SCUOLA DI DOTTORATO
UNIVERSITÀ DEGLI STUDI DI MILANO-BICOCCA

Dipartimento di / Department of

Informatica Sistemistica e Comunicazione

Dottorato di Ricerca in / PhD program Informatica Ciclo / Cycle XXX

Web Information Foraging using Multi-agent Systems

Cognome / Surname Drias Nome / Name Yassine

Matricola / Registration number 811389

Tutore / Tutor: Prof. Stefania Bandini

Supervisors: Prof. Gabriella Pasi, Prof Samir Kechid

Coordinatore / Coordinator: Prof. Stefania Bandini

ANNO ACCADEMICO / ACADEMIC YEAR 2017/2018

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Yassine Drias
October 2017

Acknowledgements

Reaching the end of the study for this PhD dissertation, I would like to thank the people, without the help of which this work would not be possible.

First of all, I would like to tender my heartfelt thankfulness to my supervisors **Prof. Samir Kechid** and **Prof. Gabriella Pasi** for their priceless supervision, continuous inspiration and enthusiastic support in every stage of this work. Their deep insight into problems, imaginative ideas, technical guidance and sufficient encouragement were a constant source of inspiration and motivation. Their critical comments, eye for detail and persistent encouragement strongly motivated me to strive hard to achieve the high targets set in completion of this thesis.

I would also like to thank all of my fellows from the *Department of Computer Science* at the *University of Sciences and Technology Houari Boumediene* and my fellows from the *Department of Computer Science, Systems and Communications* at the *University of Milano-Bicocca*. It has been a pleasure working and interacting with them during this period. I would also like to thank all the members of the *Laboratory of Research in Artificial Intelligence (LRIA)* and the members of the *Information Retrieval Laboratory (IR LAB)*.

This work wouldn't be possible without the continuous support of my family. I owe them my deepest gratitude for their continuous guidance, encouragement and support during my entire life.

Abstract

The main purpose of search engines, which support the task of *Information Retrieval*, is to provide the users with an easy method to find information on the Web. Despite their advantages, Web search engines still have numerous limitations. Several works have been done to enhance information access systems and offer a better user experience in different fields such as distributed and contextual information retrieval, link analysis and exploratory search.

This thesis deals with *Web Information Foraging*, which is a recent information access paradigm that aims at discovering paths leading to relevant information on the Web. The principal goal of the research undertaken during the Ph.D. has been to design and implement effective and efficient Information Foraging systems based on multi-agent technology. For this purpose, we investigated Information Foraging in different domains including health, scientific publications and social media.

To address this issue, we proposed a modular architecture with two important phases for the Information Foraging system to be developed. The first phase is a learning process, which aims to localize the most relevant Web pages that might interest the user according to his/her interests. This can be performed on a fixed instance of the Web. The second phase takes into account the openness and dynamicity of the Web and it consists of an incremental learning starting from the results of the first phase and reshaping the outcomes taking into account the changes that occurred on the Web. The whole system offers a tool to help users access information online easily. The development of the system went through three important steps, where different concepts and technologies were used in order to achieve both effectiveness and efficiency and also to make the system able to operate on various environments.

First, we proposed a *Swarm Intelligence*-based approach to simulate the information foraging process as described in the *Information Foraging Theory*. We carried out our proposal using *Bee Swarm Optimization (BSO)* for the sake of implementing an automatic information foraging process grounded on the humans' information foraging behavior. Knowing that BSO is a swarm intelligence approach involving reactive agents and that the concept of Information Foraging is based on nature, we thought this could be one of the most appropriate ways to tackle the problem of Web Information Foraging. To validate our proposal, experiments were

conducted on *MedlinePlus*, a website dedicated to the medical domain and which contains information on over 1000 diseases and health conditions. The results were promising and showed the ability of Information Foraging to access relevant information on the Web based on a user's interests. In order to improve the outcomes of the first resolution attempt, we decided to employ *Ant Colony Optimization (ACO)* hybridized with taboo search instead of BSO for the implementation of the Information Foraging. Indeed, in Information Foraging we deal with a graph structure, which represents the Web, and therefore ACO algorithms[1] could offer a better performance than BSO to tackle the issue. Extended experiments were conducted on *MedlinePlus* including a comparison study between both implementations (BSO and ACO).

As a second step, we decided to focus more on the efficiency and the scalability aspect of our system. A dynamic information foraging approach was proposed based on game theory and more precisely on normal-form games. We modeled the issue of Web information foraging as a game played by a set of self-interested agents that aim to reach relevant Web pages in a short time. We considered a pure strategy, a mixed strategy and a fully mixed strategy respectively for three kinds of players. We applied our new approach on the scientific publications domain, where the goal was to find relevant publications, indexed on repositories such as *DBLP* and *ACM*, based on a certain user's interests. Experiments were performed on large datasets including the *Citation Network Dataset* which contains more than 2.3 million scientific publications. The results confirmed once again the ability of our approach to find relevant information in an effective and efficient way. They also showed the capacity of our system to work on very large datasets without compromising the outcomes' quality or the response time. Furthermore, we undertook a comparative study between our approach and previous information access approaches such as classical Information Retrieval.

Finally, the third part of this work is dedicated to Information Foraging on social networks. Nowadays, people are becoming more dependent on social media in their daily life. They use them to share information and interact with each other, which highly contributes to the growth of the volume of online public data. We proposed an approach that offers social media' users the possibility of getting relevant and credible information in a rapid and effective way. We adapted our Web Information Foraging system so that it can work on social graphs. In addition to considering the users' interests, the system takes advantage of their social relations and interactions to better define their information needs and therefore find more relevant information. We built our dataset using real data we extracted from the information sharing network *Twitter*. The produced results consist in surfing paths leading to

[1] ACO algorithms were first developed to solve the *Traveling Salesman Problem*, which was modeled as an undirected weighted graph.

relevant information taking into account the user's interests and the credibility of the foraged information.

Keywords: Web Information Foraging, Multi-agent systems, Information Access, Swarm Intelligence, Game Theory, Self-interested agents, Social Networks.

Table of contents

List of figures

List of tables

Introduction

The democratization of the Internet allowed billions of users to access the Web content since the early 2000's. The creation of *forums* and *blogs* has provided Web users with the opportunity not only to access but also to generate and share their own content on the Web. With the advent of the Web 2.0 and of social media, the user-generated content is becoming more important and significant. The impressive growth of content available to users is illustrated by the increase of volume of available data online [54] [WWW]. In order to handle the exponential growth of the Web, the development of new large-scale information access techniques is required.

Research in information access aims to identify, obtain and make use of information useful to specific needs in an effective way [3]. Several research efforts are spent nowadays to the objective of defining systems that support users in accessing relevant information in an effective way. Some important tasks related to the issue of information access are: *Information Retrieval*, *Information Filtering*, and *Information Foraging*.

The usual way in which users address the task of accessing information relevant to their needs is by means of Web search engines, which support the task of Information Retrieval (IR). When using search engines, users describe their information needs by means of a keyword-based query, which is evaluated on a corpus of documents previously indexed by the system. The retrieved documents are ranked based on their estimated relevance to the query. In [11], the author pointed out that Web queries can be classified in three categories based on the user intent. They can be *navigational*, when users aim to reach a particular Web page; *informational*, when the goal is to get information on a certain topic, or *transactional* if the purpose is to reach Web pages where further interaction would take place.

More recently, the *Exploratory Search* paradigm has been introduced, to address situations in which users intend not only to find relevant documents, but they are also interested in learning, discovering and understanding complex and new topics [39]. Besides querying, Exploratory Search systems are also based on Web browsing strategies. The idea is to go beyond returning a set of ranked results as an answer to a query [36, 66].

Web Information Foraging: A new information access paradigm

A recent paradigm related to *Exploratory Search*, and which aims at discovering paths leading to relevant information on the Web is Information Foraging. The idea of Information Foraging is grounded on the *Optimal Foraging Theory* [65] developed by anthropologists to model the animals' behavior while foraging food. The *Information Foraging Theory* was first developed in [51]. The authors established their study on the similarity between the animals' food foraging behavior and the behavior of humans while seeking information on the Web. The theory is based on the assumption that, when searching for information, users rely on their senses to perceive the *information scent* that helps them to reach their goal just like animals do when they follow the scent of their preys.

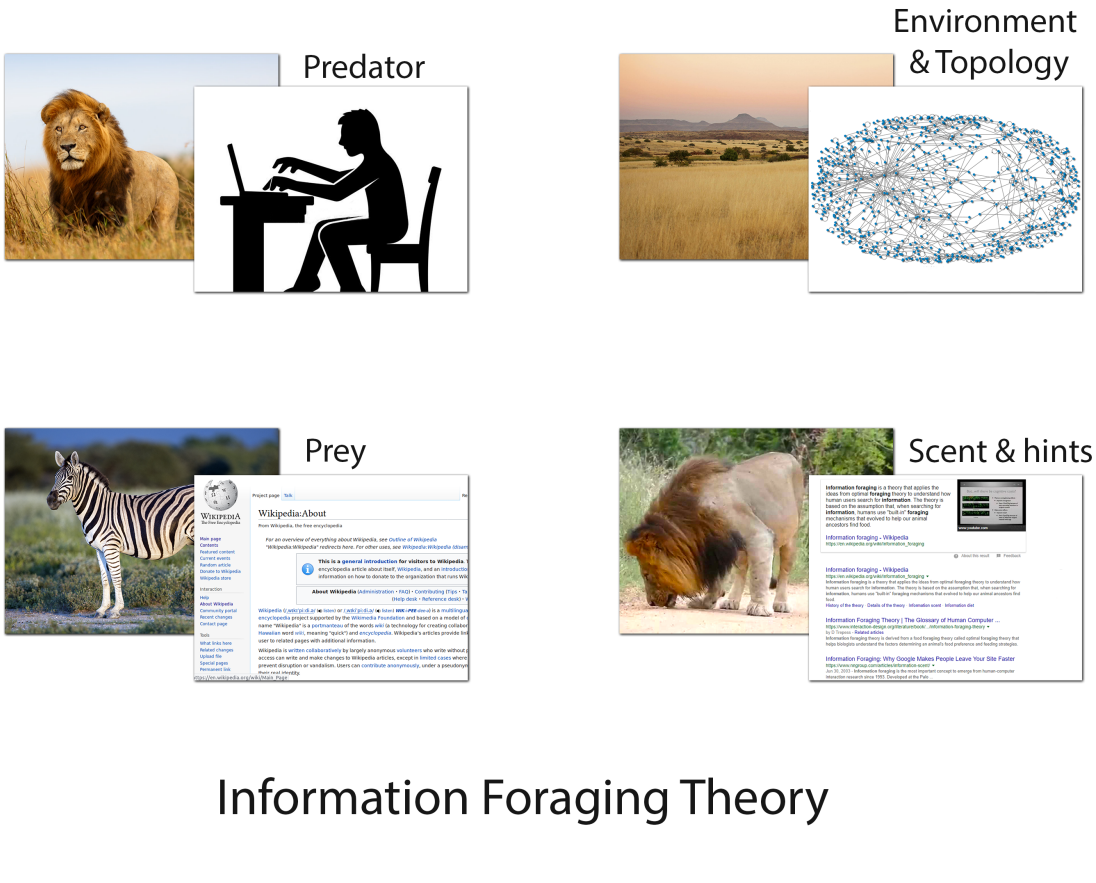

Fig. 1 Analogy between Information Foraging and Food Foraging

Elements	Food Foraging	Information Foraging
Actors	Predator	Web User
	Prey	Relevant information
Trigger	Hunger	Information needs
Environment	Nature, wilderness	Web structure
Cues	Scent of the prey	Hyperlinks, icons, titles

Table 1 Food Foraging analogy with Information Foraging

Figure 1 and Table 1 illustrate the analogy between information foraging and animals' food foraging. In fact, information foraging can be seen as a projection of the food foraging process, where the predator is a Web user looking for relevant information (prey). In order to get this information, the user navigates on the Web (environment) and makes use of some hints and cues (scent).

With the major progress experienced by the Web recently, the availability of an intelligent system capable of foraging information from the Web in an efficient and effective way has become a crucial concern. Information foraging can be seen as a simulation of Web users' surfing behavior. Nowadays its importance resides in the fact that the Web is in an incessant growth and the human ability to explore the astronomical amount of data on it is relatively limited. Besides, tackling such issue is very welcomed in domains like health, business, finance and science. The potential users not only will spend less time in getting access to the needed information but they can even get it in real time.

Motivations of the thesis work

With the information explosion, information access has become one of the hottest topics in computer science and working on it is getting more challenging everyday. Information foraging is considered to be one of the most important concepts that emerge from human-computer interaction research [62, 50]. It has received significant attention both in academic and industrial research during the past decade.

Web users are actively in search for information thanks to Web search engines such as Google, Yahoo!, Bing, etc. To this purpose the users apply foraging strategies in order to maximize their information gain. The information foraging theory offers the possibility to simulate the user's behavior when searching for information and thus, to estimate in an automatic way the amount of information gain a user would get from visiting certain Web pages.

In addition to yielding a new analysis methods for human-computer interaction, the information foraging theory provides a new path of research for Web users' information search behavior. At this point, research in this field is still deepening. There are many concerns for further study such as better predicting surfing paths leading to relevant information, developing an entire information access system based on the information foraging theory, exploring collaborative foraging approaches and employing information foraging to resolve daily life problems. In the present dissertation, we are motivated to address some of these issues and contribute with our research to design reliable solutions.

Problem Statement

The goal of the proposed research is to design and develop a Web information foraging system, which is capable of discovering in an automatic way the surfing paths that Web users would follow while seeking information on the Web. The developed system must fulfill the following essential key objectives:

1. It has to be able to work on different setups and environments, especially on the Web structure.
2. Since we are dealing with the Web, the system should be scalable so that it can deal with big volumes of data.
3. It must be applicable to various contexts related to information access such as health, scientific domains, social networks, etc.
4. Last but not least, its performance should be better than or at least comparable to other existing information access approaches.

Figure 2 showcases the inputs of a Web Information Foraging System and the outputs it produces. The user's interests comprise her/his information needs and are considered as the initiators of the information foraging process. In order to satisfy these information needs, the system forages information on Web structures that can be either websites, repositories, social networks, etc. The outcomes of the system should be a list of surfing paths leading to relevant information and ranked according to their relevance to the user.

Proposed Research and Contributions

The question to be asked is whether we can propose an effective and an efficient information access approach based on information foraging ? In order to answer this matter, we investi-

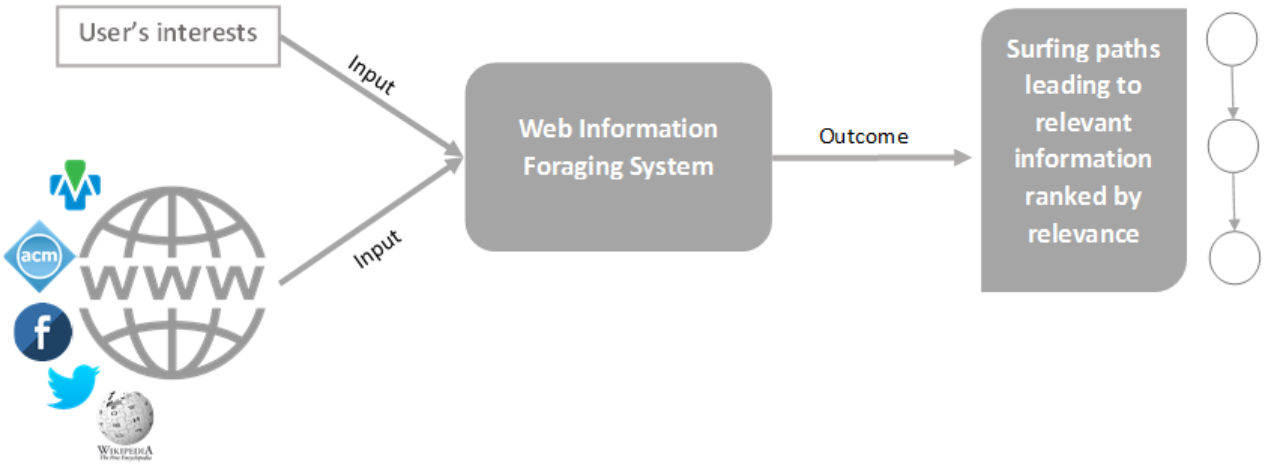

Fig. 2 Inputs and Outputs of a Web Information Foraging System

gated information foraging on different domains such as health, scientific publications and social networks platforms. To that end and to conceive such a system, Multi-Agents, Swarm Intelligence and Game Theory were explored and deployed in this thesis.

Multi-agent systems are composed of multiple interacting intelligent agents (computational entities) to some degree autonomous and able to cooperate, compete, communicate, act flexibly, and exercise control over their behavior within the frame of their goals. Multi-agent systems can be used to solve complex problems that are difficult or impossible for an individual agent or a monolithic system to solve. They are the enabling technology for a wide range of advanced applications relying on distributed and parallel processing of information, data, and knowledge relevant in domains ranging from industrial manufacturing to e-commerce and health care. We propose in Chapter 2 a Web information foraging model based on multi-agent systems. First, we present three Web surfing strategies that we model using artificial agents. The agents are meant to navigate on the Web following a certain strategy in order to forage information. We present after that the multi-agent architecture we use to design our Web information foraging system. It includes two main phases, which are both based on a group of agents that work together in order to find relevant information in an optimal time.

Swarm Intelligence (SI) deals with natural and artificial systems composed of a set of individuals that coordinate using decentralized control and self-organization. In particular, the discipline focuses on the collective behaviors resulting from interactions. SI systems consist of a population of simple agents (individuals) interacting locally with each other and with their environment. The inspiration often comes from nature, especially from biological systems. The agents follow simple rules, and although there is no centralized

control structure dictating how individual agents should behave, local, and to a certain degree random, interactions between such agents lead to the emergence of "intelligent" global behavior, unknown to the individual agents. Examples of SI systems include ant colonies, swarms of bees, flocks of birds and fish schooling. In Chapter 3, two swarm intelligence approaches are exploited to implement our Web information foraging system. To our best knowledge, this is the first time SI approaches are used to model information foraging. As a first step, we propose an implementation using *Bee Swarm Optimization* in which the agents are modeled as artificial bees. We take advantage of the collective behavior of the artificial bees in order to locate relevant information on the Web. This first attempt gave us the opportunity to study in depth the information foraging problem as an optimization problem. As a result, we took a further step with the aim of improving the performance of our system by proposing a new implementation using *Ant Colony Optimization (ACO)*. In fact ACO algorithms are more suitable for Web information foraging since they were designed to work on graph structures. Both models were implemented and evaluated on a medical Website to the aim of providing patients with relevant medical articles based on their diagnosis and their medical conditions.

Game theory is the science of strategy, it attempts to determine mathematically and logically the actions that *players* (agents) should take to ensure the best outcomes for themselves in a wide array of games. It studies mathematical models of conflict and cooperation between the playing agents. Game theory is mainly used in economics, political science, psychology, logic as well as computer science. We exploit the properties and tools offered by game theory in Chapter 4 to design a Web information foraging system based on *self-interested* agents. Once again, this is the first time game theory and self-interested agents are used to address Web information foraging. We took advantage of game strategies to better model, formalize and implement real users' Web surfing behaviors with self-interested agents. In addition to that, we implemented a Naïve Bayes text classifier to perform a prepossessing phase aiming at classifying Web pages based on their content prior the launch of the foraging process in order to optimize the latter. The designed system was evaluated on scientific publications repositories.

Social network sites (SNSs) are Web-based services that allow individuals to (1) construct a public or semi-public profile within a bounded system, (2) articulate a list of other users with whom they share a connection, and (3) view and traverse their list of connections and those made by others within the system. The nature and nomenclature of these connections may vary from site to site [8]. Social networks consist of social actors (such as friends, acquaintances, and coworkers) connected by interpersonal relationships. SNSs are generally hosted on a website or on an online service that facilitates the communication between the

members of the network. They allow the creation of virtual communities of people with a common interest and give them the opportunity to create and share information, resources, ideas and career interests. User-generated content, such as text posts or comments, digital photos or videos, and data generated through all online interactions, are considered as the lifeblood of social networks. In fact, the concept of forums, blogs and more recently social networks gave users from the large public the possibility to not only access content on the Web but also to generate their own. Since the user-generated content highly influenced and accelerated the online data volume growth, we think that exploiting social networks using information foraging would be a benefit to the information access community. Chapter 5 of this dissertation focuses on extending the proposed Web information foraging system in order to make it capable of working on social networks platforms. Experiments were conducted on real data extracted from a social network, which constitutes another novelty since this is the first Web information foraging system to be tested on a real social network.

Outline of the manuscript

The the manuscript is structured as follows. In Chapter 1, we discuss related works and recent literature on Web information foraging. We propose in Chapter 2 a Web information foraging model, which will be used for the design of a Web information foraging system. In Chapter 3, we describe the system we developed exploiting *Swarm Intelligence* and more precisely, *Bee Swarm Optimization* and *Ant Colony Optimization*. In Chapter 4, we present an enhanced version of our Web information foraging model using *Game Theory* and *Self-Interested agents*. In Chapter 5, we propose and extension to our model in order to make it able to operate on social networks and information sharing platforms. Finally we summarize our thesis work and we discuss some potential future axes.

Chapter 1

Literature Review

1.1 Introduction

In this chapter of the dissertation, a thorough review of a collection of previous works that we judge among the most related to *Web Information Foraging* will be presented.

As stated previously, Information Foraging is a recent paradigm that emerged in the late 90's. Several works have been done since the early 2000's with the purpose of implementing the theoretical studies on information foraging. Most of these works were focused on modeling and formalizing the process of information foraging and testing it on limited datasets and domains. In the following literature review, we address three essential aspects, namely: the Information Foraging Theory, Information Foraging using Multi-agent Systems and Information Foraging on social networks.

1.2 From Food Foraging to Information Foraging

The concept of optimal foraging was introduced well before the emergence of the Web. Anthropologists studied the animal's food foraging behavior around 1970. The task of foraging is grounded on the *Optimal Foraging theory (OFT)* [64], which paved the way to the *Information Foraging theory (IFT)* [51]. The authors in [14] apply the ideas from the optimal foraging theory to understand how humans search for information. The theory is based on the assumption that, when searching for information, humans use *built-in* foraging mechanisms that evolved to help our animal ancestors find food. This mechanism relies on the concept of information scent. In fact, Web users count on the information scent they get at each Web page they visit in order to guide their navigation.

Information foraging (IF) shares the same goal as *Information Retrieval (IR)*. While IR uses a structured process (Web pages indexing and query-page matching), IF consists in navigating from one page to another for the same purpose. The first studies that treated this topic were developed at the *Palo Alto Research Center (Xerox PARC previously)* by the team of *Stuart Card* and *Peter Pirolli*. They worked on a theory of IF using the analogy between the analysis of animals searching for food and the behavior of humans collecting information online. Since then, IF has stimulated the interest of researchers and a number of works have been done these last years.

1.3 Web Information Foraging

The authors of [35] gave their vision on what will be the next biggest technological progress of the century. They presented the Wisdom Web as the new challenge of Web intelligence. On the other hand, they gave meanings to the term "wisdom", followed by a scenario illustration. They ended their article by describing the fundamental capabilities of the Wisdom Web such as the autonomic Web support, the semantics and meta-knowledge.

Within the same context, *Liu* in his talk at *IJCAI'03* [33], has suggested new directions for research in the new field of Web Intelligence (WI) that has emerged a decade ago from artificial intelligence and information technology. The main goal of WI is to develop theories and technologies towards using optimally the connectivity of the Web. It deals then with three major levels:

- The communication and security protocols level.
- The interface level providing an interface for human interaction.
- The knowledge level where the Web is considered rather as a data/knowledge base.

The speaker then predicted that the paradigm that will raise interest in the research domain for the next era is the Wisdom Web. He described the objectives of the Wisdom Web and its most important concerns. He went on with two recent challenges, which are respectively information foraging and ubiquitous communities. A few works have been done on human Web surfing by [26] and [15]. The speaker then exposed the work he did with *Zhang* on a model for information foraging agents on the Web and which is described next.

The study held in [59] presented a cognitive framework for information foraging based on the SNIF-ACT model [52] hybridized with some other models used for Information Foraging. The proposal aimed to facilitate the user's choices and preferences on a website by simulating her/his online buying behavior. The framework was tested on a watch store

website. However, the authors did not give enough details in this article, especially on how the user's preferences and interests were modeled. Furthermore, no experimental results were shown in order to validate the proposal and evaluate its performance.

In [26] the author described in a clear and nice manner the recent concepts, methods and applications of text mining. The chapter on Web mining contains a rich documentation of notions that concern the new developments of Web technologies such as Web topology, sites popularity, sites ranking and propagation of metadata to co-links. The authors of [15] introduced the concept of social information foraging and its understanding. They explored models for social IF and focused on the importance of the benefits of cooperative foraging.

The authors in [41] proposed a tool-based approach for designing and evaluating Web based information environments. They pointed out to two kinds of obstacles that may occur during the foraging process. The first is caused by inadequate scent emitted by proximal cues and can be detected at a Web page level, while the second can be faced at a website level and is due to flaws in the information architecture in Web-based systems. In order to tackle these problems, the authors introduced the *Information Scent Evaluator (InfoScent Evaluator)* and the *Auto Card Sorter (AutoCardSorter)*. Both tools aim to help Web designers in conceiving well-structured websites. The authors tested their system on a university department's website although they didn't perform validation tests involving real Web users to confirm the accuracy of their outcomes. Also, the authors did not talk about the scalability of their system, and whether it can run on voluminous websites or not.

Finally the study held in [5] shows the importance of information foraging by giving an interesting and concrete application to the domain of tourism. The authors proposed a model for users of a tourism website using Web usage and content mining techniques.

1.4 Information Foraging with Agents

The study held in [52] considered the Web as a semantic space and tries to predict navigational choices of Web users. A central notion called *information scent* simulating the mutual relevance between the user's goal and Web pages' content was introduced. This model is based on the concept of semantic similarity of information. The authors tested their model on a database of selected Web search tasks collected by a survey of over 2000 Web users. Then they solicited four users to perform two Web search tasks in order to compare their surfing outcomes with those of the proposed SNIF-ACT (Scent-based Navigation and Information Foraging in the ACT architecture) model. The results show that the measure of information scent is able to generate good estimations to Web user interaction by predicting which links the users would click on and also the time when they decide to leave the website. SNIF-ACT

is well described in [48, 49]. It is a production system model using information scent as a core concept to construct rules for activating and stopping the information foraging process. The prediction of navigational choices for Web users with particular goals is made using spreading activation networks that are constructed a priori with no free parameters to be estimated from user data. These networks can be constructed thanks to statistical estimates obtained from appropriately large and representative samples of the linguistic environment.

The authors of [46] designed an agent-based information foraging model to simulate the evolution of scientific domains. For this purpose, they introduce several notions like the preferences and the motivation of the scientists and also the maturity and receptivity of the scientific domains. The authors found out that there is a strong relationship between the rationality state of the population, the openness on the growth of scientific domains and the distribution of scientific domains.

A negotiating information foraging agent, which models the evolution of information needs and the negotiation process of information seeking was proposed in [61]. The authors paid particular attention to the "goal evolution" aspect of a Web user, which can be seen in their opinion as a negotiation act since the user adapts its information needs to the outcomes yielded by the system. The proposed system was not implemented yet at the publication time of the article, therefore the authors were not able to test its performance and no concrete experiment was conducted. Moreover, the user goal modeling seems complex and hard to elaborate considering that the system has to generate the goal hierarchy by starting from the less specific need of information and then trying to find more specific sub-goals.

In [35] and [34], the authors proposed an agent-based model for IF and validated it by using empirical Web log datasets. They considered Web topology, information distribution and interest profiles in building a Wisdom IF agent. They found out that the unique distribution of the agent's interests leads to regularities in Web surfing, and that Web regularities are interrelated. They also undertook an interesting study on three categories of users according to their interests and familiarity with the Web: A random user who does not have a well defined intention in surfing, a rational user with an objective but who may not be completely familiar with the Web environment, and a recurrent user who is familiar with the Web and who has a goal form Web surfing. The result was that, independently from the kinds of users, the regularities of Web surfing are the same, which means that the user's ability to predict surfing chains leading to relevant information is predominant.

Strong regularities in Web surfing were also studied in [25] from a theoretical point of view. The authors proposed a model for studying surfing behaviors, and the experiments they held showed common surfing behaviors. The study conducted by [24] showed that Web

pages are distributed over the sites according to a universal power law, which is an example of the strong regularities.

In [37], the authors noticed that the objective of a surfing user can be either browsing, purchasing or achieving a goal, and that each of these types of surfers considers different factors during the navigation task. They focused their research on the goal attainment objective and especially on contact information submission on long tail Web sites using the information foraging theory (IFT). They tested several IFT hypotheses related to goal achievement on long tail Web sites using an inductive approach to learn cognitive factors and information scent. Experiments based on clickstream data taken from small business sites were performed and they were able to determine the sought hypotheses.

Nowadays, studies involving information foraging are applied in several domains such as e-commerce [28], computer graphics [67], programming [31, 32], tourism [27, 40] and scientific knowledge [46].

1.5 Information Foraging on social networks

In the position paper in [58], the authors described some potential sense making techniques to exploit microblogging for knowledge workers. They took into consideration two tasks: (1) information foraging and active exploration (when the user is seeking for information) and (2) awareness and passive monitoring (when information is delivered to the user). The authors suggested the use of features such as social context, information filtering and sentiment analysis in order to get more effective foraging results.

The author of [68] presented a survey on the advances of information foraging theory and its adaptation to the Web and social networks. The author claimed that the researchers working on information foraging aim to pave the way for an approach that allows users to find, share and synthesize knowledge easily and more effectively. This can be done by conceiving models able to predict how the users navigate on the Web. Applying the information foraging theory on social media in order to allow the users to get credible information in an effective way is one of the most interesting new directions according to the author.

In [45] the Autoregressive Integrated Moving Average model was used in order to study the temporal dependencies that may exist in a group of information foraging users. The authors considered that the information foraging process goes through two phases: seeking and handling. Information seeking refers to search for information, while information handling refers to the incorporation of the found information into existing knowledge. The results of this study showed the presence of temporal dependencies related to information seeking and information handling. However, no such dependencies were detected between

the individual information foraging (i.e., initially held by one member) and the common information foraging (i.e., initially held by a group of members).

Lamprecht et al. [30] studied decentralized-based search using ontologies as background knowledge. They investigated ways of modeling the navigational behavior of human users in information networks using the information foraging model. *Wikipedia* was considered as a case study in this work. A set of 100 medical articles were selected in order to construct a sub medical information network. The authors also resorted to *wiki games* with the aim of extracting navigation paths from Wikipedia. Furthermore, they used four biomedical ontologies to guide the search. The results highlighted the advantages of using ontologies in information networks and showed their ability to improve the surfing process. However, the size of the information network used for experiments is relatively small as it only contains 100 nodes.

