces

The ant colony metaphor for searching continuous design spaces proposed by Bilchev and Parmee [3] references the double barriers experiment and the work of Dorigo as its inspiration. This algorithm starts by selecting a 'nest' in a continuous n-dimensional space that is found by running a global search process on the search domain. Vectors are projected at random from this position into the local area around and a series of random jumps are made from these initial vectors until a termination criterion is met. If the resultant position is better than the initial vector than the initial vector is replaced. This process is continually repeated with higher quality vectors allocated more computational resource than poorer quality vectors. This algorithm is essentially a local search algorithm which probabilistically selects a solution from a population of solutions and makes a random change to it. The original solution is replaced if the random change is beneficial, and the probability of selecting this solution as a starting point is adjusted positively or negatively based on the result of the random change.

2.2 Information Retrieval

The general task of information retrieval (IR) is searching for information within documents. Document is a general term which refers to unstructured records in a relation database or pages in World Wide Web. Regularly, the documents are not stored directly in the information retrieval system, but are instead represented in the system by document

surrogates or metadata. The goal of information retrieval is to take a query and return a set of documents relevant to the query.

The information retrieval process begins when a user inserts a query into the system. In information retrieval a query does not uniquely identify a single object in the collection. Instead, several objects may match the query, perhaps with different degrees of relevancy.

Most information retrieval systems compute a numeric score on how well each object in the database match the query, and rank the objects according to this value. The top ranking objects are shown to the user.

2.2.1 Information Retrieval Process Overview

An information retrieval system consists of several interconnected modules as shown in Figure 2.2. The two basic processes are building an index and querying the index. The index, which is an important part of information retrieval system, contains the searchable features and enables fast query answering.

Building a search index starts with a crawler. Search engines use a crawler program to find new documents in a database. In case of the World Wide Web, discovery of new documents is accomplished by starting from a set of beginning URLs and then recursively following all hyperlinks from the beginning pages. The crawling process of file share repositories is quite similar to Web crawling. Since no hyperlinks exist in file shares, the crawler simply uses the operating system directory listing command in order to find new documents. Hence, it recursively scans the directory tree using a breadth-first-search or depth-first-search approach. Other types of repositories are crawled in a similar fashion. The crawler implementation needs to know the search scope, which is highly dependent on the targeted information retrieval application. In one case, search in the full-text content is enough. In another case, search in full-text and metadata is relevant. In other scenarios, only metadata is considered. Literature retrieval tools might be restricted to search in metadata in order to protect the publisher's rights on the full-text content.

The output of the crawler is delegated to a text processing engine where several text manipulation tasks are conducted. For example, splitting sentences into words, removing of stop words, reducing words to their root form, and synonym injection. The mean of the text processing pipeline is to identify index terms which best describe the content of a document and help to differentiate it from others. An index term is a content-bearing key

which needs not be single word but may be multi-word units. For instance, the text "sony notebook" might be processed to the single index term "sony notebook" or to the index terms "sony" and "notebook", depending on what is relevant for the application field. The important point is that the same text processing pipeline must be applied to a user's query. Otherwise, difference between indexed content and queries could lead to inconsistency in the search results.

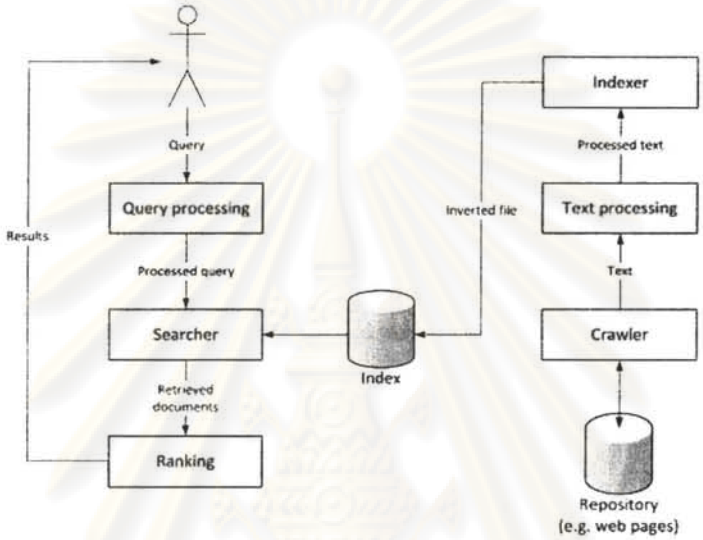


Figure 2.2: Information retrieval process

In the next step, the processed text is transferred to the indexer which builds an inverted file. The inverted file is an index structure where each indexed term is linked to the document where it occurs. The benefit of using an inverted file is that documents matching a query terms can be found efficiently.

Once the index is constructed, the information retrieval system can be queried by its users. Comparable to the text processing step, the query is first analyzed by a text processing engine in exactly the same way, before it is transformed into an internal representation. The processed query is transmitted to the searcher.

The searcher processes the query and retrieves all relevant documents from the index. At this point, the investment of building the index pays off because thousands of queries can be answered within milliseconds.

In the final step, the retrieved documents are passed to the ranking module. This module plays a crucial role, as its task is to order the results by relevance. The documents

matching best the user's information need to be located at the top of the result list.

2.2.2 Document Selection

Document selection is the collection of documents that are relevant to the user's query. Selected documents are presented based on similarity between the query and documents.

Construction Information Broker (CIB) [4] improves the retrieval process by using user's profiles and a type of collaborative information retrieval. Each profile is associated with a set of interests and preferences which include a set of keywords and information context. These profiles can be used as independent key for formulating queries. When a user uses the system, the user can select one or more standard profiles according to particular interests. The user's profile is personalized during the process of querying. When a user downloads a document and gives a high rating to the document the client agent on the local computer will extract key concepts from the document and use them to improve the user's profile. The updated user's profiles will be communicated to the server. The user can also explicitly modify the user profile by creating new interests, adding keywords and giving weight to existing keywords.

2.3 Evidence for Document Relevance

The relevance of a document [5] can be based on various sources of evidence. This section gives a brief introduction to several other sources of evidence.

Scoring metrics, which are evidence for document relevancy, can be classified into two groups as query dependent evidence and query independent evidence. Query dependent evidence means that the relevancy of documents depends on the query. Query independent evidence is a static value which is associated with a document. The PageRank value is a query independent metric. Similarly, a metric which scores recent documents higher than older ones, is also a query independent metric because it relies on the document's last modification date to calculate relevancy.

Craswell et al [6] presented three methods for combining query independent evidence with a query dependent baseline. In the first approach, rank_based, the baseline score and the query independent scores are transformed into two rankings. The merged score is computed based solely on the items' position on the order of the items. A benefit of

this approach is that power law scores can be combined with linear scores without introducing a potential bias towards a few pages having a very large score and the majority of pages having no score. This method could be applied to combine the PageRank's power law distributed scores with the tf-idf scores. In the second approach, language modeling prior, a prior is calculated for each item which is then combined with the language modeling probability by multiplication. The last approach, relevance score adjustment, uses a linear combination of the baseline score and the static score to calculate the ranking of results.

2.3.1 Content Evidence

Content evidence covers any information which can be gathered from the indexed object such as full text, meta fields, structural information and anchor text. The full text content of objects is a standard source for relevance ranking used by almost all search engines.

Data provided in meta fields is often considered to be of higher quality than data in full text content. For example, the query word "JAVA" is found in the title meta field of a document d1 and the same word is also found in the full text content of a document d2, then d1 is considered more relevant than d2.

Similar to meta fields, text within certain structural parts of documents is considered more relevant. For example, a document heading gives a good summary of the document. Therefore, if a word is matched in both heading and paragraph, the match in the heading should be considered more relevant.

Links, which occur in a document, could be described as part of structural evidence. The label of a link gives valuable information about the target's content. In other words, an anchor's text is like a tag that gives a good summary of the target's content.

2.3.2 Context Evidence

Context evidence such as IP address, cookies, session, location and user id is often used by adaptive systems to adjust the ranking of results. Common examples encountered in web search engines are called location based services. For instance, a user who search for "Greek restaurant" are shown restaurants from user's area. The location is automatically deduced by mapping the user's IP address to the geological position.

2.3.3 Time Evidence

Timestamps are an important part of many information objects. Typically, the relation between a timestamp and an information objects reflects an event which occurred at the specified time. Common events are creation, modification, and deletion of objects.

Time has an important role when searching for information. In case of news, users are mostly interested in the latest articles. For instance, a user might only be interested in the latest news about "JAVA." This knowledge is often incorporated in a search engine's ranking heuristic. While this assumption might be suitable for the greater part, it ignores users who are looking for older news. According to previous example, a user might not be interested in the current technology of JAVA but be interested in the JAVA evolution.

A better approach would be to consider the distribution of relevant documents over time. According to Li and Croft [7] there are three types of queries. The first type of query has a uniform distribution of relevant documents over time. The second type of query favors very recent documents, and the third type of query has most relevant documents within a specific period in the past. The authors' method outperforms the baseline models. In particular, it outperforms the linear combination method used by most commercial search engines.

2.3.4 Hyperlink Evidence

Link information becomes one of the most important sources of evidence for ranking results. However, this kind of evidence is not very useful in the part of intranets due to structural differences. Further, other file formats, such as PDF, Word, Excel, etc, usually lack a comparable linkage structure. In order to extract hyperlinks, the need is to apply text analysis. Because there are barely explicit URLs in non-HTML documents, extracting citations with Natural Language Processing (NLP) techniques is a difficult task.

2.3.5 URL Evidence

A URL contains valuable information for the ranking of search results. The length of a URL could show the authority of a page. If two pages have similar content, then the page with the shorter URL has probably a higher authority than the other. Pages located at the top can be considered more general than pages located in deeper hierarchies.

2.6.6 Feedback Evidence

Implicit feedback describes evidence which can be gathered through observation of a user's interactions with the search engine. The extracted evidence can be query dependent as well as query independent.

Closely related to implicit feedback is explicit user feedback. In contrast, explicit feedback is usually of a higher quality of representing user's interests but not as abundant as implicit information. The information can be used in the same manner as is done with implicit feedback.

2.4 Personalization

Personalization is the process of presenting the right information to the right user at the right moment by building, managing, and representing information customized for individual users. Web Personalization is a set of actions that can tailor the web experience to an individual user or a group of users. Typical web experiences are users browsing patterns, sequence of pages visited by a user, usual query patterns, etc. Most web personalization systems are based on some types of user's profile. A user's profile is a collection of personal data associated to identify topics of interest to a specific user. The user's profile may include demographic information, such as name, sex, age, occupation, interests, and preferences. In order to construct individual user's profile, information may be collected explicitly by the user or implicitly gathered by software agents. Moreover, information may be collected on the user's client machine or gathered by the application server itself.

Explicit user information collection relies on personal information input by the user, therefore the main problem with explicit construction of user's profile is that user may provide inconsistent or incorrect information. While the user's profile remains static, user's interests and preferences may change over time. Construction of the user's profile often places a burden on the user which may increasingly become inaccurate.

Implicit user information collection does not require any additional intervention by the user during profile construction process and provides an unbiased way to collect information. Kelly and Teevan [8] offer the type of information about user which can infer from the user's behavior. Browsing histories are a common source of information from user's interests and preferences to be extracted. These browsing histories contain the web pages

visited by the user and the times and dates of the visiting. Hence browsing histories can provide personalized services. The drawback of this technique is only collecting the user's browsing histories from a single computer. Nevertheless, the users could share their browsing histories from multiple computers or install the same proxy server on each computer they use habitually. Moreover, the users could share their browsing histories by a login system, using the same user's profile in several locations.

Numerous personalization approaches use agent to collect user's information during browsing. The agent is implemented as either a stand-alone application which includes browsing abilities or a plug-in to an existing browser. The agent is installed in the user's computer, so it is able to capture the user's activities while the user browses. Furthermore, the agent can collect richer information on web page such as URLs visited, time spent on each web page, bookmarking, and downloading. Letizia [9] was one of the first system that interactively collect and exploit implicit user's feedback based on formerly visited web pages. The system suggests links on the current web page that user might interests. Personal WebWatcher is another example of using agent that performs more significant tasks such as highlighting relevant hyperlinks to the user, recommending URLs, or refining search keywords.

The disadvantage of browser agent is that the system requires the users to install a new application on their computer and use it during browsing instead of a standard browser. The browser agent focuses on collecting users' information as they browse. The browser agent captures and shares the users' activities on their computer, which is called client-side approach. All client-side approaches place some burden on the users in order to install a new application on their computers.

In contrast, the other approach collect user's information from user's activities while interacting with the site that provide personalized services. This is called server-side personalization. This approach places no burden on the user and can silently collect the information by cookies, logins, and session IDs. User's information can be collected from two main sources, browsing activity on the site and search interactions.

