Multi-Agent User-Centric Specialization and Collaboration for Information Retrieval

by

Abdelniser Mooman

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2012

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abdelniser Mooman

Abstract

The amount of information on the World Wide Web (WWW) is rapidly growing in pace and topic diversity. This has made it increasingly difficult, and often frustrating, for information seekers to retrieve the content they are looking for as information retrieval systems (e.g., search engines) are unable to decipher the relevance of the retrieved information as it pertains to the information they are searching for. This issue can be decomposed into two aspects: 1) variability of information relevance as it pertains to an information seeker. In other words, different information seekers may enter the same search text, or keywords, but expect completely different results. It is therefore, imperative that information retrieval systems possess an ability to incorporate a model of the information seeker in order to estimate the relevance and context of use of information before presenting results. Of course, in this context, by a model we mean the capture of trends in the information seeker's search behaviour. This is what many researchers refer to as the personalized search. 2) Information diversity. Information available on the World Wide Web today spans multitudes of inherently overlapping topics, and it is difficult for any information retrieval system to decide effectively on the relevance of the information retrieved in response to an information seeker's query. For example, the information seeker who wishes to use WWW to learn about a cure for a certain illness would receive a more relevant answer if the search engine was optimized into such domains of topics. This is what is being referred to in the WWW nomenclature as a 'specialized search'.

This thesis maintains that the information seeker's search is not intended to be completely random and therefore tends to portray itself as consistent patterns of behaviour. Nonetheless, this behaviour, despite being consistent, can be quite complex to capture. To accomplish this goal the thesis proposes a Multi-Agent Personalized Information Retrieval with Specialization Ontology (MAPIRSO). MAPIRSO offers a complete learning framework that is able to model the end user's search behaviour and interests and to organize information into categorized domains so as to ensure maximum relevance of its responses as they pertain to the end user queries. Specialization and personalization are accomplished using a group of collaborative agents. Each agent employs a Reinforcement Learning (RL) strategy to capture end user's behaviour and interests. Reinforcement learning allows the agents to evolve their knowledge of the end user behaviour and interests as they function to serve him or her. Furthermore, REL allows each agent to adapt to changes in an end user's behaviour and interests.

Specialization is the process by which new information domains are created based on existing information topics, allowing new kinds of content to be built exclusively for information seekers. One of the key characteristics of specialization domains is the seeker centric - which allows intelligent agents to create new information based on the information seekers' feedback and their behaviours. Specialized domains are created by intelligent agents that collect information from a specific domain topic. The task of these specialized agents is to map the user's query to a repository of specific domains in order to present users with relevant information. As a result, mapping users' queries to only relevant information is one of the fundamental challenges in Artificial Intelligent (AI) and machine learning research. Our approach employs intelligent cooperative agents that specialize in building personalized ontology information domains that pertain to each information seeker's specific needs. Specializing and categorizing information into unique domains is one of the challenge areas that have been addressed and various proposed solutions were evaluated and adopted to address growing information. However, categorizing information into unique domains does not satisfy each individualized information seeker. Information seekers might search for similar topics, but each would have different interests. For example, medical information of a specific medical domain has different importance to both the doctor and patients. The thesis presents a novel solution that will resolve the growing and diverse information by building seeker centric specialized information domains that are personalized through the information seekers' feedback and behaviours. To address this challenge, the research examines the fundamental components that constitute the specialized agent: an intelligent machine learning system, user input queries, an intelligent agent, and information resources constructed through specialized domains.

Experimental work is reported to demonstrate the efficiency of the proposed solution in addressing the overlapping information growth. The experimental work utilizes extensive user-centric specialized domain topics. This work employs personalized and collaborative multi learning agents and ontology techniques thereby enriching the queries and domains of the user. Therefore, experiments and results have shown that building specialized ontology domains, pertinent to the information seekers' needs, are more precise and efficient compared to other information retrieval applications and existing search engines.

Keywords: Information retrieval, Multi-agent, Specialized Agent, Reinforcement learning, search engines, specialized domains, personalization, IR system, user's feedback, NLP, WordNet, relevance information and feedback, semantic web, ontology, clustering, classifications.

Acknowledgments

I would like to express my deep and sincere gratitude to my supervisor, Professor Otman Al-Basir. His broad knowledge and logical way of thinking have been of great value to me as well as his understanding, encouraging, constructive comments that have provided me with personal guidance and a good base for writing my current thesis.

