World-Wide Web Information Discovery via Relevance Feedback

Yue Che Wang, Kenneth

Department of Systems Engineering
& Engineering Management

The Chinese University of Hong Kong

Submitted in partial fulfillment of requirements for the degree of Master of Philosophy

June 1998

Abstract

Web information discovery is a process to locate unfamiliar information on the World-Wide Web. It discovers Web documents based on the tradeoff between two main objectives: exploration and exploitation of the Web information space. Generally, browsing and searching are two human interaction paradigms used on the Web. As more information is made available on the Web, those manual interactions become less feasible, time-consuming and frustrating. The aim of our research is to meet the tradeoff objectives using an automatic intelligent search paradigm and relevance feedback techniques.

We develop a new approach for information discovery on the Web. Our system allows the user to specify the topic profile (information need) by means of various topic profile specifications. An entire example page or an index page can be accepted as the input for the discovery. This is particularly useful when the user does not know how to express the information need. It makes use of a simulated annealing algorithm to automatically explore new Web pages. Simulated annealing algorithms possess some favorable properties to fulfill the discovery objectives. Information filtering techniques are adopted to evaluate the content-based relevance of each page being explored. The Web documents can be evaluated automatically by assigning meaningful, realistic relevance scores. The system is domain independent and does not rely on any prior knowledge or information about the content of Web pages. By employing automatic textual analysis techniques, it has no need of index database maintenance. Furthermore, our approach allows users to provide relevance feedback. A model for relevance feedback is developed as a learning mechanism so as to make the system more adaptive to the variability of the document content and the user's need. Three forms of the relevance feedback model, namely the positive page feedback, the negative page feedback and the positive keyword feedback are proposed. For the positive page feedback handling, the full-text and the page-attribute feedback strategies are developed.

Extensive experiments have been conducted to demonstrate the discovery performance achieved by our discovery system. The results show that the average precision values for the discovery processes using the example pages as input are higher than those only using keywords as input. The input by the index page has also shown its strength in the information discovery on the Web and is comparable with the example page input. The page-attribute based positive page feedback model proposed has been shown to be preferable to the full-text strategy.

Keywords: *WWW, information discovery, information filtering, simulated annealing, relevance feedback.*

論文摘要

「萬維網資訊開發」乃是在萬維網(World-Wide Web)中發掘資訊，並基於資訊探測及資訊利用兩項開發原則中取得平衡的一項過程。一般而言，萬維網使用者普遍採用瀏覽和搜尋兩種人手方法使用萬維網，但以上兩種方法隨著萬維網蓬勃的資訊發展而變得浪費時間，實行上會遇上困難。本研究課題旨在使用自動化智能搜尋方法及相關回應技術，在附合以上兩項原則下進行資訊開發。

我們現正爲「萬維網資訊開發」發展一套嶄新的系統，這系統能令使用者透過不同形式的「論題剖像詳述」(Topic Profile Specification)表達對資訊的需求，使用者可以將整篇網頁或索引頁(Index Page)輸入系統，這些輸入法針對使用者在表達資訊需求出現困難時提供協助。由於模擬冶煉演算法(Simulated Annealing Algorithm)擁有一些優越特性以附合開發原則，系統將利用此演算法進行自動網頁探測，而資訊過濾法(Information Filtering) 則應用於評估網頁在文字上與資訊需求之相關性，繼而爲每一網頁編配相關值。這系統沒有受知識範圍所控制，亦毋需預先設定任何關於網頁的資料，透過自動化文字分析技術，系統毋需建立索引資料庫。此

外，這系統容許使用者提供相關回應(Relevance Feedback)，並利用這些相關回應作爲學習機制，從而使它更能適應萬變的網頁內容及使用者的資訊需求。

實驗已對這系統進行測試，結果發現以整篇網頁輸入法或索引頁輸入法的開發結果，準確性高於只利用關鍵字，相關回應也能提高系統的準確性。

Acknowledgement

I would like to take this opportunity to thank all those who have helped in the production of this thesis, directly or indirectly.

I would first like to express my sincere gratitude to my thesis advisor, Prof. Wai Lam for his constructive criticism, guidance, support. I would also like to thank for his patience, enthusiasm, and being accessible at all times. All these are strong catalysts for the completion of this thesis.

I'd like to thank Prof. K.F. Wong and Prof. K.P. Lam for their precious comments given on this thesis. Next, I would like to thank all the staff in the SEEM department for their diligent help. They provide me with a pleasant working environment.

Finally, I would like to express my deep gratitude to my parents for their support and encouragement. I'd especially like to thank Barbara for her great love, her patience, understanding as well as sharing. Without their warm care, the thesis would not reach its current stage. Thanks to Jess, Cutter, Silvia and my school fellows who make my postgraduate studies joyful and unforgetable.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter gives a brief introduction on the World-Wide Web and brings out the discovery issues on the World-Wide Web. An intelligent content-based information discovery system is proposed. The contributions of our research are mentioned.

1.1 The World-Wide Web

The World-Wide Web (WWW or Web for short) is one of the information services on the Internet [5]. It is a distributed, heterogeneous, hypermedia system for information storage and retrieval. Originally, it was built to support the scientists at CERN, the European Particle Physics Laboratory in Geneva, Switzerland [6]. It was opened to public along with the popularity of the Internet. Basically, WWW is constructed based on the client-server architecture. The Web documents, or pages, which are composed by the Hypertext Markup Language (HTML [5]), are published (stored) in different Web servers on the Internet. These servers have their own IP addresses that can be accessed by their individual unique Uniform Resource Locator (URL [5]). The Web pages requested is transferred by the Hypertext Transfer Protocol (HTTP [5]) from the remote Web servers through the TCP/IP connections. The HTML pages will be parsed and displayed by a software tool called the Web page browser (the client). By clicking the hyperlinks embedded in the hypermedia pages through the user interface of the browser, the user can read information from pages to pages, from servers to servers by further traversing the information network hierarchies. As inspired by the architecture of the hypertext system, information is globally distributed over millions of Web servers. With the specifications of HTML and multi-platformed browser, the browsing of Web pages is made to be machine-independent.

1.2 Searching Information on the WWW

Generally, there are two human interaction paradigms [10] used on the Web in order to satisfy an information need:

1. **Browsing**. It is sometimes called exploration. It refers to the human-guided activity to explore the information network. Users may follow the hyperlinks and try to find relevant information on the fly. Users may not get a clear idea on what they want but just wandering.

2. **Searching**. It refers to the human activity on submitting a description of their desired information to a searchable index and the index will spot out the relevant information for the user.

The architecture of WWW promotes the ease for publishing Web pages on the Web servers. Information can be easily updated from time to time. It also promotes the ease of accessibility and usability. A large pool of distributed, heterogeneous, networked information is made available for people on the earth. In the recent years, both of the number of Web servers and the user population are growing exponentially. These two increasing factors have led to the growth in the amount of information put on the network. This situation is termed as "information overload" [9]. As more information becomes accessible, both of the mentioned interaction paradigms may be very time-consuming, inefficient and less feasible. Web users face the frustrating task of sorting through thousands of documents available on a particular subject. Browsing allows users to explore only a very small portion of the large WWW information space. Web users may be confused when they are following the hyperlinks down to a certain level and finally disoriented. Even they may omit some links and may never reach the useful resources. Besides, it is difficult for the users to place exact, precise descriptions of their information needs in searching. In this way, pinpointing useful and relevant information from the sparsely distributed pages are hard to achieve. Though there are several automatically generated searchable indexes created, each of them has an unique interface and a database covering different parts of the Web. Users have to try their queries repeatedly across those indexes. These indexes may return hundreds of unrelated pages [25]. To address the above problems, an intelligent, active, automatic textual analytical discovery system is crucial for facilitating the information discovery task.

1.3 Intelligent content-based information discovery on the Web

Information discovery, or resource discovery, is a process to locate relevant and useful information on a information repository. By projecting such kind of process onto the Web, it locates relevant and useful information on the Web [25]. Discovering resources in such a large information network is usually based on a tradeoff between two objectives:

1. **Exploitation**, which places a focal consideration on the currently available configuration (solution).
2. **Exploration**, which achieves a possibly better configuration (solution) from the current search space.

A Web information discovery system will be able to retrieve, filter and evaluate Web information resources with the above tradeoff considered. The discovery results are ranked and recommended to the user. As the user's browsing and searching on the Web are some repetitive tasks, the goal of the information discovery system is to traverse the Web automatically and intelligently so as to satisfy some long-term information needs. The discovery system needs to incorporate a model to evaluate the usefulness of the discovered information by analyzing the textual content of the Web pages. Furthermore, as the user interest and the online information are said to be ever-changing, the system should be able to learn the user's preference and adapt to the changes in the information space. A new WWW information discovery approach is proposed [71] and will be fully discussed in subsequent chapters.

The main contributions brought by our system include the followings :

- It is an intelligent, active information discovery system, with a new technique for Web page exploration. It learns the user information need expressed by the topic profile and keeps minimal human-guided document evaluations. Through an automatic Web page evaluation model, a relevance score is assigned to each page being examined. It recommends pages that are believed to be relevant to the topic.

- Automatic intelligent Web exploration model is developed based on a simulated annealing algorithm. Simulated annealing has some nice properties that can fulfill the exploitation and exploration objectives. The discovery problem will be treated as an optimization problem. The objective is to find a page on the Web with an optimal, or good relevance score under fixed computational resources. Our simulated-annealing-based exploration possesses some desirable properties suitable for this problem. It will be explained in Chapter 4.

- It allows a variety of topic profile specifications describing the user information need. It supports the example page profile specification and the index page profile specification. These profile specifications serve as a more descriptive basis for the system to set off on a discovery trail. The keyword profile specification, like the ones used in conventional searching indexes, is also supported. As the incoming streams of documents are transferred, information filtering techniques are adopted to measure the similarity between the topic profiles and those documents fetched.

- A relevance feedback model is developed as a learning mechanism so as to make the system more adaptive to the variability of the page content and the user's need. Page-based relevance feedback models are proposed to relieve the user from manual determination of relevant or irrelevant keywords. In addition, two relevance feedback feature extraction strategies, namely the full text as well as the page-attribute feedback strategies have been introduced.

- It does not require any prior knowledge about the information need or the content of Web pages. Therefore, it does not place any restriction on the application domain. No index database is required. Furthermore, it can incorporate with a thesaurus which can enhance the discovery performance.

- Extensive experiments have been conducted to demonstrate the discovery performance achieved by the system prototype. The results are evaluated and analyzed.

1.4 Organization of the Thesis

The rest of this thesis is organized as follows. The next chapter reviews some of the previous work. Chapter 3 provides an overview of the proposed discovery system, the document representation, and the topic profile representation. Chapter 4 discusses the content-based relevance score calculation. It also introduces the relevance feedback model for adapting to the content of the Web documents and the topic profiles. Chapter 5 discusses the automatic Web page exploration model which is based on simulated annealing techniques and how the exploration algorithm incorporates with various relevance feedback models. Chapter 6 presents the experimental results obtained from the system implemented and the results are analyzed. Chapter 7 makes some conclusions and introduces some future work.

Chapter 2

Literature Review

This chapter discusses some related research work and similar systems. In recent years, the Internet has increased its popularity. As mentioned before, the interactive and multimedia properties make the WWW applicable for publishing information. The volume of online information has tremendously increased and the trend is still in upwards direction. Advanced tools or systems should be built in order to facilitate the user to dig the pieces of relevant information out from the WWW universe.

2.1 Search Engines

Various WWW information discovery or search systems share the primary goal of digging the pieces of relevant information out from the WWW for the

Web users. The most common and popular ones are the searchable indexes (i.e., search engines). Some of such tools and systems developed have been reviewed in [24]. They allow the users to place some keyword queries and return a set of search results to the users. For example, Webcrawler [55] maintains a full-text index of the Web that can be queried for documents about a specific topic. It periodically makes use of a Web robot to discover and make index on new pages by using a graph traversal algorithm. It promotes searching in a server-breadth-first approach. Such breadth-first practice is to ensure a broader coverage in results. Besides, it supports a real-time search mode. The Lycos system [47] employs a Gnu DBM file to store partial information of pages. The information includes titles, headings, some weighty words, several lines of documents. Other examples of search engines include AltaVista [18], Hotbot [20] and Infoseek [34]. They are comprehensive, automatically generated indexes.

These searchable indexes possess some common disadvantages. They require the users to describe their interests by specific words that match those in the target Web pages. Pinkerton conducted a study on the user behavior of using search engines. He found out that only a limited number of keywords are specified in the query [55]. This will lead to thousands of "related" documents. Those keywords may have different semantics. (e.g. "Java" is a kind of coffee, but can also be a type of computer language). Even, they do not take into account that documents are organized as information network in which hyperlinks contain important clues for information discovery. In this way, they may return a large set of imprecise, or unrelated pages. Besides, most of them make use of exact keyword matching for searching and retrieval, the search results contain many duplicated hyperlinks. Furthermore, the validity of the hyperlinks stored in their databases is not validated frequently. As a result, those search results will contain many meaningless or unreachable hyperlinks.

2.2 Information Indexing Systems

Research on the Internet information extraction applying some advanced techniques like information retrieval (IR) and information filtering (IF) has begun recently [12][46][66][69]. They try to consider the meta-data, the document attributes, the hyperlink information together with the content of the document. In some of those research work concerning the Web information discovery, document information and hyperlink information about a portion of the WWW are stored in a database. This database will act as a cache. Searching in response to the query will make reference on this database first, before any traversal of the WWW information network. The database is constructed with the Web robots which pre-traverse the links and obtain the necessary information. Repository Based Software Engineering [23](RBSE), which was developed by Eichmann, is a content-based index. It discovers the Web by retaining the structure as a graph representation and a full text index of the Web pages encountered in a database. Harvest [11] is a system that gathers information from diverse repositories, builds topic-specific content indexes and builds cache for the objects from the Internet. It can be configured to automatically collect and summarize related objects from the Internet into a large collection or to collect, summarize and handle a tiny specialized collection. WISE [72] makes use of an autonomous WWW robot which visits a Web page, traverses the hyperlinks in a breadth-first manner to retrieve pages, extract keywords with hyperlink data from the pages as well as insert those keywords and hyperlink data into an index. It also makes use of information retrieval techniques to evaluate and rank the documents. These systems require occasional updating of their databases. Eguchi *et. al.* introduced a WWW retrieval system [22], similar to WISE. It is a WWW IR system with an index-database architecture. The parameters in the standard Rocchio model is dynamically tuned according to the user's retrieval behaviors.

Though this type of discovery systems apply some advanced techniques like Web robots, they share similar disadvantages as stated by Koster [38]. Due to the limitation of storage, only limited, partial features of the documents will be stored in the database. Also, in order to keep the searching valid, the databases will need occasional updating. As a result, it requires high traffic demand of network loading.

2.3 Agent-based Systems

Recently, discovery systems applying intelligent agent [29] technology have been under investigation. As discussed in [45], a software agent is a program or computing system that can operate autonomously and accomplish some specific tasks without much user intervention. Such agents may be introduced to the context of the Web information discovery so as to browse and search the Web pages on behalf of the users. Etzioni gave a brief introduction on how the intelligent agent are deployed on the Web [26]. The user interaction is minimized. They also possess the capability of adapting to user's ever-changing information need and searching behavior. They look for relevant information, analyze and evaluate pages against induced users' profiles. Hence, they are suitable for serving long-term user information needs. Chen *et. al.* summarized some techniques in artificial intelligence like neural networks, symbolic learning that can incorporate with information retrieval systems [16]. The problems as well as the framework of the Internet search was addressed. The framework proposed includes the Internet categorization using multi-layered Kohonen self-organizing feature map, concept-based search based on cluster analysis and Hopfield net associative retrieval, and an intelligent agent supporting efficient client-based search of relevant Internet information. Letizia [44] is a user interface agent that tracks user's personal behavior and attempts to anticipate items of interest by doing concurrent, automatic exploration of hyperlinks. It keeps tracks of the user's past behavior. A rough approximation of the user interest can be obtained. Rhodes and Starner developed the Remembrance Agent [57], which is a system augmenting human memory by displaying a list of documents relevant to the user current context. It runs continuously without any user intervention. Syskill & Webert [53] learns to evaluate the Web pages by analyzing the page information using several learning algorithms. Those algorithms include Bayesian classifier, nearest neighbor classifier, decision trees and similarity measurement in information retrieval. The system possesses an algorithm [54] to learn and revise the user profile so as to suggest which links a user would be interested to explore and to create a Lycos query to find pages that would interest the user [7]. WebWatcher [1] is a learning apprentice [50]. It performs look-ahead searches and suggests the user on the basis of reinforcement learning. It learns to assist by creating and maintaining a log file for each user. It improves its guidance through user feedbacks. It also incorporates machine learning methods to automatically acquire knowledge for selecting an appropriate hyperlink given the current Web page viewed by the user's information goal. Balabanović and Shoham developed a system [3] which learns to browse the WWW on behalf of a user. It surfs the WWW in a bounded amount of time, selects the best pages at a moment by the best-first search algorithm and receives a feedback from the user. The feedback evaluation will update the search and selection heuristics. Ngu and Wu proposed a Web agent, SiteHelper [52], which helps the Web users by learning and identifying their areas of interest. It works with a local Web server and indexes the Web pages on the Web server by using a keyword dictionary local to the Web server. Furthermore, it supports interactive and incremental learning based on the indexing of the Web pages. Recently, Menczer proposed the ARACHNID [48]. It is a distributed algorithm for information discovery in the WWW. The algorithm is based on a distributed, adaptive population of intelligent agents making local decisions. Unsupervised machine learning techniques, evolution techniques are used to make the agent more adaptive. B. Starr *et. al.* described the Do-I-Care agent [70], which also uses machine learning to detect "interesting" changes to Web documents previously found to be relevant. It periodically visits a user-defined list of target documents and identifies any changes since the last time it has visited. By extracting some attributes, it decides if the changes are interesting and notifies the user. Krulwich and Burkey proposed the Infofinder Agent [40] which learns user information interest from sets of messages or other online documents that users have classified. It is an intelligent agent that uses heuristics to extract significant phrases from documents for learning. The agent's induction algorithm was developed based on the non-uniformly distributed sample documents. It also learns standard decision trees for each user category which can be easily transformed into search query strings. By applying those tools, the wealth of information can be made effective. (For a more detailed list of other similar Internet spiders/agents, readers are advised to refer to the Web page [37]). But this type of systems may require domain expert knowledge.