Zhang and Ackerman [69] addressed the problem of searching for expertise in social networks. The classic study on searching in social networks is the *"small world"* experiment done by *Milgram* [38] where he found that subjects from *Nebraska* could successfully send a small packet to a target person in *Boston* with the help of common acquaintances. In other terms, people from *Nebraska* were able to reach a person in *Boston* even if they don't know each other directly thanks to intermediate acquaintance chains. The majority of network search algorithms are based on this experiment and consist in browsing the network nodes until reaching the goal node. The authors of [69] divided the searching strategies on social networks into three families. According to them, Breadth First Search (BFS) and Random Walk Search (RW) algorithms belong to a general computational family. The second family that is based on the network structure includes algorithms such as Best Connected Search (BCS), Weak Tie Search (WTS), Strong Tie Search (STS), Cosine Similarity Search (CSS) and Hamming Distance Search (HDS). The third and last family relies on information similarity and incorporates the Information Scent Search (ISS) algorithm. A social network was constructed using the *Enron dataset* of exchanged emails between employees. For each employee, the authors indexed all the sent and received messages and used the resulting keywords vector as the user's expertise profile of this person. The general computational results, the social costs along with the impact of social characteristics were compared for the eight search algorithms. The authors noticed that the most solicited people on the network are those who are highly connected to other people. Also, people with a high number of in-degrees (incoming links) have a high chance of being searched during the simulation process. Moreover, people who have more social connections may have more diverse knowledge. We can remark two limitations for this work, the first one is related to the fact that the simulation network is not the complete email social network of Enron organization. It only consists

of the management level subset of the complete network, this increases the density of the network and shortens the average searching paths. The second weakness resides in the way of constructing the user expertise profile. The authors assume that the user's expertise is represented by the keywords that belong to her/his email folder, which is not always the case.

At present, research on Web information foraging is still deepening. There are many issues that have to be further investigated such as expanding the information foraging theory and adapting it to the social media context. The area of information foraging on social media is nearly blank. Tackling this issue will enhance both information availability and access within social media. Social relations, connections and other criteria are used in the social networks information foraging system that we have proposed in Chapter 5.

1.6 Discussion

From the above literature review, we remark that the area is still in an early stage and the existing papers related to IF offer new ideas on how to develop theories and technologies about IF. The theory developed by *Pirolli et al.* is an important advancement and can stimulate future works in the field. On the other hand, the agent-based model proposed by Liu et al. can be considered as the commencement of future IF technologies.

The original contribution of the present thesis consists in developing a Web Information Foraging approach based on Multi-Agent Systems. The study done in [46] shares similar concerns with our work, however, there are many differences that we summarize in the following points: The issue addressed by the authors is the evolution of scientific domains, whereas in our case we treat Web information foraging and we aim to test it on different domains including health, scientific citations and social media. Moreover, the inputs are different. In the approach proposed in [46] the considered inputs are represented by scientific domains and scientists' problems problems, while in the approach that has been proposed during the PhD reported in this thesis we have focused on exploiting the Web structure to the aim of fulfilling information needs. The outputs are also distinct as the former study produces scientific problem domains and ours finds relevant Web pages for a certain user's interests. Furthermore, in our study we model the surfing behavior of Web users and we test it on real-world datasets using our Web Information Foraging System, whereas the authors of the former article modeled the scientist behavior using a simulation toolkit.

It is also worth noticing that the cooperative dimension in our proposal is performed by software agents whereas it is carried out at the user level in social IF by [15]. Concretely, we propose a multi-agent architecture that fits the real surfing process, and that aims at keeping good performance compared to a centralized solution. For this purpose, we investigate

intelligent approaches for constructing a Web information foraging system, including *Swarm Intelligence*, *Self-interested Agents*, *Text Classification* and *Game Theory*. The whole system consists of two phases simulating groups of animals hunting including a learning and an incremental learning step. The first step yields the best locations of Web pages a user might be interested in using a fixed Web topology, while the second considers the dynamicity of the Web to update the knowledge about the Web pages location taking into account the changes of the Web. Both phases are based on multi-agent systems.

Chapter 2

A Model for Web Information Foraging

2.1 Introduction

Web Information Foraging deals with a tricky and changing environment, which is the Web. In order to design an efficient and effective Web Information Foraging system, we consider the important dimensions of the Web, especially its complex structure and its dynamicity. In this Chapter, we propose a model for Web information foraging. We aim at developing a system capable of locating Web pages containing information relevant to a certain user's interests. A modular architecture of this system is proposed, in which we distinguish two important phases. The first is a learning process, which aims to find the most relevant pages that might interest the user based on his/her information needs. This is performed on a fixed instance of the Web. The second takes into account the openness and dynamicity of the Web. It consists in an incremental learning starting from the results of the first phase and reshaping the outcomes taking into account the changes that have arisen on the Web. The whole system should offer a tool to help users getting information on the Web using information foraging.

Studies on the Web structure including hyperlinks allow the detection of the relevant Web pages that have authority on a certain topic. For instance, Web Usage statistics employs techniques to discover patterns in the log files of Web users. Meanwhile searching the Web contents aims at presenting the most relevant Web pages to specific users' needs.

The dynamicity of the Web is one of the dimensions that is perhaps the trickiest to handle. It involves at the same time, the growth of the Web volume and the changes that are constantly operated by users. In [54], the author provides statistics of the increase rate of the amount of online data indexed by Google, which passes from 5 exabytes in 2002 to 280 exabytes in 2009. This volume is expected to double every 18 months according to [70]. In [42], the authors proved that the number of new pages augments by 8% a week and in [7], the authors give a more impressive rate, which is about 7.5 pages every second. We do not lose sight

that these statistics are provided by standard search engines that are unable to index hidden contents according to [22]. This tremendous revolution the Web is undergoing makes all studies on the Web a real challenge.

The rest of the chapter is organized as follows. In section 2.2, we expose the Web navigation model we adopt in our study and which is inspired from our own practices of real-world Web surfing. In section 2.3, we present the multi-agent system architecture we propose for our Web Information Foraging System.

2.2 Web Navigation Model

The navigation model we propose in this section attempts to simulate the real-world process of Web surfing. We first give a mathematical representation of the Web, then we formulate the statement we are addressing. In the end, we describe the mathematical model we adopt for Web navigation.

Web Representation and Notations The Web can be represented by an oriented hyper-graph $H = (P, L, T)$, which is a generalization of a graph [6, 10]. P is a set of pages or nodes, L a set of links and T a set of hyper-edges or topics. Unlike in a graph, a hyper-edge in a hyper-graph is a set of nodes. Figure 2.1 illustrates an example of a hyper-graph where :

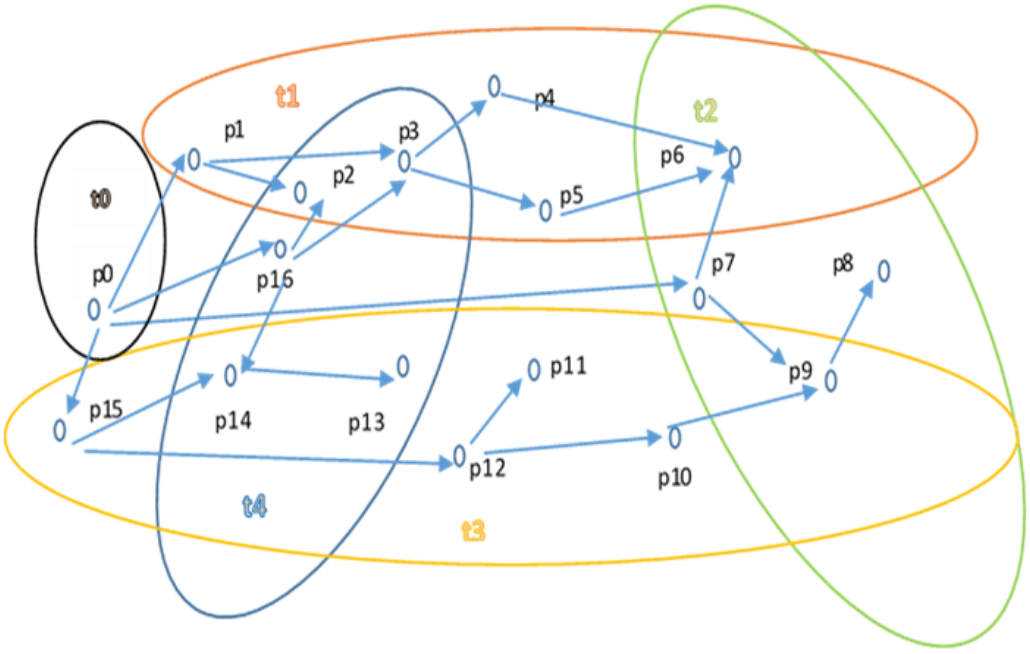

Fig. 2.1 An example of a hyper-graph H

$P = p_0, p_1, ..., p_{16}$.

$L = (p_i, p_j)$ such that $p_i \neq p_j$ and $p_i \in P$ and $p_j \in P$.

$T = t_0, t_1, t_2, t_3, t_4$.

Each topic is a subset of nodes:

$t_0 = p_0$.

$t_1 = p_1, p_2, p_3, p_4, p_5, p_6$.

$t_2 = p_6, p_7, p_8, p_9$.

$t_3 = p_9, p_{10}, p_{11}, p_{12}, p_{13}, p_{14}, p_{15}$.

$t_4 = p_2, p_3, p_{13}, p_{14}, p_{16}$.

The structure of a hyper-graph is suitable for the Web because:

- each node represents a Web page
- each edge represents a link between two pages and
- each hyper-edge represents a topic, which is a subset of P.

Problem Statement Navigating on the Web with a user's interests corresponds to visiting a branch of the hyper-graph, one node at a time starting from an initial Web page and ending with a relevant Web page, which contains information related to the user's information needs. The goal is to determine an optimal path to reach relevant Web pages for a given user's interests. For this purpose, at each click corresponding to a page p_i, the question is to find the best move to another page p_j in order to build the path.

The Model We consider three Web surfing strategies, corresponding to the three kinds of users' behaviors reported in [35] and [34]:

- A pseudo-random surfing behavior, adopted by users who don't have a clear idea about their information needs and who are not really familiar with Web surfing.
- A recurrent surfing behavior, adopted by users who are not only familiar with Web surfing but also have a precise goal with clear information needs.

- A rational surfing behavior, which concerns users who have a well defined objective but who may make some random choices while surfing due to unfamiliarity with the Web.

We propose a surfing rule for each strategy to compute the probability $P(p_i, p_j)$ of surfing from page p_i to page p_j based on the user's interests and a heuristic. When surfing, the user is guided by his/her scent that depends on his/her interest but also by a heuristic that measures the semantic similarity between the current page and the next one. We introduce then a measure called $InfoScent_i$ to estimate the information scent the user would get at a page p_i and a second one called $heur_{ij}$ that calculates the resemblance between page p_i and page p_j. Power law distributions are adopted for these quantities. Indeed, the information scent is significant for only a very few pages and is weak for many of them. In the same way, a few outgoing pages are very similar to the current page and the others are different.

Pseudo-Random Agent Strategy This strategy simulates the behavior of Web users with no strong interests in a specific topic. They roam from one Web page to another in attempt to get information on a generic topic, which leaves place to random decisions in selecting the next-level Web pages they visit. Rule 2.1 exhibits the probability of reaching a page p_j from the actual page p_i. The surfing page is drawn at random among the outgoing pages with probability $P(p_i, p_j)$, α and β are parameters allowing to weight respectively the information scent and the heuristic.

$$P(p_i, p_j) = \frac{(InfoScent_i)^{\alpha}(heur_{ij})^{\beta}}{\Sigma_{l \in N_i}[(InfoScent_l)^{\alpha}(heur_{il})^{\beta}]} \tag{2.1}$$

Recurrent Agent Strategy The recurrent strategy deals with Web users who are familiar with the Web structure and know the spots of interesting contents. Each time when they decide to forage further, they know the locations of the Web pages that closely match their interests. In this case, Rule 2.2 is considered. The system always selects the Web page, which maximizes the information scent and the heuristic value.

$$P(p_i, p_j) = \begin{cases} 1 \text{ if } p_j = argmax\{(InfoScent_i)^{\alpha}(heur_{ij})^{\beta}\} \\ 0 \text{ otherwise} \end{cases} \tag{2.2}$$

Rational Agent Strategy The rational strategy concerns users with specific topical interests in mind and they surf on the Web in order to locate the Web pages that contain information on those topics. When they reach a new Web page, they will try to decide whether or not the content matches their interests and, if not, predict which Web page at the

next level will be likely to become a more interesting one. To predict the potential next Web page they will visit, they pay attention to elements that might give a description of that page's content such as the labels of the hyperlinks.

We introduce a noise called q_0 to disturb the optimal surfing and a variable q. During the process, a value from the interval [0,1], is drawn at random for q. If it is lower than or equal to q_0, then the page with the best score is selected, otherwise another is chosen at random. Rule 2.3 models this situation.

$$\begin{aligned} &\textit{if } q \leq q_0 \\ &\quad \textit{then } P(p_i, p_j) = \begin{cases} 1 \; if \; j = argmax\{(InfoScent_i)^{\alpha}(heur_{ij})^{\beta}\} \\ 0 \; otherwise \end{cases} \\ &\quad \textit{else } P(p_i, p_j) = \frac{(InfoScent_i)^{\alpha}(heur_{ij})^{\beta}}{\Sigma_{l \in N_i}[(InfoScent_l)^{\alpha}(heur_{il})^{\beta}]} \end{aligned} \tag{2.3}$$

The first part of the rule models the fact that the user has a well defined objective from Web surfing, while the second part translates random choices s/he makes due to unfamiliarity with some websites. q_0 can be seen as the degree of rationality of the user.

Description of the variables, parameters and functions used

- V: the user's interests represented by a vector of keywords.
- q: a random variable uniformly distributed over the interval [0,1].
- $q_0 \in [0,1]$ a tunable parameter.
- Argmax: calculates the page p_j having the maximum value $(InfoScent_i)^{\alpha}(heur_{ij})^{\beta}$.
- $InfoScent_i$: the estimation of the information scent the user would get at page p_i.
- $heur_{ij}$: the similarity value between pages p_i and p_j.
- α: a parameter that controls the relative weight of the information scent.
- β: a parameter that controls the relative weight of the heuristic.
- N_i: the set of outgoing Web pages of page p_i.

- Score of a page: the semantic similarity between V and the description of the Web page (title, tags, etc).

User Information Scent Modeling The information scent the user would get at page p_i can be estimated using a quantity that increases as the user gets closer to a relevant Web page. Starting with a weak amount of information scent, each time the user moves from one page to another, a quantity relative to the quality of the outgoing page p_j is added. $InfoScent_i(t+1)$ is then computed using Formula 2.4

$$InfoScent_i(t+1) = InfoScent_i(t) + sim(V, p_j) \tag{2.4}$$

Where

- $V = (v_1, v_2, ..., v_u)$ is a vector of terms representing the user's interests.
- $p_j = (t_1, t_2, ..., t_s)$ is the description vector of Web page p_j, containing terms extracted form the title, the tags and the url of the Web page.
- $sim(V, p_j)$ is the cosine similarity between V and page p_j. It is computed with the same way as in Formula 2.5.

Heuristic The heuristic we conceive consists of the semantic similarity between the description vectors of pages p_i and p_j. It is computed using Formula 2.5.

$$heur_{ij} = sim(p_i, p_j) = \frac{\sum_{k=1}^{m} a_k b_k}{\sqrt{\sum_{k=1}^{m} a_k^2} \sqrt{\sum_{k=1}^{m} b_k^2}} \tag{2.5}$$

Where:

- $p_i = (t_{i1}, t_{i2}, ..., t_{iu})$ is the vector representing the current Web page p_i, it contains terms from the title of the page and eventually tags and synonyms.
- $p_j = (t_{j1}, t_{j2}, ..., t_{js})$ is the vector of the outgoing Web page p_j, it contains terms from the title of the page and eventually tags and synonyms.
- $sim(p_i, p_j)$ is the cosine similarity between p_i and page p_j.
- a_k and b_k are components of vector p_i and p_j respectively.

The complexity of surfing The worst case of the surfing complexity corresponds to the case where the Web structure is a fully connected graph. Starting from an initial Web page, the number of choices of the next page to visit is equal to $(n-1)$ if n is the total number of nodes or pages. To pursue the navigation, there are $(n-2)$ possible pages to select then the number of paths with a depth equal to 2 is equal to $(n-1)*(n-2)$. To go further in depth, the number of paths increases by a factor of $(n-i)$ if i is the current depth, and so on until reaching the surfing depth. Then the surfing complexity SC is expressed by Formula 2.6 where d is the surfing depth.

$$SC = (n-1)(n-2)...(n-d) \tag{2.6}$$

With the introduced measures of the information scent and the heuristic, we aim to guide the system to find relevant Web pages within a path whose length is linear in terms of the depth. This situation corresponds to the best case complexity.

2.3 A Multi-Agent Architecture for Web Information Foraging

Web information foraging (WIF) is analogous to animals hunting, where sources of information in the Web correspond to sources of food in the ecological environment. The real challenge is to understand and model the animal behavior related to natural hunting. The studies done in [20] and [21] represent a recent and rich documentation on animals hunting behaviors and pointed out the importance of cooperation group hunting. The authors claimed that a lion that hunts alone will have one successful kill every seven attempts. However, when the female lions cooperate, they catch one prey every two attempts. Each group has its own geographical territory for hunting, which may depend on many factors like weather and seasons. During winter for instance, animals find preys easily whereas in dry seasons, preys are rare. The hunting process goes through two phases:

1. The learning phase: Animals learn about the world the same way as humans do according to [21]. They use their senses to understand their environment. The group of animals also monitors its prey and contemplates its behavior in order to find the best way to attack later. Before they start hunting, they first observe the territory and try to learn the suitable places for hunting. For instance, lions know where and when their preys drink water. Hence, they wait for them at the water source at the right time. Besides the location, animals observe also:

- The kind of preys, small or big. To get enough energy to survive, the size of the predator is proportional to the one of its prey. This is not always the case but it works in general on wild animals.
- The timeline, nights or days, some preys appear only at night whereas others are busy during the day.
- The social aspects: the preys move in groups. The predator catch the most vulnerable individual such as babies.

2. The adaptation phase: Once the group learned the prerequisites of hunting, it continues observing the environment and monitor its preys so that it can adapt to changes (climate change, seasons change ...). For instance, in hot seasons, some preys migrate to another territory. Predators must change this kind of preys or eat plants instead of animals.

From these observations, we set the following hypotheses for our Web information foraging approach:

1. Consider a Multi-Agent System (MAS) to simulate a group of animals with a special interest to the cooperation feature, which is used by the animals in hunting and to maintain their survival.

2. The geographical territory of a group of animals corresponds to a topic, since the Web covers numerous topics.

3. Like the hunting group of animals, the MAS goes through two steps:

   a. The learning of relevant information sources, which satisfy the user's interests.

   b. The incremental learning of information sources to adapt to Web changes. This treatment simulates the adaptation of the group of animals to the changes of their environment.

Multi-agent systems have been widely used to address information access problems such as in [29]. However, this is the first time they are considered in Web information foraging since the majority of the previous studies used one single agent and not a whole multi-agent system. We consider in our proposal a group of artificial agents with a developed sense of cooperation in order to forage information on the Web. The group of agents simulates the process of a group of animals when hunting (looking for food). Figure 2.2 shows the architecture of the Web information system we propose. The two parts of the architecture are described in the next subsections.

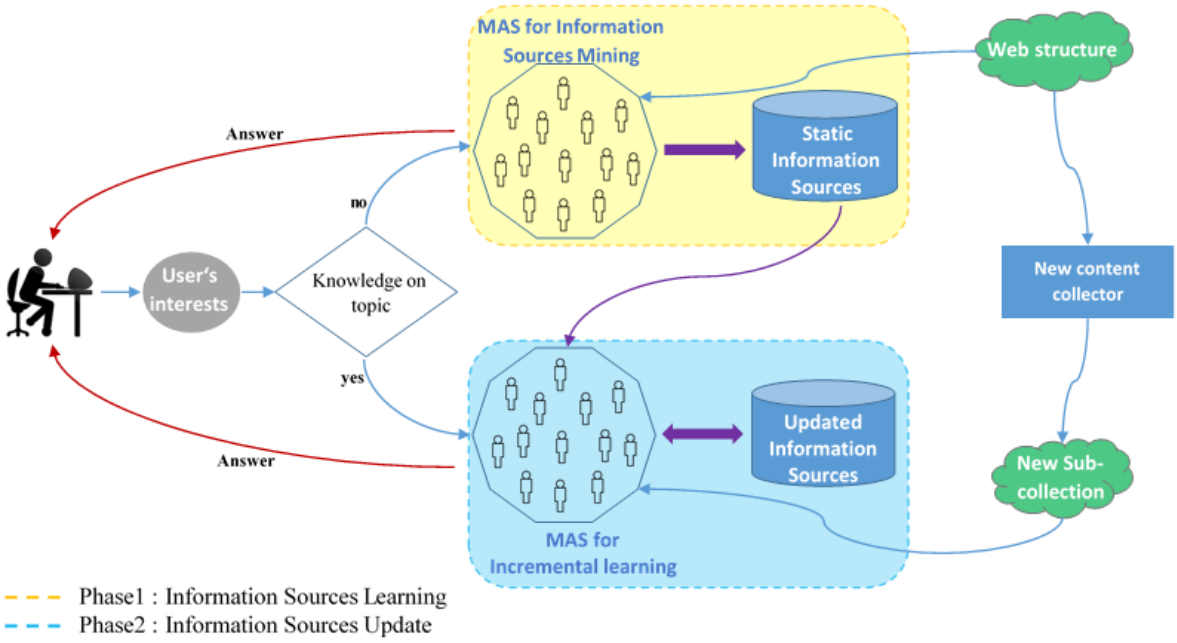

Fig. 2.2 A Multi-agent Architecture for Web Information Foraging

2.3.1 Information Sources Learning.

The task to undertake here is to investigate the Web for finding surfing paths leading to relevant Web pages in order to optimize the foraging. A multi-agent system is launched to seek potential information sources. The approach ranks the leading Web pages according to their relevance to the user in a database called "Static Information Sources" that will be exploited in the next step. The process of Information Sources mining takes into account the Web topology and the user's interests.

The search will start from an initial Web page, then guided by the foraging strategy, the agents try to find surfing paths leading to relevant pages. This phase can be implemented with various kinds of multi-agent systems. The different implementations we propose are detailed in the following chapters.

2.3.2 Information Sources Update.

The Web is a dynamic environment that keeps changing and evolving. This phase aims at making our system able to monitor the evolution of the Web thanks to the knowledge it previously acquired. It corresponds to the adaptation phase of the animal hunting process.

Unlike the learning phase that is executed just once for a given user's topical interest (the first time when the user shows interest in that topic), this phase will be triggered each time

the user shows interest in the same topic again or in a similar topic. In this case, two actions are planned: update the Information Sources database and explore the new Web pages. The first operation consists in:

- Eliminating from the database populated the learning phase, the Web pages that do not exist anymore.
- Updating the relevance score for those that got modified but are still present on the Web.
- Ranking the foraged Web pages in a decreasing order of their relevance score.

For the second one, the agents will proceed to an incremental learning to update the Information Sources at each change occurring in the Web. Concretely, the multi-agent system is launched on the new sub Web structure and the new Information Sources are integrated in the database with respect to relevance ranking.

2.3.3 Architecture for multiple information providers

The system architecture of Figure 2.2 considers only one Website but a Web information foraging system should be able to browse the entire Web for better results. In order to extend the developed system, we should integrate other information providers (websites) in the process. If we have k providers for instance, the Web information foraging program is executed on each provider then the achieved information sources are merged together in order to come up with one global database sorted in a decreasing order of the information sources score. The whole framework is depicted in Figure 2.3. Of course, the architecture remains still extensible for other new providers.

2.4 Discussion

We introduced in this chapter a new model aimed at designing a Web information foraging system. The considered Web structure was presented along with the proposed Web navigation model. Three different Web surfing strategies were studied and formalized based on the literature review we did in Chapter 1. Furthermore, we gave the general architecture of our Web information foraging system based on multi-agents technology. The architecture covers Web information foraging in both static and dynamic environments. Finally, we proposed an extension of the general architecture in order to make it able to support multiple information providers.

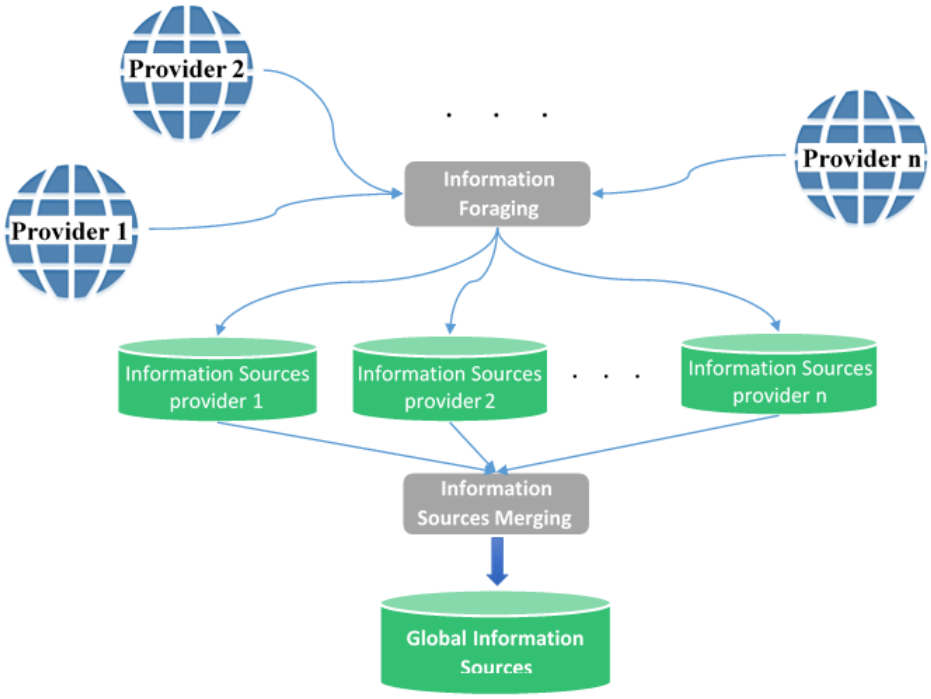

Fig. 2.3 A Multiple information providers Architecture for Web Information Foraging

The next chapter will describe the implementation of the proposed Web information foraging system using Swarm Intelligence approaches.

Chapter 3

Swarm Intelligence for Web Information Foraging: Application to the Medical Domain

3.1 Introduction

Swarm Intelligence is a bio-inspired field, which aims to mimic the behavior of groups of animals (ants, bees, fishes, bats, etc.) to solve computational problems more effectively and efficiently. The idea is grounded on the collective behavior of interacting natural/artificial agents. Swarm intelligence is based on a population of agents, which interact locally with each other and with their environment as well. Each agent operates on its local environment according to simple rules without knowing about the global effect of its actions. Swarm intelligence approaches do not only reproduce the natural behavior of swarms but they can also make use of additional domain knowledge to change the behavior of the swarm in order to solve the addressed problem.

One of the advantages of Swarm Intelligence resides in the decentralized nature of its individuals. There is no clear leader in the swarm and all agents are involved and get benefits from each other. They interact and collaborate following simple local rules, which allows the system to be self-organized. Swarm intelligence is considered as a field of evolutionary computation, since it is population-based. The members of the population interact among themselves and statistically improve themselves over generations, which makes them able to find good solutions for the problem [47].

We investigate Swarm Intelligence approaches in this chapter with the aim of proposing an implementation to our Web Information Foraging System following the model presented

in Chapter 2. First, we pay particular attention to *Bee Swarm Optimization (BSO)*, with which we implement the multi-agent system of foraging agents. As an attempt to enhance the performance of the system, we present in the second part of this chapter a different implementation using *Ant Colony Optimization (ACO)* hybridized with *Tabu Search*.

The rest of the Chapter is organized as follows. The implementation of the Web information foraging system using *Bee Swarm Optimization* is detailed in section 3.2. An improvement of this implementation is presented in section 3.3, where *Ant Colony Optimization*, which is more appropriate for the foraging process is considered as a swarm intelligence algorithm. Finally, in section 3.4 we discuss the main achievements of this part of the study as well as the benefits and the drawbacks of using swarm intelligence to implement Web information foraging.

3.2 Web pages Mining and Ranking using Bee Swarm Optimization

Bee Swarm Optimization (BSO) is an intelligent method for problem solving. It implements a multi-agent system that simulates the natural behavior of bees when searching for food.

3.2.1 Bee Swarm Optimization

Bee Swarm Optimization (BSO) was designed as a nature and bio-inspired approach to solve complex problems [18], it showed its efficiency through its adaptation to several problems such as information retrieval in [17]. BSO framework is based on the experiments held by *Seeley* in [55], who showed that when two food sources are equidistant from the hive, the bees exploit the one containing the highest food concentration. The bee exploiting the source of greater concentration makes a dance, which is proportional in strength to the source concentration. Figure 3.1 gives a better idea of the experiment.

In the Bee Swarm Optimization algorithm, an artificial bee called *BeeInit* represents a potential solution to the addressed problem and initializes a reference solution called *Sref*. A search space, namely *SearchArea* is then determined from *Sref* using a well-defined diversification generator. Several mathematical generators are proposed in the literature. They consist in determining equidistant points in a large space of data. The aim of using a diversification generator here is to better explore the search space. Each bee considers one solution from *SearchArea* as a starting point from which it performs a local search. After the bees have accomplished their search, they communicate to their congeners their best results through a table called *Dance*, which simulate the dance performed by real bees. The best

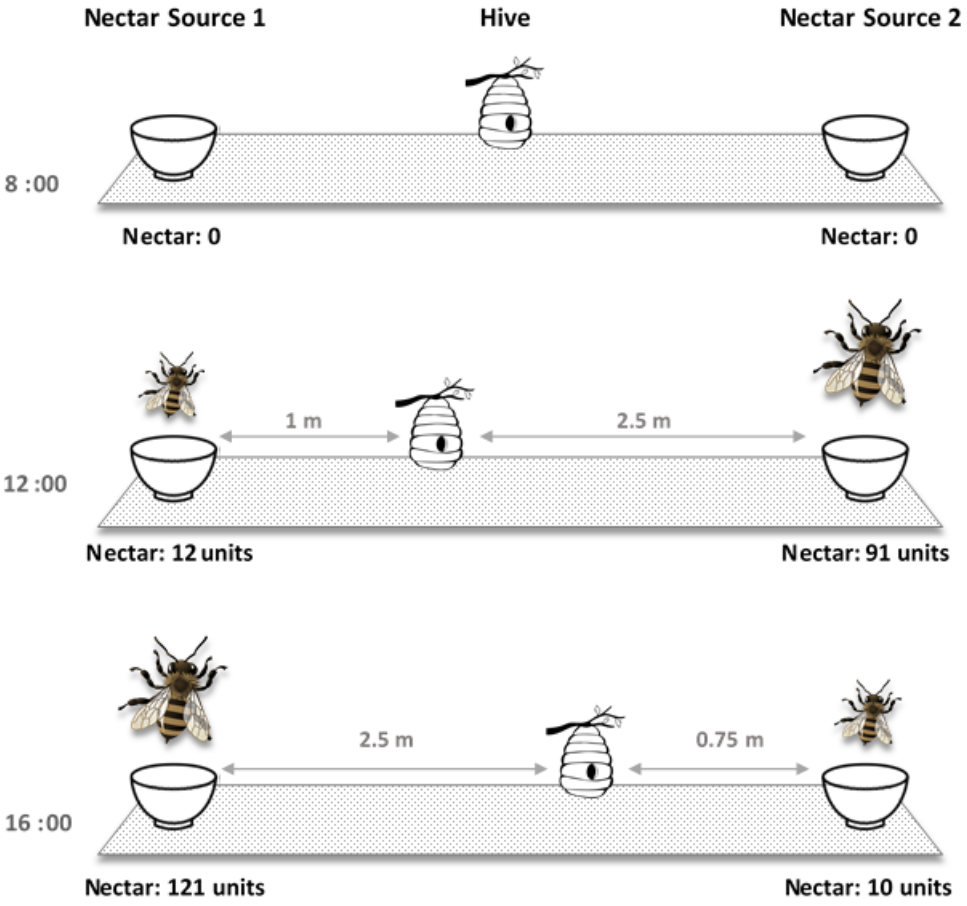

Fig. 3.1 Seeley experiment on bees food foraging

solution stored in this table is taken as the new reference solution for the next generation. The process is iterated until one of the following conditions are satisfied: the optimal solution is found, the solution quality stagnates and does not improve or the number of generations reach a certain limit determined by the physical constraints related to the machine performance. A tabu list is used to backup the reference solutions *Sref* at each iteration in order to avoid stagnation and local optima. The framework of BSO is outlined in Algorithm 1.