The server-side personalization approach has the advantage that the users do not need to install a new application on their computers to collect their information. The service is providing personalized search collecting while the users interact directly with the site. In the case of the site requires a login process, the users can use the same user's profile from everywhere, they do not need to access the site from the particular computer. On the other hand, the disadvantage of the server-side personalization approach is less available

information because only the activities at the site itself are tracked. However, several projects have been successful to provide personalized search by building the user's profile based on this information.

User's profiles which can be modified are considered as dynamic, on the contrary to static user's profiles which maintain the same user's information over time. Dynamic user's profiles that take time into consideration may discriminate between short-term and long-term interests. Short-term user's profiles signify the user's current interests while long-term user's profiles represent user's interests which are not subject to frequent changes over time. For instance, consider a programmer who uses search engines for his daily search. One day, he wants to take a vacation and uses the search engines to look for airplane ticket, hotel, car rent, etc. His profile should reflect his programming interests as long-term interests, and the vacation interests as short-term interests. Once the user returns from his vacation, he will continue his programming research, and the vacation information in his profile should be forgotten. Since they can change rapidly as users change tasks, less information is collected, so short-term user's interest is harder to identify and manage than long-term interests. Generally, the purpose of user's profile is to collect information about the subject that a user is interested, and the length of time that present this interest, in order to improve the quality of information access and infer user's intentions.

In summary, to generate user's profile for personalized search, previous studies have asked users for explicit information or collected implicit information. Nonetheless, users habitually are unwilling to provide explicit information. Implicit information has obtained users' information by observing their interactions, but implicit information will take time and may raise privacy concerns. In addition, a user's profile which generated from implicit information may contain noise because the user's interests and preferences have been estimated from user's activities and not explicitly specified.

2.4.1 Personalization Based on Search Histories

User queries are an important source in recognizing the information needs and personalizing the user's interaction. For example, if the user submits a short query, such as apple. It is not clear that the user is looking for the computer company, or a fruit. The browsing or query history could be a way to weigh the different alternatives. If the user has recently searched for an iphone, the apple query is more likely to be related to product of the computer company.

The approaches which based on search history can be organized in two groups. Firstly, offline approaches exploit history information in a distinct pre-processing step, usually analyzing relationships between queries and documents visited by users. Secondly, online approaches capture the available information and provide personalized results taking into consideration the last interactions of the user. However, an offline approach can implement more complex algorithms because there are less urgent time constraints.

2.4.1.1 Offline Approaches

An innovative personalized search algorithm is the CubeSVD algorithm [10] based on the click-through data analysis. This technique is suitable for the typical scenario of web searching, where the user submits a query to the search engine. The search engine returns a ranked list of the retrieved web pages. Finally the user clicks on pages of interest. After a period of usage, the system will have recorded useful click-through data that could be assumed to reflect users' interests and preferences.

The offline approaches address two challenges of web search. The first concerns the study of the complex relationship between user, the query, and the visited web pages. The second challenge faces the problem of unclear information, a user generally submits a short queries and visits few pages. In this case, recognizing relationships among the information becomes a hard task to carry out.

2.4.1.2 Online Approaches

Speretta and Gauch [11] developed the misearch system, which improves search accuracy by creating user's profiles from their query histories and selected search results. These profiles are used to re-rank the results by giving more importance to the documents related to topics contained in their user profile. In their approach, user's profiles are represented as weighted concept hierarchies. The open directory project (ODP) is used as the reference concept hierarchy for the profiles. Google has been chosen as the search engine to personalize through a software wrapper that monitors all search activities. For each individual user, two different types of information are collected. The first type is the submitted queries and the second type is the snippets of the selected search results by the user.

Koutrika and Ioannidis [12] presented an online approach where user's interests and

preferences are represented by a combination of terms connected through logical operators. These operators are used to transform the queries in personalized versions to be submitted to the search engines. An evaluation shows that when this personalization approach is applied, the users satisfy their needs faster compared with a traditional search engine.

The ability to recognize user interests in a completely non-invasive way, without installing software or using proxy servers, and the accuracy obtained from the personalized results, are some of the main advantages of this approach. Moreover, ranking does not depend on a global relevance measure, but ranking is computed from the context of user's interactions.

2.4.2 Personalization Based on Rich Representations of User Needs

This section presents three prototypes of personalized search systems based on complex representations of user needs. They are mostly based on frames and semantic networks, two AI structures developed in order to represent concepts in a given domain, and the related relationships between them. Even though these prototypes share some features, the mechanisms employed to build the profiles and the way the needs are represented are fairly different.

2.4.2.1 ifWeb

ifWeb [13] is a user model-based intelligent agent capable of supporting the user in web navigation, retrieval, and filtering of documents taking into account specific information needs expressed by the user with keywords, free-text descriptions, and web document examples. The ifWeb system exploits semantic networks in order to create the user's profile.

The user profile is represented as a weighted semantic network and each node corresponds to terms found in documents and textual descriptions. Network's arcs link pairs of terms that co-occurred in some document. The use of the semantic network and of the co-occurrence relationships allows ifWeb to overcome the limitations of simple keyword matching.

The user profile is updated and refined by relevance feedback provided by the user. ifWeb presents a collection of documents to the user who selects the ones that meet user's needs. Then, ifWeb extracts the information to update the user profile from the documents

on which the user expressed some positive feedback. Moreover, the prototype includes a mechanism for temporal decay called rent, which lowers the weights associated with concepts in the profile that have not been reinforced by the relevance feedback mechanism for a long period of time. This technique allows the profile to be kept updated so that it always represents the current interests of the user.

2.4.2.2 Wifs

The *Wifs* system [14] is capable of filtering HTML or text documents retrieved by the search engine ALTAVISTA in response to a query input by the user. This system evaluates and reorders page links returned by the search engine, taking into account the user model of the user who typed in the query. The user can provide feedback on the viewed documents, and the system uses that feedback to update the user model accordingly.

The *Wifs* system has been evaluated to determine the effectiveness of the user profile in providing personalized reordering of the documents retrieved by ALTAVISTA. Considering the whole set of documents retrieved by the search engine following the query, three relevance sorting structures are taken into account based on results provided by ALTAVISTA, *Wifs* and the user. The evaluation considered 15 working sessions and 24 users. The ordering of the first 30 results was considered. It shows that the system provides roughly a 34% improvement when compared to the search engine's non-personalized results.

2.4.2.3 InfoWeb

InfoWeb [15] is an interactive system developed for adaptive content-based retrieval of documents belonging to web digital libraries. The distinctive characteristic of InfoWeb is its mechanism for the creation and management of a stereotype knowledge base, and its use for user modeling. A stereotype contains the vector representation of the most significant document belonging to a specific category of users, initially defined by a domain expert. InfoWeb uses the stereotypes exclusively for the construction of the initial user model. The user's profile evolves over time in accordance to the user's information needs, formulated through queries, using an explicit relevance feedback algorithm that allows the user to provide an assessment of the documents retrieved by the system.

The InfoWeb prototype is specifically designed for digital libraries with an established

document collection and the presence of a domain expert. Nevertheless, some of the proposed techniques, e.g., stereotypes and automatic query expansion, can be also adapted to vast and dynamic environments, such as the Web.

2.5 Summary

Chapter 2 has given a background of four research fields which impact the contents of this thesis. The first is the research field of ant colony optimization. Ant colony optimization algorithms are particularly used to solve combinatorial optimization problems and have been adapted to many research fields. According to the double barriers experiment and the work of Dorigo, both of them inspired search algorithms which is one of the most interested subject in this thesis.

The second research field reviewed in this chapter was information retrieval (IR), which presents the overview of information retrieval process and document selection. Information retrieval examines the search query to provide the relevance of search results, which is the focus of this thesis. Consequently, this research field is an essential part of this thesis.

Evidence for document relevance is the third research field. In order to achieve the accurate user's profile, various evidences should be collected and analyzed.

Finally, personalization was presented. As personalization is the process of presenting the right information to the right user at the right moment, this research area becomes an indispensable part in this thesis.

All of the research fields detailed above influences personalization of search profile using ant foraging approach.

CHAPTER III
UNDERLYING TECHNOLOGIES

After an introductory given in the previous chapters, a closer look will be given at some technologies developed in thesis field. This chapter will present technologies that influence the design of personalized search which will be grouped into five areas.

In the first part, technologies that can be used to describe the documents in World Wide Web are introduced. Metadata can be used to characterize the user's profiles. Then, user's profiling, which is the method describing the characteristics of the user, is presented. In the third part, text processing to reduce its complexity by converting the text to indexing terms is explained. Next, classification which handles large amount of information is described. Finally, semantic relatedness that is a measuring of the semantic concepts is explained.

3.1 Metadata

Metadata is structured information that describes, explains, locates, or otherwise makes it easier to retrieve, use, or manage an information resource. Metadata is often called data about data or information about information. Metadata schemes have been developed to describe many types of textual and non-textual objects including published books, electronic documents, art objects, educational and training materials, and scientific datasets.

There are three main types of metadata:

1. Descriptive metadata describes a resource for purposes such as discovery and identification. It can include elements such as title, abstract, author, and keywords.
2. Structural metadata indicates relation between compound objects that are put together, for example, what is relation between each data in the same category.
3. Administrative metadata provides information to manage a resource, such as when and how it is created, file type, and who access it. There are several subsets of administrative data, two that are listed as separate metadata types are:
    - Rights management metadata deals with intellectual attribute rights.
    - Preservation metadata contains information needed to archive and preserve a resource.

Metadata can describe a collection, a single resource, or a component part of a

larger resource. Catalogers make decisions about whether a catalog record should be created for a whole set of volumes or for each particular volume in the set, so the metadata creator makes similar decisions. Metadata can be embedded in a digital object or stored separately. Metadata is often embedded in HTML documents and in the headers of image files. Storing metadata with the object it describes ensures the metadata will not be lost and the metadata and object will be updated together. Also, storing metadata separately can simplify the management of the metadata itself and facilitate search and retrieval. Therefore, metadata is commonly stored in a database system and linked to the objects described.

3.1.1 Metadata Features

A reason of creating descriptive metadata is to facilitate discovery of relevant information. Moreover, metadata can organize electronic resources, facilitate interoperability and legacy resource integration, provide digital identification, and support archiving and preservation.

Organizing Electronic Resources, as the number of Web-based resources grows rapidly, total sites are increasingly useful in organizing links to resources based on audience or topic. Lists of web pages can be built as static web pages, with the names and locations of the resources in the HTML. However, it is more efficient and increasingly more common to build these web pages dynamically from metadata stored in databases.

Digital Identification, most metadata schemes include elements such as standard numbers to uniquely identify the object to which the metadata refers. The location of a digital object may also be given using a file name, URL, or some more persistent identifier. Persistent identifiers are preferred because object locations often change, making the standard URL invalid. In addition to the actual elements that point to the object, the metadata can be combined to act as a set of identifying data, differentiating one object from another for validation purposes.

Archiving and Preservation, metadata is key to ensuring that resources will survive and continue to be accessible into the future. Archiving and preservation require special elements to track the roots of an object, to detail its physical characteristics, and to document its behavior in order to emulate it on future technologies.

Many international organizations have worked on defining metadata schemes for digital preservation, including the National Library of Australia, the British Cedars Project, and a joint Working Group of OCLC and the Research Libraries Group (RLG).

3.1.2 Metadata for Search

Metadata that supports the search function consists of any information which helps a user discover an information resource. Author, title, and subject indices are traditional metadata elements for searching, but these data are typically not the optimal search indices. Therefore, metadata elements could contain imagery, models, and related bibliographic materials in a heterogeneous distributed archive maintained by several investigators.

Table 3.1: Summary of Metadata Elements for the Search Function

Spatial Indices	Temporal Indices	Thematic Indices
Geographic region	Collection period	Collection name
Vertical range	Calendar/clock time	Data type
Horizontal position	Process time	Variable name
Vertical position	Event relationships	Related variables
Topological, metric relationships		Keywords
		Variable description
		Data collectors/authors

The search indices should be multidimensional and include various spatial, temporal, and thematic indices. A spatial index can take several different forms which might comprise an index for two dimensional space, three dimensional space, topological relations and metric relations. The temporal index might involve calendar as well as process time and temporal relationships. Thematic indexes should allow searches on any type of data collection, data collector, thematic variables as well as measures of variable similarity. Table 3.1 provides a potential list of metadata elements for search.

Search should be possible on any metadata element so all metadata may be considered relevant to search. Strictly search metadata encompasses the information required to find a data set which meets a set of criteria, but can overlap with evaluation. In some senses a data set is evaluated by its ability to satisfy a minimum set of evaluation criteria.