2.4 Information Filtering Systems

Information filtering (IF) concerns about a variety of processes involving the discovery of information to people who need it. Belkin and Croft [4], Foltz, and Dumais [27] defined IF as the counterpart of information retrieval (IR) [63][64]: They share many similarities . An IF system is an information system designed for unstructured data. Primarily, it deals with textual information. In fact, it should be more general that multi-media data can also be handled. In most cases, those kinds of data involve large amount of incoming streams. As a result, filtering is often meant to imply the selection or elimination of data from a dynamic incoming streams of data. The filtering process will be based on the descriptions of individual information preferences, often called profiles

[45]. Such profiles typically represent long-term and stable interests. Both of the profiles and the data streams will act as the input of the filtering systems. Binkley and Young developed the RAMA [8], which proposed an architecture for the general Internet information filtering. It makes use of the personal information profile, to filter Internet information such as the USENET news, or the information from an anonymous FTP server.

Like other classical IR systems, the initial issue is to obtain a representation of each textual document and the profiles suitable for a machine to digest. A document (profile) representation could be a list of extracted words considered to be significant. The systems will have some evaluation mechanisms to compare the incoming document streams (in document representation) against the specially represented profiles. Those documents that are found to be complied with the profiles will be extracted from the streams and returned to the users. The performance of the filtering will be measured by the precision and recall percentages [63]. As a whole, those IF systems are assumed to be used repeatedly by persons with long-term goals or interests.

The evaluation mechanism is referred as the information retrieval models, or ranking algorithms in some literatures [28][63][64]. There are three commonly-used methodologies, namely the Boolean, the vector space and the probabilistic retrieval models. The Boolean model allows the user to input a simple query such as a sentence or a phrase. The system will retrieve information based on the concept of an exact match of the profile representation with one or more document representations. The result of the comparison operation in the Boolean retrieval is a partition of the incoming streams into a set of relevant documents, and a set of irrelevant documents. A drawback of this model is that it does not allow any form of relevance ranking of the filtered document. On the contrary, presenting filtered document according to the degree of relevance will make the system more usable and effective. The vector space model [63] is proposed and has been incorporated with the IF systems as described in [42]. It treats the documents and the profiles as vectors in a multidimensional space. The elements in those vectors (the terms in the profiles and the documents) can be weighted in order to reflect the importance. Vectors can be compared by some correlation similarity measurements. The more similar a vector representing a document is to a profile vector, the more likely that the document is relevant to that profile. The probabilistic retrieval model is developed by Robertson and Sparck Jones [61]. It ranks the documents in the order of their probabilities of relevance to the profile, given all the evidence available. Such evidence may be the statistical distribution of terms in the datastreams. To achieve optimal performance, documents should be ranked according to their probability of relevance to the profile.

Apart from the aforesaid classical information retrieval models, the study of using logic in information retrieval model has begun. From 1986 onwards, Van Rijsbergen proposed some work on logical information retrieval [59][60]. Both of the classical and non-classical logic are defined and used to model an information retrieval system. The use of logic for IR modeling is still in its infancy and on-going.

Chapter 3

Overview of the Proposed Approach

This chapter presents an overall view of our discovery approach. It also introduces the topic profile specification which is the representation of the user long-term information need or specific interest. In order to allow effective processing by our discovery system, the topic profile specification and the documents fetched from the WWW are converted to an internal representation, namely, the profile feature representation and the document feature representation respectively. Details of the representation will be given in this chapter.

3.1 System Architecture

In general, our proposed system consists of several modules, as depicted in Figure 3.1.

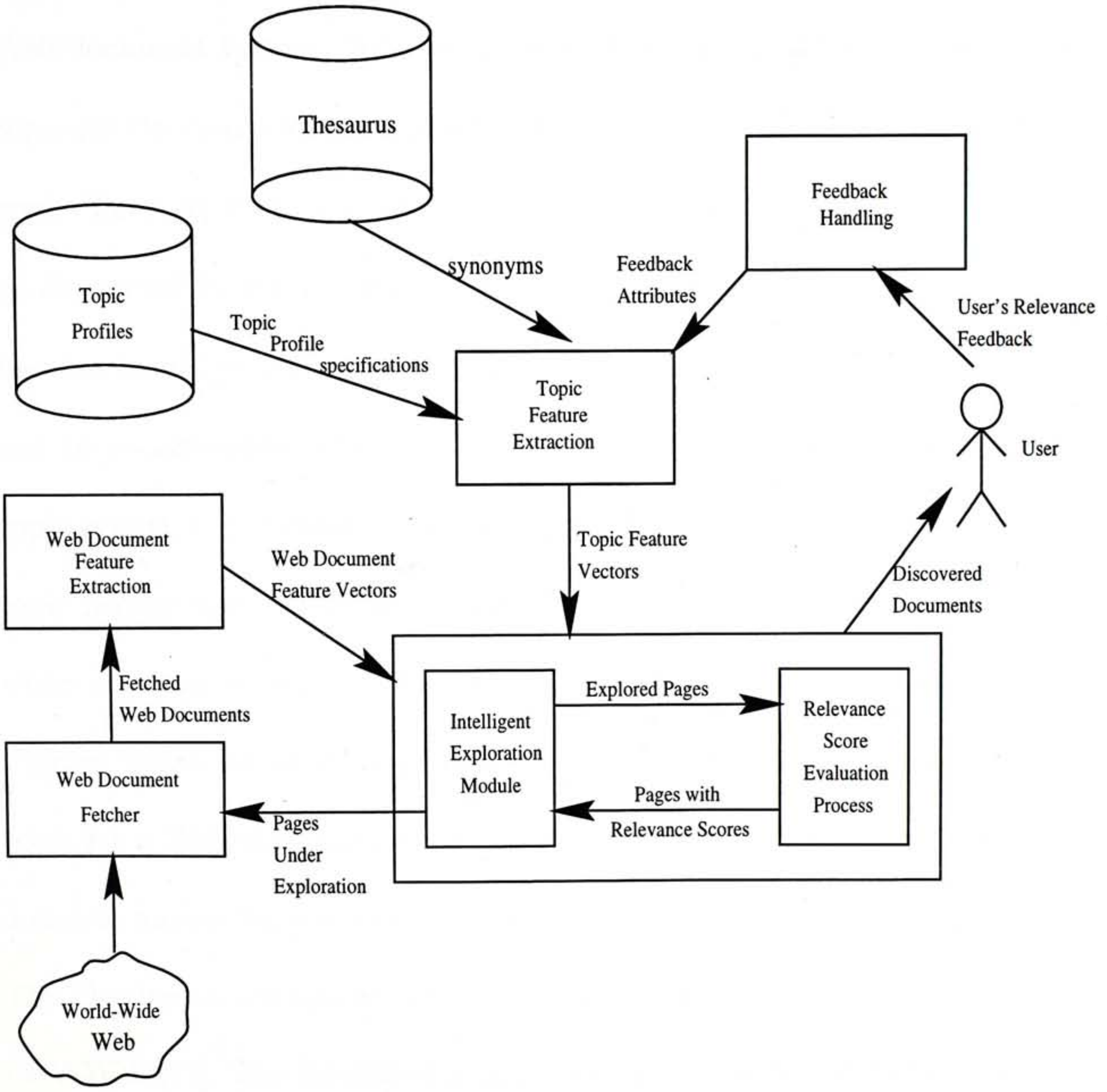


Figure 3.1: Block diagram of the Web document discovery system.

The Web Document Fetcher fetches the document from the WWW. The fetched document then passes to the Web Document Feature Extraction in order to extract the necessary document features. Those features are basically the stemmed words from the fetched pages with the stop words removed. The Web document feature vector is created. The topic profiles expressed by the topic profile specifications represent the topic of the documents to be discovered. They represent some long-term information needs or specific topics to be discovered from the WWW. Topic feature vectors are formed by extracting features from the topic profile specifications. At the same time, a thesaurus can be incorporated with the discovery system to expand the words in the topic profile, e.g "football" can be described by another word, "soccer". The topic feature vectors may be augmented, if any, by a thesaurus so as to get a wider coverage of pages with similar topic. The Relevance Score Evaluation Process makes use of information filtering techniques to compute a relevance score for a Web page against the topic features. This relevance score acts as a metric measuring how close a Web document is related to the topic profile. The filtering model appears to be effective in information retrieval or filtering on WWW [73]. The Intelligent Exploration Module will determine which Web documents will be explored next. We propose a simulated annealing algorithm to achieve this task. It will decide where to traverse down the Web network.

The system will return a set of discovered documents found to be relevant to the topic profile. Users may optionally give positive and negative feedbacks on the discovered documents. The feedback handling module will extract the necessary feedback attributes from the relevant and non-relevant documents so as to refine the topic profile vector. With respect to the design of the discovery system, it possesses several advantages. Pages can be arranged according to the order of relevance score. The topic feature vectors can be easily modified, making it possible to adapt to the document space dynamically. This dynamic changes will be easily invoked by the relevance feedback processes.

The main features of our proposed approach are summarized as follows:

1. The discovery system is designed based on new models of information filtering and intelligent search for automatic Web page evaluation and exploration.

2. The Web documents are automatically evaluated against the user information need. This is accomplished by information retrieval techniques in the Relevance Score Evaluation Process shown in Figure 3.1. A relevance score is assigned to each document encountered.

3. A simulated annealing algorithm is employed in the component called Intelligent Discovery Module. Simulated annealing is adopted since it has some desirable properties that are suitable for exploiting as well as ex-

ploring Web documents automatically. It tries to find a document from the WWW with an optimal, or good, relevance score under fixed computational resources.

4. A variety of topic profile specifications are supported. An example page can be fed into the system as input explicitly. This is useful when the user may not know how to specify their information need. Likewise, an index page which contains some hyperlinks of similar example pages can be submitted to the system. In this case, the index page specifies a set of example pages with similar topics.

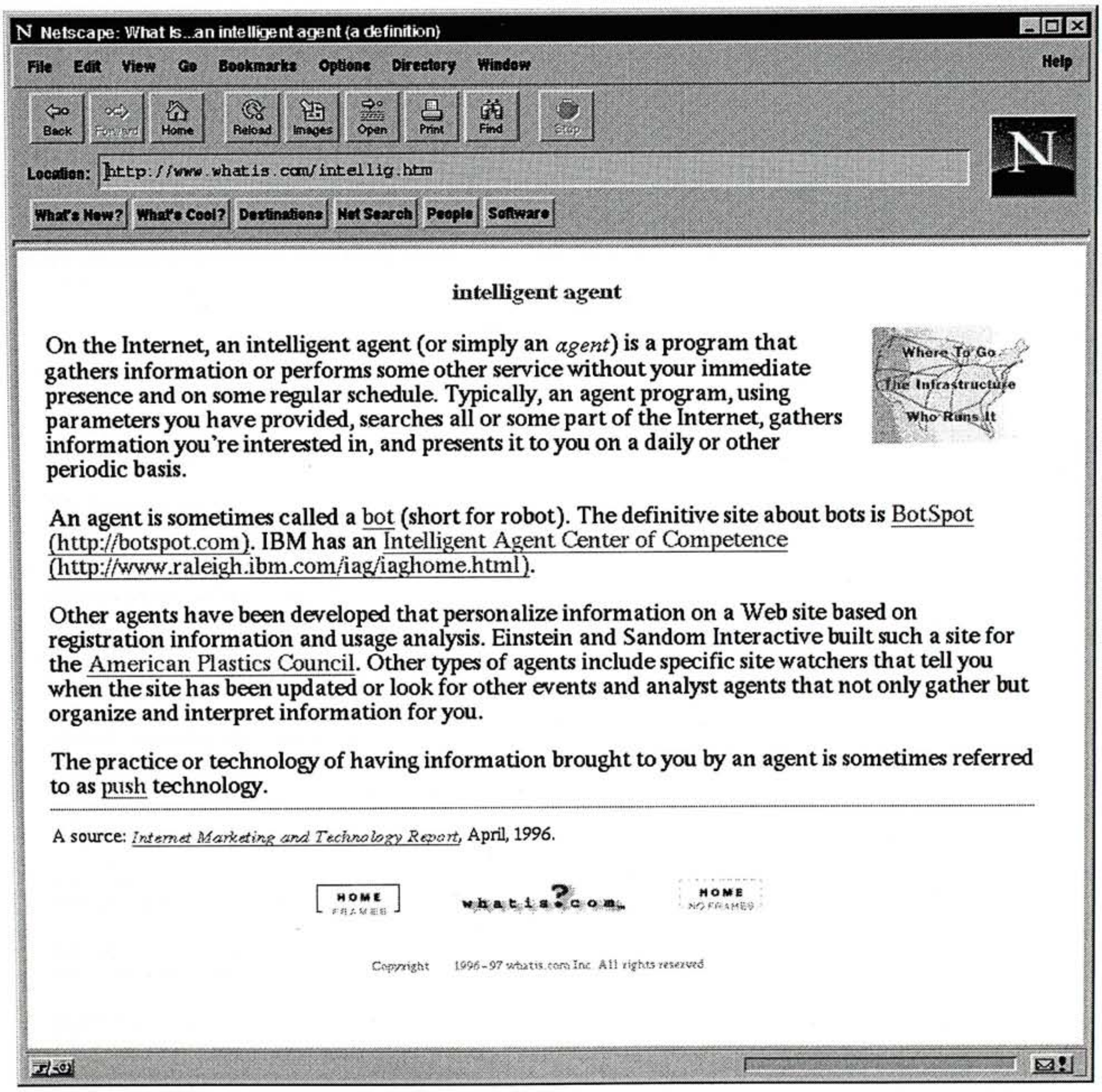
5. A learning model based on relevance feedback is developed. At any point of operation, users may suggest which documents are relevant or irrelevant. As a result, there is no need for the user to supply appropriate keywords. Two relevance feedback feature extraction strategies, namely the full text relevance feedback and the page-attribute feedback are proposed.

6. The system does not require any prior knowledge about the information need. No indices or databases have to be maintained. Besides, it can incorporate with a thesaurus which can increase the coverage of the search results.

3.2 Topic Profile Specification

The topic profile specification allows users to specify a certain topic or interest. The system allows several means for a user to specify profiles. One form is an example Web page specification. It is particularly useful when Web users do not get a clear idea on which keywords they should supply for the query. It is also useful when they want to dig out some similar pages described by the example page. For example, a user may use the example page, as shown in Figure 3.2, to request the discovery system to find documents about intelligent agent.

Our system also allows users to specify the profile by means of an index page. This index page is a plain text file written in HTML containing some Web page addresses having a similar topic. It may be an arbitrary chosen Web page containing hyperlinks, or it can be explicitly created by the user. For instance, the user may find a document about intelligent agent, which contains several outgoing hyperlinks leading to other intelligent agent resources as shown in Figure 3.3. Or the user may also prepare manually a self-made index page


Netscape: What is...an intelligent agent (a definition)
File Edit View Go Bookmarks Options Directory Window Help
Back Forward Home Reload Images Open Print Find Stop
Location: http://www.whatis.com/intellig.htm
What's New? What's Cool? Destinations Net Search People Software
Where To Go
The Infrastructure
Who Runs It

intelligent agent

On the Internet, an intelligent agent (or simply an *agent*) is a program that gathers information or performs some other service without your immediate presence and on some regular schedule. Typically, an agent program, using parameters you have provided, searches all or some part of the Internet, gathers information you're interested in, and presents it to you on a daily or other periodic basis.

An agent is sometimes called a bot (short for robot). The definitive site about bots is BotSpot (http://botspot.com). IBM has an Intelligent Agent Center of Competence (http://www.raleigh.ibm.com/iag/iaghome.html).

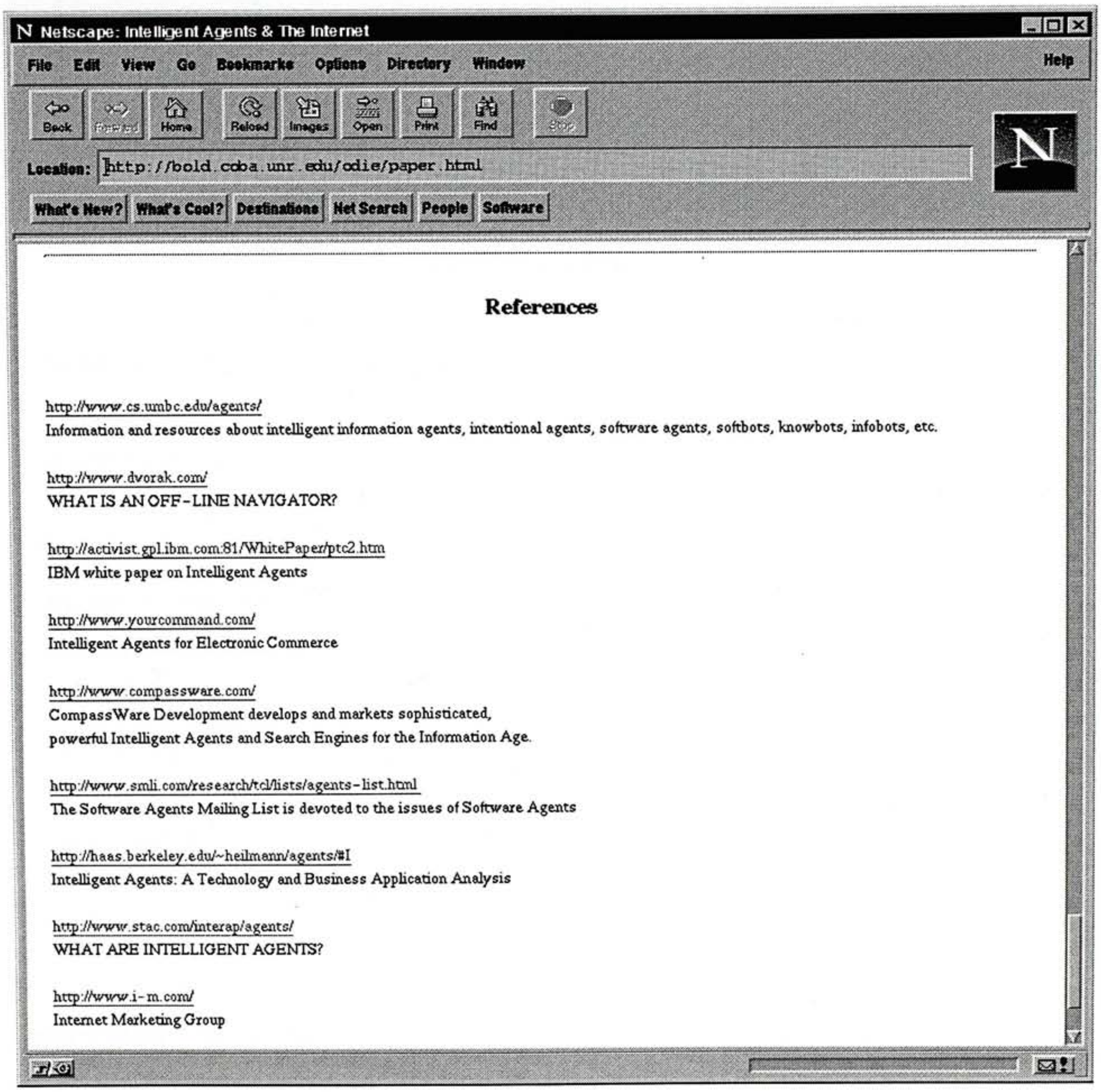
Other agents have been developed that personalize information on a Web site based on registration information and usage analysis. Einstein and Sandom Interactive built such a site for the American Plastics Council. Other types of agents include specific site watchers that tell you when the site has been updated or look for other events and analyst agents that not only gather but organize and interpret information for you.