Algorithm 1 BSO

```
Input: a search space
Output: the near optimal solution
begin
    1. Sref:= BeeInit;
    (* a solution drawn randomly from the search space *)
    2. while stop condition is not reached do
        2.1. insert Sref in TL the Tabu List;
        2.2. determine SearchArea from Sref;
        (* use a diversification generator from the literature *)
        2.3. assign for each bee a solution of SearchArea;
        2.4. for each bee do
            2.4.1. search in the corresponding area;
            2.4.2. insert result in Dance;
        2.5. Choose the best solution from Dance and assign it to Sref;
end;
```

3.2.2 Foraging Information on the Web with BSO

In Web information foraging, the search space for the colony of bees will be a Web structure (Web graph). Recall that as we have seen in the previous chapter, a topic delimits the territory of the group of animals when doing the analogy with animals food foraging. In BSO, artificial bees encapsulate solutions, which consist in our case in surfing paths. They try to discover Web surfing paths that end with a Web page that has good relevance based on the user's interests. The adaptation of BSO to Web information foraging is called BSO-WIF and is outlined in Algorithm 2.

The first statement consists in initializing the process. It creates a surfing path of depth *d* starting form a Web page drawn at random. The depth is an empirical parameter and its value is defined during the experiments. Instruction 2 loops until a maximum number of iterations is reached or when a surfing path with a satisfactory score is found. Instruction 2.1

Algorithm 2 BSO-WIF

```
Input: A Web structure and the user's interests
Output: Surfing paths ending with a relevant Web pages
begin
    1. Sref:= BeeInit, a surfing path drawn randomly starting from
    the homepage;
    2. while stop condition is not reached do
        2.1. insert Sref in TL the Tabu List;
        2.2. searchArea = a subset of k surfing paths starting at the
        homepage and such that the first node is not connected
        to the first node of Sref;
        2.3. assign for each bee a surfing path of SearchArea;
        2.4. for each bee do
            2.4.1. search locally for a better surfing page using
            formula (2.1), (2.2) or (2.3);
            2.4.2. insert result in Dance;
        2.5. Choose the path with the highest score from Dance and
         assign it to Sref;
end;
```

consists in inserting *Sref* in the taboo list in order to avoid revisiting the same surfing paths. Instruction 2.2 computes the set of surfing paths that will be assigned to bees (instruction 2.3). It is drawn randomly from the set of outgoing links representing topics that are distant from each other, which means that they are not linked directly. Instruction 2.4.1 consists in determining a better surfing path using one of the transition rules define in Chapter 2: (2.1), (2.2) or (2.3) depending on whether a pseudo-random, a recurrent or a rational surfing strategy is adopted. Instruction 2.4.2 allows the artificial bees to store their results in table *Dance*. The surfing path with the highest score among those stored in *Dance* is assigned to *Sref* in order to reiterate the process.

3.2.2.1 Solution Encoding and Evaluation

The search space considered by the swarm of bees is the set of all possible surfing paths. The bees generate solutions and work on improving them. In our case a solution is a surfing path, which consists in a collection of connected Web pages. The artificial bees have the task of discovering surfing paths leading to potential relevant Web pages that might satisfy the user's information needs.

The evaluation of a surfing path quality is performed by applying the score function f to the last Web page on the path. The score of a Web page is the similarity between the user's interests vector and the description vector of the Web page. The score function is computed as follows:

$$f(s) = sim(V, p_j) = \frac{\sum_{k=1}^{m} a_k b_k}{\sqrt{\sum_{k=1}^{m} a_k^2}\sqrt{\sum_{k=1}^{m} b_k^2}} \tag{3.1}$$

Where:

- s : a surfing path.
- $V = (v_1, v_2, ..., v_u)$ is the vector representing the user's interests.
- $p_j = (t_1, t_2, ..., t_s)$ is the description vector of the last Web page p_j on the surfing path.
- $sim(V, p_j)$ is the cosine similarity between the vectors representing respectively the user's interests and the Web page p_j.
- a_k and b_k are components of vector V and p_j respectively.

3.2.3 Information Sources Incremental learning

The system architecture we proposed in the previous chapter includes two phases: the first one operates on a static Web structure whereas the second acts on a dynamic structure. All the results of the system interactions with the user are stored in a database called *Static Information sources*, which contains the previously foraged information sources for different topics. When the user shows interest in one of those topics again, the system makes use of the knowledge it already has on that topic thanks to the *Static Information sources* database and updates it taking into consideration the changes that have occurred on the Web since the last interaction. In other terms, instead of launching BSO-WIF on the whole Web structure, the system starts from the surfing paths foraged during the previous request. This way of processing allows to gain time and makes the system respond in real-time. The main tasks the system has to carry out is to update the information sources database by deleting the Web pages that do not exist anymore in the new version of the Web structure. The second task is to forage new surfing paths leading to relevant Web pages on the new sub-structure that has emerged since the last interaction. BSO is called to search new information sources on the new sub-structure. Here two observations can be made, the first one resides in the lapse of time that separates two consecutive foraging processes, the larger the gap is, the larger will the response time be. The second phenomenon concerns the amount of changes undergone

on the Web. The same facts as for the separation duration are deduced. When considering both parameters simultaneously we introduce the notion of changes speed. The new Web sub-substructure is determined thanks to a process that we call *Changes Collector*, which uses Web crawling algorithms [44] in order to detect and gather new created pages on the Web structure. Algorithm 3 describes the incremental learning of information sources called *IL-WIF*. The Web information foraging process on the new sub-structure yields a shorter runtime than when launching it on the whole structure.

Algorithm 3 IL-WIF

Input: A Web structure, the user's interests, the information sources database
Output: Surfing paths ending with relevant Web pages
Begin

1. **for** each Web page in the *Static Information Sources* database **do** :
   1.1. **if** the Web page does not exist anymore in the current Web structure **then** delete it from the database;
2. Call BSO-WIF for the new sub-structure;
3. Insert the Web pages returned by BSO-WIF in the database with respect to the relevance ranking;

End

3.2.4 Experimental Results

The main goal of our experiments was to address a real-word problem in a specific domain and evaluate the contribution of Web information foraging. For this purpose, we considered the domain of health and used our system in order to help patients find relevant medical articles based on their diagnosis or on some symptoms.

3.2.4.1 Description of the real-world Dataset

Extensive experiments were performed on the online medical website of the U.S. National library of Medicine called *MedlinePlus*. It includes three parts:

- Health Topics,
- Drugs and Supplements,
- Videos and Cool Tools.

It provides information on over 1000 diseases, health conditions and wellness issues. MedlinePlus' health topics are regularly reviewed, and links are daily updated. This suits

the system we propose for Web information foraging and helps us to evaluate both the static learning and the incremental learning phases.

The content of the MedlinePlus website is available in an XML format on http://www.nlm.nih.gov/medlineplus/xml.html. The XML files include Web pages related to different medical topics and are generated on a daily basis. The number of nodes was equal to 1903 on July 23 2014 and the XML file had a volume of 27 Mb. This version of MedlinePlus is used for testing the static learning phase.

Each topic is specified by a title and contains the following elements:

- a URL,
- an identifier,
- the language of the topic (English or Spanish),
- the date of its creation,
- eventually, tags specifying among others, topic synonyms and a translation to other languages,
- a full summary,
- related topics, which are internal links to similar topics
- and links to external sites.

Only links to related topics are exploited because they belong to the MedlinePlus website. External sites are ignored as they direct to pages outside the website. In order to clarify this structure, Figure 3.2 provides a fragment of MedlinePlus. The empty space is inserted for the purpose of comparing this version with the one of January 1st 2015 to show the evolution of the Website. Indeed in the latter version, there is another external site specification in the empty space.

3.2.4.2 Results for the Information Sources Learning

The whole system was implemented using Java Eclipse help System Base, version 2.0.2 on a PC with an Intel core I5-3317U Processor (1.70 GH) with 4 GB of RAM.

The results reported in this subsection concern the information sources learned from a static Web structure. Table 3.1 expands the dataset specifications along with the optimal parameters of BSO-WIF.

```
<health-topic title="Aborto" url="http://www.nlm.nih.gov/medlineplus/spanish/abortion.html"
id="2238" language="Spanish" date-created="10/31/2006">
<also-called>Aborto terapéutico</also-called>
<also-called>Interrupción del embarazo</also-called>
<full-summary>&lt;p&gt;Un aborto es un procedimiento para interrumpir un embarazo. Se utili

&lt;p&gt;La decisión de interrumpir un embarazo es muy personal. Si piensa someterse a un a
<group url="http://www.nlm.nih.gov/medlineplus/spanish/pregnancyandreproduction.html" id="2
<group url="http://www.nlm.nih.gov/medlineplus/spanish/femalereproductivesystem.html" id="4
<language-mapped-topic url="http://www.nlm.nih.gov/medlineplus/abortion.html" id="122" lang
.....
<site title="Embarazo: Qué hacer cuando su embarazo es inesperado" url="http://familydoctor
<information-category>Spanish/Español</information-category>
<information-category>Asuntos relacionados</information-category>
<organization>Academia Americana de Médicos de Familia</organization>
</site>




<site title="Ponerle fin a un embarazo" url="http://familydoctor.org/familydoctor/es/drugs-
<information-category>Spanish/Español</information-category>
<information-category>Resúmenes</information-category>
<organization>Academia Americana de Médicos de Familia</organization>
</site>
....
</health-topic>
```

Fig. 3.2 A fragment of the XML version of MedlinePlus.

Number of Web pages	1903
Dataset Size	27 Mb
Number of bees	20
Number of generations	30
Tabu list size	20
q_0	0,8
α	2
β	1
d	3

Table 3.1 Dataset specifications and empirical parameters values

Different users' interests were experimented for each category of surfing strategies. The results we focused on, are: the most relevant Web page, its URL, its score, the surfing depth and the surfing time. Tables 3.2, 3.3 and 3.4 exhibit the surfing outcomes for respectively the pseudo-random, the recurrent and the rational agent strategies. The results confirm the Web regularities concerning the surfing depth and the type of surfing strategy. Indeed, the surfing depth does not exceed a certain threshold except for cases where there is no Web pages related to the user's interests in the Website. Those cases are considered as outliers since their surfing depth values are distinctly separate from the surfing depth values of the rest of the results. One definition of outlier is any data point more than 1.5 interquartile ranges (IQRs) below the first quartile or above the third quartile [23]. We see that, for instance, in Table 3.2, the surfing depth threshold is determined by the Web page's score using the outlier formula:

$Score(outliers) : 1.5 * (Q3\_score - Q1\_score)$

Where: $Q1 = 0.75$ and $Q3 = 1$ are respectively the first and the third quartiles.

All the Web pages that have a score greater than or equal to 0.75 are acceptable. The surfing depth threshold is then equal to 11 and all the Web pages that do not comply with this constraint are considered as outliers. The surfing paths provided by the system for the users' interests "Skin, Allergies" and "Anorexia" are then considered as outliers. When the score is null such as for "Ebola", no relevant Web page for this user's interest is found.

For the recurrent agent strategy, the threshold is smaller compared to the pseudo-random strategy, which is foreseeable. Table 3.3 displays a value of 3 corresponding to a score greater than or equal to 0.5 and only the surfing path of the user's interest "Anorexia" remains an outlier. Table 3.4 shows a behavior for the rational agent, which is similar to that of the recurrent agent.

From all these three tables, we find predominant the system ability for predicting surfing paths leading to relevant Web pages. Figure 3.3 illustrates these experimental results through boxplots drawn for each surfing strategy. We observe that, in fact the recurrent and the rational agents behave the same way with respect to the navigation chain while there is a little shade for the pseudo-random agent.

Figure 3.4 provides more precision concerning the surfing behavior of the three kinds of agents. It shows that when a relevant Web page exists for a certain user's interests, the three agents behave almost the same. However, when there is no relevant page that can satisfy the user's interests, the pseudo-random agent deviates a bit from the trajectory of the other agents.

User's interests	Most relevant Web page		Score	Surfing depth	Surfing time (ms)
	Title	URL			
Pain, Abdominal	Abdominal Pain	*/abdominalpain.html	1.0	2	170
Diabetes	Diabetes	*/diabetes.html	1.0	1	168
Heart, Diseases	Heart Diseases	*/heartdiseases.html	1.0	11	197
Medicines	Medicines	*/medicine.html	1.0	5	204
Cancer	Cancer	*/cancer.html	1.0	1	178
Anemia	Anemia	*/anemia.html	1.0	2	222
Obesity	Obesity	*/obesity.html	1.0	1	210
Gastroenteritis	Gastroenteritis	*/gastroenteritis.html	1.0	4	256
Skin, Allergies	Skin Cancer	*/skincancer.html	0.5	12	1737
MCI	Mild Cognitive Impairment	*/mildcognitiveimpairment .html	1.0	2	240
X-Rays	X-Rays	*/xrays.html	1.0	3	325
Recovery, surgery	After Surgery	*/aftersurgery.html	1.0	3	171
Pimples	Acne	*/acne.html	1.0	1	445
Ebola	/	/	0.0	/	/
Hypersensitivity	Allergy	*/allergy.html	1.0	2	211
Poor, Blood, Iron	Anemia	*/anemia.html	1.0	6	169
Dog, Bites	Animal Bites	*/animalbites.html	1.0	6	217
H5N1	Bird Flu	*/birdflu.html	1.0	2	233
High, blood, pressure, medicines	High Blood Pressure	*/highbloodpressure.html	0.75	2	1655
Anorexia	Eating Disorders	*/eatingdisorders.html	0.5	20	7167
Dermatitis	Rashes	*/rashes.html	1.0	2	264
Myopia	Refractive Errors	*/refractiveerrors.html	0.5	4	255

* : http://www.nlm.nih.gov/medlineplus

Table 3.2 Experimental Results for different users' interests for the random agent strategy

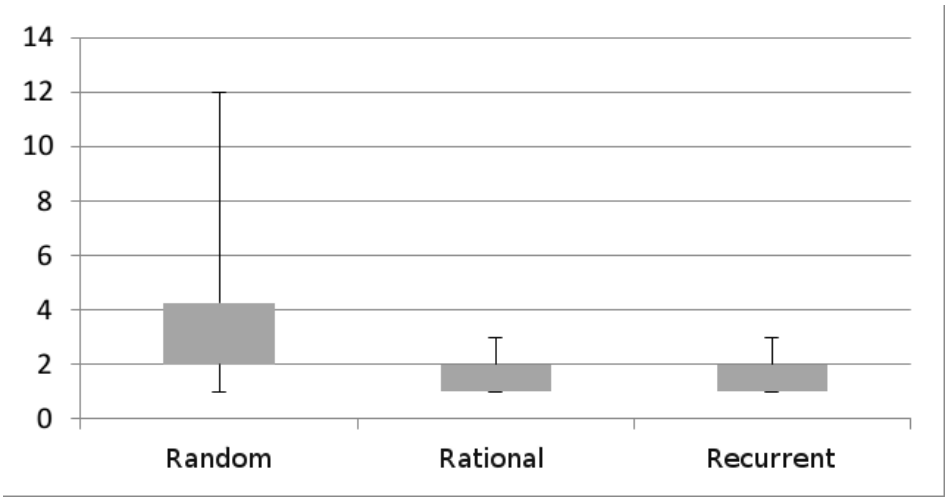

Fig. 3.3 Boxplots for surfing depth for the three experienced strategies.

User's interests	Most relevant Web page		Score	Surfing depth	Surfing time (ms)
	Title	URL			
Pain, Abdominal	Abdominal Pain	*/abdominalpain.html	1.0	2	122
Diabetes	Diabetes	*/diabetes.html	1.0	1	140
Heart, Diseases	Heart Diseases	*/heartdiseases.html	1.0	1	155
Medicines	Medicines	*/medicine.html	1.0	1	136
Cancer	Cancer	*/cancer.html	1.0	2	138
Anemia	Anemia	*/anemia.html	1.0	3	135
Obesity	Obesity	*/obesity.html	1.0	1	140
Gastroenteritis	Gastroenteritis	*/gastroenteritis.html	1.0	1	219
Skin, Allergies	Skin Conditions	*/skinconditions.html	0.5	2	844
MCI	Mild Cognitive Impairment	*/mildcognitiveimpairment .html	1.0	2	275
X-Rays	X-Rays	*/xrays.html	1.0	2	256
Recovery, surgery	After Surgery	*/aftersurgery.html	1.0	2	191
Pimples	Acne	*/acne.html	1.0	1	301
Ebola	/	/	0.0	/	/
Hypersensitivity	Allergy	*/allergy.html	1.0	1	98
Poor, Blood, Iron	Anemia	*/anemia.html	1.0	2	152
Dog, Bites	Animal Bites	*/animalbites.html	1.0	1	138
H5N1	Bird Flu	*/birdflu.html	1.0	1	140
High, blood, pressure, medicines	High Blood Pressure	*/highbloodpressure.html	0.75	3	1291
Anorexia	Body Weight	*/bodyweight.html	0.15	20	5012
Dermatitis	Rashes	*/rashes.html	1.0	1	188
Myopia	Refractive Errors	*/refractiveerrors.html	0.5	1	259

* : http://www.nlm.nih.gov/medlineplus

Table 3.3 Experimental Results for different users' interests for the recurrent agent strategy

User's interests	Most relevant Web page		Score	Surfing depth	Surfing time (ms)
	Title	URL			
Pain, Abdominal	Abdominal Pain	*/abdominalpain.html	1.0	2	104
Diabetes	Diabetes	*/diabetes.html	1.0	1	133
Heart, Diseases	Heart Diseases	*/heartdiseases.html	1.0	1	195
Medicines	Medicines	*/medicine.html	1.0	1	162
Cancer	Cancer	*/cancer.html	1.0	2	128
Anemia	Anemia	*/anemia.html	1.0	1	193
Obesity	Obesity	*/obesity.html	1.0	1	240
Gastroenteritis	Gastroenteritis	*/gastroenteritis.html	1.0	2	224
Skin, Allergies	Skin Conditions	*/skinconditions.html	0.5	2	1375
MCI	Mild Cognitive Impairment	*/mildcognitiveimpairment .html	1.0	2	244
X-Rays	X-Rays	*/xrays.html	1.0	3	305
Recovery, surgery	After Surgery	*/aftersurgery.html	1.0	3	203
Pimples	Acne	*/acne.html	1.0	2	465
Ebola	/	/	0.0	/	/
Hypersensitivity	Allergy	*/allergy.html	1.0	1	137
Poor, Blood, Iron	Anemia	*/anemia.html	1.0	3	161
Dog, Bites	Animal Bites	*/animalbites.html	1.0	1	123
H5N1	Bird Flu	*/birdflu.html	1.0	1	128
High, blood, pressure, medicines	High Blood Pressure	*/highbloodpressure.html	0.75	1	1394
Anorexia	Body Weight	*/bodyweight.html	0.15	20	6904
Dermatitis	Rashes	*/rashes.html	1.0	1	197
Myopia	Refractive Errors	*/refractiveerrors.html	0.5	1	264

* : http://www.nlm.nih.gov/medlineplus

Table 3.4 Experimental Results for different users' interests for the rational agent strategy

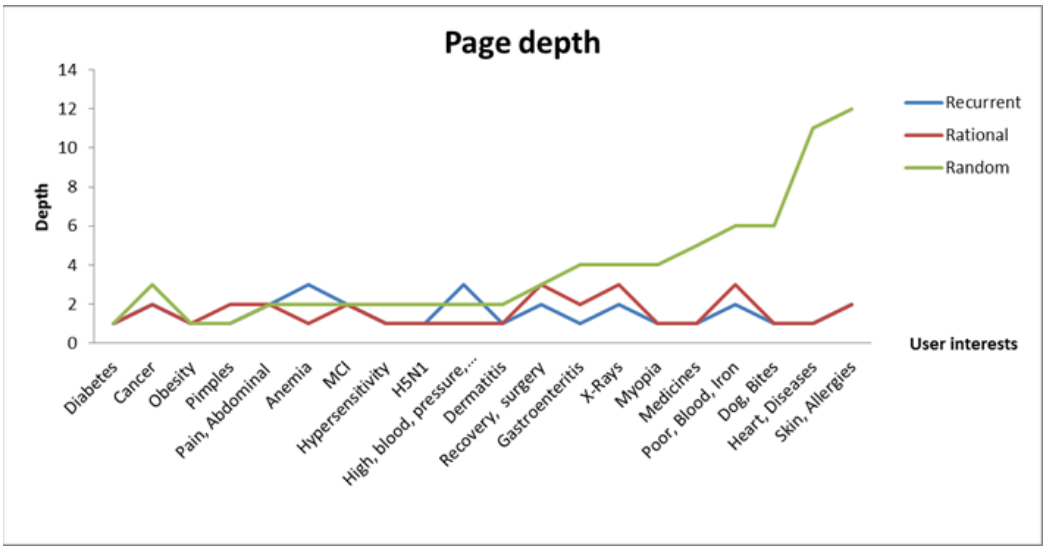

Fig. 3.4 Surfing depths for recurrent, rational and random strategies.

Concerning the surfing time, Figure 3.5 shows the average response time for the pseudo-random agent, the rational agent and the recurrent agent respectively. We can summarize the results in three points:

- the response time is very short for the three strategies and complies to the real-time constraint.
- the order of magnitude for the surfing time is the same for the three strategies.
- the recurrent agent is faster than the rational agent, which is at its turn faster than the random agent.

These results are foreseeable and confirm the power of our proposed Web surfing model.

Figure 3.6 gives a schematic view of the surfing time for the three strategies and for all the experienced users' interests. It emphasizes through the time dimension the discussed results about outliers and the agents behaviors. The response time is very short and doesn't exceed the 450 milliseconds except for the users interests "Skin, Allergies" and "High, blood, pressure, medicines".

3.2.4.3 Experiments on the Incremental Learning of Information Sources

The following experiments deal with the update of the information sources following the Web evolution. To perform such tests, we consider the configuration of *MedlinePlus* of the 1^{st} January 2015 and we compare the Web information foraging results with those of the

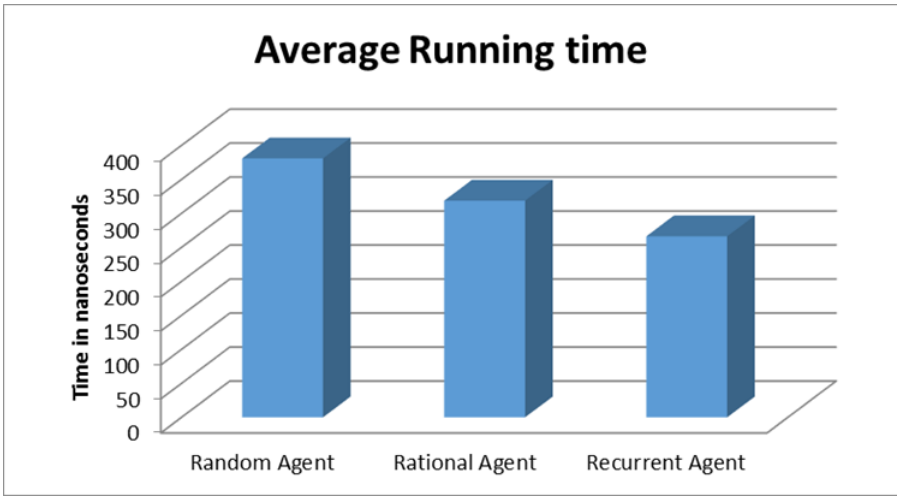

Fig. 3.5 Average response time for the pseudo-random, the rational and the recurrent surfing strategies.

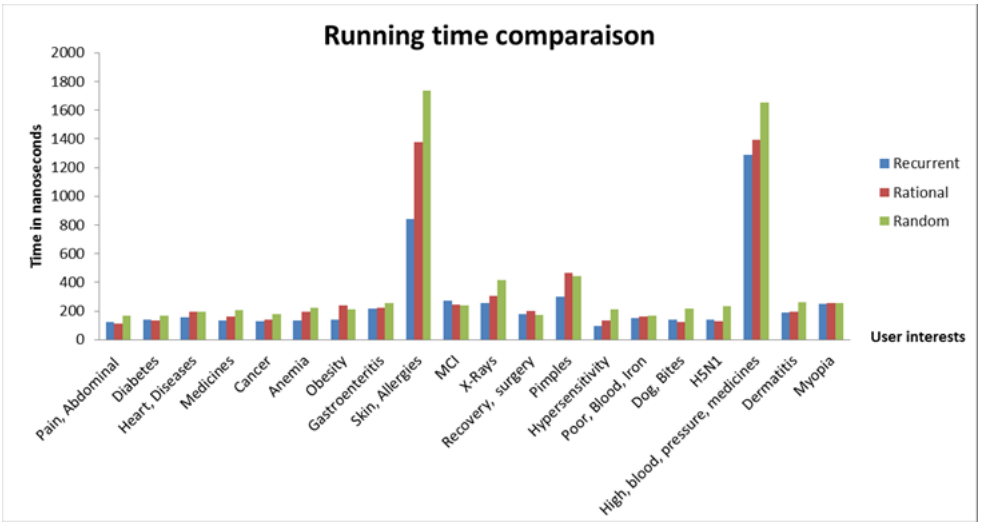

Fig. 3.6 Response time comparison between the pseudo-random, the rational and the recurrent surfing strategies for each user's interests.

previous configuration using the same users' interests. An example of *MedlinePlus* fragment for that day is shown in Figure 3.7.

```
<health-topic title="Aborto" url="http://www.nlm.nih.gov/medlineplus/spanish/abortion.html"
id="2238" language="Spanish" date-created="10/31/2006">
<also-called>Aborto terapéutico</also-called>
<also-called>Interrupción del embarazo</also-called>
<full-summary>&lt;p&gt;Un aborto es un procedimiento para interrumpir un embarazo. Se utili

&lt;p&gt;La decisión de interrumpir un embarazo es muy personal. Si piensa someterse a un a
<group url="http://www.nlm.nih.gov/medlineplus/spanish/pregnancyandreproduction.html" id="2
<group url="http://www.nlm.nih.gov/medlineplus/spanish/femalereproductivesystem.html" id="4
<language-mapped-topic url="http://www.nlm.nih.gov/medlineplus/abortion.html" id="122" lang
......
<site title="Embarazo: Qué hacer cuando su embarazo es inesperado" url="http://familydocto
<information-category>Spanish/Español</information-category>
<information-category>Asuntos relacionados</information-category>
<organization>Academia Americana de Médicos de Familia</organization>
</site>
<site title="Opciones en el embarazo: La crianza de un bebé, la adopción y el aborto" url="
<information-category>Spanish/Español</information-category>
<information-category>Asuntos relacionados</information-category>
<organization>Colegio Estadounidense de Obstetras y Ginecólogos</organization>
</site>
<site title="Ponerle fin a un embarazo" url="http://familydoctor.org/familydoctor/es/drugs-
<information-category>Spanish/Español</information-category>
<information-category>Resúmenes</information-category>
<organization>Academia Americana de Médicos de Familia</organization>
</site>
.....
</health-topic>
```

Fig. 3.7 A fragment of January 1st 2015 version of MedlinePlus.

The fragment was indeed modified during the lapse between the 23^{rd} July 2014 and the 1^{st} January 2015. The comparison with Figure 3.2 shows the introduction of a new site having as title *"Opciones en el embarazo: La crianza de un bebé, la adopcion y el aborto"*, written in Spanish.

In general, the last version underwent several changes such as adding new Web pages (topics) and editing some of the old ones. For instance the total number of pages went up from 1903 on July 24 2014 to 1909 on January 1st 2015. The XML file size also increased to reach 27.2 Mb in the last version. The rest of the changes were related to the Web pages' content and especially to the external links, a few of them disappeared and the majority were newly added. The external links correspond to the scientific production of articles and translate the constant evolution of the Web but since they do not belong to *MedlinePlus*, we could not experiment them. The multiple information providers architecture could handle this issue but with other structured Websites, which are unavailable right now.

New Web pages concerning *Cervical Cancer*, *Prostate Cancer Screening* and *Ebola*, which provoked a lot of noise in the media during the period of this study, were created as internal pages during that short interval of time. In order to observe the benefit of the incremental learning phase, we should consider a long period of time for *MedlinePlus*. For websites such as those dedicated for news for instance, the evolution is more important and the incremental learning phase will be really a windfall.

The execution of the incremental learning algorithm for Web information foraging, namely IL-WIF produces the results shown in Table 3.5 for the pseudo-random agent strategy. We find unnecessary to show numerical results for the two other strategies since we have shown through the first series of experiments that the three agents have similar behaviors. Table 3.5 shows that the response time of the incremental learning algorithm is very short and the results are provided in real-time even for the new information sources like the Web page about "Ebola" where it is equal to 1 millisecond.

In order to see the benefit of the incremental learning, the static learning for the version of November 1, 2014 and for the pseudo-random agent strategy was executed to compare both approaches, Table 3.6 exhibits the results. The first positive outcome is that the incremental learning finds the same Web pages (information sources) as the static learning. The second more important point is that the runtime of the incremental learning is almost null whereas for the static learning it takes longer.

3.3 Web pages Mining and Ranking using Ant Colony Optimization and Tabu Search

3.3.1 Ant Colony Optimization

Ant colony optimization (ACO) is a population based swarm intelligence approach that can be used to find approximate solutions to complex optimization problems. In ACO, a set of software agents representing artificial ants search for good solutions to a given problem by simulating the behavior of natural ants while searching for food. The ants move from the nest to a food source and when reaching the latter, they alert their congeners by means of a stigmergic communication asking them for help to transport the food to the nest. According to ethologists, this communication is performed thanks to the pheromone the ants deposit on the ground to orient their congeners toward the trail containing an important amount of food. Figure 3.8 illustrate the ants's food foraging process using pheromone as a means of communication.