3.2 User Profiling

In general, the selected pages of keywords query are done from the pages returned as search results by the search engines. This method selects the search results from matching between keyword query and pages. Therefore, this method has a problem in that not all the selected keywords have to do with the user's interests and preferences. User Profiling, which gathers the information about the user's interests and preferences, is important for this problem. User profiling is an approach towards the personalized system where user's profile including user's interests and preferences can be accessed during the process. Moreover, users with the long-term interests and preferences can be grouped, and feedbacks of one person can serve as the guideline for information delivery to other users in the same group.

User profiling is also the key process of many other applications. For instance, the recommendation systems mainly depend on user profiles in terms of similarity and differences to provide particular suggestions. The personalized web search engine can construct user profiles from browsing history and consequently provide personalized results to match the information needs of individuals.

User profiling process commonly consists of three main steps. First, an information collection process is used to collect user's information. The second and third steps focus on user profile representation and construction from the user's information, respectively.

3.2.1 Collecting Information About Users

The first step of a profiling technique collects information about users. Collecting Information process consists of two main phases. The first phase, a basic requirement is to be able to identify user. The next phase is to collect user's information. It may be collected by explicit user input or implicit software agent gathering. Depending on how the information is collected, different user's information may be extracted.

3.2.1.1 Methods for User Identification

User identification is one of the most challenging steps in the process of collecting information about individual user. In case of email service, the user is identified exactly by user's email address. However, in case of web users, it is difficult to identify which page

download belongs to which user because the same user can use multiple computers and more than one user can use the same computer. Hence, user identification is an essential ability for any system to represent individual user. There are five basic approaches to user identification: software agents, logins, enhanced proxy servers, cookies, and session IDs.

The first three techniques are more accurate but they require active participation of the user. Software agents are small programs that reside on the user's computer, collecting user's information and sharing with a server by some protocol. This approach is the most reliable because there is more control over the implementation of the application for identification. However, it requires user's participation in order to install the desktop software. The next most reliable method is login-based system because the users identify themselves during login. The identification is generally accurate and the users can use the same profile from any location. On the other hand, the users have to create an account by a registration process and login and logout each time when they visit the site. This is a burden on the user's part. Enhanced proxy servers can also provide reasonably accurate user identification. Nevertheless, they have several disadvantages. They require users to register their computer with a proxy server. Thus, they are generally able to identify users connecting from only one location, unless users voluntarily register all of the computers they use with the same proxy server.

The final two techniques, namely, cookies and session IDs, are less invasive methods. For tracking user's interests and preferences, cookies are the most common way of client side data storing. The first time that a browser client connects to the system, a new userid is created. This ID is stored in a cookie on the user's computer. When they revisit the same site from the same computer, the same userid is used. This places no burden on the user at all. However, the main problem with this type of information collection is, when the user uses more than one computer, each location will have a separate cookie and a separate user profile. When more than one user uses the computer and all users share the same local userid, they will all share the same inaccurate profile. Finally, if the user clears their cookies, they will lose their profile altogether because a new cookie will be assigned to the user and if the user has cookies turned off on their computer, identification and tracking is not possible. Session IDs are similar but there is no storage of the userid. Each user begins each session with a blank slate but their activity during the visit is tracked. In this case, no permanent user profile can be built but adaptation is possible during the session.

To summarize, cookies are extensively used and effective because cookies are the least invasive, requiring no actions on the user's parts. Login-based system is better

accuracy and consistency to track user's information across sessions and between computers. Consequently, a good compromise is to use cookies for current sessions and provide optional logins for users who choose to register with a site.

3.2.1.2 Methods for User Information Collection

Personalization on the web must be able to distinguish individuals' need or groups of users' need by collecting information about users. What information is available for a personalized system to infer a user's information need? Obviously, the user's query provides the most direct evidence. However, since a query is often extremely short, the user's profile based on a keyword query is certainly impoverished. An effective way to improve user's profile is to ask the users to explicitly specify what user's interests and preferences are. Unfortunately, users are usually reluctant to make the extra effort to provide their information. Thus, it is very interesting to study how to infer a user's information need based on any implicit information, which naturally exists through user interactions and thus does not require any extra user effort.

User's information is collected by users filling out surveys or selecting interests and preferences as explicit information or tracking user's activities as implicit information. Explicit information gathering has some disadvantages that only represent certain user's interests and preferences in time. However, user's interests and preferences are likely to change. Finally, user's information is collected only from the given information. Although implicit information gathering, user's information can be collected from unforeseen information.

3.2.2 User Profile Representations and Constructions

There are many ways to represent a user profile. One can place a set of rules in the profiles. Others, WebMate and Alipes used keyword vectors for the user profiles representation. For each keyword is associated to weight value which tells the importance of the particular keyword. This approach is useful in the process of keeping user's profiles up to date. Each way of representing user's profiles has its own advantage. However, the most studies focus more on the representation than on the maintenance of the user's profiles.

The concept of a user's profile usually refers to a set of interests, preferences, and

information to deliver customized capabilities to the user. User's profile is normally represented as sets of weighted keywords, semantic networks, or association rules. Keyword profiles are the simplest to build, but they fundamentally have to capture and represent all words by user's interests, then they require a large amount of user feedback. On the contrary, concept profiles are trained on examples for each concept. They begin with an existing mapping between words and concepts. Thus, they can build profiles which are less user feedback.

3.2.2.1 Keyword Profiles

The most common representation for user profiles is sets of keywords. These can be automatically extracted from web pages or directly provided by the user. Weights, which are usually associated with keywords, are numerical representations of user's interests and preferences. Each keyword can represent a topic of interests and preferences or keywords can be grouped in categories to reflect a more standard representation of user's interests. An example of a weighted keyword based user's profile is shown in Table 3.2.

Table 3.2: A keyword based user's profile

Category Name	Keyword	Weight
Sport	Football	0.87
	Tennis	0.54
Music	Jazz	0.36
	Blossom	0.72
	Rock	0.81
Food	Noodle	0.63
	Pizza	0.49

Keyword based profiles are initially created by extracting keywords from web pages collected. The simplest type of keyword based profile produces a single keyword profile for each user. The other type is building multiple keyword profiles for each user, one per interest area. Consider a user interested in programming and cooking. A keyword will point towards the middle of these two topics, creating a picture of a user interest in programmers

who cook, or a user interest in cooking menu which programmers like. In contrast, by using a pair of vectors, the user profile more accurately represents the user's two independent interests.

Amalthaea is one of many systems that create keyword based user's profile by extracting keywords from web pages. Each user's profile is represented in the form of a keyword vector and each search result that is returned by the search engine is converted to similar weighted keyword vector. Then, these vectors are compared to the user's profile using the cosine similarity formula and only the search results which are closest to the user's profile are passed to the user. Weighted keyword vectors have been used in many fields, such as personalized online newspaper, web page recommender, browsing assistant, and recommender system.

WebMate is a personal agent for web pages browsing and searching that focuses on collecting information about the users as they browse or perform other activities to build the user's profiles. WebMate accumulates information about user's interests and preferences when the user evaluates the page currently being browsed. When the user clicks on an icon that WebMate places on each page, WebMate adds information to the user's profile. The user's profiles contain multiple keyword vectors per one user's interest. WebMate provides recommendations to the user in two ways. The first method sends queries that are formed by using a keyword based profile to the search engines. The second method searches sites contained in WebMate's resource list having terms in the keyword user's profile. Once the user profile is changed, the user can prompt WebMate to recommend sites to the user.

However, keyword profile is generally seen as an absolute of words with no provision to capture the semantic meaning. In addition, learning the keyword based user's interests and preferences is a time consuming process which consists of collecting information across multiple search categories to create the user's profile. Moreover, the words collected solely from the user do not allow the system to determine the user's intention when a new search topic is encountered.

3.2.2.2 Semantic Network Profiles

A semantic network is a knowledge representation scheme involving nodes and links between nodes. The nodes represent objects or concepts and the links represent relations between nodes. The semantic network can be used either to represent knowledge or to support automated systems for reasoning about knowledge. Figure 3.1 represents the

simplest form of a semantic network.

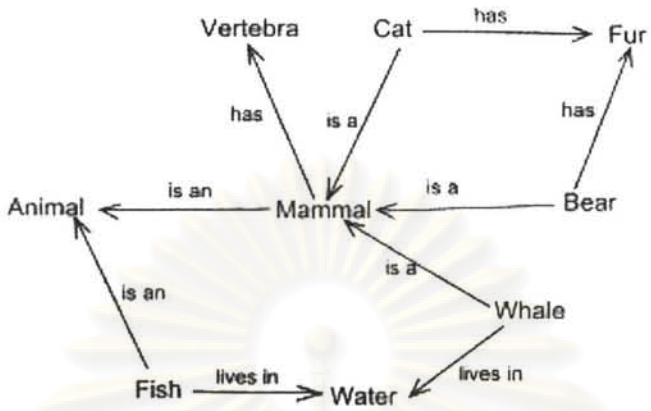


Figure 3.1. An example of the semantic network

An example of semantic network is Gellish model [16]. Gellish English, with its Gellish English dictionary, is a formal language that is defined as a network of relations between concepts and names of concepts. A Gellish network is a network of (binary) relations between things. Each relation in the network is an expression of a fact that is classified by a relation type. Each relation type itself is a concept that is defined in the Gellish language dictionary. Each related thing is either a concept or an individual thing that is classified by a concept. The definitions of concepts are created in the form of definition models that together form a Gellish Dictionary.

A semantic network profile represents a weighted semantic network in which each node represents a concept. It is typically built by collecting explicit information from users. This profile captures semantic relationships between concepts or words used and can be used to discover inherent relationships between nodes. Semantic network profile is similar to keyword based profile where keywords are extracted from the users. The difference is semantic network profile adds extracted keywords to a node in the network. Semantic user's profile can explicitly model the relationship between particular words and concepts. Hence, it can deal more effectively with synonym of natural language. However, it also places a barrier to the effortlessness of system structure.

The SiteIF project uses a word based semantic to represent user's profiles. They found that representing individual words as a node in the semantic network was not accurate enough to differentiate word meanings. Instead, they used information in WordNet

to group related words together in concepts, called synonym sets or synsets. They represents a user's profile as a semantic network in which the nodes are synsets. The arcs are co-occurrences of the synset members within a document of interest to the user. Node and arc weights represent the user's level of interest.

InfoWeb is a filtering system for online digital libraries documents. It builds semantic network based profile which represent long-term user's interests. Each user's profile is represented as a semantic network of concepts. In the beginning, each semantic network contains a collection of unlinked nodes in which each node represents a concept. As more users' information is gathered, the user's profiles are improved to include additional weighted keywords associated with the concepts. These keywords are stored in subsidiary nodes and linked to their associated concept nodes.

3.2.2.3 Concept Profiles

Concept profile is based on metadata vocabulary and concept vocabulary. It functions as a specification of concepts that specifies the meaning of a set of concepts and provides information about how each concept is functionally tied to other concepts. It also functions as an assemblage of concepts by clarifying the semantic relations between concepts and how to combine the functions of concepts into a single framework. Although this concept profile is constructed based on the elements from existing standards, it exists independently of any particular element used in any standard. Thus, this profile provides a context for using concepts in a generalized way that can be consistently applied to the establishment of any element relationship.

Concept based profile is similar to semantic network based profile in the meaning that both are represented by conceptual nodes and relationships between nodes. However, the nodes in concept based profile represent abstract topics considered interesting to the user, rather than specific words or sets of related words. Thus, the concept based profile requires some way of determining which concepts a user is interested based on user's feedback. Also, concept profiles are similar to keyword profiles because both are represented as vectors of weighted features, but the features represent concepts rather than words or sets of words.

Most concept based profiles automatically derive user's interests and preferences by exploring the contents of the user's browsed documents and search histories. The user profile is represented as a set of categories. For each category contains a set of keywords

with weights. The categories stored in the user profiles serve as a context to disambiguate user queries. If a user's profile shows that a user is interested in certain categories, the search engine can be diminish by providing suggested search results according to the user's preferred categories.

Gauch et al [11]. proposed a method to create user's profile from user browsed documents and use reference ontology to develop the hierarchical user profiles. A classifier is employed to classify user browsed documents into concepts in the reference ontology.

The OBIWAN project [17] represents user's profiles as a weighted concept hierarchy built from a reference concept hierarchy. The project focuses on personalized search to validate the quality of the user's profiles produced. The project built the user's profiles based on browsing histories which collected by proxy servers or capture from desktop screens.

3.3 Text Processing

Documents and queries are represented as vectors with a tf-idf weighting schema and their correspondence is calculated using the vector similarity measure. However, using every word of a document as an index term is not a good approach because not all words are equally significant. Terms with high frequency in any document of the collection, such as the, or, and, a, etc, are not useful index terms. Therefore, a document is processed by one or more analyzers. An analyzer is a combination of several text operations like tokenization, stopword removal, and stemming.