I would also like to extend my sincere gratitude to the members of my doctoral committee, Dr. David Chiu, Dr. Fakhri Karray, Dr. Hamid R. Tizhoosh, and Dr. Krzysztof Czarnecki for the time they spent reading this thesis and for their significant remarks and comments.

I also thank Dr. Brian Maguire of Computer Science Department, University of Regina for his valuable help during my initial stage of graduate studies.

I would also like to thank the Pattern Analysis and Machine Intelligence (PAMI) research laboratory team for the friendly atmosphere they have created, which was conducive to my studies. Special thanks go to Dr. Mohamed Kamel and friends Naser Younis, Gaith Zaied, Unmar Shabaz, Carol Sherman, Sara Clemens, and John Zagata for their support.

Lastly, and most importantly, I would like to thank my family. I owe a great deal of gratitude to my mother and my son Ahmed Mooman, as sources of my strength. I would also like to express my great appreciation to each of my brothers and their families for their continuous understanding and unlimited encouragement. It would have been impossible for me to finish this work without their moral support.

To My Parents

Contents

List of Tables

List of Figures

Glossary

MAPIRSO Multi-Agent Personalized Information Retrieval with Specialization Ontology

SIA Specialized Intelligent Agents

SA Specialized Agent

SLA Specialized Learning Agent

RL Reinforcement Learning

RLA Reinforcement Learning Agent

NLP Natural Language Process

TREC Text REtrieval Conference

TNIST National Institute of Standards and Technology

IA Intelligent Agent

QL Q-Learning

SARSA State Action Reward State Action

MAS Multi-agent system

MLA Multi Learning Agent

SMAS Specialized Multi-agent System

LDA Latent Dirichet Allocation

PLSA Probabilistic Latent Semantic Analysis

SAD Specialized Agent Domain

NLM National Library of Medicine

Chapter 1

Introduction

The Internet is undoubtedly one of the most important technologies of the modern world. Not only has it made our lives easier than ever before, but also plays a very important role in how our world communicates and operates. With the introduction of applications such as E-mail, instant chat and voice conversation, the Internet has brought global communication to the fingertips of users. These applications have not only made communication easy, but have also facilitated daily interactions among people around the globe. The Internet has significantly altered aspects of life commerce, employment, medicine, security, transportation, and entertainment, revolutionizing our lives in many ways. The Internet is one of the greatest inventions of our generation, prompting some people to suggest it has ushered in a new revolution as important as the industrial revolution.

The primary purpose of the Internet is information sharing. As soon as the Internet came into being, information could travel across the world almost instantly. This alone has impacted the practicalities of almost every industry in the world. It has changed the shape of administration, shortening the time it takes for documents to move from one place to another, making industry more productive.

The Internet has had a significant impact on the communication between people. Social media websites like Facebook, Myspace and Twitter have revolutionized the way we organize our social lives, while websites like YouTube and iPlayer have changed the shape of our entertainment.

1.1 Motivations and Inspirations

Due to the unprecedented ease of accessibility to the Internet and the exponential growth in information available to its users, it has become increasingly difficult for any information retrieval system to find relevant responses to user queries. Current information retrieval systems, such as search engines, can quickly locate information in response to a user query; however, the user is expected to browse through the response to determine if it contains information that is deemed relevant to what he/she is looking for. This process can be a tedious, and often, frustrating task. Moreover, even if it is determined by the user that the response does not contain relevant information, it does not mean the information the user is looking for is not available on the Internet.

Due to the growing amount of information available, conventional search engines inspite of their great success and contribution to resolve information retrieval (IR) problems, retrieve a countless number of documents solely based on keyword matching. Users are required to search through a large amount of information to select relevant information.

Various approaches have attempted to address this problem by clustering information into various categories through techniques such as the web mining methodologies applied by the "CLUSTY.COM" search engine. Other approaches use domain specific search engines to extract relevant information available on the Web to a particular domain, such MEDLINE (Medical Literature Analysis and Retrieval System Online) - an on-line database and search engine. Grammatical query enhancement and search auto-completion are solutions used by Google, Yahoo, and Bing search engines which are aimed at improving the information retrieval process. Personalization is another approach used to limit the amount of information retrieved by narrowing the information retrieved based on the user's profiles and preferences. Personalized content retrieval aims at improving the retrieval process by building personal profiles of individual users. However, the information retrieved can still contain a large number of results, which are based on matching keywords with the user's profiles - a user's profile may change at any given time and not all user preferences are relevant in all situations. Since human preferences are complex, and heterogeneous, categorizing information into unique domains does not satisfy each individualized information seeker. There has been notable success in looking at which words co-occur in articles in order to predict such group belongings.