The practice or technology of having information brought to you by an agent is sometimes referred to as push technology.

A source: *Internet Marketing and Technology Report*, April, 1996.

Copyright 1996-97 whatis.com Inc. All rights reserved

Figure 3.2: An example page specification describing the topic "Intelligent Agent".


N Netscape: Intelligent Agents & The Internet
File Edit View Go Bookmarks Options Directory Window Help
Back Forward Home Reload Images Open Print Find Stop
Location: http://bold.coba.unr.edu/odie/paper.html
What's New? What's Cool? Destinations Net Search People Software

References

http://www.cs.umbc.edu/agents/
Information and resources about intelligent information agents, intentional agents, software agents, softbots, knowbots, infobots, etc.

http://www.dvorak.com/
WHAT IS AN OFF-LINE NAVIGATOR?

http://activist.gpl.ibm.com:81/WhitePaper/ptc2.htm
IBM white paper on Intelligent Agents

http://www.yourcommand.com/
Intelligent Agents for Electronic Commerce

http://www.compassware.com/
CompassWare Development develops and markets sophisticated,
powerful Intelligent Agents and Search Engines for the Information Age.

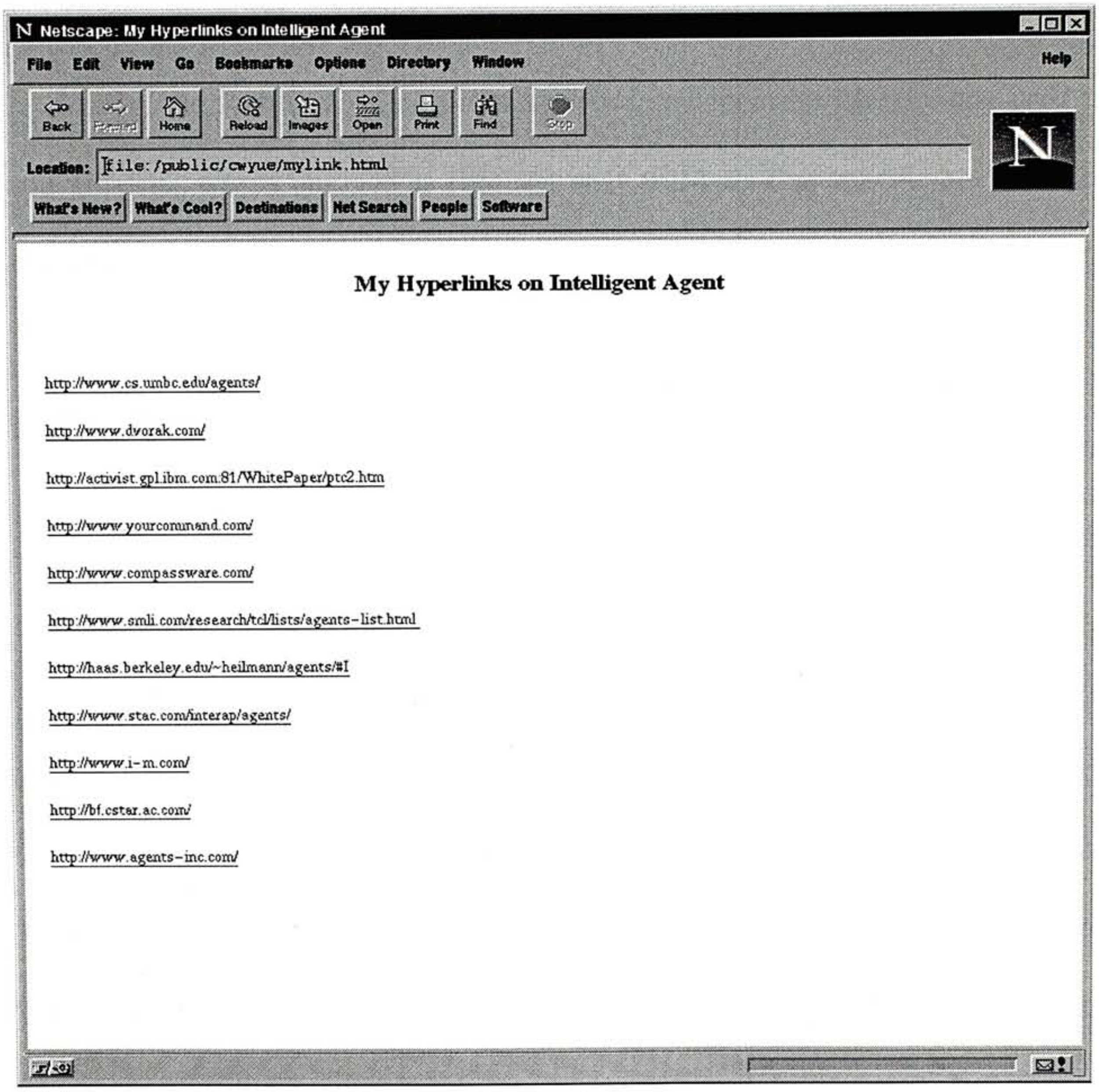
http://www.smli.com/research/tcl/lists/agents-list.html
The Software Agents Mailing List is devoted to the issues of Software Agents

http://haas.berkeley.edu/~heilmann/agents/#I
Intelligent Agents: A Technology and Business Application Analysis

http://www.stac.com/interap/agents/
WHAT ARE INTELLIGENT AGENTS?

http://www.i-m.com/
Internet Marketing Group

Figure 3.3: An index page specification describing the topic "Intelligent Agent".


Netscape: My Hyperlinks on Intelligent Agent
File Edit View Go Bookmarks Options Directory Window Help
Back Forward Home Reload Images Open Print Find Stop
Location: File:/public/cwyue/mylink.html
What's New? What's Cool? Destinations Net Search People Software

My Hyperlinks on Intelligent Agent

http://www.cs.umbc.edu/agents/

http://www.dvorak.com/

http://activist.gpl.ibm.com:81/WhitePaper/ptc2.htm

http://www.yourcommand.com/

http://www.compassware.com/

http://www.smli.com/research/tcl/lists/agents-list.html

http://haas.berkeley.edu/~heilmann/agents/#I

http://www.stac.com/interap/agents/

http://www.i-m.com/

http://bf.cstar.ac.com/

http://www.agents-inc.com/

Figure 3.4: A self-made index page specification describing the topic "Intelligent Agent".

by including the hyperlinks. An example of such self-made index page is shown in Figure 3.4. In essence, the index page topic profile specification provides a means for submitting multiple "document-queries" to the system.

Lastly, the system also supports keyword specification like the one used in conventional search engines as the user may want to give specific keywords to describe the profile.

3.3 Text Representation

We employ the vector space model to represent Web documents and topic profiles. In vector space representation, documents and queries will be transformed into vectors in some hyper-space. A distance metric is defined over the space. When a query is received, it is translated to its vector representation. Document vectors in the proximity of the query vector are retrieved in response to the search. The advantage of using the vector space model is that it supports partial matching. A raw Web document can be served as the input of the system explicitly. Once the text is transformed to a vector, the defined distance metric can be applied. An intelligent query can be constructed. By taking a further leap, several documents can be taken as one query. A group of documents with similar topic can form a more descriptive query class for document filtering.

3.3.1 Profile Feature Representation

Before any further processing, certain distinctive features need to be extracted from the topic profile specification. Those features, composed of words, are automatically extracted to represent the information need. The stop words, which appear due to grammatical purpose (e.g. the, of, a, to), will be removed. The remaining words will further be automatically stemmed to form the word-stems. These pre-processed words are usually known as "terms".

To deal with an example page submitted by a user as shown in Figure 3.2, the system makes use of all the words, with the HTML tags and the programming scripts removed as features. The word-stems and their corresponding term frequencies are extracted as shown in Table 3.1. Likewise, those text enclosed by the <TITLE> HTML tag will be transformed to their word-stems, with their term frequencies shown in Table 3.2. Those term frequencies of the ordinary terms and the title terms are the elements for the profile. Back to our example just given, it can be observed from the tables that the word-stems like "agent", "intellig", and "bot" can be extracted.

If an index page is given, all of the pages specified in the index page will be fetched. Words found in those fetched Web pages that are highlighted by the special HTML tags such as boldface, italic, underlined, or words enclosed by heading tags are extracted to form the topic feature vector by the same

Word-Stem	Term Frequency	Word-Stem	Term Frequency
agent	10	interpret	1
inform	6	regul	1
site	5	botspot	1
internet	3	brought	1
intellig	3	defin	1
techn	3	ibm	1
gather	3	einstein	1
anal	2	interact	1
program	2	person	1
bot	2	specif	1
present	2	market	1
interest	1	april	1
regist	1	organ	1
council	1	plast	1
updat	1	search	1
push	1	provid	1
typ	1	period	1
watch	1	part	1
cent	1	sandom	1
call	1	sourc	1
basi	1	reserv	1
base	1	servic	1
simpl	1	web	1
refer	1	paramet	1
perform	1	copyright	1
right	1	robot	1
daily	1	typic	1
schedul	1	usag	1
report	1	develop	1
event	1	includ	1
americ	1	practic	1
short	1	built	1
compet	1		

Table 3.1: The word-stems with the corresponding term frequencies representing the example page shown in Figure 3.2.

Title Word-Stem	Term Frequency
defin	1
agent	1
intellig	1

Table 3.2: The title word-stems with the corresponding term frequencies representing the title of the example page shown in Figure 3.2.

extraction model. The term frequencies for those extracted terms are multiplied by the degree of importance for different HTML tags. The degree of importance for the HTML tags adopted in our discovery system is shown in Table 3.3. For the keyword profile specification, the keywords are transformed to terms in the topic feature vector directly.

HTML Tag	Degree of Importance
$<b>$	5
$<big>$	5
$<em>$	5
$<h1>$	6
$<h2>$	5
$<h3>$	4
$<h4>$	3
$<h5>$	2
$<h6>$	1
$<i>$	5
$<u>$	5
$<strong>$	5

Table 3.3: The degree of importance for the HTML tags adopted in our discovery system for augmenting the various term frequencies.

With those terms, the topic profile is expressed by a topic feature vector as follows:

$$V_Q = \{w(Q, q_1), \ldots, w(Q, q_m)\} \tag{3.1}$$

where each component $w(Q, q_i)$ represents the weight of the corresponding term q_i among the m terms extracted. These weights represent the importance of the terms in a topic profile specification. Once the topic feature vector is formed, the corresponding synonyms of those terms can be drawn from the thesaurus. The dimension of the topic feature vector can then be expanded. The interpretation of the term weights in the topic feature vector will be discussed in Chapter 4.

3.3.2 Document Feature Representation

The documents retrieved from the remote sites also need to be converted to feature vectors. Like the one associated with the topic profile specification, all of the words except the HTML tags and the programming scripts are automatically extracted, stemmed, with the removal of stop words. These pre-processed words are grouped to form the document feature vector D,

$$V_D = \{w(D, d_1), \ldots, w(D, d_n)\} \tag{3.2}$$

where $w(D, d_j)$ represents the weight of term d_j upon the n terms extracted from the document. Likewise, these weights represent the importance of the terms in the corresponding document. With both of the query and the Web pages retrieved in the form of vector, relevance score can be calculated by a specially designed similarity function. The term weight definition, relevance score evaluation process and the relevance feedback model will be discussed in Chapter 4.

3.4 Advantages of the Topic Profile Specifications

Through our proposed example page and index page topic profile specifications, a large vector of keywords can be extracted from the specifications, just like the one shown in Table 3.1 and Table 3.2. The combination of keywords can provide sufficient descriptive information describing about the information need. In this way, the precision of the discovery system can be increased, as comparing with the information searching approaches used in the search engines. In those search engines, usually the user may only submit a few of simple keywords to the engines but a large set of imprecise results will be returned. The precision is affected. If an equivalent precision has to be achieved by the search engines, ones should submit such a large number or dimensions of terms (features) to those search engines. Such practice will be impossible in reality. Besides, search engines cannot handle such a large vectors though such vectors may be entered as query. Our proposed approach can produce an effective query for the user. Experiments showing the results achieved by the topic profile specifications will be shown in Chatper 6.

Chapter 4

Relevance Score Evaluation Process and Relevance Feedback Model

This chapter discusses the relevance score evaluation process and the relevance feedback model. A specially designed similarity function is developed for the relevance score evaluation between the topic profile specification and the documents fetched. Three relevance feedback models with two feature extraction schemes on these models are proposed.

4.1 Term Weights

After a topic profile specification has been transformed to the topic feature vector, the system starts to explore the WWW. Once it encounters a document fetched by the Web Document Fetcher, the document feature vector is produced and a relevance score against the topic feature vector is determined. This relevance score measures how closely a Web document is relevant to the topic profile. Basically, this relevance score is calculated based on the term weights in the topic feature vectors and the document feature vectors. The various terms weights are determined by the TF-IDF scheme [63]. The "TF" is the term frequency, whereas the "IDF" stands for the inverse document frequency [64] of a term. This formulation enhances the terms which appear in fewer documents, while degrading the terms occurring in too many documents. As a result, the document-specific features will be highlighted while the collection-wide features are diminished in importance on the contrary. The inverse document frequency (IDF) is denoted by the following formula:

$$I(x) = \log(\frac{N}{f(x)}) \tag{4.1}$$

where N is the number of documents in the information space (document collection) and the function $f(x)$ is usually known as document frequency which is the number of documents among the N documents that contain the term x.

As we can see, most of the Web documents have titles, which are those text enclosed by the <Title> tag in the corresponding HTML-written file. Most probably, they may be treated as alternative features denoting the topic of the documents. We make use of this characteristic to augment the classical TF-IDF function so as to make the system more suitable for the WWW environment. The weight of a term is calculated as follows:

$$w(a,b) = \frac{(f(a,b) + \beta f'(a,b)) \times I(b)}{\sqrt{\sum_{\forall c \in a}((f(a,c) + \beta f'(a,c)) \times I(c))^2}} \tag{4.2}$$

In the TF-IDF function shown in Equation 4.2, $w(a,b)$ represents the combined weight of the term b in the document a. The function $f(a,b)$ denotes the term frequency, which is the number of occurrence for the term b on the document a. The term $f'(a,b)$ is introduced to represent the term frequency of the term b in the title of document a. The denominator of Equation 4.2 is a normalization factor, which is the Euclidean vector length of the document. The parameter β associated with the function $f'()$ determines the degree of importance for the title term weight. The underlying principle for this design is to emphasize those terms which take part in the document title. Through this formulation, all of the term weights for a document will be determined.

The topic feature vectors based on the three types of topic profile specification can be constructed using the same formulation but different extraction models. According to the example page given by the user, all of the words are used as features, with the HTML tags and the programming scripts removed.

Those terms will be checked for existence against the title of the example page. If an index page is given, all of the documents specified in the index page will be fetched. Words found in those fetched Web pages that are highlighted as boldface, italic, underlined, or words enclosed by heading tags are extracted. Their corresponding term weights will be determined by comparing with the titles of all fetched documents. For the keyword profile specification, the keywords are transformed to the terms with the corresponding weights composing the topic feature vector directly.

4.2 Document Evaluation through Relevance Score

With the topic feature vector and the document feature vector defined, the system assigns an estimated relevance score to a document encountered through a similarity function. This similarity function $S(D, Q)$ for a document vector D against the topic profile Q consists of two components, namely the textual content score $S_c(D, Q)$ and the hyperlink score $S_h(D, Q)$. The textual content score, based on the Jaccard's similarity function, is computed as follows:

$$S_c(D, Q) = \frac{\sum_{j=1}^{M} w(Q, q_j) w(D, q_j)}{\sum_{j=1}^{M} w^2(Q, q_j) + \sum_{j=1}^{M} w^2(D, q_j) - \sum_{j=1}^{M} w(Q, q_j) w(D, q_j)} \tag{4.3}$$

The function $w(X, y)$ is the combined term weight for the term y on a document X, as defined in Section 4.2.

Based on the information architecture of the WWW, we observe that the outgoing hyperlinks from a Web document provide another dimension to determine the similarity other than the ordinary text content. This scenario is depicted by Figure 4.1. If two documents A and B are found to be containing some identical outgoing hyperlinks, it is very likely that they are dealing with the same topic. This concept can be projected to our topic profile specifications and the target document being evaluated. To handle this situation, the hyperlink score, $S_h(D, Q)$, is introduced. The score is a ratio between the number of common hyperlinks found in the topic profile specification and the document being explored. The interpretation assumes that the more common hyperlinks they contain, the more similar they will be. It is computed as follows:

$$S_h(D, Q) = \frac{|D_h \cap Q_h|}{|D_h \cup Q_h|} \tag{4.4}$$

From Equation 4.4, the term D_h represents the set of outgoing hyperlinks on the document D. For the example page topic profile specification, Q_h will be the set of hyperlinks appeared on the example page. If an index page is given, the set of hyperlinks on those subordinate pages under the index page will form the set Q_h. The keyword form of topic profile specification will leave the Q_h as an empty set.