3.3.1 Tokenization

Tokenization is the process of splitting the sentences of a text document into separate tokens. However, considering only sentences would not yield optimal results. Punctuation marks, quotation marks, exclamation marks, quote signs, hyphens, and many other characters must be considered when processing the character stream. Special attention has to be given to punctuation characters, hyphens, digits and letter case. A punctuation mark might indicate the end of a sentence or it might be an integral part of the word. For example, considering the word "ISBN:0072395591" which refers to an international standard book number. Removing the punctuation character would put the colon out of context. However, a query for "ISBN:0072395591" will still return the document as the query

is processed by the same tokenizer. The last point mentioned is the letter case. For example, "Toyota Car Rent" and "toyota car rent", this problem is in general ignored, and the sentence is either made lower case or upper case.

3.3.2 Stopword Removal

Words, which have a high frequency across the document corpus, are stopwords and must be removed. High-frequency words as "a", "the", and "is" are not good discriminators as they usually occur in almost all documents. Another benefit gained by removing stopwords is the size reduction of the index structure. Deciding on which words to include in the stopword list is thus a crucial task. Many different stopword lists exist and the inclusion or exclusion of stopwords is often dependent on the targeted corpus. Indeed, in a corpus about logic words like "and", "or", and "not" would be considered relevant.

3.3.3 Stemming and Lemmatization

Stemming is the process of removing prefixes and suffixes from a word. Consider for instance the words "personal", "personalized", "personalizing", and "personalization" These words have a similar meaning and can thus be conflated into a single term by removing the suffixes "-ed", "-ing", and "ion", yielding the stem "personal." Stemming can reduce complexity by reducing the number of indexed terms and hence the size of the index structure. Another advantage is relevant documents can be found regardless of the used query word variation.

Despite its advantages, stemming can also raise new problems. There are cases, where words with a different meaning have the same stem. As an example consider the words "new" and "news", which obviously have different meanings. They have the same stem as "new."

The different stemming algorithms being described in literature focus mostly on suffix removal because most word variations are introduced through suffixes. The most popular suffix removal algorithm is the one developed by Porter [18]. It is simple, fast, elegant, and yields a similar performance as more complex algorithms.

Lemmatization is closely related to stemming. While stemming uses an algorithmic approach based on heuristics, lemmatization is based on vocabularies and morphological analysis of words. Lemmatization returns only the base of a word form as given in the

dictionary, namely the lemma. For instance, lemmatizing the word "saw" yields either "see" or "saw" depending on whether the used token was a verb or a noun. In contrast, the heuristics used in stemming algorithms might conflate the word to "s." Therefore, lemmatization provides a higher quality in terms of retaining a word's semantic. The improvement comes at the cost of higher implementation efforts as well as a slower runtime of the algorithm.

The use of a stemming algorithm is not obligatory. In fact, the stemming algorithm reduces costs of storage space and disputes about the benefits of stemming. Many search engines ignore stemming completely because they can match other methods such as query expansion. Query expansion simply means to add additional terms to the query. Each word is expanded by its variants, achieving a similar effect as stemming. For example, the query "personal" is expanded to "personal OR personalized OR personalizing OR personalization", so that all variants are covered. However, this approach can become expensive in terms of computation time when long queries are processed.

3.4 Classification

Classification is one approach to handling large amount of information. It challenges to organize information by classifying documents into the best matching concepts from a predefined set of concepts. A number of methods for text classification have been developed, each with a different approach for comparing the new documents to the reference set. Classification has been applied to newsgroup articles, Web pages, and other online documents. In metadata, a classification is the descriptive information for an arrangement or division of objects into groups based on characteristics which the objects have in common.

3.4.1 Text Classification

Text Classification or Categorization (TC) is the task of automatically assigning a text document to one or more predefined categories based on its contents. Nowadays, the major approach in Text Classification is Machine Learning. A general inductive process automatically builds a text classifier by learning. It observes the characteristics of a set of previously classified documents as a training set. These characteristics are used to classify new documents.

Different types of Text Classification tasks can be distinguished. From a category, there have two different types between single-label and multi-label classification. In Single-label Text Classification, exactly one category must be assigned to a document. On the other hand, any number of categories may be assigned to a document in Multi-label Text Classification.

Text Classification tasks can also be differentiated by the structure of the predefined categories set. In flat categorization, the predefined categories are treated in isolation and there is no structure defining the relationships among them. Most of the studies in Text Classification have focused on flat classification which has become a well-established research area, along with many good classifiers being developed. In hierarchical categorization, the predefined categories are organized in a hierarchical structure that reflects relations between them. Most hierarchies are organized in tree-like structures. For example, there are parent-child relationships between categories. In hierarchical classification, there can distinguish between cases where all documents belonging to a child category also belong to the parent called strong subsumption. In contrast, cases where a child category has documents that do not belong to its parent category are called weak subsumption.

3.4.2 Multi-Label Classification

Many classification methods, such as Naïve Bayes, SVM, and Logistic Regression, are of the single-label type. Research on multi-label classification has received much less attention.

The most popular approach for multi-label classification is binary. A separate classifier is learned for each category Ci. The original data set is transformed into |C| data sets. The data set for each category Ci contains all examples of the original data set, labeled as *c* if the labels of the original example contained *c*, and as ¬*c* otherwise. For the classification of a new instance *x*, this method outputs as a set of labels the union of the labels predicted by the |C| classifiers.

However, this method has two main problems. First, it assumes independence of categories which is not always true. There may be strong dependence between categories, in particular in hierarchical classification. Also, relations between categories on the same level can exist. For example, the following categories have some dependency, 'Politics' and 'Unrest, Conflicts and War', 'Environmental Issue' and 'Health'. In such cases, association of

an item with one category may influence its probability to be associated with a related category. But the binary approach cannot model such relations. Second, a number of binary classifiers have to be learned which may cause memory problems and time-consuming because each new instance should be processed by all |C| classifiers.

In addition, less popular approach in multi-label classification is to consider different set of labels that exist in the multi-label data set as a single label. It learns single-label classifier for C` categories, where C` is the power set of initial C categories. One of the negative aspects of this method is that it may lead to data sets with a large number of classes and few examples per class.

3.4.3 Hierarchical Multi-Label Classification

Hierarchical classification has two main advantages compared to flat classification. Firstly, it enables easy location of required categories when there are a significantly large number of categories. It is much easily to search among some high-level categories and among some related sub-level categories than to perform a general search among all existing categories. Secondly, it reflects the intuition of relatedness of topics that are close to each other in the hierarchy.

Two approaches were adopted by existing hierarchical classification methods as big-bang and top-down level-based approach. In the big-bang approach, a document is classified into a category in the category tree by a classifier in one step. In the top-down level-based approach, one or more classifiers are constructed at each level of the category tree, and each classifier works as a flat classifier at that level. Initially, a document will be classified by the classifier at the root level into one or more lower level categories. It will be further classified by the classifiers at the lower level categories until it reaches one or more final categories, which can be leaf categories or internal categories.

One of the important works on hierarchical Text Classification is Koller and Sahami [19]. They divide the hierarchical classification task into a set of smaller classification tasks, each of which corresponds to some split in the classification hierarchy. They show that this approach enables obtaining significantly higher accuracy compared to a massive classifier.

3.5 Semantic Relatedness

Measuring the semantic relatedness of concepts is an interesting problem in Natural

Language Processing (NLP). Several approaches attempt to approximate human judgment of relatedness. This section presents a WordNet–based measures of semantic relatedness.

3.5.1 WordNet

The creators of WordNet refer to it as an electronic lexical database. This is a convenient but very complex resource. WordNet can be visualized as a large graph or semantic network, where each node of the network represents a real world concept. For example, the concept could be an object like a car, or an entity like a programmer, or an abstract concept like music, and so on.

Table 3.3: Relations between synsets defined in WordNet

Relation	Description	Example
Hypernym	is a generalization of	furniture is a hypernym of chair
Hyponym	is a kind of	chair is a hyponym of furniture
Troponym	is a way to	amble is a troponym of walk
Meronym	is part/substance/member of	wheel is a (part) meronym of a bicycle
Holonym	contains part	bicycle is a holonym of a wheel
Antonym	opposite of	ascend is an antonym of descend
Attribute	attribute of	heavy is an attribute of weight
Entailment	entails	ploughing entails digging
Cause	cause to	to offend causes
Also see	related verb	to resent to lodge is related to reside
Similar to	similar to	dead is similar to assassinated
Participle of	is participle of	stored (adj) is the participle of "to store"
Pertainym of	pertains to	radial pertains to radius

Every node consists of a set of words, each representing the real world concept associated with that node. Thus, each node is essentially a set of synonyms that represent the same concept. For example, the concept of a car may be represented by the set of words car, auto, automobile, motorcar. Such a set, in WordNet terminology, is known as a synset. A synset also has associated with it a short definition or description of the real world

concept known as a gloss. The synsets and the glosses in WordNet are comparable to the content of an ordinary dictionary.

What sets WordNet apart is the presence of links between the synsets and the edges of the graph. Each link or edge describes a relationship between the real world concepts represented by the synsets that are linked. For example, relationships of the form "a vehicle is a kind of conveyance" or "a spoke is a part of a wheel" are defined. Other relationships include is opposite of, is a member of, causes, pertains to, etc. Table 3.3 shows the list of relations defined in WordNet. The network of relations between word senses present in WordNet encodes a vast amount of human knowledge.

The synsets in WordNet are divided into four distinct categories, each corresponding to four of the parts of speech, such as nouns, verbs, adjectives, and adverbs. Most of the relationships defined between the synsets are restricted to a particular part of speech and do not cross part of speech boundaries. Exceptions are *pertains to* and *attribute* relationships that exist between adjectives and nouns. Thus, the set of relations defined on the synsets in WordNet, divide them into four almost disjoint regions.

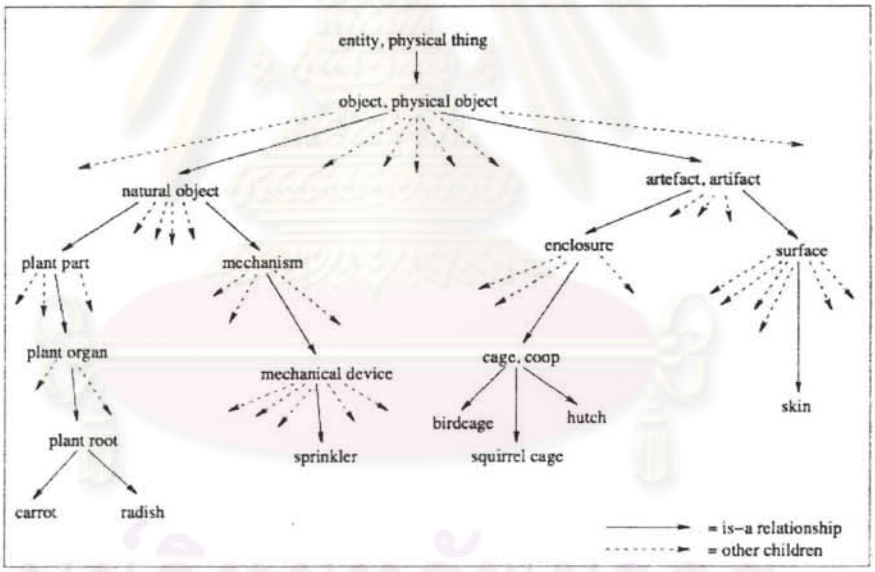


Figure 3.2: A schematic of the *is-a* hierarchy in WordNet

One of the relations in WordNet which is useful in the thesis approach is the *is-a* kind of relationship or simply *is-a*. This relationship between synsets is restricted to nouns and verbs. This relation organizes the noun and verb synsets into large hierarchies or trees.

Each tree has single root node. The more general concept nodes are ancestors of more specific concept nodes. We say that the more general concepts subsume the more specific concepts. For instance, entity is the most general concept in one of the noun hierarchies and is the root node of the tree. It subsumes other more specific concepts such as furniture, bicycle, etc, which are lower down in the tree. Similarly, furniture may subsume other concepts such as those of chair or table. There exists 9 such hierarchies in the WordNet nouns, while there are about 628 hierarchies for verbs. The large number of hierarchies in verbs is due to the fact that the verb hierarchies are, on average, much shorter and broader than the noun hierarchies. The average depth of the noun hierarchies is about 12.5 nodes, while that of the verb hierarchies is about 2.3 nodes. Each of the verb hierarchies, therefore, covers a much smaller portion of the synsets as compared to the noun hierarchies. Figure 3.2 shows an example of the is-a hierarchy in WordNet.