The challenge of locating information relevant to a user's specific needs has been well recognized by the information retrieval research community and significant progress being made to address this challenge.

1.2 Challenges

Due to the vast amount of information available on the Internet and the complexity involved in imitating human behaviour, there is a need to develop a novel IR approach that will collect category specific information that matches the information seeker's preferences. It seems there are two dimensions to this challenge. One challenge depicts the user as an ill-defined entity due to the endless possibility of user ”wants and needs”. Users may present identical queries to the information retrieval system but expect completely different responses, resulting from a variation in user needs. The other challenge results from the amount of information available on the Internet and the overlapping topics or domains this information spans. Such vast information space is making it difficult for the information retrieval system to decipher the relevance of a given response as it pertains to the user's query.

1.3 Approach and Strategy of the Research

The central goal of this research work is to introduce, construct, and demonstrate a novel approach for IR that maps users to the information that they are looking for.

To achieve this goal, this thesis proposes a Multi-Agent Personalized Information Retrieval with Specialization Ontology (MAPIRSO). The proposed system offers a complete learning framework that is able to model the end user's search behaviour and interests and to organize information into categorized domains, to ensure maximum relevance of returned responses. Specialization and personalization are accomplished using a group of cooperative learning agents. Each agent employs a Reinforcement Learning (REL) strategy to capture end user behaviour and interests. Reinforcement Learning allows agents to evolve their knowledge of end user behaviour and interests as they function to serve him or her. Furthermore, REL allows each agent to adapt to changes in an end user's behaviour and interests. Users are an essential part of building such a system since the knowledge base of a specific domain is pertinent to user needs and feedback. This type of system is scalable with the ongoing growth of information and user needs. There is much room for improvement when one utilizes an intelligent agent to build a specialized knowledge base domain based on end user desires and feedback. Humans are considered to be the best document analyzers. Their ability lies in the capability to understand document contents and judge similarities based on the understanding. For instance, knowledge domains built based on human feedback (built by humans) are more desirable than having a system base results on keywords in knowledge space (Internet).

Practical trials (which will be discussed later) clearly show that when a specialized agent builds a knowledge domain based on human feedback, retrieving precision information results are improved. This provides more motivation to exploit research findings in involving human in building an intelligent system and apply them to IR (Information Retrieval) for categorizing and building domains of a specific domain lead to be better accessing, searching, retrieving, organizing, managing, and reasoning about information they contain.

The system will be comprised of components that

- facilitate an interface layer for users to interact with the system,
- provide multiple specialized learning agents that search and map relevant information,
- learn user behaviour and construct knowledge based domains based on user participation, and sharing information among the learning agents.

The goal is to utilize and engage these techniques to build on IR systems that enhance existing methods of information retrieval. This goal can be achieved by effectively introducing specialization into the knowledge domains that incorporate both the increasing amount of available/accessible information and dynamic user needs. This goal can be subdivide into distinct strategies, accomplished in the following order:

- Build specialized knowledge base domains of users' relevant interests and needs.
- Enhance user search criteria by enriching their searching queries.
- Reinforce learning through learning users' behaviour and through learned knowledge (past experiences).
- Devise a specialized learning agent for each specific domain.
- Support collaboration among learning agents by sharing learned knowledge and retaining system efficiency and effectiveness.
- Incorporate the learning process to dynamically update the knowledge base domains.
- Retain the efficiency of the information retrieval process by distributing tasks among multi-intelligent agents of the system.

1.4 Organization of the Thesis

This thesis is organized as follows: Chapter 1 presents the defined terms and states the objective and goals of the research. It also discusses some challenges that the work faces. Chapter 2 presents the background materials and reviews related work on IR systems on the bases of its structure, and involves users in the structure and then briefly shows the techniques of information retrieval. Chapter 3 describes the framework design and subject matter of the specializations of the proposed system. Domain topic extraction and query enhancement are presented in Chapter 4. System design and implementation are addressed in Chapter 5. Experimental results are discussed and analyzed in Chapter 6. Finally, a summary of the work, and a discussion of the research contributions, findings, and recommendations for future expansions are given in the final chapter; Chapter 7.