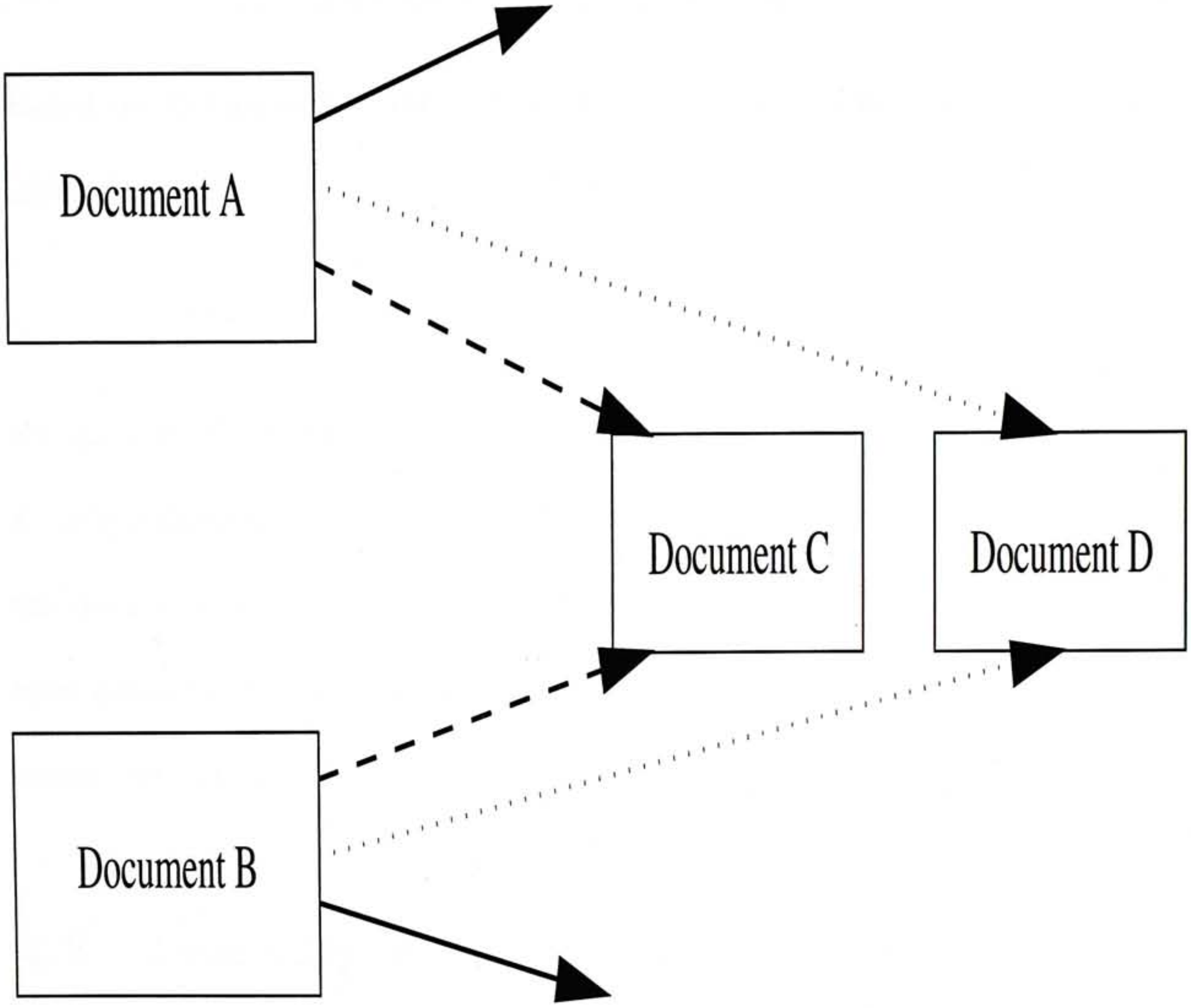


Figure 4.1: Two documents, A and B, containing some identical hyperlinks

With the above textual content score and the hyperlink score defined, the similarity function $S(D,Q)$ will be the average of the two components and is denoted as:

$$S(D,Q) = 0.5S_c(D,Q) + 0.5S_h(D,Q) \tag{4.5}$$

Based on Equation 4.5, the content-based relevance score for a document D against the topic feature vector Q, $R(D,Q)$, is formulated as:

$$R(D,Q) = (1-\alpha)S(D,Q) + \alpha\frac{\sum_{i=1}^{n} S(L_i,Q)}{n}, 0 \leq \alpha \leq 1 \tag{4.6}$$

We include the weighted average similarity functions for the child document L_i of the document D via direct hyperlinks. The rationale behind is to allow the document D, which has direct access to some relevant child documents, to have bonus on the relevance score. The parameter α is a weight for setting the importance of the child documents' average similarity score.

4.3 Learning via Relevance Feedback

Our discovery system will present a set of discovered documents for the user after a certain time interval. The result set may be too broad or contain some irrelevant resources. At that moment, the user can optionally communicate with the system by providing relevance feedback about the documents. As mentioned in Section 4.3.1, the topic feature vectors will be refined in accordance with the relevance feedback. Terms or attributes found in documents which are specified as relevant will be used to expand the topic feature vector, while those in irrelevant documents will be used to diminish the topic feature vector. The term weights in the topic feature vector V_Q as well as the hyperlinks in the set Q_h will be modified. By using this newly refined topic feature vector and the hyperlink set, a more satisfactory filtering result can be obtained.

4.3.1 Introduction to Relevance Feedback

Some IF systems make use of machine learning techniques to adapt to a model of the user's interest and using this model to find relevant documents [13]. Lewis *et. al.* introduced the Widrow-Hoff and EG algorithms as the machine learning algorithms for text retrieval, routing, categorization systems [43]. For the sake of handling the dynamic environment of IF where users are interested in more than one topic and have changing interest, the usage of the genetic algorithms (GA) [32][33][49] were investigated. IF (IR) systems incorporating the GAs were developed in the recent decade [2][14][15][67][68].

Relevance feedback may be treated as a form of machine learning or adaptation. After a set of documents is retrieved, they will be brought to the users for the evaluations. The users may evaluate the retrieval results against the information interest, or their motivating goals or even their new interest aroused by the discovery results. These evaluation processes may lead to the modification of the original queries. These refer to the relevance feedback procedures. The most well-known feedback mechanism is the standard Rocchio model [62]. The basic concept is to "move" a given query (profile) towards the relevant documents and away from the irrelevant ones by formulating a better profile. Other mechanisms are introduced in [28]. Salton and Buckley [65] investigated the relevance feedback methods which were shown to be effective in improving the retrieval performance. Kroon, *et al.* [39] proposed the "Auto-class", an algorithm for automatic classification, to reduce the lengthy time for profile learning. Through the user feedback mechanisms, the IF systems can learn more precisely what the users want. A more satisfactory filtering result, with an improved precision percentage, may be obtained.

4.3.2 Feature Extraction from the Relevance Feedback Models

Our relevance feedback model requires a set of positive feedback with a minimal negative feedback. As the vector processing model ranks the result in a descending order of the relevance scores, it is convenient to choose the topmost irrelevant documents for the feedback process. Such design facilitates three means of relevance feedback models which are also illustrated in Figure 4.2:

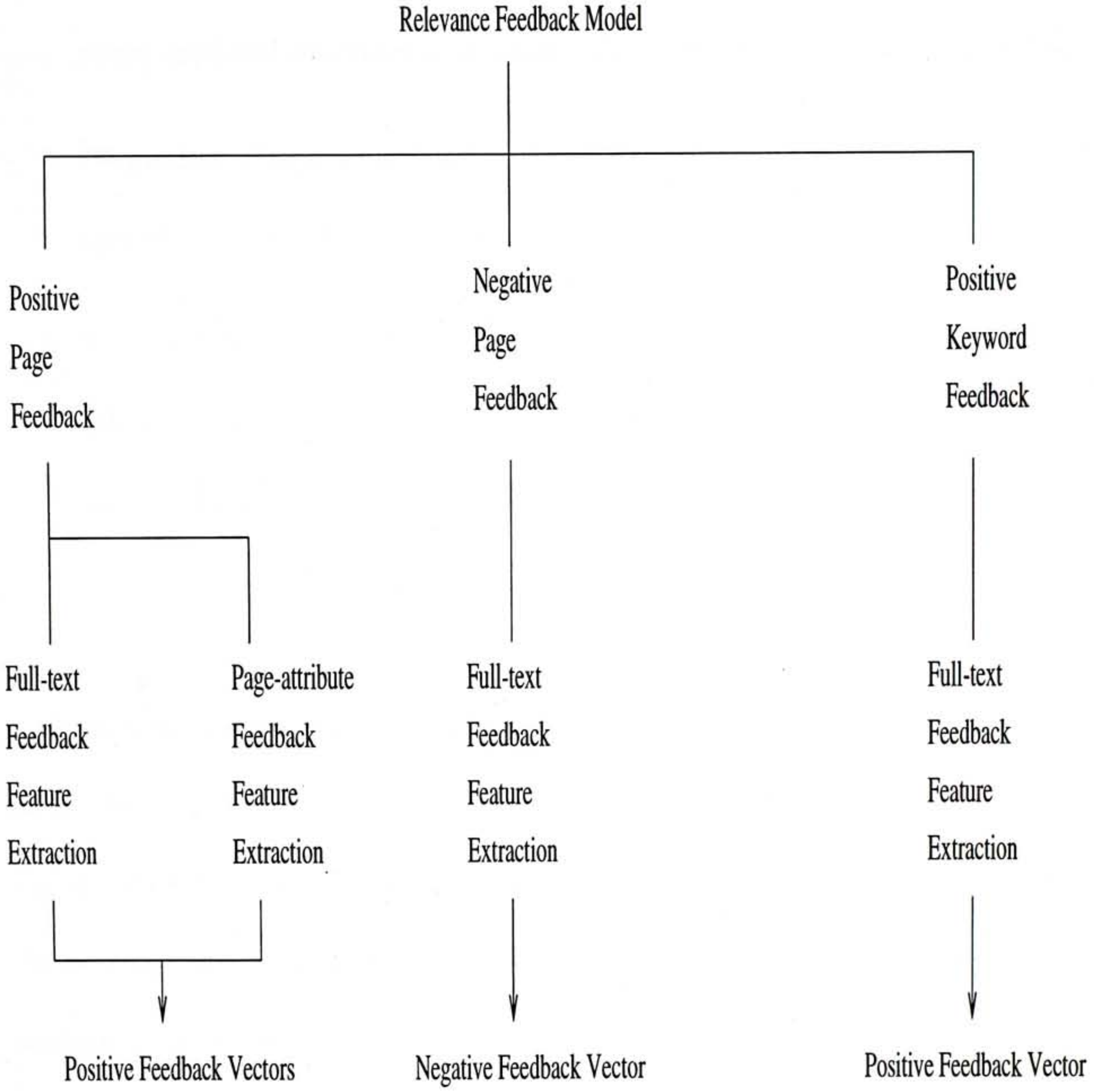

Relevance Feedback Model
Positive Page Feedback
Negative Page Feedback
Positive Keyword Feedback
Full-text Feedback Feature Extraction
Page-attribute Feedback Feature Extraction
Full-text Feedback Feature Extraction
Full-text Feedback Feature Extraction
Positive Feedback Vectors
Negative Feedback Vector
Positive Feedback Vector

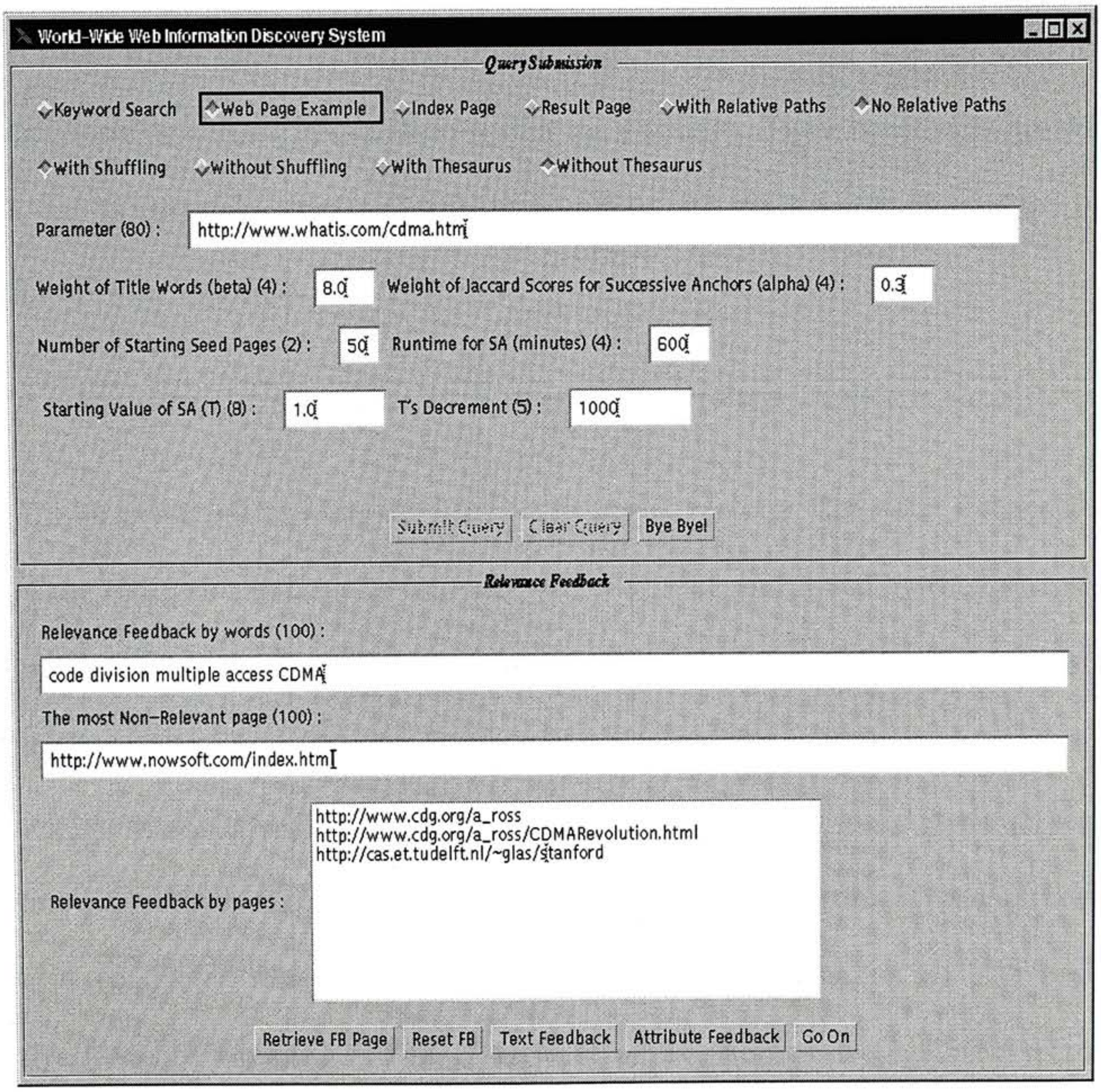
Figure 4.2: Our relevance feedback model

1. **Positive Page Feedback.** This feedback model allows the user to specify some explored documents or some other example documents as relevant.

2. **Negative Page Feedback.** This feedback model allows the user to specify the irrelevant documents.

3. **Positive Keyword Feedback.** This is the most elementary way which allows the users to suggest some keywords that are believed to be important and relevant.

Figure 4.3 depicts an example showing a user specifying the relevance feedback, through the three relevance feedback models. In order to refine the topic feature vector, features will have to be extracted from those models forming the positive and negative feedback vectors. As our system allows multiple documents to be submitted for the positive page feedback, terms will be extracted from these documents to form positive feedback vectors which will further be taken to refine the topic feature vector. There are two feature extraction strategies proposed, namely the full text feature extraction and the page-attribute feature extraction. As reflected by its name, the full text feature extraction will make use of all the words found on those positive documents and transforms to



○ Keyword Search ● Web Page Example ○ Index Page ○ Result Page ○ With Relative Paths ● No Relative Paths

● With Shuffling ○ Without Shuffling ○ With Thesaurus ● Without Thesaurus

Parameter (80) : http://www.whatis.com/cdma.htm

Weight of Title Words (beta) (4) : 8.0 Weight of Jaccard Scores for Successive Anchors (alpha) (4) : 0.3

Number of Starting Seed Pages (2) : 50 Runtime for SA (minutes) (4) : 600

Starting Value of SA (T) (8) : 1.0 T's Decrement (5) : 1000

Relevance Feedback by words (100) :

code division multiple access CDMA

The most Non-Relevant page (100) :

http://www.nowsoft.com/index.html

Relevance Feedback by pages :

http://www.cdg.org/a_ross
http://www.cdg.org/a_ross/CDMARevolution.html
http://cas.et.tudelft.nl/~glas/stanford

Figure 4.3: A user is submitting the relevance feedback.

the positive feedback vectors. This is the most elementary way to handle the positive feedback, without omitting any features.

In contrast, we find that the Web documents are composed by structured HTML, which contains various HTML tags. Usually, those tags are used for emphasizing certain information within the passages. These constructs provide a way facilitating relevance feedback handling. All those words highlighted by the HTML tags such as:

$$< b >, < i >, < h1 > \ldots < h6 >, < em >, < strong >, < u >, < big >$$

will be extracted to form the positive feedback vectors. The term frequencies of the terms in the positive feedback vector will be multiplied by the degree of importance for various HTML tags as shown in Table 3.3. The term weights of the original topic feature vector will be adjusted based on these specially created positive feedback vectors. This is the idea of the page-attribute feature extraction. Experiments on these two feature extraction strategies have been conducted and their results will be presented in Chapter 6. The positive feedback keyword will be taken to produce the positive feedback vector directly. Similarly, full-text feature extraction will be used to generate the negative feedback vector from the negative page feedback model. At the same time, the words found in those feedback vectors have to be stemmed, with the stop-words removed.