3.5.2 Measuring Semantic Relatedness

Given the enormous store of human knowledge encoded in WordNet, it has been used by many researchers in developing measure of semantic relatedness. Some use only the structure and content of WordNet to measure semantic relatedness. Other approaches combine statistical data with the structure of WordNet to give a score of semantic relatedness.

3.5.2.1 The Leacock-Chodorow Measure

An instinctive method to measure the semantic relatedness of word senses using WordNet would be to count up the number of links between the two synsets. The shorter the length of the path between them, the more related they are considered. The measure suggested by Leacock and Chodorow considers only the is a hierarchies of nouns in WordNet. Since only noun hierarchies are considered, this measure is restricted to finding relatedness between noun concepts. The noun hierarchies are all combined into a single hierarchy by imagining single root node that subsumes all the noun hierarchies. This ensures that there exists a path between every pair of noun synsets in this single tree. To determine the semantic relatedness of two synsets, the shortest path between the two in the taxonomy is determined and is scaled by the depth of the taxonomy. The following formula is used to compute semantic relatedness:

$$related_{lch}(c_1, c_2) = -log\left(\frac{shortestpath(c_1, c_2)}{2 \cdot D}\right) \quad (1)$$

where c_1 and c_2 represent the two concepts, shortest path (c_1, c_2) specifies the length of the shortest path between the two synsets c_1 and c_2, and D is the maximum depth of the taxonomy.

This method assumes the size or weight of every link in the taxonomy to be equal. This is a false assumption. It is observed that lower down in the hierarchy, concepts that are single link away are more related than such pairs higher up in the hierarchy. This simple approach, however, does relatively well, despite its lack of complexity.

Some highly related approaches attempt to overcome this disadvantage of simple edge counting by augmenting the information present in WordNet with statistical information.

3.5.2.2 The Resnik Measure

Statistical information is used to estimate the information content of concepts. The idea of information content was introduced by Resnik [20], in his paper that describes a novel method to compute semantic relatedness.

Resnik overcomes the problem of ambiguity by distributing the count of a word over all senses of the word. Therefore, if the word "bank" is encountered 50 times in the text and "bank" has 10 senses in WordNet, then each of these 10 concepts would receive a count of 5. This assumes an equal distribution of the senses in text.

Resnik defines the semantic relatedness of two concepts as the amount of information they share in common. Resnik goes on to elaborate that the quantity of information common to two concepts is equal to the information content of their lowest common subsumer. The lowest node in the hierarchy subsumes both concepts. For example, in Figure 3.2 the lowest common subsumer of carrot and radish is plant root, while that of carrot and birdcage is object.

$$related_{res}(c_1, c_2) = IC(lcs(c_1, c_2)) \quad (2)$$

where IC determines the information content of a concept and lcs(c_1, c_2)) finds the lowest common sub-suming concept of concepts c_1 and c_2.

The Resnik measure depends completely upon the information content of the lowest common subsumer of the two concepts. It takes no account of the concepts themselves. This leads to somewhat "coarser" relatedness values. For example, the concept pair car

and bicycle will have the same measure of semantic relatedness as the pair car and all terrain bicycle because both pairs of concepts have the same lowest common subsumer.

3.5.2.3 The Jiang-Conrath Measure

A measure introduced by Jiang and Conrath [21] addresses the limitations of the Resnik measure. It incorporates the information content of the two concepts, along with that of their lowest common subsumer. The measure is a distance measure that specifies the extent of unrelatedness of two concepts. It combines features of simple edge counting with those of information content introduced in the Resnik measure. The Jiang–Conrath measure is given by the formula:

$$\text{distance}_{jcn}(c_1, c_2) = IC(c_1) + IC(c_2) - (2 \cdot IC(lcs(c_1, c_2))) \quad (3)$$

where IC determines the information content of a concept and lcs determines the lowest common subsuming concept of two given concepts.

The relatedness would be undefined if there is a 0 in the denominator, which can happen in two special cases:

Case 1:

$$IC(c_1) = IC(c_2) = IC(lcs(c_1, c_2)) = 0 \quad (4)$$

$IC(lcs(c_1, c_2))$ can be 0 if the lowest common subsumer turns out to be the root node since the information content of the root node is zero. $IC(c_1)$ and $IC(c_2)$ would be 0 only if the two concepts have a 0 frequency count. In which case, for lack of data, the measure returns a relatedness of 0. c_1 and c_2 that can never be the root node since the root node is a virtual node created by us and doesn't really exist in WordNet.

Thus, in this case we return a relatedness score of 0, indicating insufficient data to assess the relatedness of c_1 and c_2.

Case 2: to handle a 0 in the denominator when

$$IC(c_1) + IC(c_2) = 2 \cdot IC(lcs(c_1, c_2)) \quad (5)$$

which is more likely to occur in the special case

$$IC(c_1) = IC(c_2) = IC(lcs(c_1, c_2)) \quad (6)$$

This usually happens when c_1, c_2 and $lcs(c_1, c_2)$ turn out to be the same concept.

Intuitively, this is the case of maximum relatedness (zero distance), and simply returning a relatedness score of 0, indicating unrelated concepts, would not be right. A more reasonable option is to return an arbitrarily high value, signifying maximum relatedness. But the difficulty is of selecting such a value.

3.5.2.4 The Lin Measure

Another measure, based on information content of concepts, is described by Lin [22]. The measure is given by:

$$related_{lin}(c_1, c_2) = \frac{2 \cdot IC(lcs(c_1, c_2))}{IC(c_1) + IC(c_2)} \quad (7)$$

For this measure, it has special handling for 0 information content values, since a 0 in the denominator in the above formula would give an undefined relatedness value. It simply returns a relatedness value of 0 if either of the two concepts have an information content of 0. This is because neither c_1 nor c_2 can be the root node. So their having an information content of 0 implies a lack of data (no frequency count for the concept). This measure has a lower bound of 0 and an upper bound of 1.

3.5.2.5 The Hirst-St.Onge Measure

Hirst and St.Onge [23] also use the rich content of WordNet to define relatedness between words. Note that this measure reports the relatedness between words and not between word senses or concepts. Unlike the above measures that considered only the *is-a* hierarchy of nouns, the Hirst-St.Onge measure actually considers all the relations defined in WordNet. All links in WordNet are classified as Upward (e.g. part-of), Downward (e.g. subclass) or Horizontal (e.g. opposite-meaning). Moreover, they also describe three types of relations between words such as extra-strong, strong and medium-strong. Any two words are related by one of these types of relations if they conform to certain rules summarized below.

Extra-strong relations are defined between two instances of the same word. Observe that this specifies a relationship between surface forms of words.

Two words are related by a strong relation under the following conditions.

- If the two words belong to the same synset in WordNet. For example, car and automobile.
- If the two words belong to two synsets connected by a horizontal link in WordNet. For example, two words that are opposite in meaning, such as hot and cold have a horizontal link between them.
- If one word is a compound word, the second word is part of the compound word and there exists an is-a relation between the synset of the first word and

that of the second word in WordNet. For example, school and private school have such a relationship.

A medium-strong relation is defined between synsets connected by a path in WordNet that is not too long and has relatively few changes in direction. The upward, downward, and horizontal classification of WordNet relations described earlier in this section, indicate the direction of the relations. The weight of any medium strong path is given by

$$\text{Weight} = C - \text{PathLength} - k \times \text{Changes in direction} \quad (8)$$

where C, k are constants. Medium-strong relations have some additional restrictions regarding the direction that the path may follow. The path between two words with the lowest weight is the one always considered. These three types of relations describe the degree of relatedness of words.

3.6 Summary

Several research fields described in Chapter 2 influences a personalization of search profile by using ant foraging. Therefore, technologies that are presented in Chapter 3 are related to personalization.

As far as the document collection of the Internet is concerned, metadata and classification are possibilities to describe the individual documents. Moreover, these technologies may be used to describe individual concepts and thus present additional information in the user's profile.

Besides the description of the information retrieved, a description of the individual user is one possibility to enable adaptation to his characteristics. Such information about a user can be held by the user profiles.

CAPTHER IV
PERSONALIZATION OF SEARCH PROFILE USING ANT FORAGING APPROACH

The focus of this work is to provide the most relevant search results to a user by personalization search according to user's profile. The approach will adopt ant colony foraging algorithm to perform both gathering the user's interests and updating the user's profile.

4.1 Approach

The approach investigates the effectiveness of personalized search based upon user profiles constructed from user browsing histories and activity at the search site itself. The reference architecture of the approach is divided in three subsystems, namely, building the user's profile, classifying the user's profile data, and personalizing the search results by means of the user's profile.

4.2 Reference Architecture

The reference consists of three subsystems as depicted in Figure 4.1.

4.2.1. Building the user's profile encompassing
   1. Annotating Web Page.
   2. Assigning Pheromone Value.
   3. Updating Pheromone.

4.2.2. Classifying the user's profile data encompassing
   1. Computing similarity value.
   2. Creating a bipartite graph.

4.2.3. Search personalization encompassing
   1. Choosing an extended word based on the user's profile.
   2. Ranking the search results obtained from the first step.
   3. Switching between short-term and long-term user's interests.

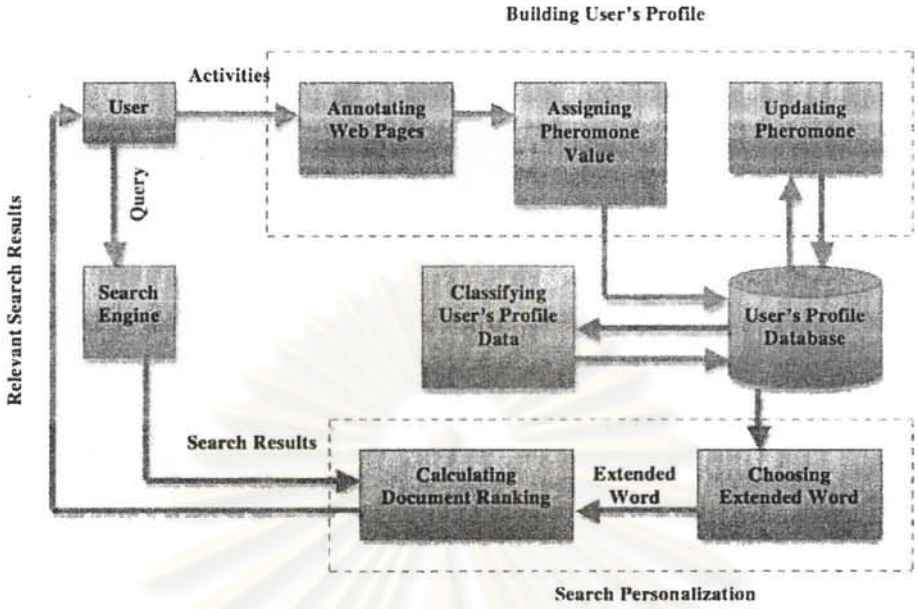


Figure 4.1: Reference of the approach

4.2.1 Building User's Profile

A user's profile represents the user's interests and preferences in order to deduce user's intention for queries. In this approach, the user's profile is constructed by transforming the information from parts of HTML document to sparse words and applying ant colony foraging behavior on the sparse words. The sparse words with the most pheromone value are used to identify the user's interests and preferences. A user's profile consists of a set of categories, each of which encompasses a set of elements and its corresponding weight. Each category denotes the user's interest in that category. The weight or score of user's interest in an element represents the significance of that element with respect to the category.

The fundamental principle of user's profile creation is inspired by the nature of ant colony foraging behavior. The ant leaves pheromone chemical as a communication means. Ants will typically choose to lay pheromone depending on the quality and quantity of food found at a source when foraging for food. Consequently, a strong pheromone path is created as soon as a profitably high value of food source is found. In general, the stronger value of pheromone it produces, the less pheromone evaporates. As food sources become depleted, it is to the advantage of the colony for the pheromone to evaporate over time. This

eliminates the possibility of ants following a strong pheromone trail to a food source that has already been diminished. It follows then that, in a situation where there is a certain probability of food randomly appearing, the colony could find new food sources.

By the same token, the user's profile is changing periodically. Thus, to solve the problem by optimizing of the pheromone level of concentration, a pheromone update is required to keep the user's profile up to date at all time. Bearing this notion in mind, assuming that web page destinations portray the food sources, the system adds the interest scores to the user's profile as pheromone gets deposited when the user visits the destination web pages. As such, the amount of pheromone being deposited depends on the user's interest of the destination web page, particularly for the pages that are located deep under the home page links of interest.

The creation of user's profile serves as a coarse-grained process that exploits pheromone deposit technique to arrive at a user's interest summary and efficient search results.