User's interests	Most relevant Web page		Score	Surfing depth	Surfing time (ms)
	Title	URL			
Pain, Abdominal	Abdominal Pain	*/abdominalpain.html	1.0	0	0
Diabetes	Diabetes	*/diabetes.html	1.0	0	0
Heart, Diseases	Heart Diseases	*/heartdiseases.html	1.0	0	0
Medicines	Medicines	*/medicine.html	1.0	0	0
Cancer	Cancer	*/cancer.html	1.0	0	0
Anemia	Anemia	*/anemia.html	1.0	0	0
Obesity	Obesity	*/obesity.html	1.0	0	0
Gastroenteritis	Gastroenteritis	*/gastroenteritis.html	1.0	0	0
Skin, Allergies	Skin Conditions	*/skinconditions.html	0.5	0	0
MCI	Mild Cognitive Impairment	*/mildcognitiveimpairment .html	1.0	0	0
X-Rays	X-Rays	*/xrays.html	1.0	0	0
Recovery, surgery	After Surgery	*/aftersurgery.html	1.0	0	0
Pimples	Acne	*/acne.html	1.0	0	0
Ebola	Ebola	*/ebola.html	1.0	1	1
Hypersensitivity	Allergy	*/allergy.html	1.0	0	0
Poor, Blood, Iron	Anemia	*/anemia.html	1.0	0	0
Dog, Bites	Animal Bites	*/animalbites.html	1.0	0	0
H5N1	Bird Flu	*/birdflu.html	1.0	0	0
High, blood, pressure, medicines	High Blood Pressure	*/highbloodpressure.html	0.75	0	0
Anorexia	Body Weight	*/bodyweight.html	0.15	0	0
Dermatitis	Rashes	*/rashes.html	1.0	0	0
Myopia	Refractive Errors	*/refractiveerrors.html	0.5	0	0

* : http://www.nlm.nih.gov/medlineplus

Table 3.5 IL-WIF Results on MedlinePlus version of November 1st 2014 for the pseudo-random agent strategy

User's interests	Most relevant Web page		Score	Surfing depth	Surfing time (ms)
	Title	URL			
Pain, Abdominal	Abdominal Pain	*/abdominalpain.html	1.0	1	173
Diabetes	Diabetes	*/diabetes.html	1.0	1	173
Heart, Diseases	Heart Diseases	*/heartdiseases.html	1.0	4	149
Medicines	Medicines	*/medicine.html	1.0	3	155
Cancer	Cancer	*/cancer.html	1.0	2	168
Anemia	Anemia	*/anemia.html	1.0	5	194
Obesity	Obesity	*/obesity.html	1.0	2	196
Gastroenteritis	Gastroenteritis	*/gastroenteritis.html	1.0	3	227
Skin, Allergies	Skin Conditions	*/skinconditions.html	0.5	6	2128
MCI	Mild Cognitive Impairment	*/mildcognitiveimpairment .html	1.0	5	291
X-Rays	X-Rays	*/xrays.html	1.0	4	491
Recovery, surgery	After Surgery	*/aftersurgery.html	1.0	2	225
Pimples	Acne	*/acne.html	1.0	1	311
Ebola	Ebola	*/ebola.html	1.0	3	298
Hypersensitivity	Allergy	*/allergy.html	1.0	2	215
Poor, Blood, Iron	Anemia	*/anemia.html	1.0	5	192
Dog, Bites	Animal Bites	*/animalbites.html	1.0	2	188
H5N1	Bird Flu	*/birdflu.html	1.0	4	227
High, blood, pressure, medicines	High Blood Pressure	*/highbloodpressure.html	0.75	2	1685
Anorexia	Body Weight	*/bodyweight.html	0.15	20	5709
Dermatitis	Rashes	*/rashes.html	1.0	2	290
Myopia	Refractive Errors	*/refractiveerrors.html	0.5	2	283

* : http://www.nlm.nih.gov/medlineplus

Table 3.6 Static learning Results for MedlinePlus version of November 1st 2014 for the pseudo-random agent strategy

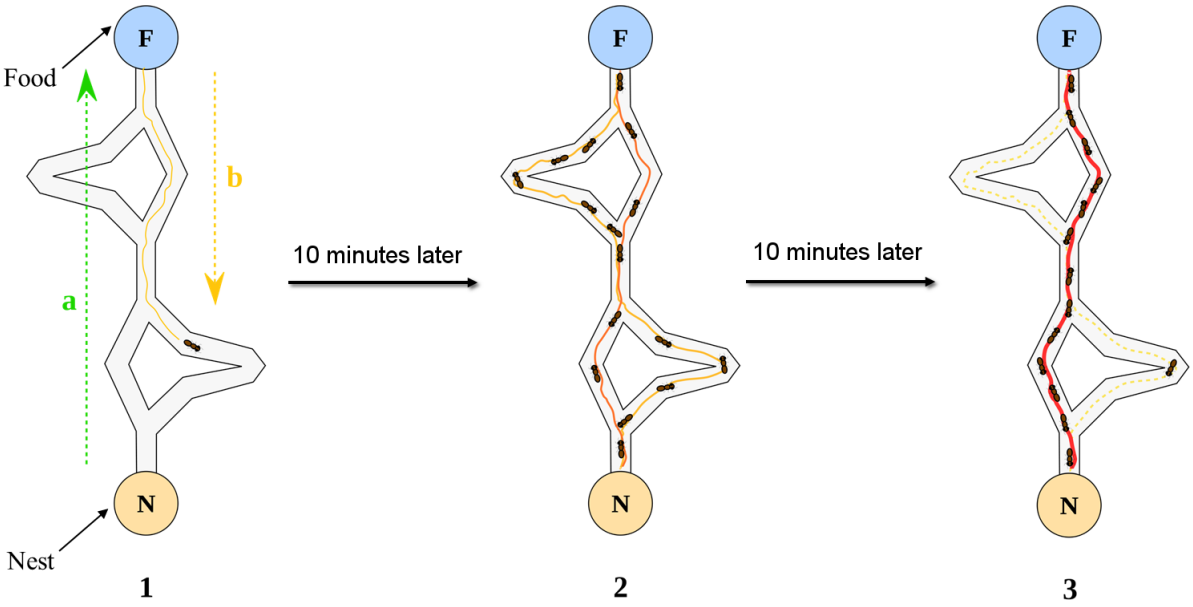

Fig. 3.8 Ants food foraging using pheromone.

Ant System [16], outlined in Algorithm 4, was the first proposed ACO algorithm. First, the pheromone initialization is proceeded by setting the values to a low number for all the ants. Each ant will then build a solution, which it will evaluate. After that, based on the evaluation outcomes the ant will perform an update to its pheromone value. At the end of each iteration the best found solution will be stored and then compared with the best global solution found during the previous iterations. Finally the best global solution is updated if a better solution is found and an offline update of the pheromone is performed for all the ants.

Algorithm 4 Ant System

```
1: pheromone initialization;
2: for i=1 to MaxIter do
3:     for k=1 to NbAnts do
4:         build a solution (S_k);
5:         evaluate (S_k);
6:         apply online delayed update();
7:     end for
8:     determine the best solution of the iteration();
9:     Apply offline-update of pheromone();
10: end for
11: return (best overall solution);
```

3.3.1.1 Pheromone table and probabilistic decision rules

The ant algorithm includes several ant generations, each generation is composed of *NbAnts* ants. Each artificial ant starts building a solution from an initial state *i* generated randomly. Recall that a solution is constructed by moving through the states of the problem using a stochastic process. The ant chooses a new state *j* from the current state's neighborhood N_i, with a probability computed by Formula 3.2:

$$P(i,j) = \frac{phero[j]}{\sum_{l \in N_i} phero[l]} \tag{3.2}$$

To each state *i* is assigned an amount of pheromone denoted by *phero[i]*. The pheromone information is initialized with a very small value in order to simulate the fact that initially the real ants deposit a very small amount of pheromone on the ground when starting their space exploration. Two structures are needed to compute the ant algorithm, a table named *Phero* to store the pheromone amount yielded by the ants each time they build a solution and a table called *sol* to save the best solution found by each ant. *phero[k]* corresponds to the pheromone amount associated to the solution found by ant *k* and *sol[k]* is the best solution determined by ant *k*. The tables are updated at each generation of ants. Besides, two variables namely *best* and *bestsol* are used to save respectively the best solution found during the current generation and the best global solution computed since the beginning of the process. During the foraging, the pheromone amount will be computed and associated to each solution found by the ants.

3.3.1.2 Updating the pheromone

The strategies of updating the pheromone simulate the evaporation of natural pheromone followed by the production of this chemical substance. The phenomenon evaporation phenomenon gives rise to rule (3.3) where the empirical parameter ρ belongs to the interval $[0,1]$ and indicates the evaporation rate. Pheromone evaporation prevents from premature convergence. An online delayed pheromone update is performed at each generation of ants, the added pheromone is calculated for each state of the solution according to rule (3.4). It is a delayed update because the pheromone assigned to a state is not updated until the ant determines the solution. For the offline pheromone update, rule (3.5) is applied. Recall that *bestsol* is the global best solution found during the previous iterations and *best* is the best solution of the current iteration.

$$phero[i] = (1-\rho) * phero[i] \tag{3.3}$$

$$phero[i] = phero[i] + \rho * f(s) \tag{3.4}$$

$$phero[k] = phero[k] + \rho * f(bestsol)/f(best) \tag{3.5}$$

Different ACO algorithms will be used to implement the three Web surfing strategies introduced previously in Chapter 2. The pseudo-random agent will be simulated using Ant System (AS), the rational agent using Ant Colony System (ACS) and the recurrent agent will be implemented by Optimal Ant Colony System (OACS) described below respectively.

3.3.2 Simulating the Pseudo-random agent using Ant System (AS)

We present here the adaptation of the Ant System algorithm to Web information foraging in order to simulate the pseudo-random user surfing behavior. As we have seen previously, the pseudo-random user is not really familiar with the web and doesn't have a well defined goal to achieve from surfing. This kind of users are only guided by their scent or motivation while surfing. We notice that the ants' pheromone in ACO is analogous to the information scent, which allows the user to move from one Web page to another during the surfing process. Therefore, we adapt the formula presented in subsection 3.3.1.1 to Web information foraging. Formula 3.6 defines the probability of moving from a Web page p_i to one of its outgoing Web pages p_j following the pseudo-random surfing strategy.

$$P(p_i, p_j) = \frac{phero[j]}{\sum_{p_l \in N_i} phero[l]} \tag{3.6}$$

In AS, each artificial ant is guided only by the pheromone and will simulate the behavior of the pseudo-random user. The adapted version of Ant System to Web information foraging (AS-WIF) is outlined in Algorithm (5).

3.3.2.1 Building and improving a solution

Each ant performs the task of exploring the best surfing path leading to relevant information in a sub-collection of the Web structure. The method designed for this purpose is described through Algorithm 6. Each ant merely initializes a solution with pages of probability P defined in Formula (3.6). The neighborhood of a page p_i is a set of Web pages that are connected to p_i via a hyperlink. Pages that are from the same neighborhood generally share some related topics.

Algorithm 5 AS-WIF

Input: a Web structure G; user's interests;
Output: bestsol, a surfing path leading to a relevant Web page;

procedure AS-WIF
  for i=1 to m **do** phero[i]= 0.1; ▷ pheromone initialization
  end for
  select at random a solution s from G; ▷ a surfing path namely s
  best := bestsol := s;
  for i=1 to MaxIter **do**
    for k=1 to NbAnts **do**
      sol[k] := build_AS();
      update the online pheromone using Formulas (3.3) and (3.4);
      if $f(sol[k]) > f(best)$ **then** then best := sol[k]; ▷ f is the evaluation function (3.1)
      end if
    end for
    if $f(best) > f(bestsol)$ **then** bestsol := best;
    end if
    apply offline-update of pheromone using Formula (3.5);
  end for
  return (bestsol);
end procedure

Algorithm 6 Build_AS

Input: a Web structure G;
Output: best_s : a surfing path with leading to a Web page;

procedure BUILD_AS(var s)
  build a surfing path s with Web pages chosen using Formula (3.6);
  $best\_s := s$;
  generate at random a number r from $[0,1]$;
  for each page p_i in s **do**
    compute $P = P(p_i, p_j)$ using Formula (3.6); ▷ for every p_j in the neighborhood of p_i
    if $P > r$ **then**
      $sol := tabu\_search(p_j)$;
      if $f(sol) > f(best\_s)$ **then** $best\_s := sol$;
      end if
    end if
  end for
  return(best_s)
end procedure

After its construction, each solution undergoes an improvement of its quality by applying the procedure *tabu_search* (Algorithm 7). Tabu search undertakes a search for a solution starting from each Web page of the surfing path i.e. the solution to be improved. The best solution of all the sought paths is kept in *best_s*. The intensification phase starts when the number of iterations without improving the solution quality reaches some limit. The quality of the solution is computed using Formula (3.1) presented in subsection 3.2.2.1.

After applying the intensification strategy, a diversification technique is launched by choosing the less recently used moves and thus directing the search to new regions of the space. In order to set the stop condition we need a variable namely *no-improve*, which informs about the number of successive iterations without improvement of *best_s*. The variable *max-no-improve* is the maximum number of iterations without improvement before starting the intensification process. The procedure *tabu_search* outlined below calls both the procedure *neighbor(p_i)* that returns the best neighbor of p_i *(according to Formula 3.6, 3.7 or 3.8 depending on the surfing strategy* which is not tabu nor satisfies the aspiration criterion and the procedure *update-t-length()*, which updates the tabu list length.

Algorithm 7 Tabu_search

Input: p_i: Web page;
Output: best_s: the best surfing path;

```
1:  procedure TABU_SEARCH(var p_i : Webpage)
2:      add p_i to best_s;
3:      no-improve = 0;
4:      while (no-improve < max-no-improve) do
5:          p_i := neighbor(p_i) using Formula (3.6);
6:          update-t-length();
7:          if (f(p_i) > f(best_s)) then add p_i to best_s;
8:          else
9:              no-improve =: no-improve + 1;
10:         end if
11:     end while
12:     if (no-improve = max-no-improve) then
13:         best_s := Intensification (p_i);
14:         best_s := diversification (best_s);
15:     end if
16:     return (best_s);
17: end procedure
```

3.3.3 Simulating the Rational Agent using Ant Colony System (ACS)

The second designed ACO algorithm is the Ant Colony System for Web Information Foraging (ACS-WIF). Its main difference with the previous one resides in the decision rules used when moving from one Web page to another and the procedure of building solutions which is called build_ACS (Algorithm 8). While in AS the ants were guided exclusively by the pheromone, in ACS we introduce a new heuristic that brings a knowledge on our problem which is Web IF and will help the ants to take better moves. This new ants behavior will be a simulation of the rational agent which is more familiar with Web surfing than the pseudo-random one. The Web familiarity of the agent is translated by the introduction of a heuristic function, which is the same heuristic we used in the Bee Swarm Optimization approach in Section 3.2. Recall that it measures the similarity that may exist between two Web pages.

Algorithm 8 Build_ACS

Input: a Web structure G;
Output: best_s : a surfing path leading to a relevant Web page;

```
1:  procedure BUILD_ACS
2:      build a surfing path s with Web pages chosen using Formula (3.7);
3:      best_s := s;
4:      generate a random variable q;
5:      if (q ≤ q0) then
6:          for each page pi in s do
7:              sol := tabu_search(pi);
8:              if ( f(sol) > f(best_s) ) then
9:                  best_s := sol;
10:             end if
11:         end for
12:     else
13:         for each page pi in s do
14:             compute probability P = P(pi, pj) by rule (3.7);
15:             generate a random value r ∈ [0, 1];
16:             if (P > r) then
17:                 sol := tabu_search(pi);
18:                 if ( f(sol) > f(best_s) ) then
19:                     best_s := sol;
20:                 end if
21:             end if
22:         end for
23:     end if
24:     return (best_s)
25: end procedure
```

We introduce a tunable parameter q_0, which represents the degree of rationality of the agent. The Web surfing probabilities are defined using Formula 3.7 for the rational agent.

$$\begin{aligned} &\textit{if } q \leq q_0 \\ &\quad \textit{then } P(p_i, p_j) = \begin{cases} 1 \text{ if } p_j = argmax(phero[j]^{\alpha}(heur_{ij})^{\beta}) \ \textit{ for } p_j \in N_i \\ 0 \text{ else} \end{cases} \end{aligned} \tag{3.7}$$

else

$$P(p_i, p_j) = \frac{phero[j]^{\alpha} \ (heur_{ij})^{\beta}}{\Sigma_{p_l \in N_i} phero[l]^{\alpha} \ (heur_{il})^{\beta}}$$

The probability $P(p_i, p_j)$ in (3.7) is computed using the quantity of pheromone and the heuristic function. α and β are empirical parameters that control respectively the importance of these two components. This probability is used to select a neighbor of a Web page both for *Build_ACS()* and *Tabu_search()*.

In other words, the ant decides stochastically to consider the best solution found in the neighborhoods of the solutions being treated during the current iteration when $q \leq q_0$, otherwise a Web page is drawn at random.

3.3.4 Simulating the Recurrent Agent using Optimal Ant Colony System (OACS)

Optimal Ant Colony System is a special variant of ACS that doesn't contain a noise parameter (q_0) which means that the artificial ants will always choose the Web page with the highest score among the others during the foraging process. Each ant in OACS simulates the behavior of the recurrent agent that is familiar with the Web and has a well defined goal from surfing.

The main difference with the two previous algorithms resides in the solution building method (Algorithm 9) and the decision rule outlined in Formula 3.8 which is used to select new Web pages in both *Build_OACS()* and *Tabu_search()* and .

$$P(p_i, p_j) = \begin{cases} 1 \textit{ if } p_j = argmax\{(phero[j])^{\alpha}(heur_{ij})^{\beta}\} \ \textit{ for all } p_j \in N_i \\ 0 \textit{ otherwise} \end{cases} \tag{3.8}$$

Algorithm 9 Build_OACS

Input: A Web structure G;
Output: best_s : a surfing path leading to a relevant Web page;

```
1: procedure BUILD_OACS
2:     build a surfing path s with pages chosen using Formula (3.8);
3:     best_s := s;
4:     for each page p_i in s do
5:         sol := tabu_search(p_i);
6:         if ( f(sol) > f(best_s) ) then
7:             best_s := sol;
8:         end if
9:     end for
10:    return (best_s)
11: end procedure
```

3.3.5 Information Sources Incremental Learning

Similar to what we did with BSO, instead of launching ACS-WIF on the whole Web structure every time, the system first checks if it already has some knowledge about the user's interest and if it is the case it tries to exploit that knowledge. This way of processing allows to gain time and optimize the foraging. Algorithm 10 is a second version of the incremental learning using ACS-WIF.

Algorithm 10 IL-WIF

Input: a Web structure, the user's interests, the static information sources database;
Output: Surfing paths leading to relevant Web pages;

```
1: procedure IL-WIF
2:     for each Web page in the database do
3:         if the Web page does not exist anymore in the Web structure then suppress it
       from the database;                          ▷ update the information sources database
4:         introduce the new pages in the Web structure;
5:         call ACS-WIF for the new sub-structure;
6:         insert the Web pages returned by ACS-WIF in the database with respect to the
       relevance ranking.
7:     end for
8: end procedure
```

3.3.6 Introducing the user's profile

In order to get more accurate and adequate results for each user depending on her/his need of information, we make use of the concept of user profile. The main purpose behind using user

User	User Profile		
	Interest	Preferred language	Period
User1	Asthma, respiration diseases	English	Recent
User2	Ebola	Spanish	All time
User3	Body weight	English, Spanish	Old

Table 3.7 Example of different users' profiles

profiles is to adapt the foraging outcomes to the user preferences. We took into consideration three aspects to model the user's profiles:

- The user's interests: it represents the different topics and domains that the user is interested in.
- The user's preferred languages: it describes the languages spoken by the user.
- The period: represents the time of publication of the information (Web page). The users can specify if they want to get recent or old publications on a certain topic.

The implementation details for each component of the user's profile are given as follows:

a) The Interest: The surfing process will highly depend on the user's interests as they represent the user's information needs. We implement the user's interests using the bag-of-words model.

b) The Preferred Languages: A list of strings is used to express the languages that the user can understand. A Content-based filtering is then applied on the information foraging outcomes in order to select Web pages that are written in one of the preferred languages of the user.

c) The Period: This parameter will most likely have an impact on the raking of the Web pages. The users can precise whether they want the new or the old Web pages to be displayed first, if the period is not specified the program will not take the creation date of the Web pages into consideration during the ranking process.

Table 3.7 shows three different users' profiles. For example, the first user who is a native English speaker wants to get the recent publications on Asthma and respiration diseases.

3.3.7 Experimental Results

Extensive experiments were performed on *MedlinePlus* the website of the U.S. National library of Medicine, which was described in subsection 3.2.4.1.

3.3.7.1 Results for the Information Sources Learning

The whole system was implemented using Java Eclipse help System Base, version 2.0.2 on a PC with an Intel core I5-3317U Processor (1.70 GH) with 4 GB of RAM.

The results of this subsection concern the information sources learned from a static Web. The Optimal parameters that we used for ACO-WIF are the following:

- Number of ants : 25
- Maximum number of generations : 20
- Tabu list size: 30
- q_0 : 0.8
- α : 2
- β :1

Then different users' profiles were experimented for each category of surfing strategies. The results we focused on, are: the Web page, its URL, its score, the surfing depth and the surfing time presented in milliseconds. Tables 3.8 and 3.9 exhibit the encoding of the users profiles' and the surfing results respectively.

These codes are used in table 3.10 which presents the surfing outcomes for respectively the pseudo-random, the recurrent and the rational agent strategies. The results once again confirm the Web regularities concerning the surfing depth and the type of surfing strategy. As seen with BSO, the surfing depth does not exceed a threshold except for cases where the system does not find relevant Web pages to satisfy the user's interests. All the Web pages that have a score greater than or equal to 0.37 are acceptable. The threshold is then equal to this number and all the Web pages that do not comply with this constraint are considered as outliers. Anorexia" is then considered as an outlier. When the score is null such as for "Ebola", no relevant Web pages for this user's interests is found.

Figure 3.9 illustrates these experimental results through a radar graph drawn for each strategy. We observe that, in fact recurrent and rational agents behave the same way with respect to navigation chain whereas the pseudo-random agent deviates a little bit and occupies the biggest area in the graph. The empirical surfing depth is quite adequate with the results reported in [25] using statistical approach and the outcomes of [34] using and agent-based method, where they both determine that surfing paths are usually short.

Figure 3.10 provides more precision concerning the surfing behavior of the three kinds of agents. The example of profile $P22$ yields a score of 0.15 which is an outlier because it is

Code	**Profile**		
	User's Interests	**Language**	**Time**
P1	Angina	English	All time
P2	Blood, Pressure	English	All time
P3	Paralysis	English	All time
P4	Influenza	English	All time
P5	Bulimia	English	All time
P6	Phobia	English	All time
P7	Asthma	English	All time
P8	Migraine	English	All time
P9	Pain, Back	English	All time
P10	Radiotherapy	English	All time
P11	Myopia	English	All time
P12	Skin, Allergies	English	All time
P13	MCI	English	All time
P14	Recovery, surgery	English	All time
P15	Pimples	English	All time
P16	Ebola	English	All time
P17	Hypersensitivity	English	All time
P18	Poor, Blood, Iron	English	All time
P19	Dog, Bites	English	All time
P20	H5N1	English	All time
P21	High, blood, pressure, medicines	English	All time
P22	Anorexia	English	All time

Table 3.8 The different users' profiles used in the experiments

Code	Web page	
	Title	URL
A1	Angina	*/medlineplus/angina.html
A2	Low Blood Pressure	*/medlineplus/lowbloodpressure.html
A3	Paralysis	*/medlineplus/paralysis.html
A4	Flu	*/medlineplus/flu.html
A5	Eating Disorders	*/medlineplus/eatingdisorders.html
A6	Phobias	*/medlineplus/phobias.html
A7	Asthma	*/medlineplus/asthma.html
A8	Migraine	*/medlineplus/migraine.html
A9	Back Pain	*/medlineplus/backpain.html
A10	Radiation Therapy	*/medlineplus/radiationtherapy.html
A11	Refractive Errors	*/medlineplus/refractiveerrors.html
A12	Skin Cancer	*/medlineplus/skincancer.html
A13	Mild Cognitive Impairment	*/medlineplus/mildcognitiveimpairment.html
A14	After Surgery	*/medlineplus/aftersurgery.html
A15	Acne	*/medlineplus/acne.html
A16	ebola	*/medlineplus/ebola.htm
A17	Allergy	*/medlineplus/allergy.html
A18	Anemia	*/medlineplus/anemia.html
A19	Animal Bites	*/medlineplus/animalbites.html
A20	Bird Flu	*/medlineplus/birdflu.html
A21	High Blood Pressure	*/medlineplus/highbloodpressure.html
A22	Eating Disorders	*/medlineplus/eatingdisorders.html
A23	Body Weight	*/medlineplus/bodyweight.html
A24	Skin Conditions	*/medlineplus/skinconditions.html

* : http://www.nlm.nih.gov/

Table 3.9 Outcomes Encoding

Profile	**Random Agent**				**Recurrent Agent**				**Rational Agent**			
	auth.	score	depth	time	auth.	score	depth	time	auth.	score	depth	time
P1	A1	1.0	1	255	A1	1.0	1	206	A1	1.0	1	220
P2	A2	0.66	3	1342	A2	0.66	1	521	A2	0.66	1	791
P3	A3	1.0	1	213	A3	1.0	1	198	A3	1.0	1	210
P4	A4	1.0	5	216	A4	1.0	1	108	A4	1.0	1	129
P5	A5	1.0	2	298	A5	1.0	1	180	A5	1.0	1	183
P6	A6	1.0	5	282	A6	1.0	1	222	A6	1.0	2	294
P7	A7	1.0	2	129	A7	1.0	1	102	A7	1.0	1	127
P8	A8	1.0	2	373	A8	1.0	1	299	A8	1.0	2	347
P9	A9	1.0	3	152	A9	1.0	1	116	A9	1.0	1	125
P10	A10	1.0	6	407	A10	1.0	2	185	A10	1.0	3	218
P11	A11	0.5	4	255	A11	0.5	1	259	A11	0.5	1	261
P12	A12	0.5	12	1692	A24	0.5	2	840	A24	0.5	2	1335
P13	A13	1.0	2	241	A13	1.0	2	271	A13	1.0	2	242
P14	A14	1.0	3	174	A14	1.0	2	152	A14	1.0	3	170
P15	A15	1.0	1	440	A15	1.0	1	299	A15	1.0	2	434
P16	/	0.0	/	/	/	0.0	/	/	/	0.0	/	/
P17	A17	1.0	2	210	A17	1.0	1	98	A17	1.0	1	136
P18	A18	1.0	6	170	A18	1.0	2	144	A18	1.0	3	163
P19	A19	1.0	6	215	A19	1.0	1	133	A19	1.0	1	122
P20	A20	1.0	2	233	A20	1.0	1	140	A20	1.0	1	130
P21	A21	0.75	2	1653	A21	0.75	2	1282	A21	0.75	1	1394
P22	A22	0.15	20	7163	A23	0.15	20	5001	A23	0.15	20	6901

Table 3.10 Experimental Results for different users' profiles for the three surfing strategies

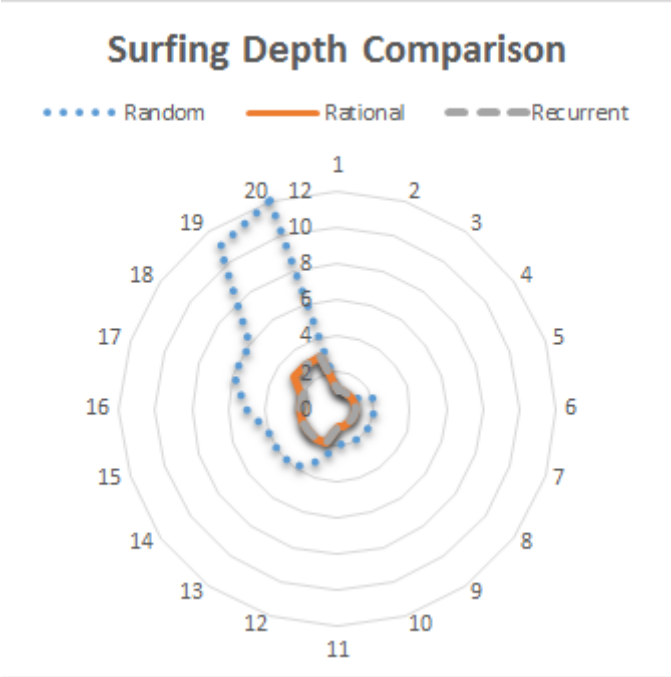

Fig. 3.9 Surfing depth radar graph for the three experienced strategies.

less than the threshold 0.37. This situation can correspond for the animal hunting to the case of starvation, when animals do not find preys and end up eating plants instead.

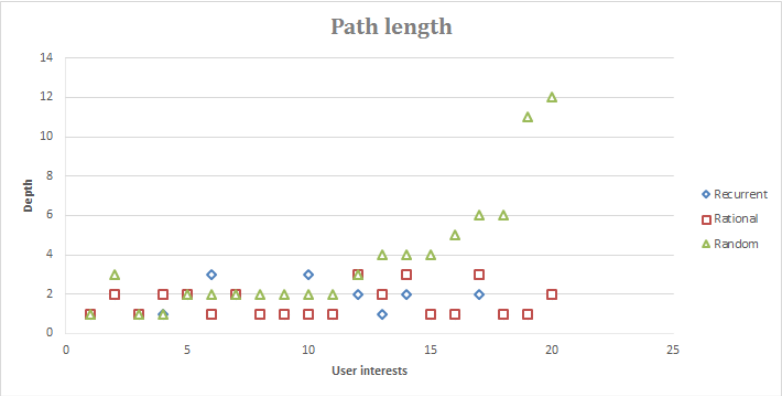

Fig. 3.10 Surfing depths comparison for recurrent, rational and random strategies.

Concerning the surfing time, Figure 3.11 shows the average runtime measured in milliseconds for respectively, the pseudo-random agent, the rational agent and the recurrent agent.

Figure 3.12 gives a schematic view of the surfing time for the three strategies and for all the experienced users' interests. It emphasizes once more in a clear way through the time dimension, the discussed results about the depth threshold, the outliers and agents behaviors.

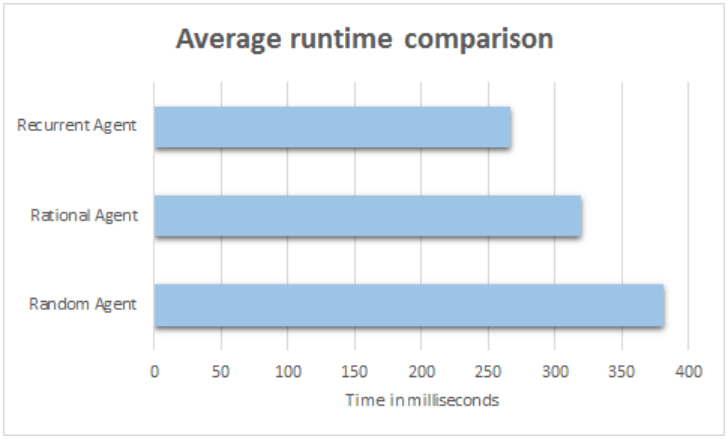

Fig. 3.11 Time means for random, rational and recurrent strategies.