4.3.3 Topic Feature Vectors Refinement

As mentioned above, the positive feedback vectors together with the negative feedback vector will be generated. The weights of the terms in these two kinds of vectors are calculated by Equation 4.2. The terms extracted from various models are associated with different weights. The topic feature vector V_Q and the set of hyperlinks originated from the topic profile specification represented by Q_h will be updated. Mathematically, our feedback processing model is depicted by,

$$\left\{ \begin{array}{ccc} V_Q^{j+1} & = & V_Q^j + \sum_{i=1}^{n} Y_i - N, \\ Q_h^{j+1} & = & (Q_h^j \cup (Y_i)_h) - N_h, \\ \text{for } i = 1, 2, \ldots, n & , & j = 1, 2, \ldots \end{array} \right.$$

where the vector V_Q^j depicts the topic feature vector at filtering iteration j. The positive feedback vector Y_i represents the relevant document i. The negative feedback vector N represents the most irrelevant document. Similarly, the term Q_h^j is the hyperlink set from the topic profile at filtering iteration j, whereas $(Y_i)_h$ is the hyperlink set from relevant document Y_i. The hyperlink sets of the topmost irrelevant document N_h is excluded from the set Q_h^{j+1}. Furthermore, the titles of the positive feedback pages will augment the title of the topic profile specification, whereas the negative page title will be used to adjust the title in an opposite manner. For those terms whose weights are worth negative are removed in V_Q^{j+1}.

In addition, those terms found in positive feedback and the negative feedback vector have another purpose. Terms extracted from the positive feedback page and the positive feedback keywords will be used to generate some starting documents. These positive terms may introduce a new information space for exploration of more related resources. They provide a known, positive direction of exploration. On the other hand, the negative feedback page are taken to stop any further exploration. The Intelligent Web Exploration deals with this task and it will be discussed in Chapter 5.

Chapter 5

Intelligent Web Exploration

This chapter presents the technique for exploring the WWW information network according to the relevance scores calculated. A simulated annealing algorithm is proposed so as to accomplish the two aforesaid discovery objectives: exploration and exploitation. The relevance feedback combined with the discovery algorithm is discussed.

5.1 Introduction to Simulated Annealing

Simulated Annealing (SA) was introduced in the early 1980s by Kirkpatrick, Gelatt and Vecchi [36]. It is a general purpose function optimization algorithm based on an analogy between the simulation of the annealing of solids and the problem of solving combinatorial optimization problem [41]. The SA approach has been studied intensively in the fields of mathematics, statistics, physics, and many other areas. It has been applied to many problems in artificial intelligence and machine learning. Wherever SA is applied, the main purpose is to avoid the problem of trapping in a local optima. This problem is the most important one encountered in conventional hill climbing search [58]. Hill climbing is a search technique that exploits the current configurations. Being a greedy algorithm, it is very likely to be trapped in a local optima. Random search is another extreme which performs exhaustive search, without considering the best solution found so far. SA is an algorithm in between these two extremes. Generally speaking, SA is a kind of heuristic search strategy as well as a variation of hill climbing process [17]. At the beginning of the process, some downhill movements are allowed. Movements to worse states are accepted so as to avoid local optima and to cover a wider area of the search space. As the algorithm progresses, the frequency for the downhill movement is gradually reduced. These properties are suitable for the WWW information discovery. An example of the SA algorithm, which is also adopted in this research, is described as follows:

1. **Initial configuration:**

   A finite solution space S is initialized and a cost function C is assigned. The function C is a real number to each solution. The algorithm is initialized with a high value for the control parameter T, usually known as the temperature, and with an initial solution. According to the selection between a current solution and an alternative solution based on some heuristics, a new sequence of solutions will be generated.

2. **Cooling and generating new configurations:**

   The value of the temperature T is gradually decreased as the algorithm progresses according to a cooling schedule. At each value of the temperature, a sequence of solutions are generated. The cost for each such solution is computed. Let ΔE be defined as

   $$\Delta E = C_{\text{current}} - C_{\text{alternative}} \tag{5.1}$$

   where C_{current} and $C_{\text{alternative}}$ stand for the current solution and an alternative solution respectively. The ΔE is usually known as the energy level. The alternative solution is accepted and used to generate a sequence of solutions according to the distribution following probability P_a:

   $$P_a = Min(1, \exp(-\frac{\Delta E}{T})) \tag{5.2}$$

   The current solutions are always accepted while the alternative transitions might or might not be, depending on the energy level ΔE and the temperature T.

3. **Termination:**

The algorithm terminates at some small value of the temperature T. The current solution is then taken as the solution to the problem at hand.

To implement the SA algorithm, it is necessary to select an annealing schedule which consists of three components. The first component is the initial value to be used for temperature T. The second one is the criteria for deciding when the temperature of the system should be reduced. The third component is the amount by which the temperature will be reduced each time it is changed. If the cooling occurs too rapidly (a steep decrease of the temperature curve), a local but not global optimum is more likely obtained. However, if a slower schedule is used, a global optimum is more likely obtained.

5.2 Intelligent Web Exploration by Simulated Annealing

We make use of simulated annealing which is an advanced search technique to develop the intelligent exploration strategy. Simulated annealing possesses a desirable property of finding an optimal solution statistically [21][30][35]. It is adaptive and capable of finding high quality solutions. All these features make it an attractive method for the automatic traversal in the huge, heterogeneous WWW information network.

For every Web document we encountered, we compute a relevance score according to the mathematical formulations introduced in Chapter 4. Therefore, the discovery problem becomes an optimization problem with the objective of finding a set of nodes from those subgraphs with high relevance scores. It can be observed that exploring from the set of documents with higher relevance scores may provide a better chance for searching a solution as those pages probably contain more related information as well as outgoing hyperlinks. Using only this strategy reduces to the best-first search. However, the WWW can be viewed as a large set of enormous cyclic subgraphs. The various documents are the nodes of the subgraphs while the hyperlinks are the arcs. Those subgraphs may or may not be interconnected. It is common that two or more documents that are of similar topic residing in two different WWW subgraphs. There may not be any connection in between the subgraphs as shown in Figure 5.1. On the other hand, it may be possible for a less favorable documents to have some outgoing hyperlinks somehow leading to some relevant documents. Therefore, only relying on the best-first strategy is not adequate for discovering relevant documents. This situation corresponds to local optimal solutions. An exploration strategy based on SA is a feasible strategy in tackling this problem.

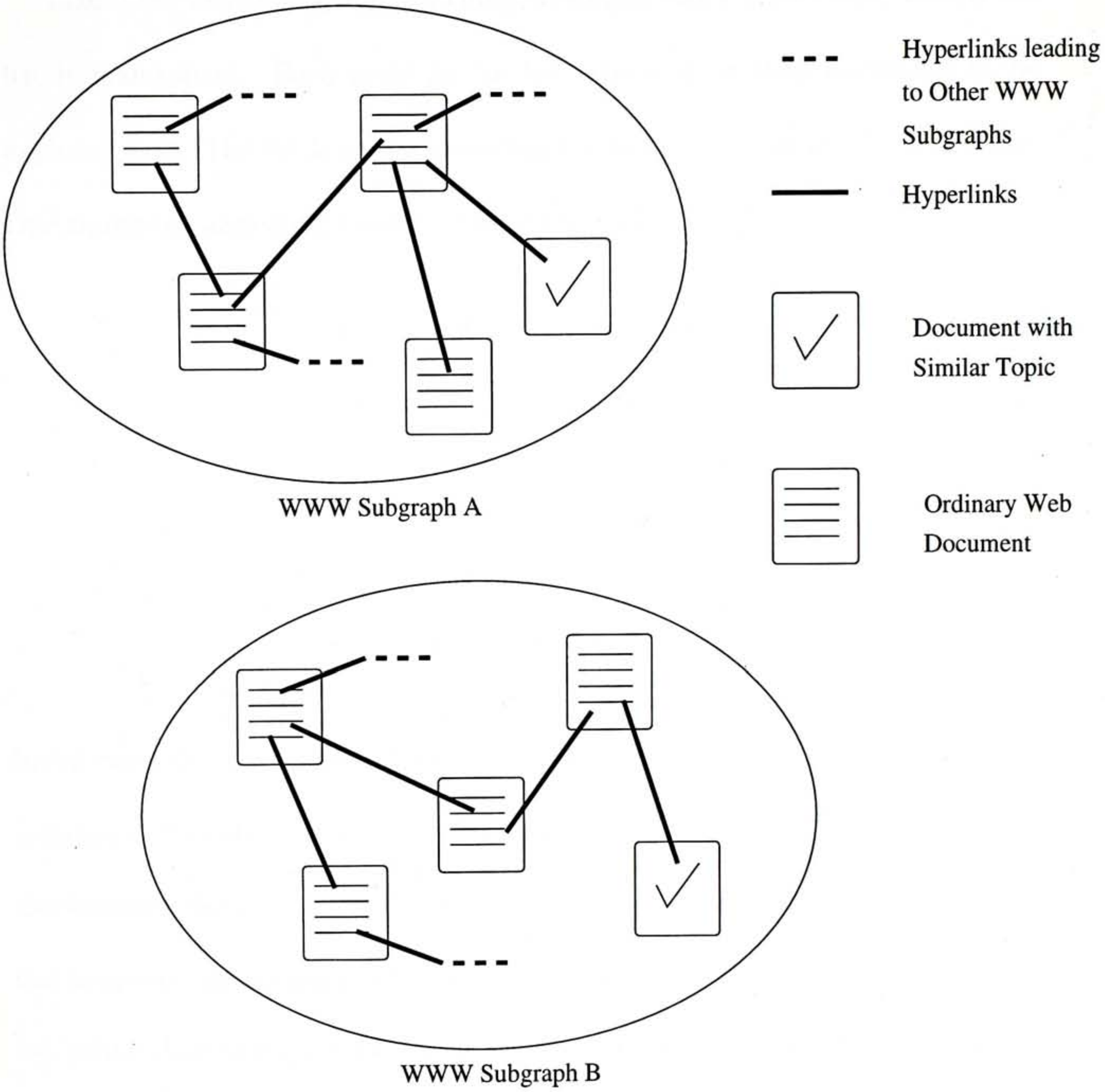

Hyperlinks leading to Other WWW Subgraphs
Hyperlinks
Document with Similar Topic
Ordinary Web Document
WWW Subgraph A
WWW Subgraph B

Figure 5.1: Documents with similar topic residing in two different WWW subgraphs.

5.2.1 Mathematical Setting of the Discovery Process

Like other heuristic graph searching strategies in AI, a list called the OPEN list is maintained. Each node in the list represents a Web document to be explored next. The list is sorted according to the corresponding relevance score. Our simulated annealing model is formulated as:

$$\begin{aligned} \Delta E &= R(P,Q) - R(M,Q), && (5.3) \\ T_k &= T_0/\log(\phi k), && (5.4) \\ p\prime &= min(1, \exp(-\frac{\Delta E}{T_k})), && (5.5) \\ & \text{for k=1,2,} \ldots, \\ & \phi \geq 0 \end{aligned}$$

In our case, the cost function is the relevance score evaluation function $R()$. The document P is the first document ranked in the OPEN list, which possesses the highest relevance score. The current solution is set to this document. As the simulated annealing model allows some explorations, a page in the OPEN list other than the topmost ranked document, depicted by M, is chosen for evaluation. It is the alternative solution mentioned in the above example. The term ΔE is the change of energy level. T_k is the temperature at time k. The symbol ϕ is the temperature decrement step size. The product of ϕ and k is controlled so that it will not be 0 or 1. Our annealing schedule is expressed in

Equation 5.4. The drop in the temperature will have direct influence on the probability $p\prime$ for starting the exploration from a random document.

5.2.2 The Entire Exploration Algorithm

With the setting introduced in Section 5.2.1, we can present the entire exploration process incorporated with the simulated annealing algorithm. Here are the steps of the algorithm:

```
1. Create the topic feature vector from the various topic profile
   specifications.  Submit to several search engines to obtain a
   set of Web document $\mathcal{P}$.

2. Form the OPEN list with each node representing the
   documents in the set $\mathcal{P}$, with the relevance
   scores sorted.

3. Determine the $\Delta E$.

4. Compute the probability $p\prime$ and generate a random
   real number ranged from 0 to 1.

5. If the random number $\leq p\prime$,

     • explore from the document $M$.
     • remove the node representing document $M$ from the OPEN list.
     • expand the OPEN list with the relevance scores of
       subordinate documents referenced by $M$.

   else,

     • explore from the document $P$.
     • remove the node representing document $P$ from the OPEN list.
```

```
        • expand the OPEN list with the relevance scores of
          subordinate documents referenced by $P$.

  6. Revise $T_k$, repeat step 3 until filtering deadline expired.

  7. Return the document discovered.
```

The probability p' is interpreted as the probability for exploring from the random page selected. At the beginning of the discovery process, the probability p' gets a larger value. Exploring from the random page will be highly probable. This scenario is equivalent to exploration. As p' decreases and approaches zero, it will be less probable to explore from the random page but will be more concentrate on exploring from the "best-so-far" page. In this case, exploitation with some interleaving exploration takes place. The "deadline" is the time when the user-specified discovery duration expires and at which a set of discovered documents is suggested to the user. By using the simulated annealing techniques proposed and an appropriate annealing schedule, a balance between the two information discovery objectives, exploitation and exploration, can be maintained.

5.3 Incorporating with the Relevance Feedback Model

It has been mentioned in Section 4.3.3 that the positive terms extracted from the two positive feedback specifications will be used to generate new exploration origins. Technically speaking, this can be accomplished by submitting some of those terms to some searchable indexes in order to obtain several starting points leading to new Web subgraphs. As the OPEN list is keeping track of the documents to be explored in future, the hyperlinks returned from the searchable indexes will be inserted into the OPEN list. As a result, those origins will get some chances of being explored in future. Related documents against the refined topic feature vector may be obtained. Likewise, the negative page feedback specification will stop any further exploration from the undesirable documents. The hostname part [1] of the negative feedback document is extracted. The hostname is then verified with the OPEN list. The documents in the OPEN list with the hostname being equal to the extracted hostname, or any document resides in the Web site specified by the extracted hostname, will be removed from the OPEN list. This will prevent further traversal from these unpromising origins.

[1] Hostname is usually the first component of an URL, delimited by the single slash, e.g. `www.se.cuhk.edu.hk` is the hostname for the document `http://www.se.cuhk.edu.hk/aboutseem.html`.

Chapter 6

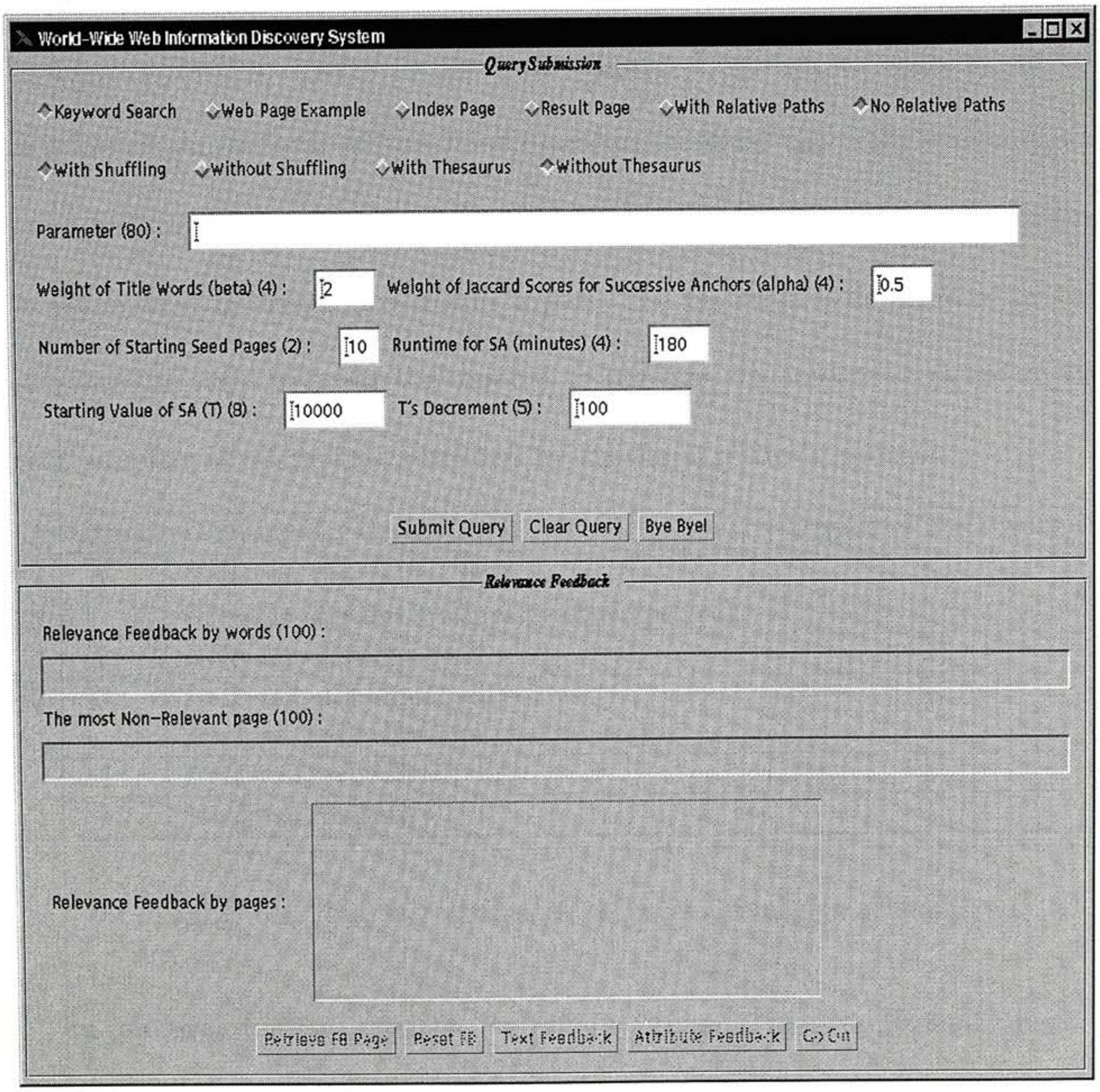
Experimental Results

Several sets of experiments have been conducted to evaluate our discovery model. This chapter presents the design of the experiments as well as the results obtained. Analysis of the results is also conducted.

6.1 The Design of the Experiments

A prototype of our information discovery system has been implemented. The graphical user interface of our system prototype is shown in Figure 6.1.