4.2.1.1 Annotating Web Page

When a user visits the destination web page, information must be extracted to annotate web page contents. Using full text documents to annotate web page takes considerable amount of time. Hence, the approach extracts information from parts of HTML document. Since a web page is a semi-structured document, annotation of web page contents is determined by the structure of the web pages. To confine the size of search space, the proposed approach considers only three kinds of tags and attributes from HTML pages, namely, URL, tag<title>, and tag<meta name="description">. URL is selected because not only navigation path can be traced from the URL, but also web page contents are usually related to their source. The URL strings accurately describe what is contained in each folder by means of descriptive words to enhance intuitive meaning to the user as shown in Figure 4.2. The tag<title> and tag<meta name="description"> provide descriptions of the web pages. The tag<title> gives a brief definition of the web page, whereas the tag <meta name="description"> provides a concise explanation of the content of web page. The proposed approach looks for the most redundant words to apply annotation of the page. For example, if the most redundant word is football, one criterion on page annotation is to choose the web page annotated with football. Another criterion is to employ page type classification that is determined by the pheromone count. The procedure

will be described in subsequent sections. At any rate, a systematic procedure for participating candidate word consideration that is produced by tags and attributes is the root word, disregarding all derivatives thereof.

Figure 4.2: A dichotomy of a URL

In this approach, URL, tag<title>, and tag<meta name="description"> are segmented into tokens. The procedure breaks non-alphanumeric characters, conjunction words, and stop words to create smaller tokens, and applies Porter stemming algorithm to transform each token to a root word. An indicative statistics, that is, word density is computed from these tokens to gather web page annotation statistics.

Word density can be computed from the equation:

$$density = ((Nkr * Nwp) / Tkn) * 100 \quad (9)$$

where Nkr denotes word frequency, Nwp is number of word occurrences in a phrase, and Tkn is the total number of words.

Table 4.1: Sample web page with URL, tag, and META tag

URL	Tag<title>	Tag<meta name="description">
http://news.bbc.co.uk/sport2/hi/football/default.stm	BBC SPORT | Football	The latest BBC Football news plus live scores, fixtures, results, tables, video, audio, blogs and analysis for all major UK and international leagues.

The sample web page annotation is shown in Table 4.1. Each URL, tag, and meta tag are segmented into tokens and assigned weight derived from the previous density statistics as shown in Table 4.2.

Table 4.2: Weight and density of individual token

Token	Weight	Density
news	1,1	10.526
sport	1,1	10.526
hi	1	5.263
football	1,1,1	15.789
:	:	:
league	1	2.263

4.2.1.2 Assigning Pheromone Value

The analogy of quality of the food source that affects the amount of pheromone deposit gives rise to the pheromone value of user's interest in the web page. Typically, most users prefer a direct link to the web page that they are interest in. However, if they cannot find sufficient information required to reach the designated web page, they will surf through the web pages to find the desired information. Consequently, counting the path along the URL measures the user's interest in the web page. The frequency and the access time of visiting the same type of web pages can also be used to evaluate user's concentration. Moreover, the selected search results can denote the relevance of user's intention. These four factors constitute the pheromone value of the designated web page.

One essential ant foraging behavior occurs when ants found a good quality food source. They will congregate at the food source to acquire as much food as they can. Thus, heavy pheromone will be deposited along the path to food source. By this analogy, the approach also considers the selected web pages acquired from search results as an additional factor of pheromone value calculation.

In creating a new user's profile, the above information so obtained is inadequate to infer what the user real interests are. Fortunately, the access time of visiting web pages and the selected web pages can make up "short-term" user's interests. By assigning higher weight to increase the amount of pheromone deposit, the level of information in the user's profile will quickly become steady for inference of user's current interest.

Although user's interest diverts over a period of time, the frequency of visiting the same type of web pages can show "long-term" user's interest. Switching between short-term and long-term user's interest is determined by the amount of pheromone deposit,

which will be further elaborated in next section. The rationale behind this observation is to render a newly created user's profile reaching steady state as soon as possible in proportional to the selected web pages (or pheromone deposit).

This approach, the number of paths along the URL and the frequency of visiting the same type of web pages (component I) make up pheromone value for long-term user's interests. For short-term user's interests, the access time of visiting the same type of web pages and the selected search results (component II) constitute pheromone value. In addition, the most recently accessed time of visiting the same type of web pages are added highest pheromone value.

Table 4.3 shows pheromone deposition of visit. Each node represents the web page annotated from the previous step. The pheromone deposit denotes the user's interests which are considered from the factors of pheromone value calculation. The highest pheromone deposit of short-term is travel node as shown in the table, reflecting higher current user's interest in travel topic than the rest of the topics under investigation. The uppermost pheromone deposit of long-term is technology node. It means that the most ordinary user's interest is technology.

Table 4.3: Pheromone deposition of visit

Short-term					Long-term			
No	Node	Component I	Component II	Pheromone	No	Node	Component I	Pheromone
1	Travel	4	62	48.694	1	Technology	37	31.689
2	MacBook	3	57	41.946	2	Football	29	22.217
3	Technology	37	19	37.568	3	System Analysis	29	22.217
4	Football	29	17	27.128	4	Car	8	2.421
5	System Analysis	29	11	21.438	5	Camera	5	0.966

4.2.1.3 Updating Pheromone

As user's interests and preferences always change over time, the value of pheromone must be updated, preferably in real-time, to keep the user's profile up-to-date. According to ant colony behavior, deposit and evaporation of pheromone must be proportionated. If the

destination has abundant food source, many ants will go there and lay pheromone which results in strong pheromone deposit and lower evaporation rate along the path. On the other hand, for a low quantity of food source, the path will make a weak pheromone path having high evaporation rate because few ants will visit the area.

When the rate of pheromone evaporation for the node becomes 1, or the highest of the rate of pheromone evaporation, that node will be deleted from the user's profile. The fact is that the user has lost interest in that topic.

The formula for pheromone value update, or equivalently the user's interest score, can be determined as follows:

$$\tau'_d = (1 - \rho)\,\tau_d \qquad (10)$$

where τ_d denotes the amount of pheromone on a given destination web page and ρ denotes the rate of pheromone evaporation. The equation of the rate of pheromone evaporation (ρ) is

$$\rho = 1 - (\tau_d / \Sigma\tau_d) \qquad (11)$$

In this thesis, we adjust the pheromone differences to accommodate subsequent computations by the equation:

$$\tau'_d = (1 - \rho)\,\tau_d^{\alpha} \qquad (12)$$

4.2.2 Classifying User's Profile Data

After annotating the web pages and collecting user's interest to be archived in the user's profile, some of this information may be similar by category, others may be different. Organizing this information for fine-grained scrutiny is a necessary requirement to keep track of the objects of similar properties, as well as their relationships if they exist. In other words, it is an issue of how this information should appropriately be classified. In so doing, performance of the personalization process will improve. The approach employs WordNet to establish a user's preference word-list, and Leacock-Chodorow measure [24] for semantic similarity in the classification process.

Leacock-Chodorow measure deals with semantic similarity by only considering the IS-A relation. To determine semantic similarity of two synsets, the shortest path between the two synsets in the taxonomy is determined and scaled by the depth of the taxonomy. The following formula computes semantic similarity:

$$\text{Sim LCH}(a,b) = -\log(\text{length}(a,b) / (2 * D)) \qquad (13)$$

where length denotes the length of the shortest path between synset *a* and synset *b*, and D denotes the maximum depth of the taxonomy.

Leacock-Chodorow measure assumes a virtual top node dominating all nodes and will always return a value greater than zero, as long as the two synsets compared can be found in WordNet. Leacock-Chodorow measure gives a score of 3.583 for maximum similarity that is the similarity of a concept and itself.

Table 4.4: Similarity value of word-list from user's profile

	football	cruise	tour	tennis	travel	resort	sport	game	seafood
football	-	1.1239	1.2040	1.8971	1.3863	1.7430	*2.5903*	2.3026	1.1239
cruise	1.1239	-	1.9459	1.1239	*2.6391*	2.2336	1.7228	1.3863	0.8557
tour	1.2040	1.9459	-	1.2040	*2.6391*	1.5404	1.5404	1.6094	1.2910
tennis	1.8971	1.1239	1.2040	-	1.3863	1.7430	2.3026	2.3026	0.9808
travel	1.3863	*2.6391*	*2.6391*	1.3863	-	*2.6391*	1.9459	1.743	1.2040
resort	1.7430	2.2336	1.5404	1.7430	*2.6391*	-	2.0794	2.3026	1.3863
sport	*2.5903*	1.7228	1.5404	*2.3026*	1.9459	2.0794	-	*2.5903*	1.6094
game	2.3026	1.3863	1.6094	2.0794	1.7430	2.3026	*2.5903*	-	*2.3026*
seafood	1.1239	0.8557	1.2910	0.9808	1.2040	1.3863	1.6094	2.3026	-

The first step of classifying the user's profile is to compute similarity value from Equation (13) by setting up a word-list matrix created from the user's profile. A word- pair similarity value so computed indicates the closeness between the designated word-pair. The results are shown in Table 4.4.

Table 4.5: Category and its element

Category name	Element
sport	football, tennis, game
travel	cruise, tour, resort
-	seafood

The second step creates a bipartite graph designating the classification of word relations that build from the most similarity value of each designated word-pair. Each designated word-pair, which is selected to build a bipartite graph, is called word-list. The upper hierarchy represents category name which is derived from the nodes having

common characteristics. The lower hierarchy of category name represents the corresponding elements.

Figure 4.3: An example of a bipartite graph with category name and its elements

Figure 4.3 depicts a sample bipartite graph of category name and its elements. To check whether an element belongs to the right category, the relationship between category name and its element must exist. If any element does not have a word-list relation, the element does not belong to its category and will be removed from the category and placed on unclassified category as shown in Table 4.5. For example, in Figure 4.3, sport– football, sport – tennis, sport – game, and game – seafood are element-category pairs. It appears that sport – game – seafood are related. However, sport – seafood does not contain in word-list relationship pair, thus seafood does not map to sport category.

Table 4.6. Sport category comparison of Yahoo, Google, and directory

Sport directory		
Yahoo directory	Google directory	Directory
recreation> sport>football	sport> football	sport> football
recreation> sport>tennis	sport> tennis	sport> tennis
recreation> game	game	sport> game

Table 4.7. Travel category comparison of Yahoo, Google, and directory

Travel directory		
Yahoo directory	Google directory	Directory
recreation> travel>cruise	recreation> travel>specialty travel>cruise	travel> cruise
recreation> travel>tour	recreation> travel>tour	travel> tour
recreation> travel>resort	recreation> travel>loading>resort	travel> resort

To test the proposed approach classification capability, we compared directory with Yahoo and Google directories. The results are close to both Yahoo directory and Google directory searches as shown in Table 4.6 and 4.7, but are smaller and less complex than

those of Yahoo and Google. Our directory contains all relevant category names and their elements that are easy to comprehend.

4.2.3 Search Personalization

A typical search query often ends up with plentiful results that contain few relevant ones. To reduce unwanted search results, the above user's profile classification can be exploited to establish a search personalization mechanism. Search personalization is based on user's interest subjects (described by single word) having the highest amount of pheromone deposit. By reordering the pheromone deposit, all subjects (words) can be arranged according to their relevance to suit the user's interests and preferences.

The proposed approach supports word query between nouns, verbs, adjectives, and adverbs. Word query can be a word or collocation of words in the form of a sequence of words that go concurrently for a specific meaning such as "system analysis and design". However, the proposed approach does not support sentence query. One may contend that the more query words used, the clearer the (meaning of) query. Nevertheless, most users do not like to enter too many words just to look for a piece of information, typically about 3 words. As such, adding one or two "key" words to form an extended word (to be described subsequently) entails a keyword search approach that yields high performance and useful results. This is because the extended word will help narrow down search theme which enables the search engine to recognize the user's interests and preferences.

Search personalization is then carried out in three steps. The first step is to choose an extended word based on the user's profile to make a new query which is more relevant to the user. The second step is to rank the search results obtained from the first step. The results are sorted in descending order. The last step is to switch between short-term and long-term user's interests to regulate the account on search personalization process. The procedural details are described below.

4.2.3.1 Choosing Extended Word for Making a New Query Keyword

To analyze if a word in the user's profile can be used as an extended word, all words are first classified and formed a bipartite graph. Each node of the bipartite graph is assigned a weight which is the pheromone deposit of user's interest. All words in the bipartite graph hierarchy form an extended word list to match the input query search

results, irrespective of individual word position. This is the first step of the search personalization process. Figure 4.4 illustrates a user's profile classification from the bipartite graph and the corresponding pheromone value. There are three matching scenarios to consider:

1. Matching category name with the query. The query matches a category name having the highest pheromone deposit. The element becomes the extended word.
2. Matching element with the query. The query matches an element that belongs to one or more categories. If the element belongs to one category, the element will match with its own category. Otherwise, the element will match with the category that has the highest pheromone deposit.
3. No matching word between the query and user's profile word-list. This scenario will choose a word having the highest pheromone deposit in the user's profile to be an extended word.