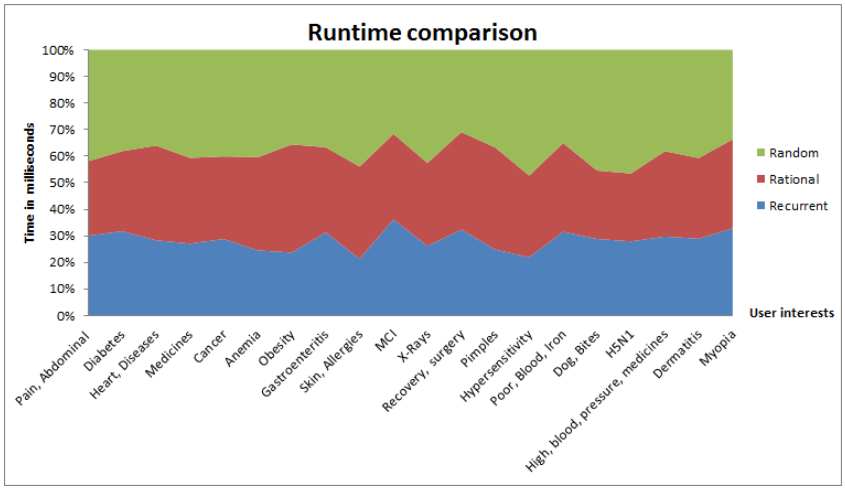

Fig. 3.12 Comparison of respond time for random, rational and recurrent strategies for each user's interests.

Profile	Incremental Learning				Static Learning			
	Auth.	Score	depth	time(ms)	Auth.	Score	depth	time(ms)
P1	A1	1.0	0	0	A1	1.0	1	266
P2	A2	0.66	0	0	A2	0.66	3	1381
P3	A3	1.0	0	0	A3	1.0	1	215
P4	A4	1.0	0	0	A4	1.0	5	222
P5	A5	1.0	0	0	A5	1.0	2	296
P6	A6	1.0	0	0	A6	1.0	5	284
P7	A7	1.0	0	0	A7	1.0	2	141
P8	A8	1.0	0	0	A8	1.0	2	389
P9	A9	1.0	0	0	A9	1.0	3	162
P10	A10	1.0	0	0	A10	1.0	6	405
P11	A11	0.5	0	0	A11	0.5	4	255
P12	A12	0.5	0	0	A24	0.5	6	2146
P13	A13	1.0	0	0	A13	1.0	5	292
P14	A14	1.0	0	0	A14	1.0	2	225
P15	A15	1.0	0	0	A15	1.0	1	310
P16	A16	1.0	1	1	A16	1.0	3	300
P17	A17	1.0	0	0	A17	1.0	2	210
P18	A18	1.0	0	0	A18	1.0	5	190
P19	A19	1.0	0	0	A19	1.0	2	194
P20	A20	1.0	0	0	A20	1.0	4	229
P21	A21	0.75	0	0	A21	0.75	2	1679
P22	A22	0.15	0	0	A23	0.15	20	5713

Table 3.11 Comparison between the Incremental Learning and the Static Learning results using ACO

3.3.7.2 Experiments on Information Sources Incremental learning

The execution of the incremental learning algorithm for Web information foraging *IL-WIF2* on the 1st of January 2015 version of *MedlinePlus* produces the results shown in Table 3.11 for the pseudo-random agent strategy. In order to evaluate the achieved improvements with the incremental learning, the static learning for the pseudo-random agent strategy was executed on the version of November 1, 2014 of *MedlinePlus* to compare both approaches, Table 3.11 exhibits the results.

3.3.7.3 Validation of the model

In order to validate our model, we experiment real surfing performed by the help of students similar to what we did with BSO. We assigned different user's interests to 20 students, then

User's interests	Real users' outcomes		Recurrent Agent's outcomes	
	Surfing Depth	Web page	Surfing Depth	Web page
Renal, disease	3	Kidney Diseases	1	Kidney Diseases
Cat, Scratch	4	Cat Scratch Disease	4	Animal Bites
Childhood, Immunization	2	Childhood Immunization	1	Childhood Immunization
Pollen, allergy	3	Hay Fever	1	Hay Fever
Fractures	2	Fractures	1	Fractures
Endocrine	3	Endocrine Diseases	4	Endocrine Diseases
Brain, attack	3	Stroke	1	Stroke
CVA	5	Stroke	1	Stroke
Walking, Problems	2	Walking Problems	1	Walking, Problems
Conjunctivitis	3	Pinkeye	2	Pinkeye

Table 3.12 Results comparison between real users and our Recurrent Agent

we asked each student to find a relevant Web page on the July version of MedlinePlus. The same users' profiles was used during the tests with the language set to *English* and the period to *All time*. Table 3.12 illustrates the results of the conducted experiment and shows the difference between the outcomes of the students' surfing and those of our system using the recurrent surfing strategy. The students' surfing depth was calculated as the mean of their results while the last Web page on the surfing path was defined by taking the most frequent ending page.

We observe that the last Web pages on the surfing path and the surfing depths found by the students and our program are generally similar. However, when we deal with complex and ambiguous users' interests our system is more effective as it finds shorter surfing paths ending with relevant Web pages . It can be explained by the fact that the real users may have a relatively limited knowledge on medical terminology in contrast to our system thanks to the health topics synonyms provided by *MedlinePlus*. For example, they may not know that *pollen allergy* is also called *hay fever* or that *CVA* is the abbreviation for Cerebrovascular Accident. We also notice that our system is more efficient and the response time is substantially shorter than the surfing time of real users.

3.3.7.4 ACO versus BSO for Web Information Foraging

In this part we compare the application of ACO and BSO to Web information foraging. For this purpose, we test the two approaches on the same set of users' interests.

Figure 3.13 represents the execution time comparison between ACO and BSO, it shows the ability of ACO to response in a shorter time than BSO. Figure 3.14 illustrates the average scores achieved by both approaches. The figure shows that ACO algorithms were able to achieve a better solution quality compared to BSO one.

We can conclude that ACO is more effective and more efficient than BSO for Web information foraging. This is due to the fact that ants' behavior is more appropriate for Web navigation. In fact, in ACO algorithms the ants build their solutions starting from an initial state and moving to other states following certain rules. This is exactly what real users do when they start surfing form an initial Web page and then they move to other Web pages constructing a surfing path. However, in BSO bees consider a solution and try to improve it, which corresponds to taking an already constructed surfing path and working on make it better. Furthermore, as stated previously ACO algorithms are better adapted to work on graph structures.

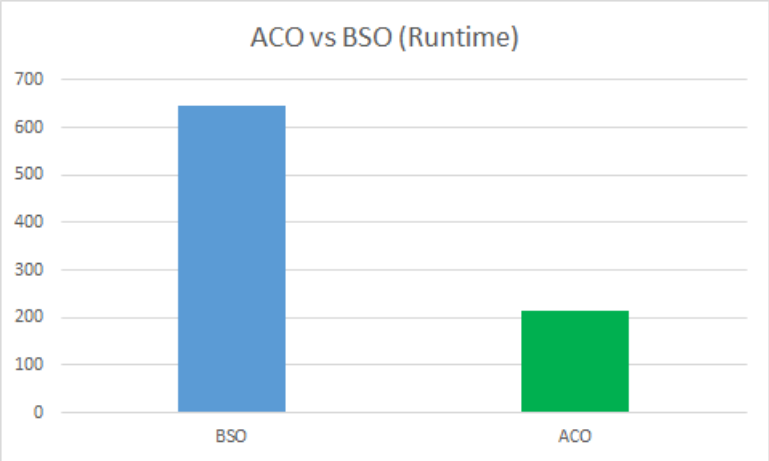

Fig. 3.13 Comparison between the efficiency of ACO and BSO.

3.4 Discussion

We proposed in this chapter two implementations for our Web information foraging system based on Swarm Intelligence approaches. Unlike previous works that used a single agent surfing model, our approach which is inspired from nature and biological psychology, adopts an analogy with animals groups hunting, which simulates real-world surfing. The second original aspect of our proposal is the use of a real-word medical website for the experiments instead of an artificial generated dataset [34].

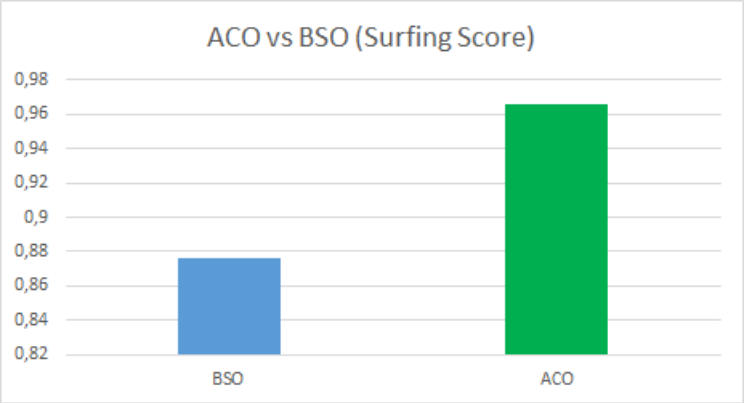

Fig. 3.14 Comparison between the effectiveness of ACO and BSO.

On the other hand, we considered several dimensions of the Web and the most important one, its openness. The system architecture we designed includes two phases simulating the animal hunting for finding foods. These steps are performed consecutively and translate efficiently the hunting process of a group of animals. The first phase consists in learning locations of the most relevant Web pages that might interest the user. It corresponds to the learning process of interesting regions an animal might use for hunting. From the implementation point of view, the system performs learning process on a static Web structure. The second phase focuses on the openness of the Web and uses the results of the first phase to initialize an incremental learning, which consists in updating the most relevant Web pages taking into account the dynamic changes of the Web.

We implemented the system using two different Swarm Intelligence approaches. The first one is *Bee Swarm Optimization* and the second one is a hybridization between *Ant Colony Optimization* and *Tabu Search*. Both approaches were applied to the domain of health and were tested on the U.S. national library of medicine. The work we undertook includes several parts, among them:

- The implementation of BSO;
- The implementation of ACO;
- The hybridization of ACO with Tabu Search;
- The adaptation of BSO and ACO to the information sources learning and detection;
- The incremental learning.

At our best knowledge, there is no study on Web information foraging using Swarm Intelligence algorithms. This idea is inspired from nature where animals hunt together in

groups and rarely alone. The results are promising especially for validating certain strong Web regularities but also for a broad spectrum of other applications than the medical one that we undertook.

Despite the satisfying results, we noted a few issues when implementing Web information foraging using Swarm Intelligence approaches. The first is related to modeling and formalizing the foraging behavior. In fact, the Web surfing strategies rules had to be adapted to the Swarm Intelligence approach properties. For example, in ACO we included the pheromone computation in the decision rules of the three surfing strategies. Moreover, the score of a surfing path (potential solution) was based on the semantic similarity between the user's interests and the description vector of the last Web page on the path. The latter contains information extracted from the title of the page and tags (if available), which means that the actual content of the page is not considered. Finally, the developed system scored a good performance on *MedlinePlus*, which contains around 2000 Web pages. However, when we tried to increase the dataset size we noticed a remarkable slowdown in the performance.

In order to cope with these limitations, we propose in the next chapter a new approach and implementation employing *Game Theory*, *Self-interested Agents* and *Text Classification*.

Chapter 4

Self-interested Agents for Web Information Foraging: Application to Scientific Publications

4.1 Introduction

Multi-agent systems are widely used to simulate complex and distributed systems. In the previous chapter, we used them to developed a medical Web information foraging system. The implementation of the system was insured with Swarm Intelligence approaches and more precisely BSO and ACO. We propose in this chapter a new approach to dynamic information foraging on the Web using *self-interested agents*. The use of *Game Theory* in the implementation of the self-interested agents will permit a better modeling of the Web navigation behavior both from a formal and a practical point of view.

A self-interested agent has his own description of which states of the world he likes and acts in an attempt to achieve these states. Being a self-interested agent does not necessary mean to be in conflict with other agents. In fact, self-interested agents can share common goals and cooperate to achieve them.

The book of [56] treats a comprehensive subject on multi-agent systems and their game theoretic foundation in particular. The authors tackle the issue of agents' coordination in problem solving. They explain that while the area of multi-agent systems is not synonymous with game theory, there is no question that game theory is a key tool to master within the field. According to the authors, the Web can be viewed as the ultimate platform for interaction among self-interested and distributed computational entities. They present some modern-day game theory representations such as the normal and the extensive forms and highlight the

use of self-interested agents in game theory. Several efforts were published on the use of self-interested agents to model issues from various domains. The work in [13] is an example that deals with the concern of the way of cutting a cake or sharing a good in general, which is a question related to sharing payoff in a game. Using self-interested agents, the authors developed deterministic and randomized methods to determine a truthful, pareto-efficient and fair sharing.

In this chapter, a dynamic Web Information Foraging approach is proposed. This is the first time Self-interested Agents and Game Theory are used to address Web information foraging. We tackle the issue as a game played by a set of self-interested agents that aim at reaching relevant Web pages in a short time. We introduce three kinds of players along with the strategies followed by each one of them. In fact, the actions performed by the players would highly depend on the strategy they adopt. We propose a pure strategy, a mixed strategy and a fully mixed strategy respectively for the three kinds of players. The players share a common goal, which is satisfying a specific information need and work together in order to achieve it. The cooperative aspect between the players implemented by self-interested agents allow to get information from the Web in an effective and efficient way. Furthermore, we discuss the dynamic aspect of our information foraging approach which is ensured thanks to the modular multi-agent system architecture. In order to test our new Web information foraging system, we performed extensive experiments on the *Citation Network Dataset*, which includes more than 2.3 millions scientific publication. One of the main concerns we had with our previous implementation in Chapter 3 was related to the scalability of the system and its efficiency in large datasets. A pre-processing step was conducted with the aim of classifying the publications according to the *2012 ACM ontology*, the purpose being to make our approach scalable and more efficient. We also compared the results of our approach to previous information access approaches. The experimental results are promising.

The rest of this chapter is organized as follows. Normal-form games representation is described in section 4.2. Section 4.3 discusses the use of normal-form games for the sake of foraging information on the Web. In section 4.4, we give details about the classification of information sources, which guarantees the scalability of our approach. We then explain how we support the dynamic aspect of the Web in section 4.5. Section 4.6 presents the experiments we held and the obtained results on the Scientific Citations Network Dataset, which is explored and tested for the first time using Web information. Finally, we conclude in section 4.7 by discussing the new contributions brought in this chapter.

4.2 Normal-form games representation

The following representation is based on game theory to model self-interested agents. The normal-form representation is arguably the most fundamental in game theory. It is a description of a game that includes all perceptible and conceivable strategies, and their corresponding payoffs, for each player. A strategy is a complete plan of actions for every stage of the game, regardless of whether that stage actually arises in play. A payoff function for a player is a mapping from the cross-product of players' strategy spaces to that player's set of payoffs, i.e. the payoff function of a player takes a strategy profile (that is a specification of strategies for every player) as an input and yields a representation of payoff as an output. A finite n-player normal-form game is a triple (N, A, u), where:

- N is a finite set of n players, where each player is indexed by i and $i = 1..n$;
- $A = A_1 \times ... \times A_n$, where A_i is a finite set of actions available for player i. Each vector $a = (a_1, ..., a_n) \in A$ is called an action profile;
- $u = (u_1, ..., u_n)$ where $u_i : A \mapsto R$ is a real-valued utility (or payoff) function for player i.

There are different restricted classes of normal-form games that can serve either for coordination or competition purposes. Among those classes we find common-payoff games, zero-sum games and constant-sum games. In our work, we pay a particular attention to common-payoff games due to the coordination property they offer.

In common-payoff games, also called pure coordination games or team games, the players have no conflicting interests. Their main challenge is to coordinate on an action that maximizes the profit to all. In fact, all the players get the same payoff for the same action profile. i.e.: For $a \in A_1 \times ... \times A_n$ and any pair of players i, j, it is the case that $u_i(a) = u_j(a)$. This kind of representation is adopted for our self-interested agents because we aim to promote the coordination between them in order to increase the performance of the Web foraging.

4.3 Normal-form games for Web Information Foraging

In this section, we explain how we use game theory to forage information from the Web. We consider three kinds of players based on their behaviors as they were defined in Chapter 2: a pseudo-random surfing behavior, a recurrent surfing behavior and a rational surfing behavior.

The game consists of a Web graph which represents the players' world and the objective is to find relevant Web pages based on a user's interests. We consider the bag of words

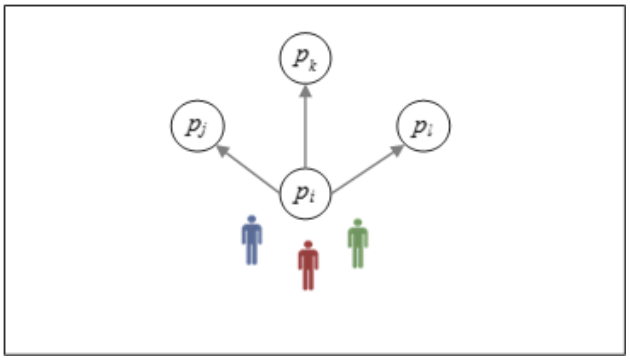

Fig. 4.1 An example of a possible state of the game

model in order to represent the user's interests using a vector denoted by V. Starting from an initial Web page (a node), the players have to find a path leading to a relevant Web page. For this purpose, we model the three players using self-interested agents which are set to make choices that allow them to move from the current Web page to a more relevant one. The agents work as a team and coordinate their actions in order to achieve their common goal and get a maximum pay-off.

Figure 4.1 illustrates an example of a state of the world during the game. The three players (agents) are located on the Web page p_i which has three outgoing pages: p_j, p_k and p_l. The agents are set to move from the current page p_i to one of its outgoing pages. The choice of the next page depends on the strategy followed by each player.

The normal-form of the game showed on Figure 4.1 is represented by the tuple (N, A, u):

- $N = Rand, Rec, Rat$;
- $A = A_{Rand} \times A_{Rec} \times A_{Rat}, A_{Rand} = A_{Rec} = A_{Rat} = \{move(p_i, p_m), p_m \in O_i\}$;
- $u = (u_{Rand}, u_{Rec}, u_{Rat})$.

Where:

- *Rand, Rec, Rat* represent respectively the pseudo-random player, the recurrent player and the rational player;
- O_i is the set of outgoing pages of the page p_i.

4.3.1 Representation of a Web page

Web pages constitute a part of the world in the game and they are the sources of information. In our model, we associate with each Web page p_i two term vectors. The first vector denoted

by pd_i represents the description vector of the page p_i and contains terms taken essentially from the title and the tags (if available) of the Web page. This vector constitutes an overview of the general topic of the page. The second vector denoted pc_i is the content vector of the page p_i. It groups terms taken from the whole Web page content which makes it more representative of the information provided by the Web page. This representation is inspired by the Web surfing process of real users. Generally, a user who is interested in a certain topic browses the Web with the goal of reaching relevant Web pages that satisfy his information needs. Assuming that the user starts the Web surfing from an initial page, s/he will first check the content of the page (represented by pc_i) in order to see if it satisfies her/his information needs. If the user is not completely convinced by the information s/he got from this Web page then s/he will look for other potential relevant Web pages to visit based on the description of the outgoing pages from the current Web page (represented by pd_i).

4.3.2 Players' strategies for Web information foraging

As we have seen in the second Chapter, not all Web users surf on the Web with the same way. Their navigation behavior and strategy depend on their familiarity with the Web and computer science technologies, and also on their purpose behind Web surfing. A player's strategy in game theory can be seen as an algorithm that tells the player what to do in every possible situation. In this Chapter, we introduce three kinds of players inspired by the Web surfing strategies presented in Chapter 2 and implemented using Swarm Intelligence approaches in Chapter 3. Each kind of players has a specific behavior, which characterizes him and defines his actions while playing the game.

We will explain next the proposed strategies followed by the three kinds of players. We will also give the corresponding transition rule for each one of them.

4.3.2.1 Modeling the Pseudo-Random player using a mixed strategy

This agent simulates users that are not familiar with the Web and who are browsing it without having a well-defined interest. The player chooses the next page randomly according to some probability distribution without taking into consideration its content. However, he has a minimum degree of rationality that gives him the ability to avoid pages that has nothing to do with the user's interests. In game theory, such a strategy is called a mixed strategy. Although it may not be immediately obvious why a player should introduce randomness into his choice of actions, the role of mixed strategies is crucial in a multi-agent setting. We denote by $Sim(pd_j, V)$ the semantic similarity between the description vector of the Web page p_j and

the user's interests vector. The probability of the action performed by the pseudo-random player is calculated as shown in Formula (4.1).

$$P(p_j) = \frac{Sim(pd_j, V)}{\Sigma_{k=1}^{m} Sim(pd_k, V)} \tag{4.1}$$

Where m is the number of outgoing pages from page p_i.
If we consider the example on Figure 4.1 and we suppose that:
$Sim(pd_j, V) = 0$; $Sim(pd_k, V) = 0.4$; $Sim(pd_l, V) = 0.1$
Then, the probabilities corresponding to the outgoing pages would be:
$P(p_j) = 0$; $P(p_k) = 0.8$; $P(p_l) = 0.2$
The probability of choosing the Web page p_j as the next page is null due to the fact that $Sim(pd_j, V) = 0$, which means that the Web page is completely out of the user's interest area.

4.3.2.2 Modeling the Recurrent player using a pure strategy

This player simulates the case when the users are familiar with the Web and have well-defined interests. The player bases the choice of the next page on its relevance towards the user's interests. At each state of the game, only the Web page that seems the most relevant (based on its description vector) among the outgoing pages is selected. Such a strategy is called a pure strategy in normal-form games. The probability of the action performed by the recurrent player is computed using Formula (4.2).

$$P(p_j) = \begin{cases} 1 & if \;\; p_j = argmax(Sim(pd_j, V)) \\ 0 & otherwise \end{cases} \tag{4.2}$$

Considering the example of Figure 4.1, the probabilities corresponding to the outgoing pages would be:
$P(p_j) = 0$; $P(p_k) = 1$; $P(p_l) = 0$

4.3.2.3 Modeling the Rational player using a fully mixed strategy

The rational player has a specific goal behind Web surfing but he is not really used to the Web environment. He tries to choose the most relevant Web page each time but he can make random choices in certain situations due to his unfamiliarity with the Web. This strategy is also known as a fully mixed strategy in game theory. The probability of the action performed by the rational player is computed using Formula (4.3).

$$P(p_j) = \begin{cases} q_0 \;\; if \;\; p_j = argmax(Sim(pd_j, V)) \\ \frac{1-q_0}{m-1} \;\; otherwise \end{cases} \tag{4.3}$$

q_0 is an empirical parameter which represents the degree of rationality of the player.

Note that a fully mixed strategy is a special case of a mixed strategy, in which every strategy action is assigned a nonzero probability. Considering the example on Figure 4.1, if we suppose that $q_0 = 0.8$ then the probabilities corresponding to the outgoing pages would be:

$P(p_j) = 0.1;$ $P(p_k) = 0.8;$ $P(p_l) = 0.1$

4.3.3 Utility function

The utility function allows to measure the profit of the agent when performing an action. It is a mapping from the states of the world in the game to real numbers. These numbers are interpreted as measures of an agent's level of satisfaction in the given states. When the agent is uncertain about which state of the world he faces, his utility is defined as the expected value of his utility function with respect to the appropriate probability distribution over states. In our approach we adopt the common-payoff games representation which means that the value of the utility would be the same for all the players. After the three players choose the new Web page to visit according to the probabilities defined in section 4.3.2, they move to their selected Web page to check its content. We define the utility function as the maximum value of the semantic similarity between the user's interests and the content of the selected pages.

$$U = u_{Rand} = u_{Rec} = u_{Rat} = Max[sim(pc_{Rand}, V), sim(pc_{Rec}, V), sim(pc_{Rat}, V)]$$

Where pc_{Rand}, pc_{Rec} and pc_{Rat} are the vectors representing the content of the pages selected respectively by the pseudo-random agent, recurrent agent and rational agent. The most similar page to the user's interests based on its content would be chosen as the next page on the surfing path and the three agents would move together to reach it. Algorithm 11 presents the pseudo-code of the three players. The only difference between them resides on the Web page selection strategy.

Obviously, the three players are in Nash equilibrium because their action profile is built such that no single player can obtain a higher payoff, by deviating unilaterally from his profile.

Algorithm 11 Self-interested foraging agent

Input: a Web structure;
Output: Surfing paths leading to relevant Web pages;

1: **procedure** SELF-INTERESTED FORAGING AGENT
2: SP := $\emptyset$; (* SP being the surfing path *);
3: Add an initial Web page to the surfing path $SP = SP + p_i$;
4: Check the neighborhood of the current page, O_i = outgoing pages of p_i ;
5: Select a Web page p_k to visit among O_i using formulas (4.1, 4.2 or 4.3) depending on the player's strategy;
6: Check the content of the selected page p_k and compute $score =: Sim(pc_k, V)$;
7: Receive the utility value;
8: Choose the Web page with the highest score and add it to the surfing path;
9: Return to 3;
10: **end procedure**

4.4 Web Information Foraging with Classification

One issue related to Web information foraging is to locate the page from which to initiate the navigation with regards to the user's interests. For large data collections, this concern is crucial, pruning crossing inutile and meaningless pages is necessary to enhance the efficiency of the information foraging process. For instance, if the user is interested in information about health, it is unnecessary to search in pages containing information on sport. One solution would be to cluster in topical groups the data according to their semantic or context. Another objective of the classification other than guiding the foraging in the start points, is the remarkable reduction of the foraging space. A preprocessing phase consisting in classifying the data is needed for that purpose.

Classification methods are numerous, they can be supervised or unsupervised depending on whether classes exist already or not. Supervised classification considers classes to insert the objects whereas unsupervised classification generates classes as outcome. The goal of a supervised classifier is to learn a model to predict the class value for a given object. A set of data is first trained to yield classes and new data are classified based on the training set. One of the most popular methods is classification by tree induction, which is suitable for knowledge extraction as it does not need any domain knowledge or parameter setting. Algorithms such as ID3 [53], C4.5 which is a successor of ID3 and CART [9] are based on a greedy approach, in which decision trees are constructed in a top-down recursive manner. The statistical Bayesian classifier is based on Bayes' theorem. Its strength resides in its high accuracy and efficiency when applied to large datasets. A simple kind qualified as naïve is comparable in performance with decision tree classifiers. Another category of classifiers are the rules-based

classifiers, where the learned model is represented by a set of if-then rules. The rules are extracted either from decision trees or directly from the training data set. Neural networks can also be used to train data. They take long training times and are therefore not suitable for applications of large data sets. Their drawback resides in their poor interpretability, which make them less attractive for data mining. SVM (Support Vector Machine) is a recent classification method relatively to neural networks. It works on both linear and non-linear data and is applied more for object recognition and speaker identification. KNN (K-Nearest Neighbor) is also another technique widely used for classification because of its simplicity. Unsupervised methods are also numerous. We distinguish partitioning methods like k-means, PAM and CLARANS. We know that k-means is less effective but more efficient than PAM and that CLARANS has the best behavior. There are also hierarchical methods and other kinds of techniques depending on the application nature.

As neural networks and SVM models are like black boxes difficult to understand, we implement a Naïve Bayes classifier in order to classify data (Web pages) from very large collections with millions of items. Since this is a supervised method, a training set is needed in order to do the classification. In our case, we consider a taxonomy proper to our application domain and we use it as a training set. Once the Web pages classified based on their content, we apply K-means on the most pupated classes. The number of clusters is defined for each class according to its number of items. This step is performed in order to reduce the size of the most populated classes by creating clusters inside those classes with a relatively uniform distribution of Web pages. Recall that the aim of the classification and the clustering is to enhance efficiency of the information foraging process.

4.5 Extension to dynamic information foraging

The Web is a dynamic environment that undergoes multiple changes and evolvements at a high rate. The dynamicity of the Web is due to the growth of its volume and the changes that are constantly operated by users [54, WWW]. For this purpose, we adapt the dynamic Web information foraging architecture proposed in Chapter 2, to the case of self-interested agents. Figure 4.2 shows the system architecture on which we distinguish two main phases. The first one aims to localize the information sources based on a user's interests while the second performs an update to the information sources taking into account the changes that occur on the Web. The system learns automatically the appropriate information sources for each user's interests as they are introduced.

The results of the first user interaction are stored in a database containing all the information sources corresponding to the user's information need sorted in a decreasing order

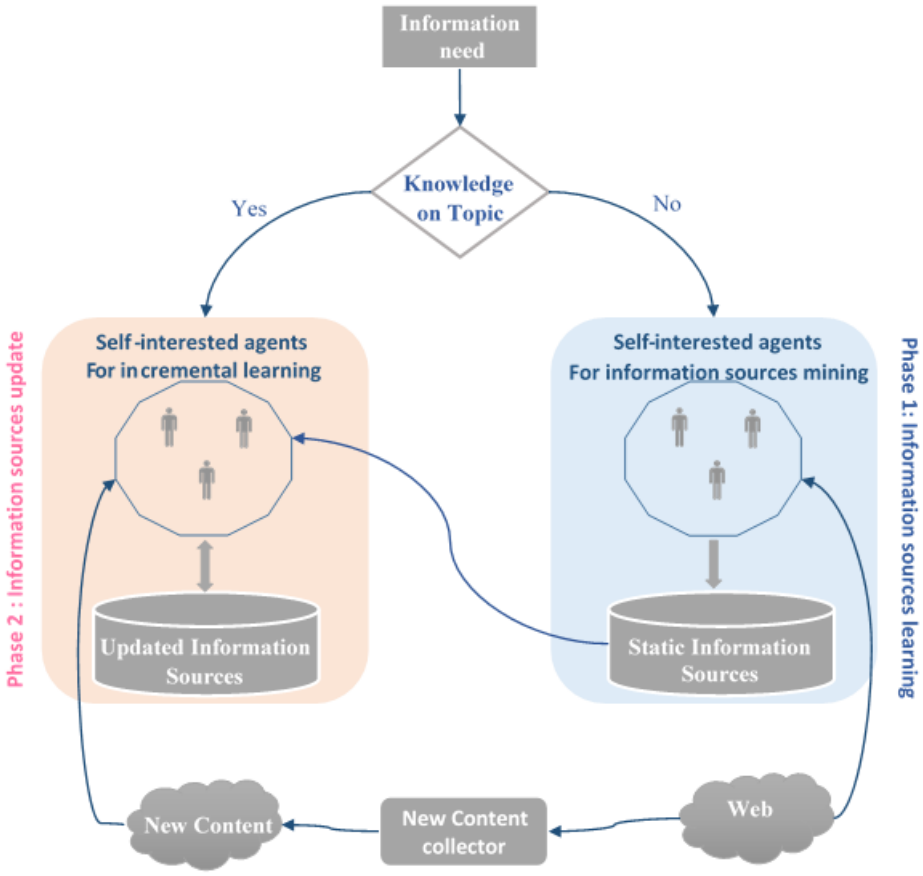

Fig. 4.2 Dynamic Web Information Foraging Architecture

according to their relevance. Once the user shows interest in the same (or a similar) topic again, the system uses the results of the previous user's request and updates them taking into consideration the changes that have occurred since the last interaction. In other terms, instead of launching the self-interested agents on the whole Web (or data collection), the system starts from the already learned information sources of the previous request. This automatic-learning allows to optimize the foraging process and reduce the response time. The two phases are described in the following subsections.