Currently, the system is able to handle English text. Certain mechanisms guarantee non-repetitive visit for each document. Hyperlinks leading to common Web sites or documents with common gateway interface scripts have been re-



◆ Keyword Search ◇ Web Page Example ◇ Index Page ◇ Result Page ◇ With Relative Paths ◆ No Relative Paths

◆ With Shuffling ◇ Without Shuffling ◇ With Thesaurus ◆ Without Thesaurus

Parameter (80) :

Weight of Title Words (beta) (4) : 2 Weight of Jaccard Scores for Successive Anchors (alpha) (4) : 0.5

Number of Starting Seed Pages (2) : 10 Runtime for SA (minutes) (4) : 180

Starting Value of SA (T) (8) : 10000 T's Decrement (5) : 100

Relevance Feedback by words (100) :

The most Non-Relevant page (100) :

Relevance Feedback by pages :

Figure 6.1: The graphical user interface of the discovery system prototype.

moved. Outdated or unreachable Web documents are ignored by assigning zero relevance scores. Several sets of experiments were set up to evaluate and verify the performance of our system in a practical environment. We use a metric based on precision to measure the performance. Precision is a common metric in information retrieval (information filtering) systems. Generally speaking, the users will judge the documents in the retrieval or discovery results according to the current information needs or topics. The precision is determined by calculating the ratio between the number of relevant documents retrieved to the total number of documents retrieved by the same retrieval process.

Rating	Document Content	Degree of Relevance
1	Relevant document title & relevant content with relevant hyperlinks	1
2	Relevant content only	$\frac{3}{4}$
3	Relevant document title with relevant hyperlinks or similar topics	$\frac{1}{2}$
4	Relevant hyperlinks only	$\frac{1}{4}$
5	Nothing relevant	0

Table 6.1: The five ratings in the user evaluation scheme

In our experimental settings, we design a scheme to evaluate how relevant a document is against the information need captured in the topic profile. The main characteristic of our scheme is the incorporation of the relevance of hyperlinks. The scheme includes 5 ratings of documents with various degree of relevance. We have studied several different sets of the degree of relevance for the 5 ratings and found that those sets of degree of importance have no effects on our document evaluation. We have chosen a typical scheme as depicted in Table 6.1. The first rating represents the most favorable documents. Documents of this rating possess relevant content, relevant title as well as outgoing hyperlinks leading to other relevant pages. The second rating includes documents relevant to the desired topic but without relevant hyperlinks connecting to other relevant documents. The third rating covers a more diverse range of documents. It includes the so-called "pointer-document" which contains a relevant title with only some of the relevant outgoing links only. Besides, it also includes documents having a similar topic with the topic profile. The documents ranked as the forth rating contain only relevant hyperlinks. Typical examples are various personal or organizational homepages with some hyperlinks leading to a relevant Web location, but its own content is not relevant to the topic profile. Those totally irrelevant documents will fall into the fifth rating with degree of relevance of zero. For each discovery output session, the first $\mathcal{U}$ documents were examined manually and their degree of relevance were determined based on the evaluation scheme. The precision for a discovery process will be computed by the following formula:

$$\text{precision} = \frac{\text{Total Degree of Relevance}}{\mathcal{U}} \tag{6.1}$$

In our experiments, the denominator $\mathcal{U}$ of Equation 6.1 is set to 50. The parameter β in Equation 4.2 (i.e., the degree of importance for the title term frequency) is set to 8. Since the document title usually provides descriptive and direct information on what the document is about, the degree of importance for the title terms are made superior to those tags defined in Table 3.3. The parameter α in Equation 4.6 (i.e., the weight defined for setting the importance of the child documents' average similarity score) is set to 0.3 so as to allow the parent document to have more contribution in the relevance score. Experiments were conducted to assess the discovery system's ability to fulfill and adapt to different information needs. The details of the experiments and results are shown in the following sections.

6.2 Experiments on the Effects of the Simulated Annealing Schedule upon the Discovery Precision.

The purpose of this experiment is to demonstrate the effect of the simulated annealing schedules upon the discovery precision. It also compares the precision of the discovery processes using the example page topic profile specification as well as the keyword topic profile specification.

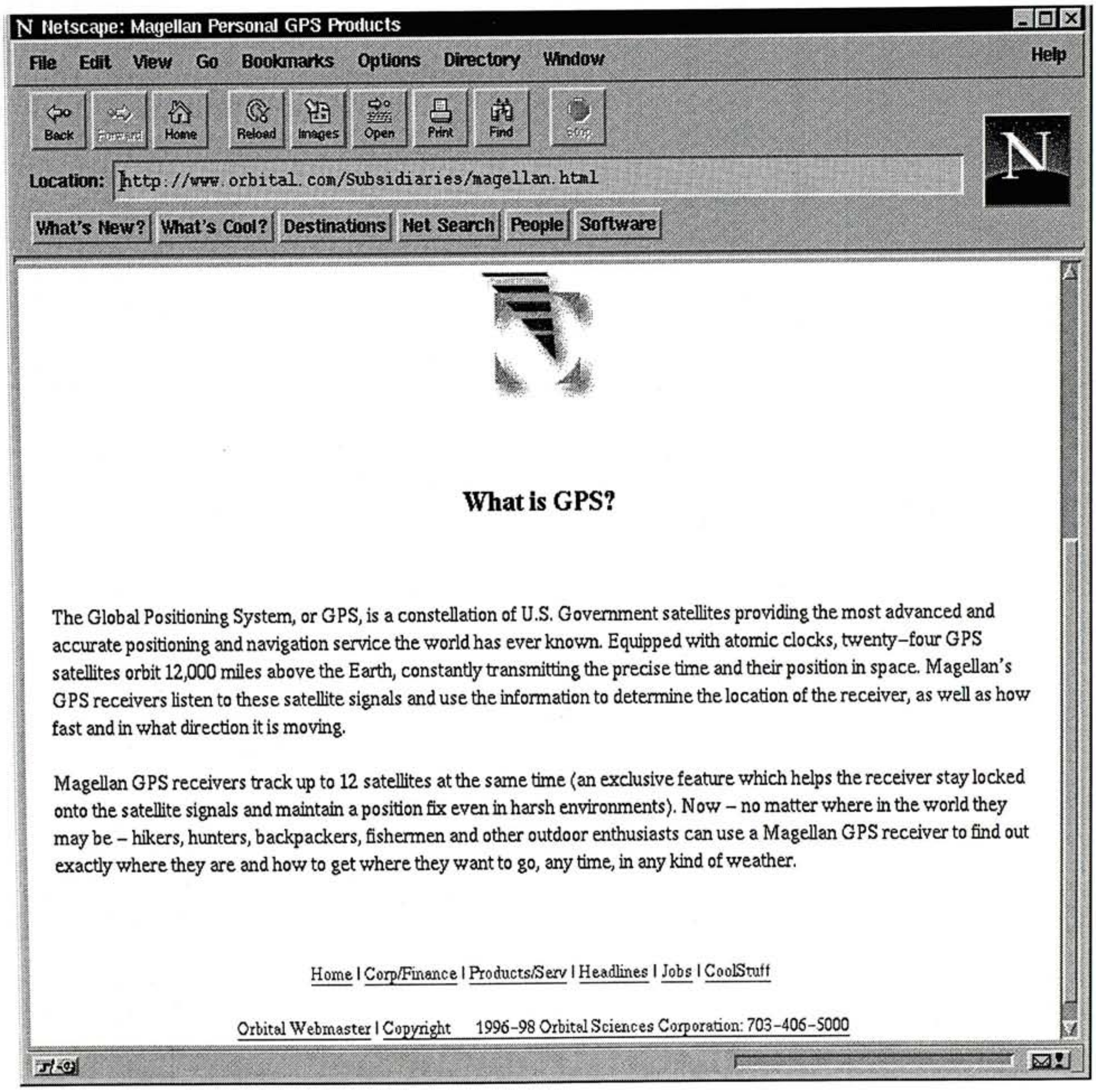
6.2.1 Experiment Setup

There were 11 randomly selected topic profiles chosen for this experiment. For each topic profile, the example page profile specification and the keyword profile specification were submitted. Figure 6.2 shows the example page topic profile specification for the Topic 4 (i.e., Global Positioning System) in the experiments. Two types of annealing schedules, namely the schedule A and the schedule B upon those profile specifications, were under investigation. The probabilities $p\prime$ in the SA processes with the schedule A were controlled to have an average around 0.9. The implication for the schedule A is to simulate the scenario of exploring from mostly random pages. In the schedule B, the average $p\prime$ is set to around 0.5. This provides both of the "best-so-far" pages and the random pages with equal opportunities for exploration.

6.2.2 Results

Table 6.2 and 6.3 show the precision values of the discovery processes upon two schedule types by the example page topic profile specification and keyword topic profile specification respectively.

For the two different types of annealing schedules, the discovery processes with the schedule B are generally providing higher precision values compared



http://www.orbital.com/Subsidiaries/magellan.html

What is GPS?

The Global Positioning System, or GPS, is a constellation of U.S. Government satellites providing the most advanced and accurate positioning and navigation service the world has ever known. Equipped with atomic clocks, twenty–four GPS satellites orbit 12,000 miles above the Earth, constantly transmitting the precise time and their position in space. Magellan's GPS receivers listen to these satellite signals and use the information to determine the location of the receiver, as well as how fast and in what direction it is moving.

Magellan GPS receivers track up to 12 satellites at the same time (an exclusive feature which helps the receiver stay locked onto the satellite signals and maintain a position fix even in harsh environments). Now – no matter where in the world they may be – hikers, hunters, backpackers, fishermen and other outdoor enthusiasts can use a Magellan GPS receiver to find out exactly where they are and how to get where they want to go, any time, in any kind of weather.

Home | Corp/Finance | Products/Serv | Headlines | Jobs | CoolStuff

Orbital Webmaster | Copyright 1996-98 Orbital Sciences Corporation: 703-406-5000

Figure 6.2: The example page topic profile specification for the topic "Global Positioning System" in the experiment.

	Search Topics	Schedule A		Schedule B	
	(Example page)	Precision	Average p'	Precision	Average p'
1.	Bicycle	0.9	0.988	0.96	0.502
2.	Computer Vision	0.175	0.991	0.75	0.51
3.	Geographic Information System	0.49	0.994	0.72	0.738
4.	Global Positioning System	0.84	0.883	0.88	0.486
5.	Global System for Mobile Communication(b)	0.85	0.684	0.585	0.829
6.	Honda Civic	0.795	0.996	1.00	0.59
7.	Honda Motorcycle	0.915	0.985	0.92	0.583
8.	Moon Apollo	0.455	0.989	0.76	0.647
9.	Palmpilot	0.8	0.776	0.96	0.493
10.	Titanic	0.99	0.967	0.92	0.628
11.	Warcraft	0.68	0.964	0.89	0.664
	Average	**0.717**	**0.9288**	**0.85**	**0.606364**

Table 6.2: The precision values of the discovery processes by the example page topic profile specification upon the two schedule types.

	Search Topics	Schedule A		Schedule B	
	(Keyword)	Precision	Average p'	Precision	Average p'
1.	Bicycle	0.83	0.992	0.705	0.451
2.	Computer Vision	0.48	0.922	0.78	0.348
3.	Geographic Information System	0.32	0.938	0.83	0.453
4.	Global Positioning System	0.81	0.863	0.76	0.677
5.	Global System for Mobile Communication(b)	0.29	0.943	0.76	0.53
6.	Honda Civic	0.84	0.841	0.9	0.424
7.	Honda Motorcycle	0.66	0.989	0.88	0.481
8.	Moon Apollo	0.79	0.852	0.78	0.521
9.	Palmpilot	0.835	0.859	0.7	0.612
10.	Titanic	0.98	0.743	0.985	0.429
11.	Warcraft	0.615	0.775	0.96	0.328
	Average	**0.67727**	**0.8834**	**0.822**	**0.4776**

Table 6.3: The precision values of the discovery processes by the keyword topic profile specification upon the two schedule types.

	Search Topics (Example Page)	Schedule A		Schedule B	
		# of discovered page	# of fetched page	# of discovered page	# of fetched page
1.	Bicycle	723	4229	294	1010
2.	Computer Vision	269	3556	134	1223
3.	Geographic Information System	494	4512	140	1432
4.	Global Positioning System	355	5117	284	2381
5.	Global System for Mobile Communication(b)	170	1424	481	2103
6.	Honda Civic	102	1702	452	2364
7.	Honda Motorcycle	581	6115	214	809
8.	Moon Apollo	94	961	208	6840
9.	Palmpilot	116	3000	312	3245
10.	Titanic	402	10044	231	1040
11.	Warcraft	132	2036	289	1155
	Average	**312.55**	**3881.45**	**276.27**	**2145.64**

Table 6.4: The number of discovered page and the number of fetched pages of the discovery processes by the example page topic profile specification upon the two schedule types.

	Search Topics (Keyword)	Schedule A		Schedule B	
		# of discovered page	# of fetched page	# of discovered page	# of fetched page
1.	Bicycle	138	807	242	1348
2.	Computer Vision	95	611	243	1924
3.	Geographic Information System	88	947	182	1844
4.	Global Positioning System	60	1236	154	2449
5.	Global System for Mobile Communication(b)	75	2933	305	3219
6.	Honda Civic	66	855	354	1713
7.	Honda Motorcycle	461	2387	200	1459
8.	Moon Apollo	236	3915	139	1046
9.	Palmpilot	351	1753	453	2193
10.	Titanic	455	5156	421	2116
11.	Warcraft	155	1228	371	1412
	Average	**198.18**	**1984.36**	**278.55**	**1883.91**

Table 6.5: The number of discovered page and the number of fetched pages of the discovery processes by the keyword topic profile specification upon the two schedule types.

with the schedule A, for all the topic profile specifications used. This is mainly due to the simulated annealing properties described in the previous chapter. Schedule A is actually simulating the random search algorithm. The high value of the average probability $p\prime$ promotes an emphasized exploration from randomly selected pages. In this case, documents with higher rank will have less chances for further exploration. Therefore, the average precision of discovery processes using Schedule A is lower than those documents under the simulated annealing schedule B.

It can also be observed that by taking the same schedule type, the average precision of the discovery processes initiated by the example page topic profile specification is higher than the ones by the keyword-based profile specification. The most probable reason for this observation is that all of the text found on the example page is used as features. More descriptive, detailed features can be obtained from example pages. When a document is fetched, a more meaningful and accurate relevance score can be calculated.

Table 6.4 and Table 6.5 show the total number of documents discovered together with the total number of documents fetched by the discovery processes of both schedules using the two topic profile specifications. An identical discovery time duration was taken. These figures are highly dependent on the network loading on the WWW.

As the annealing schedule type B can achieve the properties of exploitation and exploration of the Web documents, this type of schedule is adopted for conducting the following sets of experiment.

6.3 Experiments on the Index Page Topic Profile Specification

The purpose of the experiment is to further demonstrate how the index page topic profile specification can fulfill the information need of the user. The user is not required to formulate the keyword query, but just feed our discovery system with those index pages.

6.3.1 Experiment Setup

There were 11 selected topic profiles for evaluation. Those index pages were arbitrarily chosen on the WWW or they were manually constructed by the user him(her)self. The simulated annealing schedule B described in Section 6.2 was adopted for the discovery process, which keeps a fair exploration opportunities between the "best-so-far" pages and the random pages.

	Search Topics	Index Page Type	Average Precision	Average p'	# of discovered page	# of fetched page
1.	Bike	S	0.945	0.509	1147	3197
2.	DVD	S	0.94	0.673	200	1312
3.	Genetic Algorithm	A	0.83	0.625	668	4830
4.	Intelligent Agent	A	0.685	0.738	319	3042
5.	Motorcycle	S	0.815	0.56	171	1029
6.	Palmpilot	S	0.775	0.471	165	912
7.	Photography(a)	S	0.66	0.779	245	1592
8.	Photography(b)	A	0.705	0.734	504	2480
9.	Photography(c)	A	0.8	0.555	352	3048
10.	Titanic	S	0.78	0.499	298	1565
11.	Year 2000	A	0.495	0.464	631	4226
	Average		**0.766**	**0.601**	**427.273**	**2475.727**
	S-index Average		**0.819**	**0.698**	**445.2**	**1921.4**
	A-index Average		**0.703**	**0.623**	**494.8**	**3525.2**

Table 6.6: The experiment results of the discovery processes using the index page topic profile specification.

6.3.2 Results

Table 6.6 shows the precision obtained from our discovery system for serving the 11 topics of various information needs. In the column "Index Page Type", the letter "S" stands for the topic profile represented by the self-constructed index page topic profile specification. The letter "A" stands for an arbitrarily chosen WWW documents taken to be the index page. The values of precision and the average exploration probability p', the total number of discovered pages as well as the total number of pages fetched by our discovery system are shown.

From Table 6.6, it is observed that the average precision value of the discovery processes using the index page topic profile specifications is 0.766. The average $p\prime$ was kept at 0.601. As mentioned in Chapter 3, all of the pages specified in the index page will be fetched. Words found in those fetched Web pages that are highlighted as boldface, italic, underlined, or words enclosed by heading tags are extracted to form the topic feature vector by the feature extraction model. By comparing with the example page topic profile specification which applies the full-text feature extraction, the index page approach may use the storage efficiently as only the emphasized features are stored. The results of index page specification are found to be comparative with the example page topic profile specification. Our system is able to draw the emphasized text on the pages to form the descriptive features. By taking a closer observation in Table 6.6, the average precision for the self-made index page specification is 0.819 while the one for arbitrary-chosen index page specification is 0.703. This difference may be explained by the situation that the arbitrary-chosen index page may have "noisy" hyperlinks. Not all of the outgoing hyperlinks are describing the same topic in the "A type" index page. But for the self-made index page, the hyperlinks are constructed by the users with care.

6.4 Experiments on the Relevance Feedback with Full-Text Feature Extraction Strategy

The purpose of this experiment is to demonstrate the capability of the relevance feedback models of our discovery system. As mentioned before, there are altogether three relevance feedback models, namely the positive page feedback, negative page feedback and positive keyword feedback. For handling the positive page feedback, the full-text feature extraction strategy as well as the page-attribute feature extraction strategy are verified. This experiment focuses on the full-text strategy. The experiment in the next section is concerned with the page-attribute strategy.

6.4.1 Experiment Setup

There were 13 topic profiles submitted to the discovery system. Among various topic profiles, the example page together with the index page topic profile specifications were used for representation. According to the discovery results returned by our discovery system, relevance feedback of the result pages were given. Several relevant documents and some relevant keywords as well as the most irrelevant documents were submitted to the discovery system for the refinement of the topic feature vectors. In handling the positive page and negative page feedbacks, all of the text found on those feedback pages were used for the topic feature vector refinement.