The above matching scenarios are exemplified by the following examples.

Scenario	Query word	Extended word
1	sport	sport, football
2	palm	palm, technology
3	news	news, technology

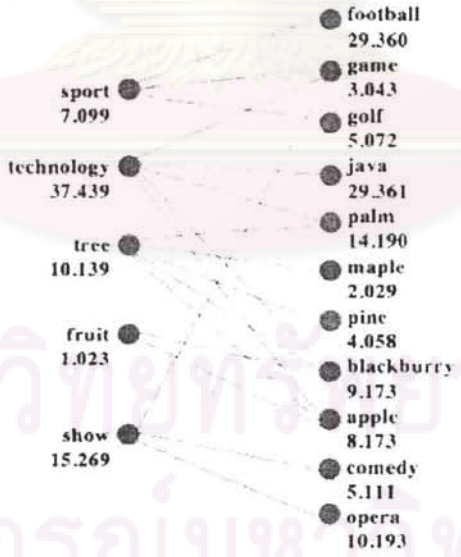


Figure 4.4: A bipartite graph of category name and its elements with pheromone value

4.2.3.2 Calculating Document Ranking

The new extended query will yield search results that provide more relevant information to the users. Relevancy is obtained from re-ranking all words in the pertinent document, whereby the most likely related document will be retrieved.

To calculate ranking of each document, the cosine similarity between the extended query and the user's profile is computed. The cosine of two vectors is a measure of how similar two vectors will be on the (0,1) scale, where 1 means completely related (or similar) and 0 means completely unrelated (or dissimilar). The cosine similarity of two vectors *a1* and *a2* is defined as follows:

$$Sim_{cos}(a_1, a_2) = \cos(a_1, a_2) \quad (14)$$

$$\cos(a_1, a_2) = dot(a_1, a_2) / \|a_1\| \|a_2\| \quad (15)$$

where a_1 denotes the extended query, a_2 denotes the term frequency of document, and $dot(a_1, a_2)$ denotes the dot product of a_1 and a_2. Term frequency, which measures how often an extended query is found in a document, is defined as follows:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}. \quad (16)$$

where $n_{i,j}$ denotes the number of occurrences of the considered term (t_i) in document d_j and $\sum_k n_{k,j}$ denotes total occurrences of all terms in document d_j.

Table 4.8: Using cosine similarity for re-ranking search results

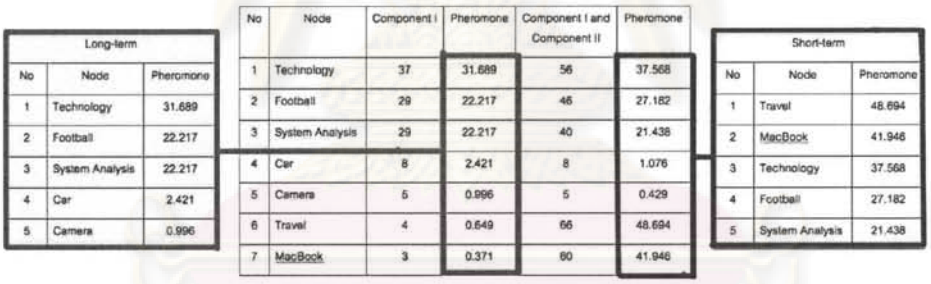
Rank	Pervious Rank	URL	Cosine Similarity
1	2	http://edition.cnn.com/TECH/	0.99228
1	9	http://www.techweb.com/home	0.99228
2	7	http://news.zdnet.com/	0.99160
3	5	http://www.nytimes.com/pages/technology/index.html	0.85749
3	10	http://www.t3.com/	0.85749
4	6	http://www.businessweek.com/technology/	0.83957
5	8	http://www.physorg.com/technology-news/	0.80717
6	1	http://news.cnet.com/	0.79262
7	3	http://www.technewsworld.com/	0.70711
7	4	http://news.yahoo.com/technology	0.70711

Table 4.8 depicts a new ranking of search results from the extended query "technology news" ordered by the most similarity to the least similarity.

4.2.3.3 Switching Between Short-term and Long-term User's Interests

Long-term user's interests are created in the same manner as their short-term counterpart. The entries are fundamentally taken from the short-term entries that occupied their top ranking for a predetermined duration. When personalized search component cannot find the desired item or items from the usual short-term user's interests list, it automatically switches to the long-term one and proceeds to function in the same manner. If the entry is found, it is copied to the short-term list. The corresponding pheromone value and duration in both lists are updated. Otherwise, a new entry in the short-term list is created. As each entry ages, its ranking follows. Upon falling below a predefined threshold limit, the entry is removed. Figure 4.5 illustrates the entry transfer and update process between both lists.

Figure 4.5: Transferring and updating process between short-term and long-term lists



4.3 Summary

The lack of personalization of the retrieval process was identified as one of the major drawbacks of existing approaches. In order to facilitate the information retrieval, identifying to the user's interests and preferences opens possibilities to improve these processes.

A building the user's profile approach is proposed to identify the user's interests and preferences. Classifying is utilized to facilitate for retrieval processes. Personalized are conducted concerning individual user's interests and preferences.

This approach copes with the changing interests and preferences of users and provides the search results which according to user's interests and preferences.

CHAPTER V
EXPERIMENTS

The experiments were carried out in different stages, namely, experimental setup, annotating web page, assigning, updating pheromone value, and re-ranking. The outcomes were measured by their precision and tested against Yahoo and Yahoo Motif.

5.1 Experimental Setup

Four sets of extensive experiments were conducted by different user's intention to the same query based on the proposed procedures described. The first set, involving a basic word "Technology" will be elucidated in the sections that follow. The remaining three sets, i.e., Zoology, Botany, and Finance were carried out in the same manner.

5.2 Annotating Web Page

Since evaluation of the approach is based primarily on the frequency of user's web access, the web pages which the users visited are annotated and found that 81% of the annotated results was similar to the page title. Figure 5.1 shows the density of each token from sample web pages. From Figure 5.1, football is the selected annotated web page because it has the highest density.

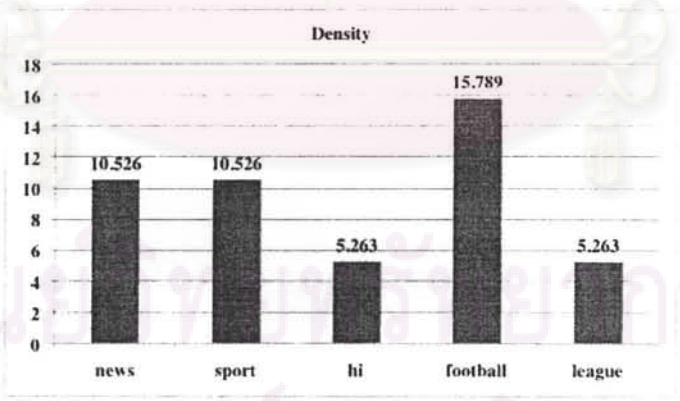


Figure 5.1. Comparison of token density

5.3 Assigning and Updating Pheromone Value

The user's interests and preferences were determined from the amount of pheromone deposit, while the user's profile was kept up-to-date by the rate of pheromone evaporation. Thus, the updated pheromone value of each node in the bipartite graph would reflect the current degree of user's interest.

Table 5.1 summarizes the amount of pheromone deposit, rate of pheromone evaporation, and pheromone of each node. The higher the amount of pheromone deposit, the higher the degree of user's interest. The rate of pheromone evaporation is used to update the user's profile. Lower pheromone rate of evaporation reflects the intense of current user's interest, whilst higher pheromone rate of evaporation signifies the topics of interest currently being faded away. The pheromone represents the actual degree of user's interest.

Table 5.1: Pheromone deposit, rate of pheromone evaporation, and pheromone of each node

No.	Node	Amount of pheromone deposit	Rate of pheromone evaporation	Pheromone
1	Technology	37	0.561	37.439
2	Football	29	0.639	29.361
3	System Analysis	20	0.738	20.262
4	Car	8	0.887	8.113
5	Tree	5	0.928	5.072
6	Game	3	0.957	3.043

5.4 Re-ranking

Re-ranking process makes use of extended word notation based on user's interest and input query word. Using as few input keywords as possible, the user's profile is searched to retrieve the word having highest interest score to be combined with the input query word, or extended word in context. Experiments were tested to compare simple query

words with extended query words, and to compare Yahoo Motif with extended query words. For instance, the input query word "palm", along with the highest interest scored word "technology", form an extended word "palm technology" for use in the search query. As a consequence, the search results yield different documents to be retrieved from a new list of URLs. This process is called re- ranking.

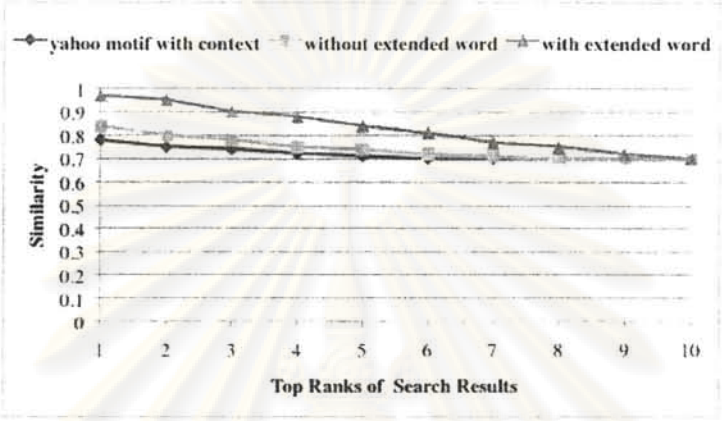


Figure 5.2: Input (simple) query: palm, Yahoo motif query: palm, extended query: palm technology

Figure 5.2 shows the result comparison between query with simple query words, yahoo motif, and extended query words. The results of the three queries show that the last query yields more relevant documents to the user than the other two queries. A closer look at the cosine similarity value of the results reveals that the new ranking from extended word is closer to 1 than the ranking without extended word and Yahoo motif with context.

5.5 Experimental Results

Effectiveness of the proposed approach relates directly to the relevancy of retrieved results. The effectiveness of personalized search is measured by precision of the ability to retrieve top-ranked results that are mostly relevant to the user's interest. The precision is defined as follows:

$$\text{precision} = \frac{\text{number of relevant documents retrieved}}{\text{total number of documents retrieved}} \quad (17)$$

For personalized search evaluation, four different user's profiles are used in each set of

experiment. Some entries could appear in more than one profile as the search went on. For example, "python" could fit either technology profile or zoology profile, "palm" could be in technology profile or botany profile, or "portfolio" could be in technology profile or finance profile. Table 5.2 and Table 5.3 compare the top-20 ranked of relevant search results based on user's profile. The first 10-ranked entries are constituted from short-term profile, while the last 10-ranked entries come from long-term profile. These two lists enhance search precision considerably. Table 5.4 shows the precision of the overall search results at different time. Those that are not relevant to the user's interest exhibit low precision values. However, as activities increase, search results improve since sufficient information is accumulated. This fact is depicted in Figure 5.3.