4.5.1 The information sources learning

The task to undertake here is to investigate the Web for the sake of finding relevant Web information sources. The agents seek relevant Web pages according to their information foraging strategies defined in Section 4.3.2. The approach will rank the leading pages according to their relevance in a database called "Static information sources" that will be used during the next step. Each time a new user's interests is introduced into the system for the first time, the foraging results are stored in this database.

4.5.2 The information sources update

Recall that the information sources update aims at making the system scalable and able to monitor the evolution of the Web. Unlike the learning phase that is executed just once for a given user's interests (when introduced for the first time), this phase will be running each time a user asks for information. For this purpose, two tasks have to be planned: update the information sources database and then explore the new Web pages. The first operation consists in:

- Eliminating from the *static information sources* database the Web pages that do not exist anymore at the current date.
- Updating the score for those that are still present.
- Sorting the pages in a decreasing order of their score (relevance).

As a second step, the agents will proceed to an incremental learning to update the information sources at each change occurring in the Web. Concretely, the agents are launched on the new sub-collecting contain the Web changed and the achieved information sources are integrated in the *updated information sources* database with respect to the predefined order.

4.6 Experiments

In this section, we describe the experiments we undertook on the Citation Network Dataset and report the achieved results. We implemented and tested our approach using Java Eclipse help System Base, version 2.0.2 on a PC with an Intel core I5-3317U Processor (1.70 GH) with 4 GB of RAM.

4.6.1 The Dataset

The Citation Network Dataset [60] gives access to scientific papers indexed by *ACM* and *DBLP* and is published mainly for research purposes. Both repositories *ACM* and *DBLP* give access to scientific publications through their websites, where each scientific publication has a corresponding Web page. The main reason behind choosing this dataset resides in its structure which is close to a Web graph structure. We explored the ACM-Citation-network V8, which was created on 02/04/2016 and contains 2,381,688 papers extracted along with 10,476,564 citation relationships. The first version of this archive included 629,814 papers with 632,752 citations. We observe that the ratio between the number of citations and the number of articles was around 1 in the first version of the dataset whereas in the considered version it is around 5. This is due to the importance of the citation relations in scientific articles and the particular attention given recently by the scientific community and journal ranking systems to this metric. This recent phenomenon allows to forage information more easily by surfing on papers that are relatively more linked than in the previous versions. Each paper is associated with a title, an abstract, a list of authors, a year, a venue, a unique index and a list of citations .

Figure 4.3 shows a preview of the dataset structure. Each field is preceded by a specific code where:

- #* indicates the paper title
- #@ for the authors
- #t for the year of publication
- #c for the publication venue
- #index for the index id of the paper
- #% for the id of citations of this paper (there are multiple lines, with each indicating a reference)

```
#*A static estimation technique of power sensitivity in logic circuits
#@Taewhan Kim, Ki-Seok Chung, C. L. Liu
#t2001
#cProceedings of the 38th annual Design Automation Conference
#index5390881220f70186a0d7ec8b
#%539087e720f70186a0d699ac
#%5390882720f70186a0d8a0a0
#%539087d420f70186a0d5e3be
#%539087e120f70186a0d6726f
#%539087c720f70186a0d56c67
#%5390b00c20f70186a0ed4f65
#%558ac6e0612c41e6b9d39eed
#%539087e120f70186a0d672f0
#%53908bde20f70186a0dc70e7
#!In this paper, we study a new problem of statically estimating the power sensi
characteristics of power dissipation due to changes in state of primary inputs.
power consumption of the circuit efficiently but also to provide potential oppor
static estimation technique for power sensitivity based on a new concept calledp
data on MCNC benchmark examples show that the proposed technique is useful and e
power consumption is 9.4\% with a huge speed-up in simulation.

#*JouleTrack: a web based tool for software energy profiling
#@Amit Sinha, Anantha P. Chandrakasan
#t2001
#cProceedings of the 38th annual Design Automation Conference
#index5390881220f70186a0d7ec8c
#%539087d420f70186a0d5eb74
#%53909ee020f70186a0e3415f
#%558ac6e0612c41e6b9d39eed
#%53908bfb20f70186a0dc9bad
#%539087c720f70186a0d578bc
#%5390882d20f70186a0d8dfc8
#%539087c720f70186a0d56c91
#!A software energy estimation methodology is presented that avoids explicit cha
for a set of bench-mark programs evaluated on the StrongARM SA-1100 and Hitachi
estimation levels. It also isolates the switch-ing and leakage components of the
```

Fig. 4.3 Dataset preview

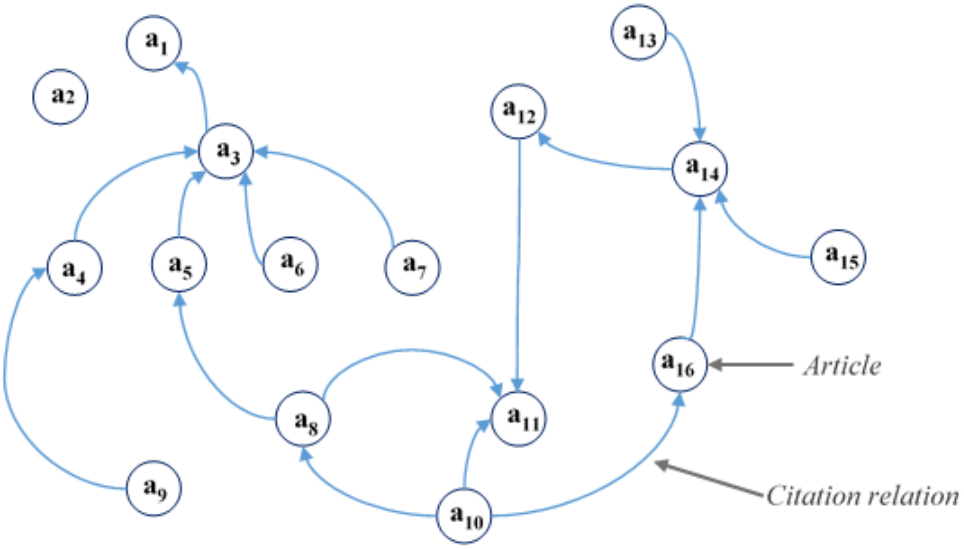

Fig. 4.4 Snapshot of the Citation network dataset graph

- #! for the abstract

Note that the abstract and the citations are not available for all the articles of the dataset. In our experiments, we only consider the articles that have an abstract. We construct a graph composed by the 1,669,237 articles that have an abstract. The articles are the nodes of the graph while the citation relations represent the edges as illustrated in Figure 4.4. The article a_3 for example is cited by the articles a_4, a_5, a_6 and a_7. Its number of citations is equal to 4. What is interesting in this work is that the outcome of the foraging process ends up not only with the most appropriate article to the user's interests but also with its number of citations. Another important aspect is that our approach of foraging is also guided by the number of citations of the papers as the latter became an important metric for publications. The more a paper has citations, the more it is selected for surfing on the graph. This way, foraging privileges paths with substantial number of citations.

The size of the file containing the dataset is about 2.2 GB which makes it hard to read and load into the random access memory with our hardware specifications. To handle a dataset with such volume, we store the citation graph in a database using *PostgreSQL* database management system [43] and then connect the database to our system. Table 4.1 provides a summary on the dataset characteristics.

4.6.2 Classification Results

Like mentioned in section 4.4, a preprocessing phase is mandatory in order to facilitate the foraging process especially when working on large data collections. For this purpose, we

Characteristic	Cardinality
Number of articles	1,669,237
Average abstract length	137 words
Average title length	9 words
Total number of citations	10,198,574
Average citations by article	6
Most cited Article	805 citations

Table 4.1 Characteristics of the Citation Network Dataset

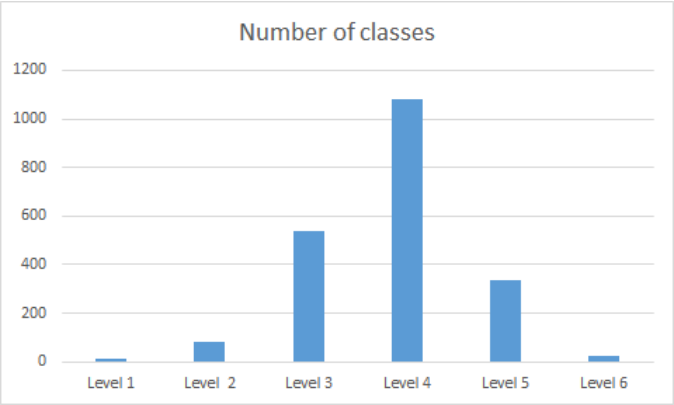

Fig. 4.5 Number of classes per level

implement a Naïve Bayes text classifier that aims to classify the 1,669,237 papers according to the *2012 ACM taxonomy* [2].

The 2012 ACM classification is a hierarchy of keywords used by the computer science community. Certain journals require a list of the ACM classification keywords related to the submissions. Figure 4.5 gives an overview of the number of classes per level. The taxonomy is composed of 6 hierarchical levels with the fourth being the one containing the largest number of keywords and classes.

We perform a classification using our Naïve Bayes classifier based on the content of the papers. The articles are distributed over 1079 classes deduced from the fourth level of the 2012 ACM classification.We consider the fourth level as it counts the largest number of classes according to Figure 4.5. For each class, we create a text file containing characterizing keywords taken from the class' name and the class names of its upper and lower levels. This files serve as a training examples for our classifier. Figure 4.6 shows the overall articles' distribution over the classes of the fourth level.

"Machine learning: Learning in probabilistic graphical models" and *"Very large scale integration design: System on chip"* are the most populated classes containing respectively 18% and 15% of the total number of the papers. This could be explained by the fact that

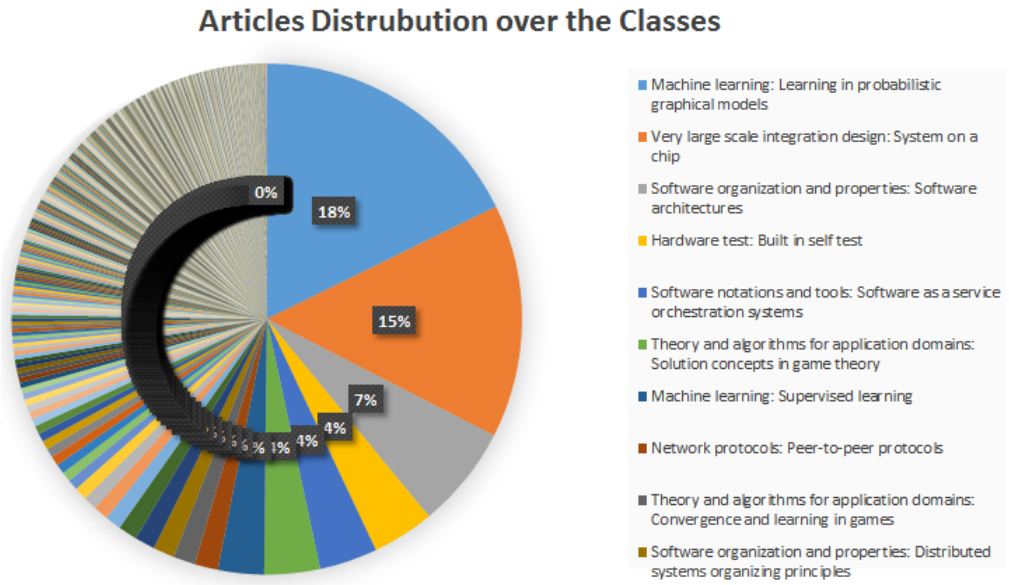

Fig. 4.6 Overall articles' distribution

there are a lot of articles about machine-learning and hardware design. Figure 4.7 and table 4.2 give more details about the classification results, they show respectively the number of articles per class and the first 3 levels for the top 10 most populated classes.

Figure 4.8 presents a box plot illustrating the number of articles of the ten most populated classes. We notice that a great number of articles are above the median, which means that the articles are far to be distributed in a homogeneous way. This phenomenon indicates that for the classes with a few articles, the foraging time will be almost instantaneous whereas in the large classes it will be more significant. In order to get a real-time Web information foraging system, we divided these large classes into sub-classes using the k-means algorithm. This information is taken into account for the selection of the user's interests in the experiments.

4.6.3 Static Web Information Foraging Results

Once the scientific articles of the dataset classified, we focused on the first part of experiments which deals with a static (non-changing) environment. This part consists in running our Web information foraging system on the whole dataset considering different users' interest (information needs). In order to assess the performance of our approach, we used the following evaluation metrics:

- The score of a surfing path, computed as follows:

$$Score(SP) = \omega_1 Sim(pd_i, V) + \omega_2 (\frac{number\ of\ citations\ of\ p_i}{max\ number\ of\ citations})$$

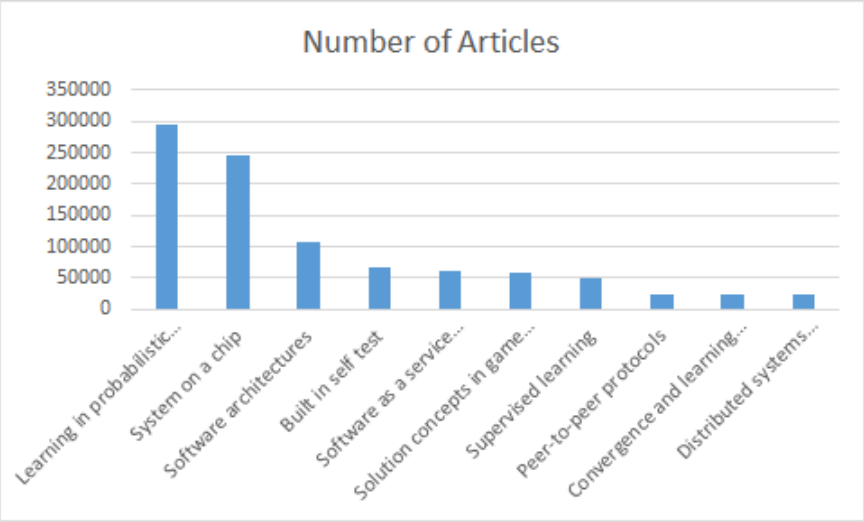

Fig. 4.7 Number of articles of the top 10 most populated classes

Class Name	First level	Second level	Third level
Learning in probabilistic graphical models	Computing methodologies	Machine learning	Machine learning approaches
System on a chip	Hardware	Very large scale integration design	Design reuse and communication-based design
Software architectures	Software and its engineering	Software organization and properties	Software system structures
Built in self test	Hardware	Hardware test	Design for testability
Software as a service orchestration systems	Software and its engineering	Software notations and tools	Development frameworks and environments
Solution concepts in game theory	Theory of computation	Theory and algorithms for application domains	Algorithmic game theory and mechanism design
Supervised learning	Computing methodologies	Machine learning	Learning paradigms
Peer-to-peer protocols	Networks	Network protocols	Application layer protocols
Convergence and learning in games	Theory of computation	Theory and algorithms for application domains	Algorithmic game theory and mechanism design
Distributed systems organizing principles	Software and its engineering	Software organization and properties	Software system structures

Table 4.2 The first three levels of the top 10 classes

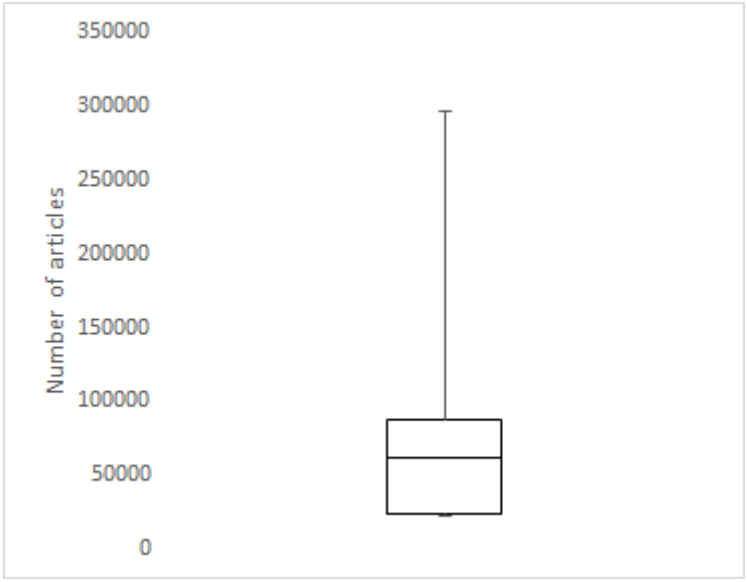

Fig. 4.8 Articles number box plot of the top 10 most populated classes

Where:

- SP is the surfing path
- Pd_i is the last page (article) on the surfing path
- V is the vector representing the user's interests
- $Sim(pd_i, V)$ is the cosine similarity between the description vector of the Web page and the user's interests
- ω_1 and ω_2 are weight parameters

- The number of articles (nodes of the graph) the systems visits in order to reach the outcome article. The articles should not necessarily be linked.
- The response time the system takes to provide results to users. To evaluate the response time for a certain user's interests, we run the system 100 times then we compute the mean time for that user's interests.

4.6.3.1 Parameters setting

We carried out some tests for the sake of finding the values of the empirical parameters that yield effective and efficient results. We run our Web information foraging system on the same set of users' interest that we generated randomly beforehand while changing the values of the empirical parameters. Figure 4.9(a)-(d) show the variation of the score and the response time according to the values of q_0 and ω_1. We fix the values as follows:

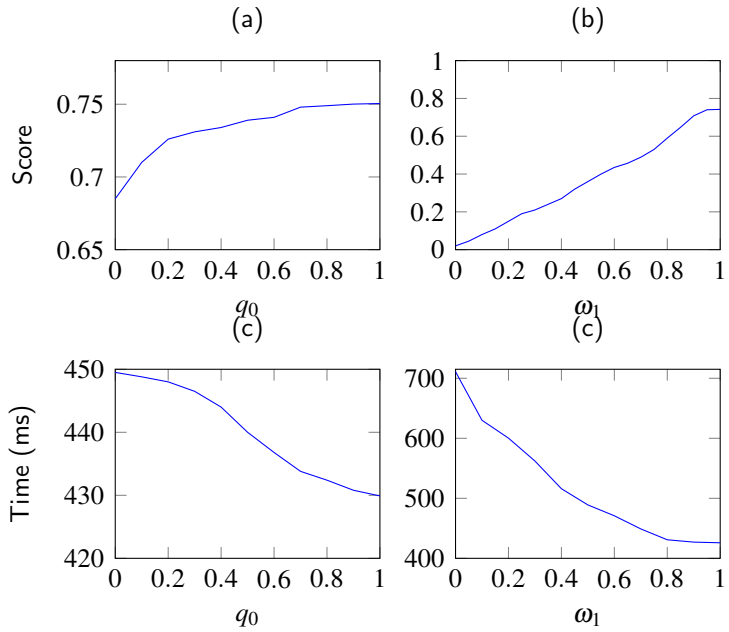

Fig. 4.9 Score and response time variation according to the values of q_0, ω_1 and ω_2

- $q_0 = 0.65$
- $\omega_1 = 0.95$
- $\omega_2 = 0.05$

4.6.3.2 Experimental Results

Table 4.3 shows the information foraging results yielded by our approach for different users' interests. The first column on the left represents the user's interests vector which is composed by a set of keywords taken from topics of the different *ACM taxonomy* classes in a homogeneous way. The second column displays the class to which the user's interests belong and the third one presents the most relevant article found by our approach. The last four columns exhibit respectively the score which represents the relevance of the result, the execution time in milliseconds, the number of visited articles before reaching the best result and the number of citations of each article. For instance, the most relevant article corresponding to the user's interests "Complexity of Turing Machines" is named "The Complexity of Small Universal Turing Machines". It has a score of 0.9 and a number of citations equal to 408, this means that the article will most likely satisfy the user's information needs because it matches his interests and it is also highly cited within the Turing Machines community.

User's Interst	Class	Most Relevant Article	Score	Time (ms)	Visited Nodes	Citations
Swarm Intelligence	Randomized search	Editorial special issue: swarm intelligence	0.62	132	12	5
Natural Language Understanding	Information extraction	Natural Language Understanding	0.95	1375	143	8
Virtual reality	Virtual reality	Virtual reality	0.95	703	83	5
NoSQL	Query languages for non relational engines	SQL vs. NoSQL	0.38	713	1	1
Human Behavior	Mobile devices	Mobile robot control using fuzzy-neural-network for learning human behavior	0.32	628	5	3
Complexity of Turing Machines	Turing machines	The Complexity of Small Universal Turing Machines	0.90	344	10	408
Advnced multi-agent systems	Multi-agent systems	Agent System Development Method Based on Agent Patterns	0.48	1153	67	2
Business Intelligence	Business intelligence	Business Intelligence	0.95	109	4	1
Tasks scheduling	Recommender systems	Scheduling PVM tasks	0.71	230	4	10
Query evaluation techniques large scale	Multimedia databases	Query evaluation techniques for large databases	0.87	472	14	238
Medical Terminology	Computational genomics	Medical Terminology Online for Exploring Medical Language	0.57	188	4	1
Cloud computing	Cloud based storage	Cloud Computing	0.95	500	38	1
Unlabeled learning using bayesian classifiers	Bayesian analysis	Bayesian classifiers for positive unlabeled learning	0.90	567	329	20
Recommender Systems	Recommender Systems	Introduction to Recommender Systems	0.79	295	7	1
Dataflow programming languages	History of programming languages	Advances in dataflow programming languages	0.78	304	17	50
Evolutionary Computation	Quantum information theory	Evolutionary Computation: A New Transactions	0.61	686	7	1
Web information retrieval	Environment specific retrieval	Information retrieval on the Web	0.97	591	142	143
Big Data	Deduplication	Ethics of Big Data	0.65	209	8	1
Wisdom Web	Mashups	World Wide Wisdom	0.69	306	11	1
Exploratory search	Randomized search	Editorial: Evaluating exploratory search systems	0.74	287	3	10
Web Design	Mashups	Easy Web Design	0.76	285	9	1
Java data structures	Data compression	Data Structures and Algorithms in Java	0.68	323	86	1
Computer Ethics	Codes of ethics	Computer Ethics	0.95	151	2	1
Cognitive Robotics	Cognitive Robotics	Towards Cognitive Robotics	0.69	386	56	10

Table 4.3 Web Information Foraging results for different users' interests

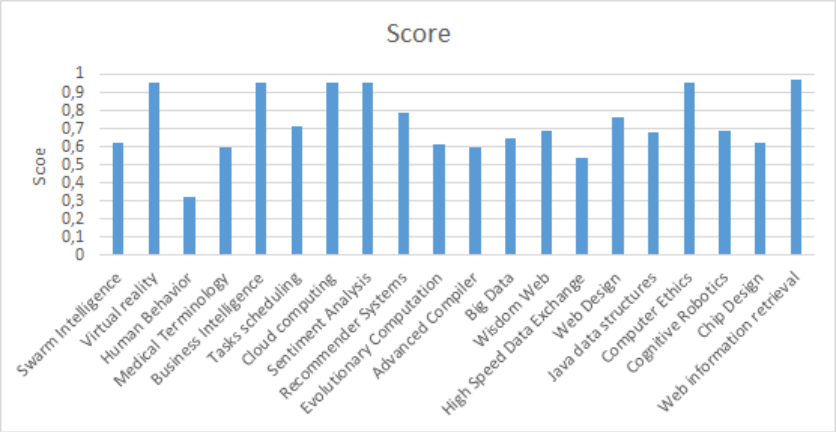

Fig. 4.10 Achieved score for a collection of users' interests

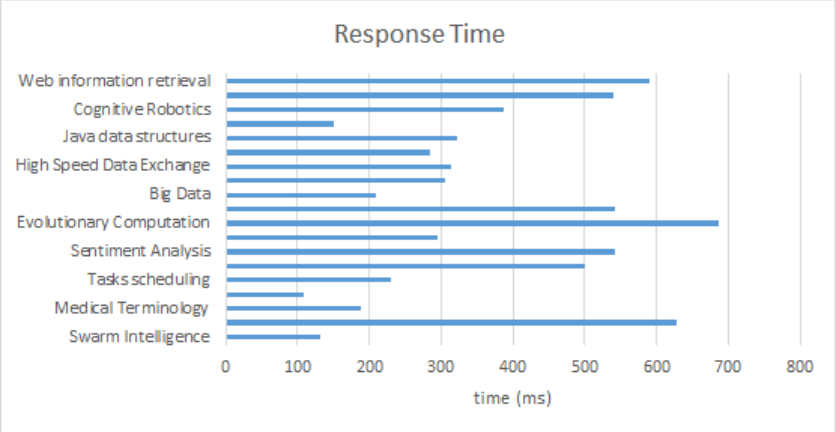

Fig. 4.11 Response time for different users' interests

Figures 4.10 and 4.11 depict a graphical view of the score and the time for the users' interests of Table 4.3.

To get a clearer idea about the performance of our approach and in order to evaluate its outcomes, we run a series of extensive tests including 200 different users' interests that seemed the most representative and appropriate. We first started by gathering random words from the content of the articles of the Citation Network Dataset, then using these words we composed a large number of users' interests with a length varying from 2 to 10 words. Table 4.4 exhibits some statistics of the evaluation metrics we use including the response time, the number of visited articles before reaching the most relevant article and the best result's score.

Metric	Cardinality
Average response Time	435 ms
Average number of visited articles	30
Average score of the best result	0.74

Table 4.4 Average evaluation metrics values

4.6.4 Comparison with other approaches

A necessary step to validate our approach is to compare it to other information access approaches previously developed and proposed in the literature. In this section, we pay particular attention to two approaches. The first concerns classical information retrieval, while the second one our Web information foraging using swarm intelligence approach presented in Chapter 3.

4.6.4.1 Classical Information Retrieval

In order to situate the advantage of our Web information foraging system relatively to Web search engines, we undertook an empirical comparison between the classical information retrieval process and our information foraging approach. For this purpose, we implemented a search engine based on the Vector Space Model (VSM) of information retrieval using *Apache Lucene*'s *Java* API. Afterwards, we compared its performance to our Web information foraging system on the Citation Network dataset.

One of the first noticeable differences resides in the way of presenting the results. Both approaches return a ranked list of results for each query. However, in classical information retrieval each result is one single article while in our approach it is a surfing path containing at least one article. In addition to showing the best result, our approach provides the entire surfing path that led to it. Table 4.5 is a good example of this difference, it shows the highest ranked result for two different queries / user's interests. For example in the second example, both approaches consider the article entitled *"Dispersed particle swarm optimization"* as the best result. In addition to showing the best result, our approach provides the entire surfing path that led to it. We remark that all the articles on the surfing path are related to the user's interest which is *"Dispersed particle swarm optimization"* and therefore, they might interest the user as well.

To get an idea about the time required by each approach to respond to user's information needs, we tested 50 users' interests generated randomly on both approaches. Figure 4.12 reports the average response time comparison between classical information retrieval and

User Interst	Classical Information Retrieval	Web Information Foraging
	Highest ranked result	Highest ranked result
lazy learning and multilabel learning	ML-KNN: A lazy learning approach to multi-label learning	Coupled dimensionality reduction and classification for supervised and semi-supervised multilabel learning → LIFT: multi-label learning with label-specific features → ML-KNN: A lazy learning approach to multi-label learning
Dispersed particle swarm optimization	Dispersed particle swarm optimization	The application of particle swarm optimisation in organisational behaviour → Integral Particle Swarm Optimization with Dispersed Accelerator Information → Dispersed particle swarm optimization

Table 4.5 Classical Web Information retrieval vs Web Information Foraging

our Web information foraging system. The results confirm the superiority of our approach in terms of efficiency.

4.6.4.2 Medical Web information foraging

For the sake of evaluating the improvements brought by our new approach, we consider a performance comparison between the new system implemented using self-interested agents and the medical Web information foraging system using hybrid ACO and Tabu search [19] proposed in Chapter 3.

We tested both systems on *MedlinePlus* then we compared their outcomes. We used the XML version of the website of July 23 2014. We took 45 users' interests as the inputs of both systems. Figure 4.13 shows the response time comparison between the two approaches. For simplification, we denote the two approaches by DWIFSA (Dynamic Web Information Foraging using Self-interested Agents) and MWIFAT (Medical Web Information Foraging using ACO and Tabu search). We notice a big gap in terms of efficiency between the two approaches. In fact, the new implementation using self-interested agents, game theory and text classification is more than three times faster than MWIFAT.

4.6.5 Dynamic Web Information Foraging

In this part we planned to experiment the dynamic aspect of our system. We first divided the dataset we used in the static experiments in two collections of articles. The first contains 1,600,000 articles and represents the static part of the Web. The second contains the 669,237

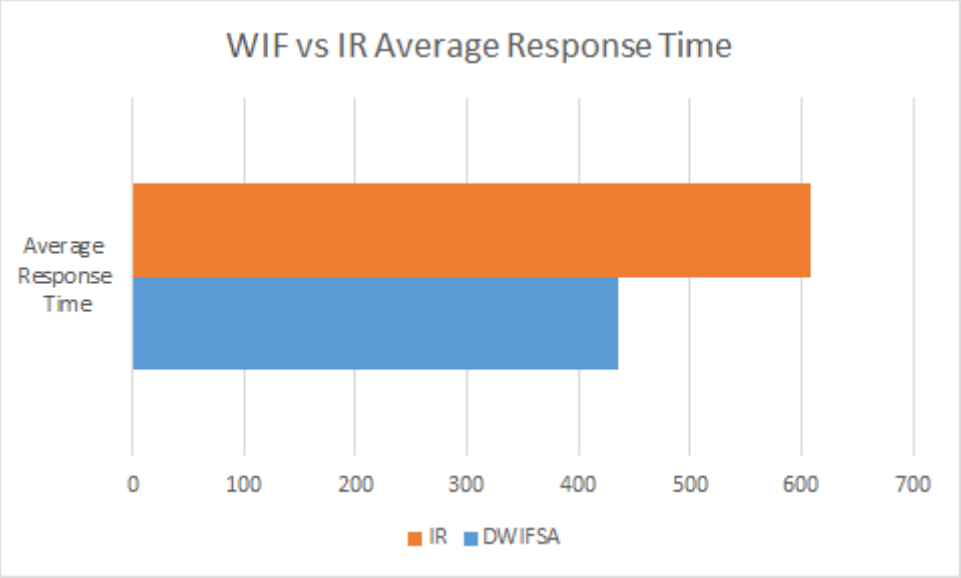

Fig. 4.12 Average Response time comparison between IR and WIF

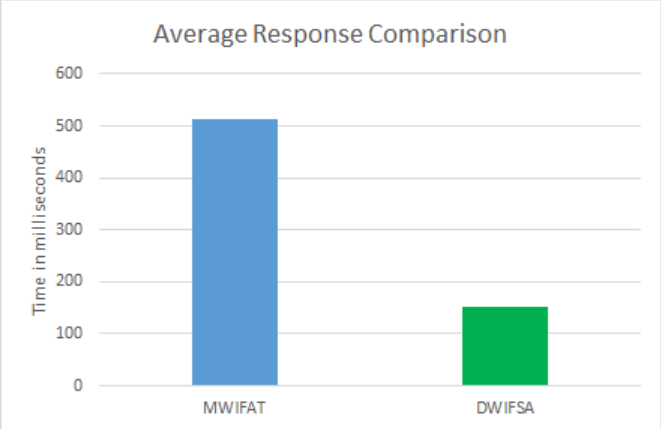

Fig. 4.13 DWIFSA versus MWIFAT - Response Time

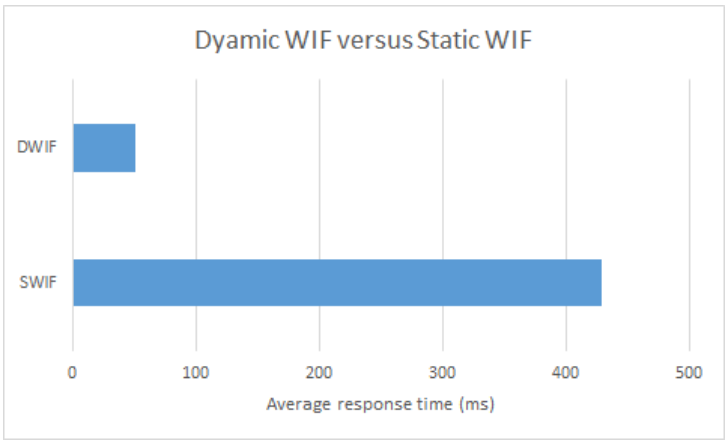

Fig. 4.14 Response time comparison

remaining articles, which we consider as the new Web content. As a second step, we run our system using 100 users' interests that we generated randomly beforehand on the static part of the dataset (1,600,000 articles). Considering that all the users' interests are introduced to the system for the first time, this step corresponds to the first phase described in section 4.5. Afterwards, we updated the content of the first collection of articles by including the 669,237 articles of the second collection in order to simulate the changes that occur on the Web. We took into consideration the citation relationships between article when combining the two parts. Finally, to reproduce the case where our system gets as inputs users interests that it already has knowledge about (phase 2, Section 4.5), we run the same 100 user's interests we previously used on the updated collection.