6.4.2 Results

	Search Topic	Initial Precision	Precision after Feedback	Precision Improvement
1.	Computer Game (Quake)	0	0.375	0.375
2.	DVD (factual, technical info.)	0.32	0.8	0.48
3.	IF – > IF, ID, IR	0.165	0.405	0.24
4.	Jeep (pajero – > prado)	0.2	0.13	0.13
5.	GIS	0.25	0.7	0.45
6.	IA – > EC	0.04	0.29	0.25
7.	MCSE	0.27	0.31	0.04
8.	Motorcycle(honda,yamaha)	0.56	0.58	0.02
9.	Palmpilot (software, application)	0.605	0.7	0.095
10.	Photography – > camera	0.15	0.17	0.02
	camera – > digital camera	0.17	0.36	0.19
11.	Simulated annealing	0	0.34	0.34
12.	Subaru	0.22	0.63	0.41
13.	Software Agent	0.215	0.68	0.465
	Average	**0.213**	**0.462**	**0.25**
	Improvement in Percentage			**117%**

Table 6.7: The improvement in precision by the relevance feedback models using the full-text feature extraction strategy.

Table 6.7 and Table 6.8 shows the various search topics and the corresponding precision values. The abbreviations "DVD", "IF", "ID", "IR", "GIS", "IA", "EC", "MCSE" represent digital versatile disc, information filtering, information discovery, information retrieval, geographic information system, intelligent agent, electronic commerce and Microsoft certified system engineer respectively.

	Search Topic	Topic Profile Spec.	Pos. Pages Model	Neg. Page Model	Pos. Keywords Model
1.	Computer Game (Quake)	Example	√	√	√
2.	DVD	S-index	√	√	
	(factual, technical info.)				
3.	IF − > IF, ID, IR	A-index	√	√	√
4.	Jeep (pajero − > prado)	Example	√		√
5.	GIS	A-index	√	√	√
6.	IA − > EC	S-index	√		√
7.	MCSE	Example	√	√	√
8.	Motorcycle(honda,yamaha)	S-index	√	√	√
9.	Palmpilot (software, application)	S-index	√		√
10.	Photography − > camera	S-index	√		√
	camera− >digital camera		√		√
11.	Simulated annealing	Example	√	√	√
12.	Subaru	A-index	√		√
13.	Software Agent	S-index	√	√	

Table 6.8: The distributions of the feedback models submitted for the topic profile refinement using the full-text feature extraction strategy.

The improvements achieved by the relevance feedback models are demonstrated. The "√" sign in Table 6.8 indicates the type of the relevance feedback model the user used. The "− >" sign indicates the change in the information need during document examination. The 10^{th} search topic demonstrated that there were 2 switches of information need during the discovery process. It switched from "photography" to "camera" and from "camera" further to "digital camera".

The result shows that the initial average precision is 0.213. After the submissions of relevance feedback, the average precision becomes 0.462, with an improvement of 0.25 (i.e, 117%). Some of the search topic received a zero for initial precision. This indicates the situation that those discovery processes might be trapped in a gigantic Web location (site) due to the heavy network loading.

Most of the evaluations occurred at the same site and the precision values were affected. Meanwhile, as shown by the results, improvement in precision can be achieved through the relevance feedback models.

6.5 Comparisons of the Relevance Feedback Feature Extraction Strategies

This set of experiment is the extension of the experiment conducted in Section 6.4. The page-attribute strategy will be verified against the full-text approach. The improvements in precision made by the two relevance feedback feature extraction strategies are compared. Besides, the memory storage for both of the strategies are compared.

6.5.1 Experiment Setup

There were 10 topic profiles submitted to our discovery system for evaluations. Like the previous experiment, the example page and the index page topic profile specifications were used for representing the information needs. According to the discovery results, the user was required to express relevance feedbacks. In order to evaluate the two aforesaid relevance feedback feature extraction strategies, the positive page relevance feedback models were mandatory. The precision of the both strategies are compared. As the relevance feedback models refined or expanded the topic feature vector, the memory storage for those feature vectors are compared. The size of the topic feature vector represent the total number of features stored in the vector.

6.5.2 Results

Various comparisons are shown in Table 6.9 and Table 6.10. The abbreviations "TFB" and "AFB" stand for the full-text extraction strategy and the page-attribute extraction strategy respectively. The "TFV" stands for the "topic feature vector".

	Search Topic	Initial Precision	TFB Precision	AFB Precision	TFB Improv.	AFB Improv.
1.	Australia travel tourism	0.155	0.18	0.28	0.025	0.125
2.	Code Division Multiple Access	0.215	0.32	0.355	0.105	0.14
3.	Distributed Database	0.035	0.05	0.065	0.015	0.03
4.	Gold Coast	0.395	0.515	0.58	0.12	0.185
5.	Handheld PC	0.19	0.22	0.25	0.03	0.06
6.	Neural network (a)	0.38	0.45	0.445	0.07	0.065
7.	Neural network (b)	0.255	0.455	0.435	0.2	0.18
8.	Parallel computing	0.525	0.57	0.55	0.045	0.025
9.	Standard Generalized Markup Language	0.32	0.45	0.475	0.13	0.155
10.	Software agent	0.215	0.68	0.71	0.465	0.495
	Average	**0.269**	**0.389**	**0.415**	**0.121**	**0.146**
	Improvement in Percentage				**45%**	**54%**

Table 6.9: The comparison of the precision for the full-text and page-attribute feature extraction strategies on the positive page relevance feedback model.

	Search Topic	Initial TFV size	TFB TFV size	AFB TFV size
1.	Australia travel tourism	33	217	116
2.	Code Division Multiple Access	58	604	114
3.	Distributed Database	254	368	280
4.	Gold Coast	43	243	91
5.	Handheld PC	112	392	207
6.	Neural network (a)	60	575	108
7.	Neural network (b)	347	744	361
8.	Parallel computing	127	812	732
9.	Standard Generalized Markup language	66	919	130
10.	Software agent	140	1118	206
	Average	**124**	**599.2**	**234.5**

Table 6.10: The size comparison of the topic feature vectors refined by the full-text and page-attribute feature extraction strategies upon the positive page relevance feedback model.

	Search Topic	Topic Profile Spec.	Pos. Pages Model	Neg. Page Model	Pos. Keywords Model
1.	Australia travel tourism	A-index	√	√	√
2.	Code division Multiple Access	Example	√	√	
3.	Distributed Database	Example	√		√
4.	Gold Coast	Example	√	√	√
5.	Handheld PC	Example	√		√
6.	Neural network (a)	Example	√		
7.	Neural network (b)	A-index	√		
8.	Parallel computing	Example	√		√
9.	Standard Generalized Markup Language	Example	√		
10.	Software agent	S-index	√	√	

Table 6.11: The distribution of the feedback models submitted for topic profile refinement using the two feature extraction strategies.

Like Table 6.8, Table 6.11 indicates the type of the relevance feedback models the user used to express the relevance feedback. Table 6.9 shows that the average precision achieved by the full-text feature extraction strategy is 0.389, while the average precision made by the page-attribute feature extraction strategy is 0.415. Based on the various initial precision of the discovery processes, the improvement made by the full-text strategy is 0.121 (i.e., 45%). The page-attribute strategy improves the average precision by 0.146 (i.e., 54%). Furthermore, by considering the results reflected in Table 6.10, the average initial size of the topic feature vectors is 124. The size of the topic feature vector refined by the full-text strategy increased to 599.2. The size of the topic feature vector refined by the page-attribute strategy is much smaller, having the value of 234.5 as only the emphasized text in those feedback pages are drawn to refine the topic feature vectors. Those emphasized text may provide the unique features for the topic feature vector refinement. With such economic memory usage and the favorable achievement in precision improvements , the page-attribute feature extraction strategy proposed has achieved a better performance than the classical full-text strategy.

6.6 Comparisons between the Example Page and the Keyword Topic Profile Specifications

This section shows the experiments extended from the one illustrated in Section 6.2. Sometimes, a user still would like to use keywords to express the information need. This experiment was conducted to show the performance of the system in response to the keyword profile specifications. The comparisons with the example page topic profile specifications are also illustrated.

The precision values of the discovery processes using the example topic profile specification as well as the keyword topic profile specification are compared.

6.6.1 Experiment Setup

Random number of arbitrarily chosen topic profiles were selected for the performance evaluation of the example page and the keyword topic profile specifications. The simulated annealing schedule A and B were adopted. In the keyword topic profile specification, the keywords describing the search topics were explicitly submitted to the discovery system.

6.6.2 Results

	Search Topic (Example Page upon Schedule A)	Precision
1.	Bicycle	0.9
2.	Camera	0.205
3.	Computer Vision	0.175
4.	Geographic Information System	0.49
5.	Global Systems for Mobile Communication (b)	0.85
6.	Global Positioning System	0.84
7.	Honda Civic	0.705
8.	Honda Motorcycle	0.915
9.	Intelligent Agent	0.65
10.	Moon Apollo	0.455
11.	Palmpilot	0.8
12.	Silicon Valley	0.7
13.	Sports Car (a)	0.5
14.	Sports Car (b)	0.655
15.	Titanic	0.99
16.	Warcraft	0.68
	Average	**0.6625**

Table 6.12: The precision of the discovery processes using the example page topic profile specification upon simulated annealing schedule A.

	Search Topic (Keyword upon Schedule A)	Precision
1.	Alien	0.71
2.	Bicycle	0.83
3.	Canon Camera	0.61
4.	Computer Pattern recognition	0.58
5.	Computer Vision	0.48
6.	Ecology	0.54
7.	Geographic Information System	0.32
8.	Geology	0.52
9.	Global Positioning System	0.81
10.	Global Systems for Mobile Communication(b)	0.29
11.	Handheld PC	0.98
12.	Honda Civic	0.84
13.	Machine Learning Artificial Intelligent	0.63
14.	Mitsubishi Lancer Evolution	0.44
15.	Mitsubishi Pajero	0.47
16.	Moon Apollo	0.79
17.	Motorcycle	0.66
18.	Nikon Camera (a)	0.71
19.	Nikon Camera (b)	0.555
20.	Pathology	0.46
21.	Palmpilot	0.835
22.	Photography	0.57
23.	Quake	0.505
24.	Science Park	0.76
25.	Titanic	0.98
26.	Warcraft	0.615
	Average	**0.634**

Table 6.13: The precision of the discovery processes using the keyword topic profile specification upon simulated annealing schedule A.

	Search Topic (Example Page upon Schedule B)	Precision
1.	Bicycle	0.96
2.	Camera	0.835
3.	Computer Vision	0.75
4.	Geographic Information System	0.72
5.	Global Positioning System	0.88
6.	Global Systems for Mobile Communication (b)	0.585
7.	Handheld PC	0.96
8.	Honda Civic	1
9.	Honda Motorcycle	0.92
10.	Intelligent Agent	0.69
11.	Moon Apollo	0.76
12.	Neural Network	0.555
13.	Palmpilot	0.96
14.	Titanic	0.92
15.	Warcraft	0.89
	Average	**0.835**

Table 6.14: The precision of the discovery processes using the example page topic profile specification upon simulated annealing schedule B.

	Search Topic (Keyword upon Schedule B)	Precision
1.	Bicycle	0.705
2.	Computer Vision	0.78
3.	Geographic Information System	0.83
4.	Global Positioning System	0.76
5.	Global Systems for Mobile Communication (b)	0.76
6.	Honda Civic	0.9
7.	Mitsubishi Pajero	0.55
8.	Moon Apollo	0.78
9.	Motorcycle	0.88
10.	Palmpilot	0.7
11.	Photography	0.635
12.	Titanic	0.985
13.	Warcraft	0.96
	Average	**0.787**

Table 6.15: The precision of the discovery processes using the keyword topic profile specification upon simulated annealing schedule B.

Table 6.12 to Table 6.15 summarize the precision values of both topic profile specifications. With different simulated annealing schedules, the example page topic profile specification can achieve more satisfactory results compared with the keyword topic profile specification. As mentioned before, the example page topic profile specification makes use of the textual content of the example page to construct the topic feature vector. This topic feature vector provides a meaningful representation of the example page by comparing with the keyword specification. The keyword specification only provides limited number of features.

6.7 Summary from the Experimental Results

Several extensive sets of experiment were conducted and the results have been illustrated in the last few sections. As the simulated annealing algorithm is applied for the intelligent exploration of the Web information, the annealing schedule is an important factor on how the discovery system digs out information from the Web. It is observed that the annealing schedule should be carefully controlled so that the discovery system is able to explore and exploit. In that case, the average precision values for those discovery processes with exploration and exploitation will be higher.

Experiments were conducted to demonstrate the results of the discovery processes using the example page topic profile specification as well as the index page topic profile specification. The average precision values of the discovery processes using the example page profile specification are compared with the discovery processes using the keyword topic profile specification. Experimental results show that those processes using example page as input can achieve higher precision values than the ones using the keyword profile specification. The reason for this observation is that all of the text on the example page is used as features. More descriptive, detailed features can be obtained from the example pages. Effective queries are constructed on behalf of the users. As a result, a more meaningful, accurate relevance score for the document found on the Web can be determined. An experiment was conducted to demonstrate the results obtained from our discovery system using the index page topic profile specification. The result shows that those discovery processes can have a comparative result with the ones using the example page topic profile specification. Our system is able to draw the emphasized text on the pages to form descriptive features.

Some experiments were conducted to illustrate the precision improvement made by various relevance feedback models. With the positive page feedback, there are two feature extraction strategies proposed. The full-text feature extraction, which is treated as a classical strategy, can have improvement on the precision values. The improvement made by the page-attribute feature extraction is even larger. By comparing the memory usage of both relevance feedback feature extraction strategies, the page-attribute strategy uses much smaller memory usage since only those emphasized text is drawn to become the unique features for refining the original topic feature vectors. With such economic memory usage and the favorable achievement in precision value improvements, the page-attribute relevance feedback achieves a better performance than the full-text strategy.

After the analysis of the experimental results, we focus on the discussion about the implementation issues of our discovery system and the observations arose from the discovery processes. The discussions on these issues would be helpful for developing a similar discovery system in future. Like other software systems, a variety of performance tradeoffs had to be made during the course of developing our discovery system. The most important tradeoff that our system faced is the balance between the memory or disk usage and the discovery time. The most time-intensive and expensive component in our system is the Relevance Score Evaluation Process module. In order to save the memory usage, various term weights and relevance scores can be calculated during runtime. Though the memory space is saved, the discovery time may be elongated as the recomputation of those scores takes time. On the other hand, some evaluation quantities can be pre-calculated and stored. Running time is shortened but memory space is sacrificed. Besides, it can be observed that our system is developed to be dynamic so that no prior knowledge is required. Everything is fetched during runtime. In this case, the discovery process will depend on the network loading of the Internet. This is a common problem that other similar discovery tools on the Web have to face. Based on our discovery processes and the discovery results, several observations are made. From the results obtained through our example page and index page topic profile specifications, we observe that most of the discovered pages are "topic-similar" to the profiles. This provides more information or suggestions for the users to discover more relevant and similar resources via the relevance feedback. And the keyword profile specifications can provide more specific relevant spots for the user's direct access. However, those personal or organizational "private homepages" which have subordinates with a variety of distinct topics may affect the quality of the discovery. Discovery on some current issues and hot topics will have effects on the discovery results, as many of those "hot" documents appear on the Web and may have many interconnected hyperlinks. Our discovery can have great harvests on those popular topics.

Chapter 7

Conclusion

This chapter summarizes what we have proposed and concludes our research. It also suggests a few extensions for our proposed discovery system.

7.1 The Aim of Our Proposed System

In recent years, WWW has experienced a gigantic expansion in terms of the volume of information and information providers. Efforts making such information resources more manageable and accessible are highly in demand. Automatic textual analytical tools such as information discovery systems are crucial. A Web information discovery system must be able to discover information in the large information network of World-Wide Web with the consideration of a tradeoff between two objectives: exploitation and exploration of Web documents. It should also be able to relieve the burden of the users from time-consuming, frustrating human-guided tasks of browsing and searching. Though there exists some searchable engines on the Web, indexing the enormous and ever-increasing number of Web documents are infeasible. The shortcomings and imprecise results from those indexes make the searching tasks less effective. The main aim and goal for this research is to fulfill the two objectives and assist the users in information discovery from the gigantic WWW in a manageable and organized way.

7.2 The Favorable Features and the Effectiveness of Our Proposed System

We have developed a new approach for discovery information from the WWW. Our system is built to fulfill the two discovery objectives mentioned above. It traverses the Web actively, automatically and intelligently. A new metric for evaluating the usefulness of the discovered information by analyzing the textual content of the Web pages is developed. This metric takes the relevance of hyperlinks into consideration.

The effectiveness of our system has been already illustrated in Chapter 6. Experiments have demonstrated the strength of the example pages and index pages over the keyword-based input. Effective queries can be built from those example pages or index pages, which is beneficial for the users when the users do not know what should be included in the queries. The side effects made by the query constructions to the discovery results can be kept to minimal. Our system can provide valid, reachable, unambiguous, related documents without requiring the users to formulate the query, but just supply the sample page(s). The same rationale has been applied to the relevance feedback handling. The system has been shown to be able to learn the user's information need via relevance feedback. The users can optionally provide relevant pages and irrelevant pages as the input for the relevance feedback. The user is not required to select which keywords are relevant or irrelevant. The page-attribute positive relevance feedback strategy, which is a new way for handling relevance feedback, has been investigated. The results have shown that it is a promising way to handle relevance feedback on the World-Wide Web environment. Furthermore, the system is constructed to be dynamic. The fetching and the evaluations of the Web documents are done at runtime. It does not require any prior knowledge about information need and does not place any restriction on the domain. Our system can overcome the shortcomings brought by various information indexing systems and search engines.

7.3 Future Work

We have demonstrated that our proposed system addresses to the issues concerning with the WWW information discovery. Our next goal is to further enhance the system so as to make the system more practical. Some suggestions are given are summarized as follows:

- The discovery results provided by our discovery system are actually an accessible list of related documents. Like the index page topic profile specification, those discovery results can be recursively submitted to our discovery system and more novel related documents can be suggested.

- The wealth of the thesaurus incorporated with our discovery system can be enlarged. Apart from the synonyms which we have already included in our thesaurus, other lexical resources like hypnonyms can be provided to the thesaurus. This may provide knowledge on the Web exploration.

- More page attributes from the HTML-written Web documents can be collected so as to make the document feature representation more meaningful and descriptive. For example, we may include the textual label of the graphical clips on the Web documents as parts of the features.

- Other machine learning algorithms may be employed in our discovery system for handling the user relevance feedbacks.