Table 5.2: Comparison of top-10 ranked relevant search results from short-term profile between Yahoo, Yahoo Motif, and the approach

Input query	Profile	Extended query	Amount of relevant results in top-10 ranked		
			Yahoo	Yahoo Motif	The approach
palm	technology	palm technology	9	2	9
python	zoology	python snake	0	9	10
palm	botany	palm tree	0	2	8
portfolio	finance	portfolio finance	6	10	9
blackberry	technology	blackberry software	2	2	10
firefox	zoology	firefox animal	0	0	6
apple	botany	apple tree	1	0	3
float	finance	float financial	2	8	7

Table 5.3: Comparison of top-10 ranked relevant search results from long-term profile between Yahoo, Yahoo Motif, and the approach

Input query	Profile	Extended query	Amount of relevant results in top-10 ranked after switching		
			Yahoo	Yahoo Motif	The approach
palm	botany	palm tree	0	2	8
python	technology	python programming	2	10	10
palm	technology	palm technology	9	2	9
portfolio	technology	portfolio technology	0	1	8
blackberry	botany	blackberry tree	0	3	7
firefox	technology	firefox browser	10	10	10
apple	technology	apple laptop	9	10	10
float	technology	float sql	6	2	9

Table 5.4: The precision of user's profiles at different time

Time	Precision			
	Technology	Zoology	Botany	Finance
T1	0.5	0.0	0.0	0.6
T2	0.6	0.2	0.0	0.6
T3	0.7	0.3	0.1	0.6
T4	0.8	0.4	0.2	0.7
T5	0.8	0.5	0.3	0.7
T6	0.8	0.7	0.4	0.8
T7	0.9	0.8	0.6	0.8
T8	0.9	0.9	0.7	0.8
T9	0.9	1.0	0.8	0.9
T10	0.9	1.0	0.8	0.9

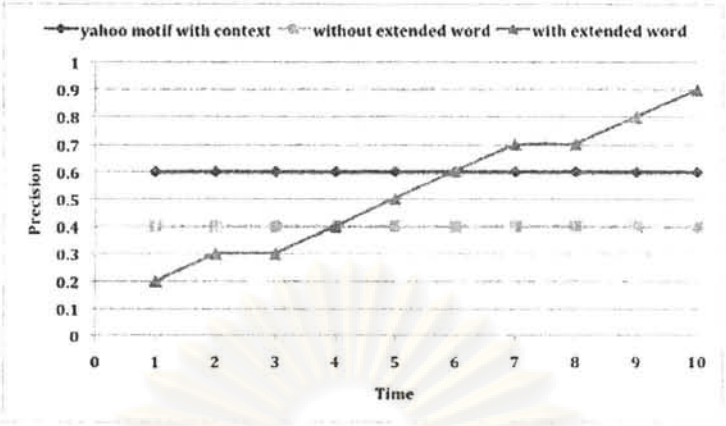


Figure 5.3: Precision of personalized searches with extended word, general search without extended word and Yahoo Motif with context data

CHAPTER VI
SUMMARY AND FUTURE WORK

Within a few years search engines have become the method of choice for accessing and retrieving information from the Internet. While search engines have existed for several years recent studies show that search for information is still a challenge. The first challenge is that information is scattered over multiple sources. The second challenge is that search engines apply a "one-size-fits-all" approach ignoring the needs of the individual. Finally, users are given no guidance through the information space. These are exactly the issues which the thesis has tried to address by using several techniques from ant colony optimization, information retrieval, personalization, user profiling, and clustering.

6.1 Summary

General search engines mostly use a "one-size-fits-all" approach which does not deliver optimal search results to a individual user. Therefore, the thesis approach has presented the feasibility of personalized web search to provide the relevant search results by means of an extended query that is constructed from the most up-to-date user's profile in accordance with the short-term and long-term interests. The short-term interest is induced by new events and vanishes quickly. On the contrary, the long-term interest generally reflects real user's interests. One way to realize this scheme is to set the short-term interest as the default search personalization and arrange the long-term interest as the secondary search. At a predetermined threshold period, short-term values are promoted to the long-term list. Older values in long-term list will eventually be discarded. Hence, search personalization will satisfy the user intent without having to resort to long strings of query.

The reference architecture consists of three major phases. First, building the user's profile based on implicitly collected information. Specifically, the information of destination web page is extracted to annotate web page content. The annotation of web page is a key task because the additional metadata is important for the adaptation of search results. Assigning and updating interest score to the user's profile as pheromone value and pheromone value update that was inspired by ant foraging behavior. The experiments investigated that the profiles converged to a stable set after approximately 350 visited web pages. Upon incorporating the visited web pages and selected web pages from the search results would converge to a stable set after approximately 100 visited web pages. The

second phase is classifying the user's profile data. The information which was extracted from destination web page was classified based on its similar properties. In so doing, performance of the personalization process will improve. The last phase personalization the search results is proposed to reduce irrelevant search results by giving extended word based on the user's profile to make a new query which is more relevant to the user.

The thesis experiments were tested on Yahoo and Yahoo Motif. The results yielded higher similarity score and precision score than those of Yahoo and Yahoo Motif, in particular, when comparisons were confined to the most relevant top-20 ranked results. Overall, 10% improvement in the top-20 similarity score and precision score from the thesis approach with the biggest improvement seen in the top ranked results. However, a notable limitation of the approach is the performance which fell slightly as the profile contained less information, thereby the short-term compensation still fell short of what was anticipated.

There are no suitable personalization algorithms that fit all search queries. Different algorithms have different strengths and weaknesses. In-depth investigation on extended words will exploit personalization algorithms to enhance the search results, whereby higher precision can be attained. Moreover, as the user gains more search experience, i.e., knowing how to select proper "search words", the precision score will increase. But this will take time to accumulate enough information before the steady state is reached. It is envisioned that additional measures could be employed to shorten the profile accumulation cycle, namely, specificity, sensitivity, and accuracy, to see if the amount of information is sufficient for profile update, thereby user's experience will improve search profile personalization considerably.

6.2 Lessons Learned

Research in the area of personalization has mostly targeted to provide relevant search results. Further research regarding the personalization of search results mainly focused on implicit feedback methods more than explicit feedback methods because implicit feedback methods do not require any additional intervention by the user during the process of constructing profiles and provides an unbiased way to collect information.

Most popular approaches collected user's information by the user explicitly, as user's interests and preferences changed over time. This placed a burden on the user and user's information which might increasingly become inaccurate. For this reason, the thesis suggested a new approach to collect implicit user's information. This technique does not

place a burden on the user's part.

Ant colony optimization has been studied for several research fields. However, ant colony optimization has not yet been well studied in this research. The reason is that ant colony optimization is used mostly to find the shortest path, as ant behavior is always finding the shortest path to the food source. Moreover, ant behavior which always changes the path to the new food source furnishes an analogy to model user's interests and preferences that are always changing as well. Hence, research in personalization of search profile using ant foraging approach was made possible.

The level of personalization was discriminated with the help of short-term and long-term user's interests and preferences. The approach identified short-term and long-term user's interests and preferences by keeping accessed time when the user visited the web pages. The last page that user accessed related with the short-term user's interests and preferences along with the page having the highest frequency on the long-term user's interests and preferences.

As a conclusion, implicit user information collection technique goes much longer way than explicit user information collection technique. It seems to be worthwhile to intensify research on implicit information gathering. Ant colony optimization is exceedingly interesting because its algorithm is flexible which can be applied to many research fields.

6.3 Future Works

The thesis study opens many possibilities for future work. In particular, several issues which could not be targeted or were not comprehensively considered in this work can be further investigated. In just the same manner, new or improved approaches can be analyzed and evaluated.

According to the experiments, building the user's profile from implicit information gathering encompasses some errors because the approach collects user's information from tags and attributes of HTML page. Occasionally, these tags and attributes do not provide the relevant and useful information. However, the approach could tolerate these errors as long as the error rate is low or discover the new approach which can extract the useful information from the web page content so that any detected errors are eliminated.

Another area of future work concerns the relation between extended words and query words. Query words are best described the user's information needs. However, it is difficult for the user to choose the suitable query words for describing the user's information needs.

The user may not know what exactly is looking for or may not use the suitable query words to describe it. Hence, the approach should provide the extended words that can describe the exactly user's information needs by considering the relationship between extended words and query words. Nevertheless, the relationships between extended words and query words may span more than one dimension such that each extended word and query word should relate to one another. This is a challenge because the relationship beyond one dimension can be very complex.

Another area of future work is reducing time consumed to reach the steady state of user's profile. As the thesis approach gains more user's information, the time to find the relevant documents will increase. However, this will take time to accumulate enough information before the steady state is reached. To solve this problem is to employ additional measures to shorten the profile accumulation cycle for user personalization research area.

REFERENCES

[1] Dorigo, M. and Stützle, T. Ant colony optimization. USA: MIT Press, 2004.

[2] Deneubourg, J.L. and Pasteels, J.M. From individual to collective behavior in social insects. London, UK: Birkhauser-Verlag, 1987.

[3] Bilchev, G. and Parmee, I.C. The ant colony metaphor for searching continuous design spaces. In Terence C. Fogarty, Proceedings of the AISB Workshop on Evolutionary Computation, pp.25-39, London, UK: Springer-Verlag, 1995.

[4] Bakis, N. and Sun, M. Intelligent Broker for Collaborative Search and Retrieval of Construction Information on the WWW. Proceedings of the International Conference on Construction Information Technology, pp.86-94, New York, NY: ACM, 2004.

[5] Sérgio N. Exploring Temporal Evidence in Web Information Retrieval. Proceeding of the Seventeenth ACM Conference on Information and knowledge management, pp.243-252, New York, NY: ACM, 2007.

[6] Craswell, N., Robertson, S., Zaragoza, H., and Taylor, M. Relevance weighting for query independent evidence. Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'05), pp.416-423, New York, NY: ACM, 2005.

[7] Li, X. and Croft, W. B. Time-based language models. Proceedings of the Twelfth International Conference on Information and Knowledge Management, pp.469-475, New York, NY: ACM, 2003.

[8] Kelly, D. and Teevan, J. Understanding what works: Evaluating personal information management tools. Seattle, WA: University of Wachington Press, 2007.

[9] Lieberman, H. Letizia: An agent that assists web browsing. In Chris S. M., Proceeding of

the Fourteenth International Joint Conference on Artificial Intelligence, pp.924-929, London, UK: Springer-Verlag, 1995.

[10] Sun, J., Zeng, H., Liu, H., Lu, Y., and Chen, Z. CubeSVD: a novel approach to personalized Web search. Proceedings of the Fourteenth international conference on World Wide Web, pp.382-390, New York, NY: ACM, 2005.

[11] Speretta, M. and Gauch, S. Personalized search based on user search histories. Proceeding of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence, pp.622-628, Washington, DC: IEEE Computer Socuety, 2005.

[12] Koutrika, G. and Ioannidis, Y. Rule-based query personalization in digital libraries. International Journal on Digital Libraries Volume 4 1 (August 2004): 60-63.

[13] Asnicar, F. and Tasso, C. ifWeb: A Prototype of UserModel-Based Intelligent Agent for Documentation Filtering and Navigation in the World Wide Web. In Evelina Lamma et al. (eds.), Proceedings of the sixth International Conference on User Modeling, pp.261-271, London, UK: Springer-Verlag, 1997.

[14] Micarelli, A. and Sciarrone, F. Anatomy and Empirical Evaluation of an Adaptive Web-Based Information Filtering System. User Modeling and User-Adapted Interaction Volume 14 2-3 (June 2004): 159-200.

[15] Gentili, G., Micarelli, A., and Sciarrone, F. Infoweb: An Adaptive Information Filtering System for the Cultural Heritage Domain. Applied Artificial Intelligence Volume 17 8-9 (September 2003): 715-744.

[16] Renssen, A.V. A generic extensible ontological language design and application of a universal data structure. Amsterdam, ZA: IOS Press/Delft University Press, 2005.

[17] Challam, V., Gauch, S., and Chandramouli, A. Contextual Search Using Ontology-Based User Profiles. Proceedings of Eighth International Conference on Computer-

Assisted Information Retrieval, pp.267-249, Hingham, MA: Kluwer Academic Publishers, 2000.

[18] Porter, M.F. An algorithm for suffix stripping. San Francisco, CA: Morgan Kaufmann Publishers, 1980.

[19] Koller, D. and Sahami, M. Hierarchically classifying documents using very few words. Proceedings of the Fourteenth International Conference on Machine Learning, pp.170-178, San Francisco, CA: Morgan Kaufmann Publishers, 1997.

[20] Resnik, P. Using information content to evaluate semantic similarity in a taxonomy. Proceedings of the fourteenth International Joint Conference on Artificial Intelligence, pp.448-453, San Francisco, CA: Morgan Kaufmann Publishers, 1995.

[21] Jiang, J. and Conrath, D. Semantic similarity based on corpus statistics and lexical taxonomy. Proceedings of International Conference on Research in Computational Linguistics, pp.13-19, Morristown, NJ: Computational Linguistics, 1998.

[22] Lin, D. An information-theoretic definition of similarity. Proceedings of the Fifteenth International Conference on Machine Learning, pp.296-304, San Francisco, CA: Morgan Kaufmann Publishers, 1998.

[23] Hirst, G. and St.Onge, D. Lexical chains as representations of context for the detection and correction of malapropisms. USA: MIT Press, 1998.

[24] Leacock, C. and Chodorow, M. Combining local context and WordNet similarity for word sence identification. USA: MIT Press, 1998.

BIOGRAPHY

Pattira Phinitkar was born in 1983. She enrolled in the Faculty of Science, Siam University and graduated with a Bachelor of Science degree, first class honors in 2005. By 2008, she received a computer graphic certification from Concordia University, Canada.

In occupation background, she joined IT Square Co., Ltd as an internship programmer in 2005. The experience gain in programming entailed her to be employed by Pakgon Co. Ltd, Software House Company in 2006.