Figure 4.14 presents the response time comparison between static Web information foraging (SWIF) and dynamic Web information foraging (DWIF). We observe that the response time becomes remarkably shorter and almost instant when taking advantage of the dynamic feature of our system. This is due to the fact that when the system gets a user's interest which is similar to one it already got at time t', the system uses the knowledge it had before and combines it with the foraging results on the new collection changes, which occurred between the current time t and t' instead of foraging on the whole collection.

4.7 Discussion

In this chapter, we presented an approach to Dynamic Web information foraging using self-interested agents. We proposed a model based on game theory and more precisely on normal-form games in order to simulate the Web surfing behavior of real Web users. This idea was inspired from the information foraging theory and from our findings following some

experiments we did on how real users surf on the Web in the previous chapter. In fact, our self-interested agents behave in the same way as real users when browsing the Web. When an agent has the choice to visit multiple Web pages, he selects the page that seems the more relevant based on its description (title, URL, icon, tags...). Once the Web page is selected, the agent visits this page and checks its content in order to compare the provided information on the page with his information need.

We also proposed a normal-form game representation and introduced three kinds of players with a game strategy for each one of them. The three kinds of players are considered as a team and coordinate together to find relevant Web pages thanks to the common-payoff configuration we adopted. We also showed that information foraging gains in efficiency when data are perprocessed and grouped in classes. The results of information foraging depend closely on those of the classification phase. We handled this step thoroughly in order to achieved scalability.

Furthermore, we extended our model to handle the Web dynamicity by proposing a dynamic Web information foraging architecture that takes into account the changes that may occur on the Web. Finally, we conducted a series of experiments on the Citation Network Dataset, which offers access to more than 2.3 millions scientific publications. To our knowledge, this is the first time an information foraging system is tested on a dataset with such size. The performed experiments show the ability of our system to find relevant Web pages in an effective and efficient way based on a user's interests. The outcomes consist in a set of surfing paths ranked by relevance. This offers the user the possibility to go more in depth with getting information on a certain topic without spending too much time on visiting a lot of Web pages. We undertook a comparative performance analysis with two information access approaches. The first one being classical information retrieval while the second one is a the medical Web information foraging system we proposed in chapter 3 and published in [19]. Our system was able to achieve better results with regards to both approaches and cope with the two main drawbacks of the previous implementation noted in Chapter 3, which are the formal representation of the information foraging process and the efficiency in large datasets.

Chapter 5

Information Foraging on Social Networks

5.1 Introduction

Nowadays, people are becoming more dependent on social networks in their daily life. They use them to share information and interact with each other, which highly contributes to the growth of the volume of online public data.

Sixty seconds might seem like an insignificant amount of time for us, however when we look at it in terms of how much of data is created on the Web, there will be a huge difference. There is no doubt that the increase in the number of connected Web users has played a big part in data's tremendous growth, but recently we witnessed the emergence of a new and more important factor. New technologies have seen the light such as social platforms which allow people to create and share information online. These new forms of sharing are delivering a sense of connectedness, adding a new value to people's lives. Twitter users for example produce around 7777 tweets per second according to Internet Live Stats [4], this number corresponds to more than 500 million tweets in one day. This huge number of tweets and social posts represents a rich source of information and one major concern that results from it, is to find a way to explore this source and get useful information from it.

In the previous chapters, we proposed different approaches to implement Web information foraging then we applied them to the domains of health and scientific publications. We propose in this chapter an approach that offers social networks' users the possibility of getting relevant and credible information in a rapid and effective way. We adapt our Web information foraging system to make it able to operate on social graphs and to take into account the users' interests and their social relations and interactions. We built our dataset

using real data we extracted from the information sharing network *Twitter* and we exploited these data to automatically define the information needs of users. The produced results consist in surfing paths leading to relevant information taking into consideration the user's interests and the credibility of the foraged information.

The rest of the Chapter is structured as follows. We propose a model for information foraging on social networks in Section 5.2. In Section 5.3 we present our social information foraging system, which is based on multi-agents technology. In section 5.4 we present the experimental results that we obtained by testing our system on real twitter data. Finally, in Section 5.5 we conclude our study and discuss some potential future axes for this work.

5.2 Foraging Information on Social Networks

Social networks started to become popular a few years ago and since then, they revolutionized the way people communicate and socialize on the Web. Their impact on the society can be noticed in our daily life, for example people don't have to go outside to buy the newspaper in the morning anymore, they can just scroll their twitter timeline to get the latest news in real time. They also don't need to call their friends each time they have news to share with them thanks to Facebook's messaging platform. All these social interactions allow faster and easier information exchange but at the meantime it generates a huge amount of data. According to IBM, 2.5 quintillion (1018) bytes of data were created every day in 2012 and most of it, comes from social media platforms.

Social networks are generally represented using social graphs, which are oriented graphs that illustrate interconnections and social relations among people, groups and organizations in social networks. Like any other oriented graph, a social graph is an ordered pair $G = (V, E)$ composed by a set V of vertices and a set E of directed edges (arrows). Each element $e \in E$ represents a relation of incidence that associates with each edge two vertices. Figure 1 illustrates the social graph structure we consider in the present work, where the labels on the vertices represent users and the labels on the edges show the relationships that exist between them. The labeled graph is defined as G = (V, E), where:

- $V = \{user1, user2, user3, user4, user5, user6\}$
- $E = \{\ (user1, user2, repliesto), (user1, user3, follows),$
  $(user1, user4, mentions), (user1, user5, follows),$
  $(user2, user4, follows), (user3, user1, follows),$
  $(user4, user4, posts), (user5, user3, mentions),$

$(user5, user5, posts), (user6, user1, re-posts),$
$(user6, user5, mentions)\ \}$

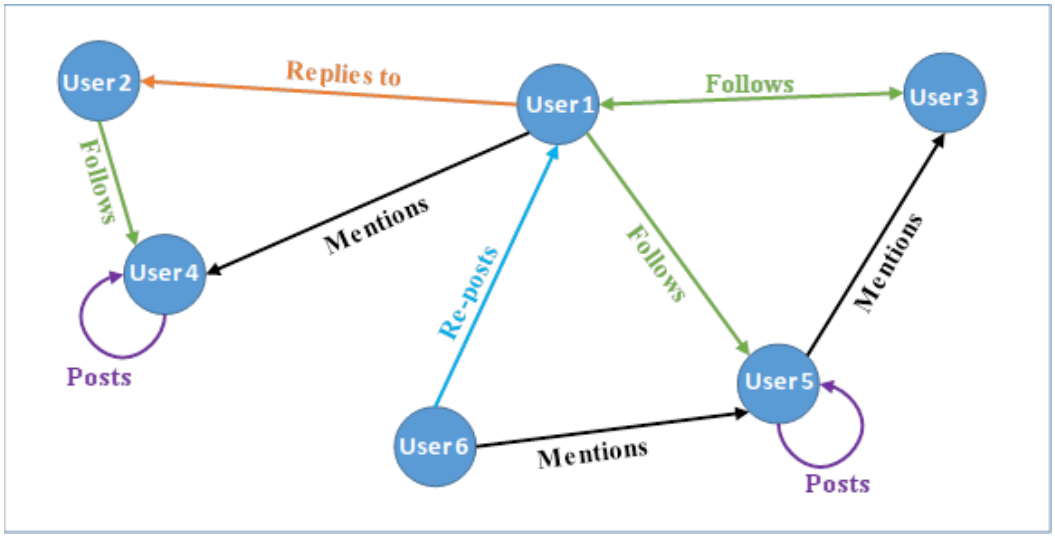

Fig. 5.1 An example of the considered social graph structure.

The social relations on Figure 5.1 are defined as follows:

- *Follows*: is a relationship between two users.
- *Posts*: is the action of publishing a post on the social network by a user. A post is represented by a self-looping edge on the graph because it only involves one user (the author of the post).
- *Mentions*: is the event of mentioning a user in a post. The edge representing this relation starts from the vertex of the author of the post and goes towards the user who is being mentioned in that post.
- *Replies to*: is an event consisting in answering a post of another user. The edge representing a *reply* goes from the vertex of the user who is writing the reply towards the vertex of the user who published the original post.
- *Re-posts*: is the event of sharing a post of another user. Like in the previous relation, the edge goes from the user who is re-posting towards the user who published the original post.

Note that we denote each edge by a triplet that indicates respectively the source vertex, the destination vertex and the social relation linking the two vertices. The edges representing

the relations *posts, mentions, replies to* and *re-posts* are the ones containing social posts (messages) generated by the users of the social network. We denote these edges by *content-sharing-edges*. Moreover, this way of representing the graph is optimal as the storage of its content is the lowest possible. Instead of using the common data structure for instance the adjacency table of the graph, a table of three triplets is built $T(|E|,3)$. The three columns contain respectively the source, the destination and the relationship between them. Each row of the table corresponds to and edge of the graph G.

Exploring social networks content would bring a significant boost to the field of information access. Our contribution consists in applying Information Foraging on social networks which offers the possibility to exploit some social features and relations in order to get relevant and credible information in real time. We pay particular attention to representing a specific user's information needs, finding relevant information corresponding to that information need, and assessing the credibility of the foraged information. The three aspects are detailed in the following subsections.

5.2.1 Modeling the user's interests

The task of information foraging is usually motivated by an information need, which can be either specified explicitly by the user by means of a query or automatically deduced using prior information on the user [12]. In the previous chapters, the user's interests were generated either randomly using a dictionary or provided explicitly by real users for experiments purpose. In this model, we consider the second case where the user's interests are automatically defined by the system based on the user's social data. We implicitly represent the information need of a user as a vector of terms V which corresponds to her/his interests extracted from her/his activities on social networks. The idea is to collect frequent keywords from the user's biography, her/his recent posts and her/his interactions with other users in order to use them to determine her/his interests. Once a user's interests defined, the Social Nets Information Foraging System starts to search relevant information to that user based on her/his vector of interests V. Algorithm 12 presents the pseudo-code of the implicit user's interests generation. h and n are tunable parameters that represent respectively the period of time during which the user's tweets are crawled and the size of the interests vector. The information foraging process is detailed in the following subsection.

5.2.2 Information Foraging

Recall that the basis of the *Information Foraging Theory* is a cost and benefit assessment of achieving a goal, where the cost is the amount of time and energy we spend during the

Algorithm 12 Building the user's interests vector

```
Input: User's id
Output: User's interest vector V
begin
    1. Create an empty text file D;
    2. Extract the user's biography from her/his social profile and put it in D;
    3. Crawl the user's posts generated during the past h hours and put them
    in D;
    4. Create a vector of terms V;
    5. Put in V the n most frequent terms form D;
end;
```

information foraging process and the benefit is the quality of the obtained information. The goal of an information foraging system is to get the highest amount of relevant information whilst spending the lowest amount of time and energy. This goal can be achieved thanks to the *information scent* concept introduced in Chapter 2 and which is able to automatically guide the foraging process towards sources containing relevant information based on an information need.

In our case, the goal of the system is to reach social posts that are relevant to a certain user based on her/his vector of interests V. The information foraging process starts from one content-sharing-edge of the social graph, then it aims at increasing the information scent by moving to other content-sharing-edges with the intent of reaching relevant posts in the end. If we consider that at time t we are on a content-sharing-edge e_i, the motivation to move to a new edge e_j relies on the increase of the information scent we would obtain in moving from the former to the latter edge. The information scent increase at time $t+1$ with respect to time t, can be modeled by a quantity proportional to the similarity between the content of the new selected edge e_j and the user's interests vector V. We propose Formula (5.1) to estimate the information scent we would get at a given content-sharing-edge e_j.

$$InfoScent(e_j) = InfoScent(e_i) + Sim(e_j, V) \tag{5.1}$$

If we suppose that e_i is the initial edge from which the foraging starts, then the information scent at e_i would be equal to the similarity between its content and the user's interests.

Among the three Web surfing strategies presented in Chapter 2, we focus on the rational surfing strategy. Formula (5.2) defines the probability $P(e_i, e_j)$ to move from a content-sharing-edge e_i to one of its adjacent content-sharing-edges e_j.

$$P(e_i, e_j) = \frac{InfoScent(e_j)^{\alpha} \times sim(e_i, e_j)^{\beta}}{\Sigma_{l \in A_i}(InfoScent(e_l)^{\alpha} \times sim(e_i, e_l)^{\beta})} \tag{5.2}$$

where α and β are parameters that control the relative weight of the *InfotScent* and the content-based similarity between the social posts on edges e_i and e_j. A_i is the set of the adjacent content-sharing-edges of the edge e_i. If we take the example on Figure 1 and we suppose that the system is currently on the edge $e_i : (user6, user1, reposts)$ then there are two possible content-sharing-edges to visit which are: $e_j : (user1, user4, mentions)$ and $e_k : (user1, user2, repliesto)$. In order to make a decision and select the next edge to visit, the system will compute the probabilities $P(e_i, e_j)$ and $P(e_i, e_k)$ according to Formula 5.2. The values of the probabilities depend on the content (posts) of the edges e_i, e_j and e_k and on the user's interests vector.

The outcome of the social nets information foraging system consists in a surfing path containing all the visited edges during the process. The path starts from an initial post (edge) and ends with a target post which is supposed to contain information relevant to the user's interests. The number of edges contained in the surfing path is called the surfing depth. A relevance score can be associated with a surfing path, by assessing the similarity of the last post in the path and the user's interest vector V using Formula (3.1).

This way of computing the score of a social post is based exclusively on the concept of semantic similarity. While it ensures to offer a good ranking based on the relevance of the post to the user's interest, it doesn't take into account whether the information contained in the post is reliable or not. In the following subsection, we explain how we address the issue of information credibility.

5.2.3 Assessing the information credibility

Social networks give users the opportunity to spread news and information on different topics, which makes them content generators and not just content reviewers. This fact raises the concern of information credibility, especially during big events such as natural disasters and social movements where it is more susceptible for wrongdoer to use social platforms to spread misinformation and rumors. In [63], the authors propose a model-driven approach based on Multi-Criteria Decision Making (MCDM) and quantifier guided aggregation to assess the credibility of information on social media. They make use of ordered Weighted Averaging aggregation operators associated with linguistic quantifiers to set the number of criteria that can be used to identify fake review on Yelp.

In the present work, we use the MCDM approach to estimate the credibility of social posts. For this purpose, we consider several social features that we define as follows:

- n_r: number of re-posts of the current post.
- n_w: number of words in the post.
- n_f: number of social connections of the user (followers, friends,...).
- n_p: number of posts of the user.
- p_v: whether the profile of the user is verified or not.
- p_p: whether the user is using the default profile picture or a custom one.

We assign afterwards for each social post e_i a binary evaluation function ϕ such as :

- $\phi_{n_r}(e_i) = 0$ if $n_r = 0$, $\phi_{n_r}(e_i) = 1$ otherwise
- $\phi_{n_w}(e_i) = 0$ if $n_w < 3$, $\phi_{n_w}(e_i) = 1$ otherwise
- $\phi_{n_f}(e_i) = 0$ if $n_f \leq 10$, $\phi_{n_f}(e_i) = 1$ otherwise
- $\phi_{n_p}(e_i) = 0$ if $n_p \leq 10$, $\phi_{n_p}(e_i) = 1$ otherwise
- $\phi_{p_v}(e_i) = 0$ if $p_v = notverified$, $\phi_{p_v}(e_i) = 1$ if $p_v = verified$
- $\phi_{p_p}(e_i) = 0$ if $p_p = Default$, $\phi_{p_p}(e_i) = 1$ if $p_p = Custom$

We also define a parameter that represents the importance of each feature in giving a good estimation of the credibility of a social post. We set the values as follows: $I_{n_r} = 0.25$, $I_{n_w} = 0.8$, $I_{n_f} = 1$, $I_{n_p} = 0.9$, $I_{p_v} = 0.1$, $I_{p_p} = 0.95$.

5.3 A Muli-agent based Social Information Foraging System

In this section we propose a multi-agents based Social Information Foraging System, which employs the information foraging strategy described in section 5.2. By this approach the foraging activity motivated by an information need is carried out by several agents. We consider the social-graph representation of social networks as previously explained, and we assume that each agent browses a part of the social graph with the aim of finding relevant information. The system is composed of two kinds of agents that can work simultaneously and in a cooperative way to address the issue of information foraging:

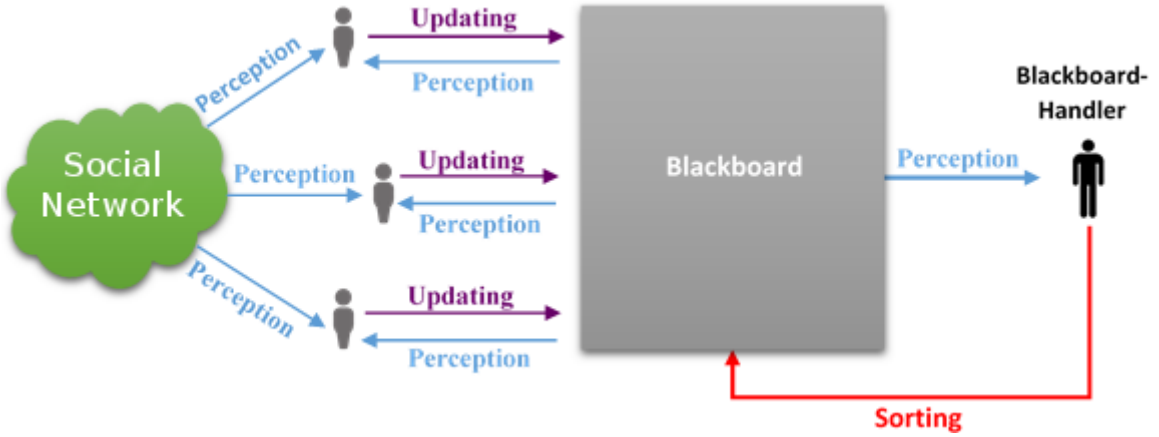

Fig. 5.2 Architecture of the Web Information Foraging Multi-Agent System.

- A group of foraging agents that have the task of foraging relevant information on the social network. Starting with the same user's interests, the agents forage information simultaneously with the goal of reaching relevant social posts;
- A coordinator agent that takes in charge the management of the system and the sorting of the partial solutions proposed by the foraging agents.

Figure 5.2 illustrates the architecture of the multi-agent system that is composed by the following modules:

Foraging Agents Each foraging agent has the task of simulating the rational behavior of a single Web user when searching for information on the Web. More concretely, the agent must determine at every click corresponding to a edge e_i the best move to another edge e_j.

At the beginning of the process, several foraging agents are launched at the same time and each of them starts its foraging from an initial node randomly selected on the Social graph. Each time a foraging agent makes a move towards a new social post using Formula 5.2, it evaluates its new partial solution (actual surfing path) and communicates it to the other agents by writing it on the blackboard. The blackboard-handler agent will then sort the solutions according to their relevance to the user's interests and will share the top solutions on the blackboard. If the proposed solution by the foraging agent is a top solution then the agent continues surfing on the same path, else the agent abandons its current surfing path and starts surfing again from a new node defined randomly. Algorithm 13 presents the pseudo-code of the foraging agents' behavior.

Blackboard System The blackboard system is an interaction model that allows information sharing between the agents in our system. More concretely, it is an area of shared

Algorithm 13 Foraging Agent

Input: a social graph structure;
Output: surfing paths ending with relevant social posts;

```
1:  procedure FORAGING AGENT
2:      Select a node at random ;
3:      Perceive the environment and check the neighborhood of the current node ;
4:      Move to a new social post using Formula 5.2 ;
5:      Add the new social post to the surfing path ;
6:      Communicate the surfing path to the other agents by writing it on the blackboard ;
7:      Wait for the blackboard-handler agent to sort the solutions ;
8:      Check the blackboard ;
9:      if the surfing path belongs to the top solutions then go to 2
10:     else
11:         Abandon the current surfing path and Go to 1 ;
12:     end if
13: end procedure
```

memory accessible in read and write mode by all the agents. The blackboard facilitates the collaboration between the agents as it offers them the possibility to exchange knowledge and share their partial solutions and suggestions for the sake of finding the best information sources that give answers to the user's interest.

Blackboard-Handler Agent The main function of this agent is to sort the solutions reported by the foraging agents in the blackboard according to their relevance to the user's interests. Each time a *foraging agent* reports a new solution in the blackboard, the blackboard-handler agent inserts it in the right position in order to keep the top solutions sorted using a *Binary Search Tree* as shown in Algorithm 14. The worst case runtime complexity of the Binary Search Tree insertion is $O(h)$, where h is the height of the tree. This means that the sorted solutions are continuously updated in real-time on the blackboard without any delay. Algorithm 15 shows the pseudo-code of the blackboard-handler agent.

5.4 Experiments

We describe in this section the experiments we undertook and we present the results we obtained using our Social Nets Information Foraging System. Note that we implemented and tested our system using Java Eclipse help System Base, version 2.0.2 on a PC with an Intel core I5-3317U Processor (1.70 GH) with 4 GB of RAM.

Algorithm 14 BinarySearchTreeInsertion(Node root, SurfingPath sp)

Input: surfing path *sp*
Output: *sp* position in the Binary Search Tree according to its relevance
1. if (root == null)
    return new Node(s)
2. if (sp.relevance $\leq$ root.key.relevance)
    root.left = Insert(root.left, sp)
3. else
    root.right = Insert(root.right, sp)

Algorithm 15 Blackboard-Handler Agent

Input: a surfing path *sp*
Output: a sorted list of surfing paths ranked by their relevance
1. Check the blackboard
2. Sort the surfing paths *sp* proposed by the foraging agents according to their relevance using Algorithm 14
3. As soon as a new solution is reported by a foraging agent insert it in the right position in the *Binary Search Tree* containing the top solutions
4. Update the top solutions list on the blackboard
5. Go to 3

5.4.1 The Dataset

The main goal of our experiments was to test our system on a social network using real-world users generated data. For this purpose, we considered the information sharing social network *Twitter*. We extracted content from twitter in order to construct multiple datasets related to different events and periods. We used *NodeXL* [57], which is a tool that offers the possibility to crawl data from different social platforms and provides access to the social graphs of the extracted data as well.

Table 5.1 presents the values of the empirical parameters used during the experiments.

Metric	Cardinality
Number of Foraging Agents	100
Top Results list size	30
Average User's Interest Size n	6
α	2
β	1

Table 5.1 Values of the empirical parameters

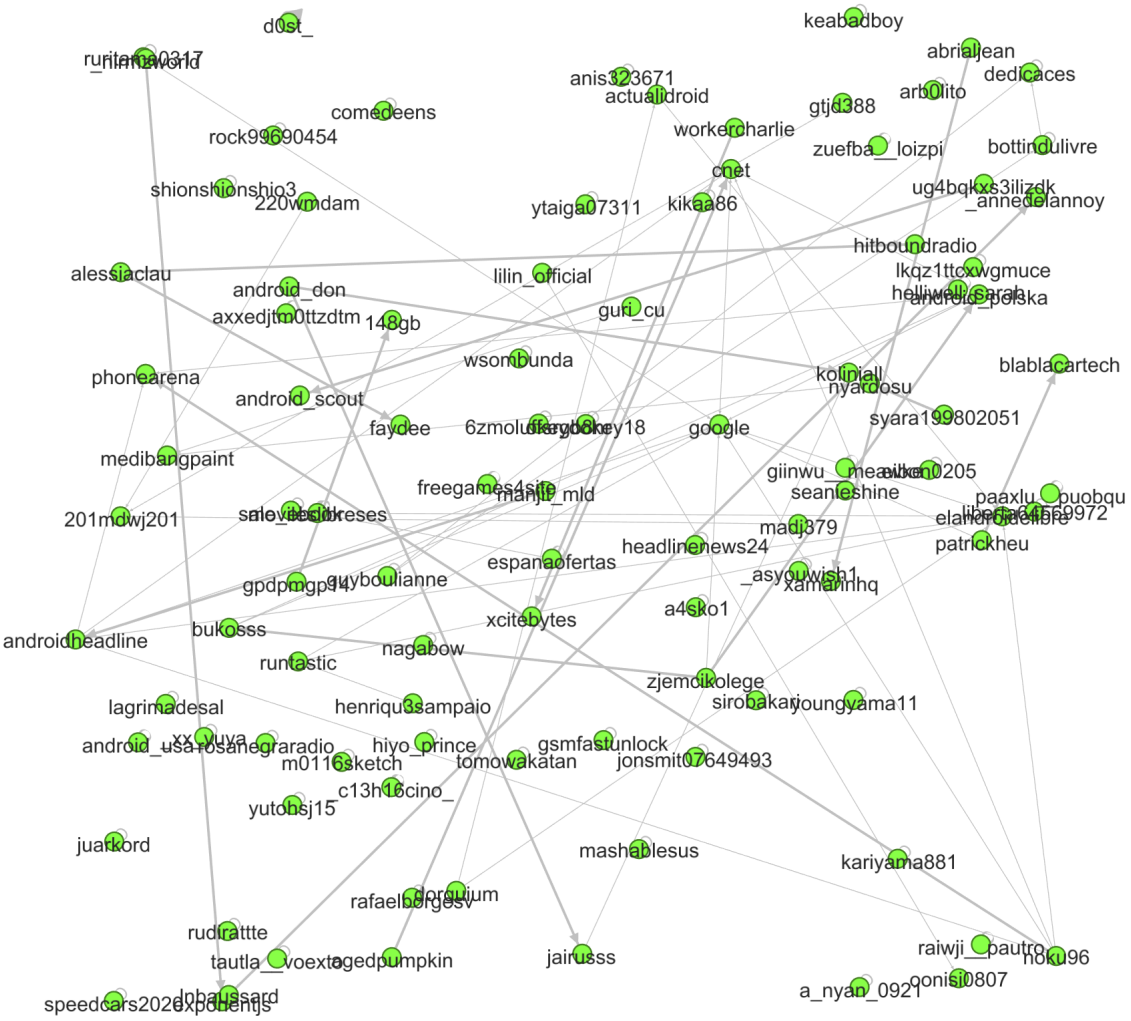

Fig. 5.3 A social graph generated with NodeXL

Figure 5.3 shows an example of a social graph generated with *NodeXL* using the keyword *"Android"*. The resulting directed graph contains 100 nodes and 118 edges. The nodes represent the users (twitter accounts), while the edges include the tweets of those users. Table 5.2 details the content of a part of the social graph. The *Source* column designates the username of the user who generated the tweet while the *Target* contains the username of the user to whom the tweet is directed. The *e-relationship* column describes the relation between the source and the target, which can be either a *tweet*, a *mention* or a *reply*. The *e-tweet* column contains the text of the tweet generated by the target user.

Source	Target	e-relationship	e-tweet
patrickheu	blablacartech	Mentions	RT @BlaBlaCarTech: Very proud to announce that we have been rewarded the "top developper" badge on our #Android app ! Thanks @google
_annedelannoy	lnbaussard	Follows	
patrickheu	google	Mentions	RT @BlaBlaCarTech: Very proud to announce that we have been rewarded the "top developper" badge on our #Android app ! Thanks @google
agedpumpkin	cnet	Mentions	RT @CNET: iPhone gives ground to Android in US and Europe https://t.co/XtvWCvIYcI
seanieshine	seanieshine	Tweet	I upgraded my Tavern and can now order: Vegetable Omelet! https://t.co/6R4NJfRQFb #android,#androidgames,#gameinsight
c13h16cino	_c13h16cino_	Tweet	I have Lumber Mill on my island! Now my island is even more awesome! https://t.co/acmQK7uqFc #android, #androidgames, #gameinsight
android_don	koliniall	Replies to	@koliNIALL @jairusss coincidence lang sorry hahaha
jairusss	koliniall	Follows	
jairusss	android_don	Follows	
android_don	jairusss	Mentions	@koliNIALL @jairusss coincidence lang sorry hahaha

Table 5.2 An example of the detailed content of a part of the social graph

5.4.2 Generating the user's interests vector

As mentioned in section 5.2.1, one of the new contributions in this chapter is the ability to implicitly generate the user's interest based on her/his social profile. Figure 5.4 shows the Twitter's official account on the homonymous social network. The figure highlights

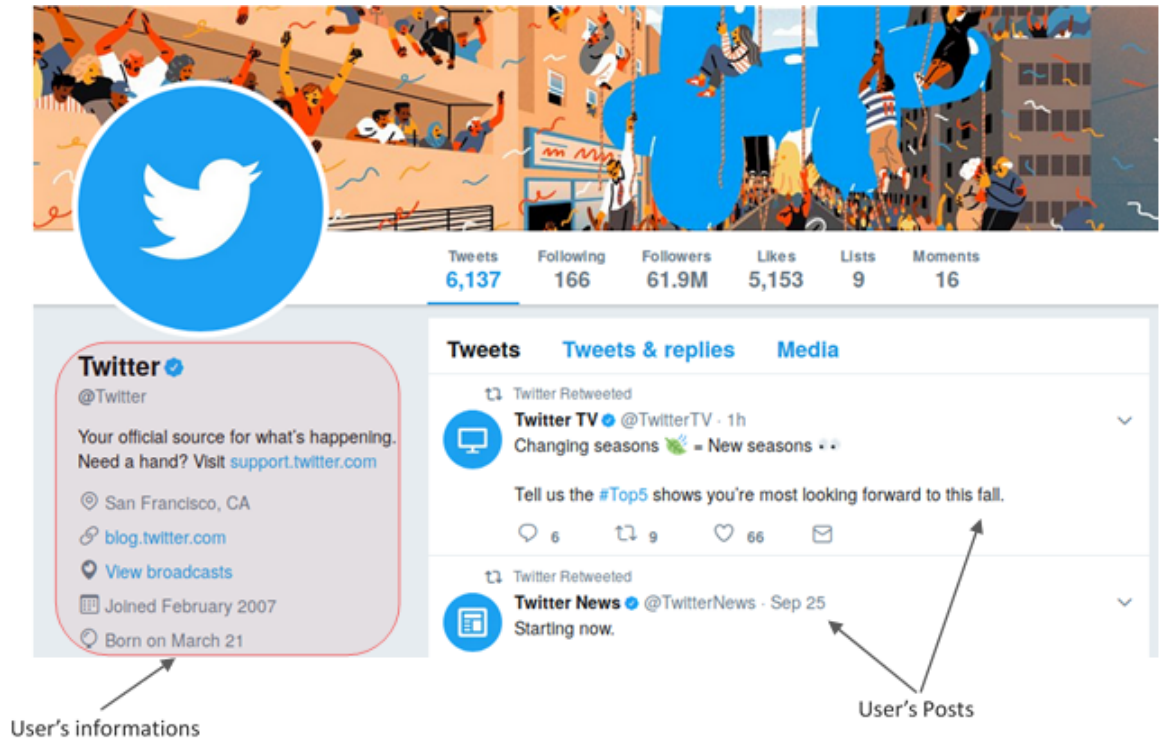

Fig. 5.4 Generating the user's interests form Twitter.

the information used to generate the interests vector including the biography and the latest tweets.