- As the main goal of our discovery system is to serve the users, a more user-friendly graphical user interface may be necessary. Different languages may be handled.

Appendix A

List of URLs for the Example Pages

The list below shows the list of URLs for the example pages used in the various example page topic profile specifications:

Search Topic	URL
Bicycle	`http://www.bicycleoutfitter.com/pages/bicycles.htm`
Camera	`http://shuya.ml.org:888/~ zhu/photo/compact/guide.html`
Code Division Multiple Access	`http://www.whatis.com/cdma.htm`
Computer Game (Quake)	`http://www.citynet.net/quake.html`
Computer Vision	`http://www.iris.swin.edu.au/sergio/cv.html`
CORBA	`http://www.whatis.com/corba.htm`
Distributed Database	`http://www.csd.uu.se/~ d93bsp/Parallell/dis_db.html`
Geographic Information System	`http://h2o.er.usgs.gov/nsdi/pages/what_is_gis.html`
Global Positioning System	`http://www.orbital.com/Subsidiaries/magellan.html`
Global Systems for Mobile Communication (a)	`http://home.clara.net/hairydog/cell4.html`
Global Systems for Mobile Communication (b)	`http://www.gsmworld.com/history/history.htm`
Gold Coast	`http://www.goldcoast.qld.gov.au/tourism.html`
Handheld PC	`http://www.lgeus.com/hpc/overview.html`
Honda Civic	`http://www.zoomsporttuning.com/civic.htm`
Honda Motorcycle	`http://www.motorcycle.com/mo/mchhonda/cbr900rr_96.html`
Intelligent Agent	`http://www.whatis.com/intellig.htm`
Jeep	`http://4wd.sofcom.com/Jeep/Jeep.html`
MCSE	`http://www.whatis.com/mcse.htm`
Moon Apollo	`http://cass.jsc.nasa.gov/pub/expmoon/apollo_landings.html`
Neural Network (a)	`http://www.cybernetics.com.au/Technology/Annfm4.htm`
Palmpilot	`http://palmpilot.3com.com/index.html`
Parallel Computing	`http://i30www.ira.uka.de/courses/Prozessorzuteilung/links/introductions.html`
Silicon Valley	`http://www.silvalonline.com/history.html`
Simulated Annealing	`http://www.ee.umr/edu/~ zz/thesis/thesis/node10.html`
Sports Car (a)	`http://sunflower.singnet.com.sg/~ turbo/`
Sports Car (b)	`http://claus-winzer.com/mmal/306.html`
Standard Generalized Markup Language	`http://www.whatis.com/sgml.htm`
Titanic	`http://www.lib.virginia.edu/cataloging/vnp/titpref.html`
Warcraft	`http://www.es.co.nz/~ toosh/war3.htm`

Table A.1: List of URLs for the example pages used in the various example page topic profile specifications.

Appendix B

List of URLs for the Arbitrarily Chosen Index Pages

The list below shows the list of URLs for the index pages arbitrarily chosen from the WWW used in the various index page topic profile specifications:

Search Topic	URL
Australia Travel Tourism	`http://www.atn.com.au/links/info.html`
EC with IA	`http://bold.coba.unr.edu/odie/paper.html`
Genetic Algorithm	`http://www.astro.gla.ac.uk/users/scott/mystuff/ga.html`
Geographic	`http://taiga.geog.niu.edu/faculty/greene/links.html`
Information Filtering	`http://www.cs.kun.nl/is/research/filter/references.html`
Intelligent Agent	`http://bold.coba.unr.edu/odie/paper.html`
Neural Network (b)	`http://www.trajecta.com/neural/links.htm`
Photography (b)	`http://www.mbay.net/~ cgd/photo/pholinks.htm`
Photography (c)	`http://www.reflector.msstate.edu/~ jtiffin/Photolinks.html`
Subaru	`http://sunflower.singnet.com.sg/~ turbo/subaru.htm`
Year 2000	`http://www.y2ktool.com/about.shtml`

Table B.1: List of URLs for the index pages arbitrarily chosen from the WWW used in the various index page topic profile specifications.

Bibliography

[1] R. Armstrong, *et al.* "WebWatcher: A Learning Apprentice for the World Wide Web". *AAAI 1995 Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, Stanford, March 1995. `http://www.cs.cmu.edu/afs/cs/project/theo-6/web-agent/www/webagent-plus.ps.Z`

[2] P.E. Baclace. "Personal Information Intake Filtering". *The Bellcore Information Filtering Workshop*, November 1991. `http://www.baclace.net/ifilter1.html`

[3] M. Balabanović, Y. Shoham. "Learning Information Retrieval Agents: Experiments with Automated Web Browsing". *AAAI 1995 Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, Stanford, March 1995.
`ftp://venera.isi.edu/sims/sss95/balabanovic.ps.Z`

[4] N.J. Belkin, W.B. Croft. "Information Filtering and Information Retrieval: Two Sides of the Same Coin?" *Communications of the ACM*, 35(12):29-38, December 1992.

[5] T. Berners-Lee. "HyperText Markup Language". *WorldWide Web Seminar.*
`http://www.w3.org/Talks/General.html`

[6] T. Berners-Lee, R.Cailliau, A. Luotonen, H.F. Nielsen, and A. Secret. "The World-Wide Web". *Communications of the ACM*, 37(8):76-82, August 1994.

[7] D. Billsus, M. Pazzani. "Revising User Profiles: The Search for Interesting Web Sites". *Proceeding of the International Multi-Strategy Learning Conference*, VA, 1996. `http://www.ics.uci.edu/ pazzani/Publications/RevUserProfiles.ps`

[8] J. Binkley, L. Young. "RAMA: An Architecure for Internet Information Filtering". *Journal of intelligent Information Systems*, 5:81-99, 1995.

[9] D.C. Blair, M.E. Maron. "An evaluation of retrieval effectiveness for a full-text document-retrieval system". *Communications of the ACM*, 28(3):289-299, 1985.

[10] C. M. Bowman, *et al.* "Scalable Internet Resource Discovery : Research Problems and Approaches". *Communications of the ACM*, 37(8):98-107,114, August 1994.

[11] C. M. Bowmam, *et al.* "The Harvest Information Discovery and Access System". *Proceedings of the 2nd International Conference on the World Wide Web*, Chicago, 1994.

[12] J. Carriére. "WebQuery: Searching and Visualizing the Web through Connectivity". *Proceeding of the 6th International Conference on the World Wide Web*, Santa Clara, 1997. `http://proceedings.www6conf.org/HyperNews/get/PAPER96.html`

[13] H. Chen. "A Machine Learning Approach to Document Retrieval: An Overview and an Experiment". *in Proceedings of the 27th Annual Hawaii International Conference on System Sciences (HICSS-27)*, Hawaii, January, 1994.
`http://ai.bpa.arizona.edu/papers/PS/hicss27g.ps`

[14] H. Chen. "Machine Learning for Information Retrieval: Neural Networks, Symbolic Learning, and Genetic Algorithms". *Journal of the American Society for Information Science*, 46(3):194-216, April, 1995.
`http://ai.bpa.arizona.edu/papers/PS/mlir93.ps`

[15] H. Chen and J. Kim. "GANNET: Information Retrieval Using Genetic Algorithms and Neural Nets". *IEEE Transactions on Neural Networks*, 1994.
`http://ai.bpa.arizona.edu/papers/PS/gannet93.ps`

[16] H. Chen, C. Schuffels, R. Orwig. "Internet Categorization and Search: A Self-Organizing Approach". *Journal of Visual Communication and Image Representation Special Issue on Digital Libraries*, 7(1):88-102, 1996. `http://ai.bpa.arizona.edu/papers/PS/som95.ps`

[17] T. Dean, J. Allen, Y. Aloimonos. *Artificial Intelligence, Thoery and Practice*, The Benjamin/Cummings Publishing, 1995.

[18] Digital Equipment Corporation. The AltaVista Search Engine Homepage. `http://altavista.digital.com`

[19] Digital Equipment Corporation. "Our Strengths, AltaVista's Latest Features". `http://altavista.digital.com/av/content/about_our_strengths.htm`

[20] Digital Equipment Corporation. The Hotbot Search Engine Homepage. `http://www.hotbot.com`

[21] R. Desai and R. Patil. "SALO:Combining Simulated Annealing and Local Optimization for Efficient Global Optimization", *in Proceedings of the* 9^{th} *Florida AI Research Symposium*, 233-237, 1996.

[22] K. Eguchi, H. Ito, A. Kumamoto. "Information Retrieval Considering Adaptation to User's Behaviors on the WWW". *Proceedings of the* 2^{nd} *International Workshop on Information Retrieval with Asian Languages*, Tsukuba-Shi, Japan, October 1997.

[23] D. Eichmann. "The RBSE Spider - balancing Effective Search Against Web Load". *Proceedings of the* 1^{st} *International Conference on the World Wide Web*, Geneva, 1994. `http://rbse.jsc.nasa.gov/eichmann/urlsearch.html`

[24] D. Eichmann. "Advances in Network Information Discovery and Retrieval". *Journal of Software Engineering and Knowledge Engineering*, 5(1):143-160, 1995.

[25] O. Etzioni. "The World-Wide Web: Quagmire or Gold Mine?" *Communications of the ACM*, 39(11):65-68, November 1996.

[26] O. Etzioni. "Moving Up the Information Food Chain, Deploying Softbots on the World Wide Web". *AI Magazine, Amermican Association for Artificial Intelligence*, 11-18, Summer 1997.

[27] P.W. Foltz, S.T. Dumais. "Personalized Information Delivery: an Analysis of Information Filtering Methods". *Communications of the ACM*, 35(12):51-60, December 1992.

[28] W.B. Frakes, R.Baeza-Yates. *Information Retrieval Data Sturctures & Algorithms*, Prentice Hall, 1992.

[29] S. Franklin, A. Graesser. "Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents". *Proceedings of the* 3^{rd} *International Workshop on Agent Theories, Architectures, and Languages*, Springer-Verlag, 1996.

[30] S. Geman and D. Geman. "Stochastic Relaxation, Gibbs Distribution and the Bayesian Restoration in Images". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721-741, 1984.

[31] M. Ginsberg. *Essentials of Artificial Intelligence*, Morgan Kaufmann Publishers, CA, 1993.

[32] D.E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA, 1989.

[33] D.E. Goldberg. "Genetic and evolutionary algorithms come of age". *Communications of the ACM*, 37(3):113-119, March 1994.

[34] Infoseek, Inc. The Infoseek Search Engine Homepage. `http://www.infoseek.com`

[35] L. Ingber. "Simulated Annealing: Practice versus Theory". *Journal of Mathematical Computer Modelling*, 18(11):29-57, 1993. `http://www.ingber.com/asa93_sapvt.ps.Z`

[36] S. Kirkpartrick, C.D. Gelatt, Jr., M.P. Vecchi. "Optimization by Simulated Annealing". *Science*, 200:671-680, 1983.

[37] M. Koster. "The Web Robots Database". Nexor Corp. `http://info.webcrawler.com/mak/projects/robots/active.html`

[38] M. Koster. "Robots in the Web: threat or treat?". *ConnXions*, 9(4), April, 1995. `http://info.webcrawler.com/mak/projects/robots/threat-or-treat.html`

[39] H.C.M. de Kroon, T.M. Mitchell, E.J.H. Kerckhoffs. "Improving Learning Accuracy in Information Filtering". *The 13th International Conference on Machine Learning, Workshop on Machine Learning meets Human Computer Interaction*, 1996. `http://www.ics.forth.gr/~moustaki/ICML96_HCI_ML/kroon.ps`

[40] B. Krulwich and C. Burkey. "The InfoFinder Agent: Learning User Interests through Heuristic Phrase Extraction". *IEEE Expert/Intelligent Systems & Their Applications*, 12(5), September/October 1997.

[41] P.J.M. van Laarhoven, E.H.L. Aarts. *Simulated Annealing: Theory and Applications*, Kluwer Academic Publilshers, Dordrecht, 1988.

[42] K. Lang. "NewsWeeder: Learning to Filter Netnews". *Proceedings of the 12th International Conference on Machine Learning*, Lake Tahoe, CA, 1995

[43] D.D. Lewis, R.E. Schapire, J.P. Callan, R. Papka. "Training Algorithms for Linear Text Classifiers". *Proceedings of the* 19th *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'96)*, Zurich, Switzerland , 1996.

[44] H. Lieberman. "Letizia: An agent that assists web browsing". *International Joint Conference on Artificial Intelligence*, Montreal, August 1995. `http://lieber.www.media.mit.edu/people/lieber/Lieberary/Letizia/Letizia-AAAI/Letizia.html`

[45] P. Maes. "Agents that Reduce Work and Information Overload". *Communications of the ACM*, 37(7):31-40,146, July 1994. `http://pattie.www.media.mit.edu/people/pattie/CACM-94/CACM-94.p1.html`

[46] M. Marchiori. "The Quest for Correct Information on the Web: Hyper Search Engines". *Proceeding of the* 6th *International Conference on the World Wide Web*, Santa Clara, 1997. `http://proceedings.www6conf.org/HyperNews/get/PAPER222.html`

[47] M.L. Mauldin, J.R.R. Leavitt. "Web Agent Related Research at the Center for Machine Translation". *Proceedings of the ACM Special Interest Group on Networked Information Discovery and Retrieval (SIGNIDR-94)*, August 1994.

[48] F. Menczer. "ARACHNID: Adaptive Retrieval Agents Choosing Heuristic Neighborhoods for Information Discovery". *Proceedings of the International Conference on Machine Learning'97*, 227-235, 1997.

[49] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, Berlin Heidelberg, 1992.

[50] T. Mitchell, *et al.* "Experience with a Learning Personal Assistant". *Communications of the the ACM*, 37(7):81-91, July 1994. `http://www.cs.cmu.edu/afs/cs/user/mitchell/ftp/cacm.ps.Z`

[51] K.J. Mock, V.R. Vemuri. "Adaptive User Models for Intelligent Information Filtering". *Proceedings of the Third Golden West International Conference on Intelligent Systems*, Las Vegas, Nevada, 1994.

[52] D. Ngu, X. Wu. "SiteHelper: A Localized Agent that Helps Incremental Exploration of the World Wide Web". *Proceedings of the* 6th *International Conference on the World Wide Web*, Santa Clara, 1997. `http://proceedings.www6conf.org/HyperNews/get/PAPER68.html`

[53] M. Pazzani, J. Muramatsu, D. Billsus. "Syskill & Webert: Identifying interesting web sites". *Proceedings of the 13th National Conference on Artificial Intelligence AAAI-96*, 1996

[54] M. Pazzani, D. Billsus. "Learning and Revising User Profiles: The identification of Interesting Web Sites". *Machine Learning*, 27:313-331, 1997. `http://www.ics.uci.edu/ pazzani/Publications/SW-MLJ.pdf`

[55] B. Pinkerton. "Finding What People Want: Experiences with the WebCrawler". *Proceedings of the* 2^{nd} *International Conference on the World Wide Web*, Chicago, 1994.

[56] J. R. Quinlan. *C4.5: Programs for Machine Learning*,Morgan Kaufmann, 1993.

[57] B.J. Rhodes, T. Starner. "Remembrance Agent. A continuously running auomated information retrieval system". *Proceedings of the* 1^{st} *International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'96)*, 487-495, 1996. `http://rhodes.www.media.mit.edu/people/rhodes/Papers remembrance.html`

[58] E. Rich, K.Knight. *Artificial Intelligence*, McGraw-Hill, 1991.

[59] C.J. van Rijsbergen. "A new theoretical framework for information retrieval". In F. Rabiti (Ed.), *Conference Proceedings of ACM/SIGIR, 1986*, Pisa, 1986.

[60] C.J. van Rijsbergen, M. Lalmas. "Information Calculus for Information Retrieval". *Journal of the American Society for Information Science*, 47(5):385-398, 1996.

[61] S.E. Robertson, K. Sparck Jones. "Relevance Weighting of Search Terms". *J. American Society for Information Science*, 27(3):129-146, 1976.

[62] J.J. Rocchio. "Relevance Feedback in Information Retrieval". In G. Salton (Ed.), *The SMART Retrieval System*, Prentice Hall, 1971.

[63] G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*, Addison-Wesley, 1989.

[64] G. Salton, M.J. McGill. *Introduction to Modern Information Retrieval*, McGraw-Hill, 1983.

[65] G. Salton, C. Buckley. "Improving Retrieval Performance by Relevance Feedback". *Journal of the American Society for Information Science*, 41(4):288-297, June 1990.

[66] J. Savoy. "An Extended Vector-Processing Scheme for Searching Information Hypertext Systems". *Information Processing & Management*, 32(2):155-170, 1996.

[67] B.D. Sheth, P. Maes. "Evolving Agents for Personalized Information Filtering". *Proceedings of the* 9th *IEEE Conference on Artificial Intelligence for Applications*, 1993.

[68] B.D. Sheth. "A Learning Approach to Personalized Information Filtering". *M.S. Thesis*, Massachusetts Institute of Technology, February 1994. `ftp://ftp.media.mit.edu/pub/agents/interface-agents/news-filter.ps`

[69] E. Spertus. "ParaSite: Mining Structural Information on the Web". *Proceeding of the* 6th *International Conference on the World Wide Web*, Santa Clara, 1997. `http://proceedings.www6conf.org/HyperNews/get/PAPER206.html`

[70] B. Starr, M.S. Ackerman, M. Pazzani. "Do I Care? – Tell Me What's Changed on the Web". *AAAI Spring Symposium*, Stanford, CA, 1996. `http://www.ics.uci.edu/~pazzani/Publications/mlia96-bstarr.ps`

[71] C.W. Yue, W. Lam. "An Intelligent Content-Based Web Document Discovery System". *Proceedings of the* 2nd *IEEE International Conference on Intelligent Processing Systems*, Gold Coast, August 1998.

[72] B. Yuwono, D.L. Lee. "A World Wide Web Resource Discovery System". *Proceedings of the* 4th *International Conference on the World Wide Web*, Boston, MA., 1995.
`http://www.cs.ust.hk/~dlee/Papers/www/www4.ps.Z`

[73] B. Yuwono, D.L. Lee. "Search and ranking algorithms for locating resources on the World Wide Web". *Proceedings of the* 12th *International Conference on Data Engineering*, New Orleans, Louisiana, February 1996. `http://www.cs.ust.hk/~dlee/Papers/www/icde96-www.ps.gz`