Table 5.3 and Table 5.4 show an example of different users' information extracted from the nodes of the social graph of Figure 5.3. The two tables were split for display purpose, they both expand social features of 10 different users including their usernames, their number of followers, the number of accounts they follow and their biography. In order to generate the users' interest vector we use their biography labelled as *"v-description"* in Table 5.3 along with their latest tweets.

Table 5.5 is an example of a user's interest vector generated explicitly using Algorithm 12 for the Google's official Twitter account during the period from September 28^{th} to October 3^{rd} 2017. The terms contained in the vector including "information", "question"and "learn" showcase the company's interest in the domain of information access. The rest of the terms are related to some of the company's products such as its own mobile operating system "android".

Id	v-followed	v-followers	v-tweets	v-favorites	v-description	v-location
runtastic	2007	38042	21157	8376	Official Runtastic Twitter account. We love reading your Tweets and hearing about your runs, workouts and progress! Twitter chats every 1st & 3rd Monday #rtchat	Linz, Austria, Europe
actualidroid	8	21	4944	0	Las noticias sobre el mundo #Android en tu timeline. Powered by @feedhi_	
google	440	13260154	8086	420	News and updates from Google	Mountain View, CA
androidheadline	231	231306	131909	113	All The Latest Breaking Android News & Rumors Covering Phones,Tablets, Apps, Reviews and More.	Los Angeles, CA
phonearena	0	54983	70451	1	PhoneArena is the ultimate source for cell phones - the latest news, in-depth professional reviews and huge phone specifications database!	
cnet	280	1070942	100496	954	CNET is the place to find out what's happening in tech and why it matters. CNET Magazine: https://t.co/ebMrur6pkM Snapchat: cnetsnaps	San Francisco
xcitebytes	489	989	2292	836	Are you a Movie Lover, Sport Fanatic or Music Enthusiast looking for the Ultimate in Unrestricted Entertainment, then meet the ROKit Media Streamers	South Africa
xamarinhq	406	36659	4104	10725	Build beautiful iOS, Android, Mac and Windows apps in C# and .NET with Xamarin.	
exponentjs	206	493	109	99	A home for experiences written with JavaScript and React Native	Palo Alto, CA
blablacartech	41	792	337	18	The Tech team of BlaBlaCar	Paris/ Warsaw

Table 5.3 Example of users' personal information extracted from Twitter - a

Id	v-joined twitter date (utc)	v-default profile	v-default profile image	v-geo enabled	v-language	v-listed count	v-verified
runtastic	10/02/2009 15:51	False	False	True	en	716	False
actualidroid	07/08/2015 19:37	True	False	False	es	13	False
google	10/02/2009 19:14	False	False	True	en	92125	True
androidheadline	25/06/2009 20:55	False	False	True	en	3538	False
phonearena	23/09/2008 10:27	False	False	False	en	1677	False
cnet	10/04/2009 17:10	False	False	True	en	23562	True
xcitebytes	22/09/2014 14:22	False	False	False	en	5	False
xamarinhq	16/05/2011 13:44	False	False	False	en	775	False
exponentjs	24/05/2015 22:58	True	False	False	en	13	False
blablacartech	27/10/2014 10:53	False	False	True	en	34	False

Table 5.4 Example of users' personal information extracted from Twitter - b

Username	User's Interests							
Google	google	parent	learn	question	andoid	information	treat	connect

Table 5.5 Example of a Users' Interests Vector

5.4.3 Defining the information credibility

We consider the 5 following features among the social features presented in tables 5.2, 5.3 and 5.4 in order to assess the credibility of the tweets:

- ***e-retweet:*** number of retweets;
- ***e-tweet:*** number of words in the tweet;
- ***v-followers:*** number of followers of the user;
- ***v-tweets:*** number of tweets of the user;
- ***v-default profile image:*** whether the user's Twitter profile is verified or not;
- ***v-verified:*** whether the user is using a default picture or a custom one.

The credibility of a tweet is the defined as described in subsection 5.2.3.

5.4.4 Foraging results

To test the performance of our Social Nets Information foraging system, we crawled data from *Twitter* during the first half of December 2015 using the keyword *"Paris"*. Recall that during this period, the *United Nations Climate Change Conference, COP 21* was being held

User's Interests	Prefered Language	Best Information Foraging Result	
		Tweet	Source Account
Planet Climate Futur Paris COP21 Deal	English	New draft climate deal emerges as Paris talks near end #COP21 https://t.co/GLVUyGzTK1 https://t.co/3xsNaSOdgT	@cop21_news
COP21 Pollution Climate Nature Life Earth better	English	As #COP21 continues, Earth breaches a feared pollution threshold. Does it really matter? #Climate https://t.co/RxJq8E1krM https://t.co	@business
Translate French Leader Paris Planet Climate Guide	English	Paris Conference Leaders Want You to Think the Planet Is Facing a Climate Change Crisis. https://t.co/S2Zyz1OE0R @DailySi	@bob4gov_now
Tourist Attraction Relax Tour Visit Vacation Paris Beach	English	Gpe_image: #Guadeloupe 🏝☼ #Antilles #Martinique #Paris #Sensation #Gi #Fwi #Westindies #Beach #Caraï… https://t.co/ZbP1Wj4OIu	@relaxincanada
Machine Learning Engineering Startup Etudiants Python	French	Développeur C++/Python/R dans un contexte "Machine Learning" @SAP #Paris https://t.co/ZKyWZH7KTq #machinelearning #Python	@SiliconArmada

Table 5.6 Information Foraging Results On Twitter

in the french capital. The size of the social graph we worked in is 16896 nodes with 17706 connections. Table 5.6 shows information foraging results on the graph for different users' interests generated implicitly by our system.

The first column of the table presents different user's interests, which were generated automatically by our system from random twitter accounts belonging to the social graph. The twitter accounts from where the interests were generated are not displayed in the table in respect to Twitter's confidentiality terms. The size of the interests vectors varies between 6 and 8 terms, which was crawled during 72 hours. The second columns of the table gives details about the preferred language of each user. The best result for each user's interests vector is shown the last two columns. Column 3 contains the most relevant tweet found by our system, while column 4 precises the source of the tweet.

The results on the table are a proof that our Social Nets Information Foraging approach is capable of finding relevant tweets with respect to the user's interest and her/his preferred language. Also, the content of the tweets provides from reliable Twitter accounts with a minimum degree of credibility.

5.5 Discussion

As mentioned in the beginning of this chapter, the issue of social networks information foraging is of an extreme importance. Indeed, information is shared by persons using social networks on a daily basis. As a consequence, these new infrastructures store relevant information related to different life domains and hot topics. Navigating on social networks seems to us of a greater concern than surfing on Web documents.

The approach of Social Nets Information Foraging proposed in this chapter was intended to complete our study on Web Information Foraging. We proposed a new approach to make our Web Information Foraging System able to forage information on social networks. We started our investigations on this topic by developing a model for social networks foraging. Then we designed and implemented a whole system to bring solutions to this question. Unlike the implementations presented in the previous chapters, we focused on generating the users' interests implicitly based on their data on social networks. We also payed particular attention to the credibility of the information since we are dealing with user-generated data.

We performed extensive experiments to evaluate our system. We first built a dataset using *NodeXL*, then we extracted a large social graph with more than 16000 Twitter users and 17000 connections using the keyword *Paris* during the period of December 2015. We presented some preliminary results on this subject and they are really promising, which encourages us to deepen the investigation of information foraging in social networks.

As perspectives, we plan to undertake massive experiments during different periods on various topics in order to show the usefulness of our developed system for domains such as news, climate, health and natural disasters.

Conclusions

Nowadays digital information is generated at unprecedented rates. A significant amount of this information is stored and accessed through the Web. The common way users access information is using search engines, which provide them with a simplified mechanism to discover and access information in order to satisfy their information needs.

Information access is currently one of the hottest topics in computer science. There are numerous research efforts employing different fields and disciplines with the objective of making information access easier, more effective and efficient, and more user-friendly. Among those disciplines we find *Information Retrieval* and *Information Filtering*.

Information Foraging is a relatively recent information access paradigm introduced for the first time in 1999. It aims to simulate the human behavior when seeking information on the Web. To do so, the *Information foraging Theory (IFT)* applies the ideas from the *Optimal Foraging Theory*, which was developed by anthropologists to model the animal food foraging behavior. A central concept in the IFT is *information scent*. This concept is based on the analogy between the scent animals rely on to estimate the chances of finding preys in the current area and the cues in the information environment used by humans to reach relevant information.

Several efforts have been undertaken during the past decade with the objective of implementing the IFT and developing new information foraging approaches. The majority of these works focused either on the theoretical side of the IFT or on testing information foraging on restrained environments such as companies' mailing lists, artificially generated Web logs, specific websites, etc. The goal of this dissertation was to develop using multi-agents technology an entire Web Information Foraging System (WIFS), which is capable of working on various environments in an effective and efficient way.

Research and Contributions

Based on the IFT and inspired from nature and biology, we designed in Chapter 2 an architecture for Web information foraging that adopts the analogy with animal groups' hunting. It includes two phases simulating the animal hunting strategy. These steps are

performed consecutively and translates efficiently the hunting process of a group of animals. The first phase consists in locating the most relevant Web pages based on a certain user's interests. It corresponds to the learning process of interesting regions an animal might use for hunting. The second phase focuses on the openness and dynamicity of the Web and takes advantage of the previous interaction with the user. It first checks if the system has knowledge about the user's interest. If so, it uses the results of the first phase to initialize an incremental learning, which consists in updating the most relevant Web pages taking into account the dynamic changes that occurred on the Web since the last interaction. Both phases are meant to be implemented with a multi-agent system simulating a group of animal hunting behavior.

In Chapter 3, we implemented our Web Information Foraging System using two *Swarm Intelligence (SI)* approaches. First, we considered *Bee Swarm Optimization (BSO)* that we carried out in several parts, among them:

- The implementation of BSO
- The adaptation of BSO algorithm to Web information foraging.
- Support the two phases of our system

The results achieved with BSO were satisfying and really encouraging. However, since we are dealing with the Web, which we consider as a graph, we thought of using a swarm intelligence approach that would more appropriate for graph structures.

For this purpose, we proposed a second implementation of our WIF system with a hybrid *Ant Colony Optimization (ACO)* and *Tabu Search* approach. ACO is known to be used for solving the *Traveling Salesman Problem (TSP)*, which constitutes an NP-hard problem in combinatorial optimization. Knowing that TSP is modeled using a graph structure, ACO seemed like the most appropriate SI approach to use. Indeed, this was confirmed by the good results we got with ACO and Tabu Search. We were able to improve the general performance of the system compared to the previous implementation using BSO.

We presented a new approach to Dynamic Web information foraging using self-interested agents in Chapter 4. Our main focus was to make the system scalable and test its efficiency on large datasets. We proposed for this purpose a model based on game theory and more precisely on normal-form games in order to simulate the Web surfing behavior of real Web users. This idea was inspired from the information foraging theory and from our findings on how real users surf on the Web following the previous solving attempts. In fact, our self-interested agents behave the same as real users when browsing the Web. When an agent has the choice to visit multiple Web pages, it selects the page that seems the more relevant

based on its description (title, URL, icon, tags...). Once the Web page is selected, the agent visits this page and checks its content in order to compare the provided information on the page with his information needs. We also proposed a normal-form game representation and introduced three kinds of players with a game strategy for each of them. The three kinds of players are considered as a team and coordinate together to find relevant Web pages thanks to the common-payoff configuration we adopted. Furthermore, we extended our model by adding a preprocessing phase using text classification. The goal behind it is to classify the Web pages (data collection) the system is working based on their content in order to make the foraging process faster and ensure the scalability of the system.

Social Networks have become recently among the most influential Web services in our daily life. In fact, billions of people use them to access and share text and media content, which makes them an important source of information. In Chapter 5, we proposed a new implementation of the WIF system to make it able to operate on *Social Networks* and take advantage of the social features and interactions between the users. One of the originalities of this work is to provide the system with the ability to implicitly generate the users' interest based on their social activities. We also payed particular attention to the credibility of the information since we are dealing with user-generated content. The results obtained with our system are promising in terms of the foraging process on social networks, the credibility of the foraged information and also the implicit user's interests generation.

Application Domains

One of the initial goals of this dissertation was to apply Web information foraging on different domains and environments and evaluate its contribution in each one of them.

The first idea we had was to exploit information foraging to provide patients with relevant medical articles based on their diagnosis or on some symptoms. For this purpose, we used medlineplus.com, which is a website produced by the USA National Library of Medicine. *MedlinePlus* contains around 2000 Web pages with information on more than 1000 diseases and health conditions. We generated different users' interests using a medical dictionary to check if our system is capable of finding relevant Web pages on the website based on those interests. The results were satisfying and showed the ability of the system to find surfing paths leading to relevant Web pages in a very short time.

Our second application domain was related to *Scientific Publications*. The purpose was to find relevant scientific publications based on a user's interests and taking into consideration some features such as the number of citations of the publications. We performed experiments on the *Citation Network Dataset V8*, which contains more than 2.3 million scientific publications indexed by *DBLP* and *ACM* repositories. Note that this is the first time an information

foraging system / approach is tested on a dataset with such size. The outcomes of our system confirmed its ability to work on large dataset and still be efficient and effective. The results also showed the superiority of our system over classical information retrieval in terms of efficiency.

Finally, we explored *Social media* and more precisely *Twitter*. We used our system to forage relevant information on *Twitter* according to a user's interest that we implicitly generate form her/his account. In our experiments, we generated a social graph containing more than 16000 nodes from data we crawled from *Twitter*. The achieved results demonstrate that our Web information foraging system is capable of finding relevant tweets emanating from credible accounts.

Perspectives and Future Works

The Web information foraging system we developed throughout this thesis is meant to work on different environments and mainly on Web structures. We were able to test it on a semi-structured website (MedlinePlus), scientific publication repositories (ACM and DBLP) and a social network (Twitter). We think it would be interesting to work on approaches that use *Web scraping* and *Web Content Analysis* and integrate them into the system. This will give us the possibility to extract informative content form any Web page and thus make the system able to work on any website even non well structured ones. Moreover, this addition would make performing experiments on different websites at once like presented in Section 2.3.3 possible.

Although our system already has the capacity to implicitly generate the users' interests using their personal data (online profiles and social networks accounts), it would be even better to take into consideration not only the interests but also the context and the preferences of the users. Thereby, the system would be able to construct a personal profile for each user and therefore enhance the foraging results. Furthermore, enriching the user's interests by the semantics of its keywords using a dictionary like *WordNet*, would be of a great benefit in our opinion.

Adding the social networks integration to the system was the last step in this work. We believe that this direction is promising and there are more social networks features to explore. We plan to do more research in this area to enhance the social posts credibility assessment and better investigate the social relations between users. Also, we can implement an automatic alert process that would send notifications to users whenever new relevant content is available for them.

Overall, although the field of Information Foraging has already more than a decade of study and experimentation, a significant number of areas and applications remain largely

unexplored. As previously mentioned this is the first time an entire Web Information Forging system is proposed and evaluated on big sized real-word datasets. It was one of the aims of this thesis to highlight some of the potential directions of research for Web Information Foraging systems.

Publications

The following publications have been made by the author during the PhD:

- Yassine Drias and Samir Kechid. 2014. Bees Swarm Optimization for Web Information Foraging. In Mining Intelligence and Knowledge Exploration - Second International Conference, MIKE 2014, Cork, Ireland, December 10-12, 2014. Proceedings. 189–198. DOI:https://doi.org/10.1007/978-3-319-13817-6 20.

- Yassine Drias and Samir Kechid. 2015. A Multi-agent Framework for Web Information Foraging: Application to MedlinePlus. In New Contributions in Information Systems and Technologies - Volume 1 [WorldCIST'15, Azores, Portugal,April 1-3, 2015]. 247–256. DOI:https://doi.org 10.1007/978-3-319-16486-1 25.

- Yassine Drias, Samir Kechid, and Gabriella Pasi. 2016. Bee Swarm Optimization for Medical Web Information Foraging. Journal of Medical Systems 40, 2 (2016), 40:1–40:17. DOI:https://doi.org/10.1007/s10916-015-0373-5

- Yassine Drias, Samir Kechid, and Gabriella Pasi. 2016. A Novel Framework for Medical Web Information Foraging Using Hybrid ACO and Tabu Search. Journal of Medical Systems 40, 1 (2016), 5:1–5:18. DOI:https://doi.org/10.1007/s10916-015-0350-z

- Yassine Drias and Gabriella Pasi. 2016. Web Information Foraging. In Proceedings of the 7th Italian Information Retrieval Workshop, Venezia, Italy, May 30-31, 2016. http://ceur-ws.org/Vol-1653/paper 26.pdf

- Yassine Drias and Samir Kechid. 2017. Dynamic Web Information Foraging Using Self-interested Agents. In Recent Advances in Information Systems and Technologies - Volume 1 [WorldCIST'17, Porto Santo Island, Madeira, Portugal, April 11-13, 2017]. 405–415. DOI:https://doi.org/10.1007/978-3-319-56535-4 41

- Yassine Drias and Gabriella Pasi. 2017. A collaborative approach to web information foraging based on multi-agent systems. In Proceedings of the International

Conference on Web Intelligence, Leipzig, Germany, August 23-26, 2017. 365–371. DOI:https://doi.org/10.1145/3106426.3106533

- Yassine Drias and Gabriella Pasi. 2017. A Collaborative Multi-Agent Approach to Web Information Foraging. In Proceedings of the 8th Italian Information Retrieval Workshop, Lugano, Switzerland, June 05-07, 2017. 106–115. http://ceur-ws.org/Vol-1911/20.pdf

References

[WWW] World wide web size.

[2] (2012). The acm computing classification system.

[3] (2017). "information access" in computer sciences. [online] http://www.encyclopedia.com/computing/news-wires-white-papers-and-books/information-access.

[4] (2017). "internet live stats. [online] http://www.internetlivestats.com/.

[5] Arbelaitz, O., Gurrutxaga, I., Lojo, A., Muguerza, J., Pérez, J. M., and Perona, I. (2013). Web usage and content mining to extract knowledge for modelling the users of the bidasoa turismo website and to adapt it. *Expert System With Applications*, 40:7478–7491.

[6] Berlt, K., de Moura, E. S., da Costa Carvalho, A. L., Cristo, M., Ziviani, N., and Couto, T. (2010). Modeling the web as a hypergraph to compute page reputation. *Inf. Syst.*, 35(5):530–543.

[7] Bharat, K. and Broder, A. (1998). A technique for measuring the relative size and overlap of public web search engines. *Computer Networks and ISDN Systems*, 30:379–388.

[8] Boyd, D. and Ellison, N. B. (2007). Social network sites: Definition, history, and scholarship. *J. Computer-Mediated Communication*, 13(1):210–230.

[9] Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA.

[10] Bretto, A. (2013). Hypergraph theory: Introduction. *Mathematical engineering*.

[11] Broder, A. Z. (2002). A taxonomy of web search. sigir forum. *SIGIR Forum*, 36(2):3–10.

[12] Chen, P. and Kuo, F. (2000). An information retrieval system based on a user profile. *Journal of Systems and Software*, 54(1):3–8.

[13] Chen, Y., Lai, J. K., Parkes, D. C., and Procaccia, A. D. (2013). Truth, justice, and cake cutting. *Games and Economic Behavior*, 77(1):284–297.

[14] Chi, E. H., Pirolli, P., Chen, K., and Pitkow, J. E. (2001). Using information scent to model user information needs and actions and the web. In *Proceedings of the CHI 2001 Conference on Human Factors in Computing Systems, Seattle, WA, USA, March 31 - April 5, 2001.*, pages 490–497.

[15] Chi, H. and Pirolli, P. (2006). Social information foraging and collaborative search. *HCIC Workshop, Frase CO*, 40:7478–7491.

[16] Dorigo, M. and Caro, G. D. (1999). Ant algorithms for discrete optimization. *Artificial Life*, 5-3:137–172.

[17] Drias, H. and Mosteghanemi, H. (2010). Bees swarm optimization based approach for web information retrieval. *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, 40:6–13.

[18] Drias, H., Sadeg, S., and Yahi, S. (2005). Cooperative bees swarm for solving the maximum weighted satisfiability problem. *Computational Intelligence and Bioinspired Systems*, pages 318–325.

[19] Drias, Y., Kechid, S., and Pasi, G. (2016). A novel framework for medical web information foraging using hybrid ACO and tabu search. *J. Medical Systems*, 40(1):5:1–5:18.

[20] Geraghty, P. (2012). Predator and prey: Adaptations.

[21] Goldstein, N. (2013). Animal behavior: Animal hunting and feeding. *New York: Chelsea House*.

[22] Graupmann, J., Cai, J., and Schenkel, R. (2005). Automatic query refinement using mined semantic relations. *Proceedings of the International Workshop on Challenges in Web Information Retrieval and Integration, WIRI'05*.

[23] Han, J., Kamber, M., and Pei, J. (2011). *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition.

[24] Huberman, B. A. and Adamic, L. A. (1999). Growth dynamics of the world-wide web. *Nature*, 40:7478–7491.

[25] Huberman, B. A., Pirolli, P., Pitkow, J. E., and Lukose, R. M. (1997). Strong regularities in world wide web surfing. *Science*.

[26] Iberkwe-SanJuan, F. (2007). Fouille de textes methods, outils et apllications. *Expert System With Applications*.

[27] Jordan, E. (2008). Online travel information search behaviors: An information foraging perspective. Ph.D Thesis, the Graduate School of Clemson University.

[28] Katz, M. A. and Byrne, M. D. (2003). Effects of scent and breadth on use of site-specific search on e-commerce web sites. *ACM Trans. Comput.-Hum. Interact.*, 10(3):198–220.

[29] Kechid, S. and Drias, H. (2009). Mutli-agent system for personalizing information source selection. In *2009 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2009, Milan, Italy, 15-18 September 2009, Main Conference Proceedings*, pages 588–595.

[30] Lamprecht, D., Strohmaier, M., Helic, D., Nyulas, C., Tudorache, T., Noy, N. F., and Musen, M. A. (2015). Using ontologies to model human navigation behavior in information networks: A study based on wikipedia. *Semantic Web*, 6(4):403–422.

[31] Lawrance, J., Bellamy, R. K. E., and Burnett, M. M. (2007). Scents in programs: Does information foraging theory apply to program maintenance? In *2007 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2007), 23-27 September 2007, Coeur d'Alene, Idaho, USA*, pages 15–22.

[32] Lawrance, J., Bellamy, R. K. E., Burnett, M. M., and Rector, K. (2008). Using information scent to model the dynamic foraging behavior of programmers in maintenance tasks. In *Proceedings of the 2008 Conference on Human Factors in Computing Systems, CHI 2008, 2008, Florence, Italy, April 5-10, 2008*, pages 1323–1332.

[33] Liu, J. (2003). Web intelligence: What makes wisdom web? invited talk. *Expert System With Applications*.

[34] Liu, J. and Zhang, S. W. (2004). Characterizing web usage regularities with information foraging agents. *IEEE Transactions on Knowledge and Data Engineering*, 40:7478–7491.

[35] Liu, J., Zhong, N., Yao, Y., , and Ras, Z. (2013). The wisdom web: New challenges for web intelligence (wi). *Expert System With Applications*, 40:7478–7491.

[36] Marchionini, G. (2006). Exploratory search: from finding to understanding. *Commun. ACM*, 49(4):41–46.

[37] McCart, J. A., Padmanabhan, B., and Berndt, D. J. (2013). Goal attainment on long tail web sites: An information foraging approach. *Decision Support Systems*, 55(1):235–246.

[38] Milgram, S. (1967). The small-world problem. In *Psychology Today*, volume 1, pages 62–67.

[39] Mirizzi, R. and Noia, T. D. (2010). From exploratory search to web search and back. In *Proceedings of the Third Ph.D. Workshop on Information and Knowledge Management, PIKM 2010, Toronto, Ontario, Canada, October 30, 2010*, pages 39–46.

[40] Murphy, J. and Olaru, D. (2009). How information foraging styles relate to tourism demographics and behaviours. *Journal of Vacation Marketing*, 15(4).

[41] Nikolaos Tselios, C. K. and Avouris, N. (1987). Beyond user centered design: A web design approach based on information foraging theory. *Inf. Process. Lett.*, 24(4):247–250.

[42] Ntoulas, A., Cho, J., and Olston, C. (2004). What's new on the web?: The evolution of the web from a search engine perspective. *Proceedings of the 13th International Conference on World Wide Web, WWW'04*.

[43] Obe, R. and Hsu, L. (2012). *PostgreSQL: Up and Running*. O'Reilly Media, Inc.

[44] Olston, C. and Najork, M. (2010). Web crawling. *Foundations and Trends in Information Retrieval*, 4(3):175–246.

[45] Osatuyi, B. and Mendonça, D. (2013). Temporal modeling of group information foraging: An application to emergency response. *Inf. Process. Manage.*, 49(1):169–178.

[46] Özmen, Ö. and Yilmaz, L. (2012). An agent-based information foraging model of scientific knowledge creation and spillover in open science communities. In *2012 Spring Simulation Multiconference, SpringSim '12, Orlando, FL, USA - March 26-29, 2012, Proceedings of the 2012 Symposium on Agent Directed Simulation*, page 1.

[47] Panigrahi, B. K., Shi, Y., and Lim, M.-H. (2011). *Handbook of Swarm Intelligence: Concepts, Principles and Applications*. Springer Publishing Company, Incorporated, 1st edition.

[48] Pirolli, P. (2007). Information foraging theory: Adaptive interaction with information. Oxford University Press, New York, NY.

[49] Pirolli, P. (2010). A cognitive model of information foraging on the web. In *Information Foraging Theory, Human Technology Interaction Series*. Oxford University Press, New York, NY.

[50] Pirolli, P. and Card, S. K. (1995). Information foraging in information access environments. In *Human Factors in Computing Systems, CHI '95 Conference Proceedings, Denver, Colorado, USA, May 7-11, 1995.*, pages 51–58.

[51] Pirolli, P. and Card, S. K. (1999). Information foraging. *Psychological Review*, 106(4):643–675.

[52] Pirolli, P. and Fu, W. (2003). SNIF-ACT: A model of information foraging on the world wide web. In *User Modeling 2003, 9th International Conference, UM 2003, Johnstown, PA, USA, June 22-26, 2003, Proceedings*, pages 45–54.

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

[54] Ranganathan, P. (2011). From microprocessors to nanostores: Rethinking data-centric systems. *IEEE Computer*, 44(1):39–48.

[55] Seeley, T., Camazine, S., and Sneyd, J. (1991). Collective decision-making in honey bees: how colonies choose among nectar sources. *Behavioral Ecology and Sociobiology*.

[56] Shoham, Y. and Leyton-Brown, K. (2009). *Multiagent Systems - Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press.

[57] Smith, M., Ceni, A., Milic-Fraylin, N., Shneiderman, B., Rodrigues, E. M., Leskovec, J., and Dunne, C. (2010). Nodexl: a free and open network overview, discovery and exploration add-in for excel 2007/2010/2013/2016. In *the Social Media Research Foundation*.

[58] Suh, B., Hong, L., Convertino, G., Chi, E. H., and Bernstein, M. (2010). Sensemaking with tweeting: Exploiting microblogging for knowledge workers. In *ACM conference on Human factors in computing systems, Atlanta GA*.

[59] Tahilyani, S. and Darbari, M. (2015). *Cognitive Framework for Intelligent Traffic Routing in a Multiagent Environment*, pages 67–76. Springer International Publishing.

[60] Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., and Su, Z. (2008). Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*, pages 990–998.

[61] Tao, X., Theng, Y. L., and Ting, T. (2009). Web user modeling via negotiating information foraging agent. In *Human-Computer Interaction - INTERACT 2009, 12th IFIP TC 13 International Conference, Uppsala, Sweden, August 24-28, 2009, Proceedings, Part I*, pages 436–439.

[62] Trepess, D. (2002). Information foraging theory. In *The Glossary of Human Computer Interaction*, pages 38–39.

[63] Viviani, M. and Pasi, G. (2016). A multi-criteria decision making approach for the assessment of information credibility in social media. In *Fuzzy Logic and Soft Computing Applications - 11th International Workshop, WILF 2016, Naples, Italy, December 19-21, 2016, Revised Selected Papers*, pages 197–207.

[64] Werner, E. and Hall, D. (1974a). Optimal foraging and the size selection of prey by the bluegill sunfish (lepomis macrochirus). *Ecology*, 55(5):1042–1052.

[65] Werner, E. E. and Hall, D. J. (1974b). Optimal foraging and the size selection of prey by the bluegill sunfish (lepomis macrochirus). *Ecology*, 55(5):1042.

[66] White, R. W. and Roth, R. A. (2009). *Exploratory Search: Beyond the Query-Response Paradigm*. Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool Publishers.

[67] Willett, W., Heer, J., and Agrawala, M. (2007). Scented widgets: Improving navigation cues with embedded visualizations. *IEEE Trans. Vis. Comput. Graph.*, 13(6):1129–1136.

[68] Winerman, L. (2012). In *Monitor on psychology. March 2012, Vol 43, No. 3*, page 44.

[69] Zhang, J. and Ackerman, M. S. (2005). Searching for expertise in social networks: a simulation of potential strategies. In *Proceedings of the 2005 International ACM SIGGROUP Conference on Supporting Group Work, GROUP 2005, Sanibel Island, Florida, USA, November 6-9, 2005*, pages 71–80.

[70] Zhu, Y., Zhong, N., and Xiong, Y. (2009). Data explosion, data nature and dataology. *Proceedings of the 2009 International Conference on Brain Informatics, BI'09, Springer-Verlag*.