1.5 Summary

Mapping information that pertains users' interests requires the development of an intelligent IR system that involves learning the user's behaviour and retrieving relevant information efficiently and effectively. The available information through the Internet becomes overwhelming and complex in its structure, type, and sources as well as the topic domains the information represents. Thus, learning users' behaviour can provide means for an automated intelligent learning processing that requires human involvement in the building of specialized knowledge base domains. Inspired and motivated by the increase improvement of Artificial Intelligent (AI) applications and the users desire to find useful and relevant information from the Internet, this thesis proposes a novel IR system. The proposed system integrates AI and end users to construct a personalized and specialized multi-agent learning for IR system. The implementation of the proposed approach brings an effective knowledge discovery system. The basic principles of this approach are outlined, and some of its capabilities, requirements, and challenges, are discussed. This approach presents a novel solution to mapping users into desirable information and to segmenting the growing amount of information on the Internet, using specialized knowledge domain obtained through users' feedback. This thesis proposes a Multi-Agent Personalized Information Retrieval with Specialization Ontology (MAPIRSO) that offers a complete learning framework. This framework operates in a way that resembles human intelligence by imitating specific human behaviours in order to interact with users and learn their interests and patterns of behavior over time. This proposed approach contributes to the field of information retrieval by employing both specialized agents and the end user to construct user centric specialized knowledge domains that accurately and efficiently satisfy user information needs.

Chapter 2

Background and Literature Review

This Chapter presents a brief review and background of IR systems and applications, including definitions and basic concepts, goals and the criteria based on which IR systems are evaluated. Major IR system design models, such as classical Boolean, vector space, and ranking, are discussed. This Chapter also discusses how relevant research areas, such as machine-learning, artificial intelligence (AI), reinforcement learning (RL), and Natural Language Process (NLP), contribute to automating and enhancing IR systems by adopting to variation in user needs. IR search engines, their types, and user interaction approaches are reviewed. Finally, the chapter is concluded with a review of some IR systems and applications deemed relevant to the proposed IR systems.

2.1 Information Retrieval (IR)

Information retrieval is a field concerned with the structure, analysis, organization, storage, search and retrieval of information [98, 115, 99, 1]. It involves finding and retrieving information that pertain to a user's query from within an unstructured collection of information sources. In this process, a user must express their information needs in the form of a query containing terms or keywords, enabling the system to locate and retrieve information that matches the query. While the primary focus of IR has been on text and documents, recent applications of IR have increasingly evolved to incorporate new media, such as video, photos, music and speech. Since documents that are addressed by IR are specifically text documents, the term text documents will be used throughout this thesis. While information retrieval systems have existed and been used by researchers for over three decades, through the Internet the World Wide Web search engines have become well-known examples of retrieval systems [98, 1]. Nowadays, hundreds of millions of people use information retrieval systems daily. The IR field evolved to provide access approaches to searching many forms of content in response to the many challenges faced to provide information access. It began with scientific publications and library records, but expanded into other forms of information domains (unstructured information) such as medical records, news data, and law firm data archives. IR systems work with unstructured information, in contrast to database systems, which require highly structured information, and have a formal semantics. In comparison to expert systems (an expert system is a computer system that emulates the decision-making ability of a human expert.), IR systems do not derive or generate specific answers but instead return a set of documents whose content is relevant to the user's query. Although information retrieval did not begin with the Web, in recent years, the World Wide Web has been the principal driver of IR innovation, granting millions of users access to a large scale of information. Initially, IR system processing techniques, such as information indexing, were conducted manually. However, by the late 1990s, the available information and usage had grown at such a pace as to render manual indexing logistically impossible. Accordingly, various other techniques, such as machine-learning and Natural Language Process , were adopted and integrated with IR systems to enhance and automate the majority of manual tasks of IR systems.

2.1.1 The Basic Concept of IR Systems

Information retrieval (IR) can be simply expressed as a matching process, pairing the user's information need with the information source (School of Information Studies, 1998). The standard information retrieval system comprises four elements: the query (formulated by the user), a matching algorithm, an indexing process, and information resources. A query is the representation of a user's information need and consists of text terms or keywords. A user interface interacts between the user and retrieval system; i.e., Internet Web browsers.

Indexing is the process of selecting terms to represent a text. Automatic indexing functions assign terms to the full text document by way of applying Natural Language Process (NLP), such as tokenization, removal of stop words, and stemming. Information retrieval resources are typically a collection of text documents residing either in storage (static) or through the Web (dynamic). The main goal of IR is to find relevant information within the information resources about a given topic (query) that will satisfy the end user. Retrieved documents that satisfy the given query are evaluated by the user are said to be ”relevant” information. Fig. 2.1 depicts the basic IR system architecture.

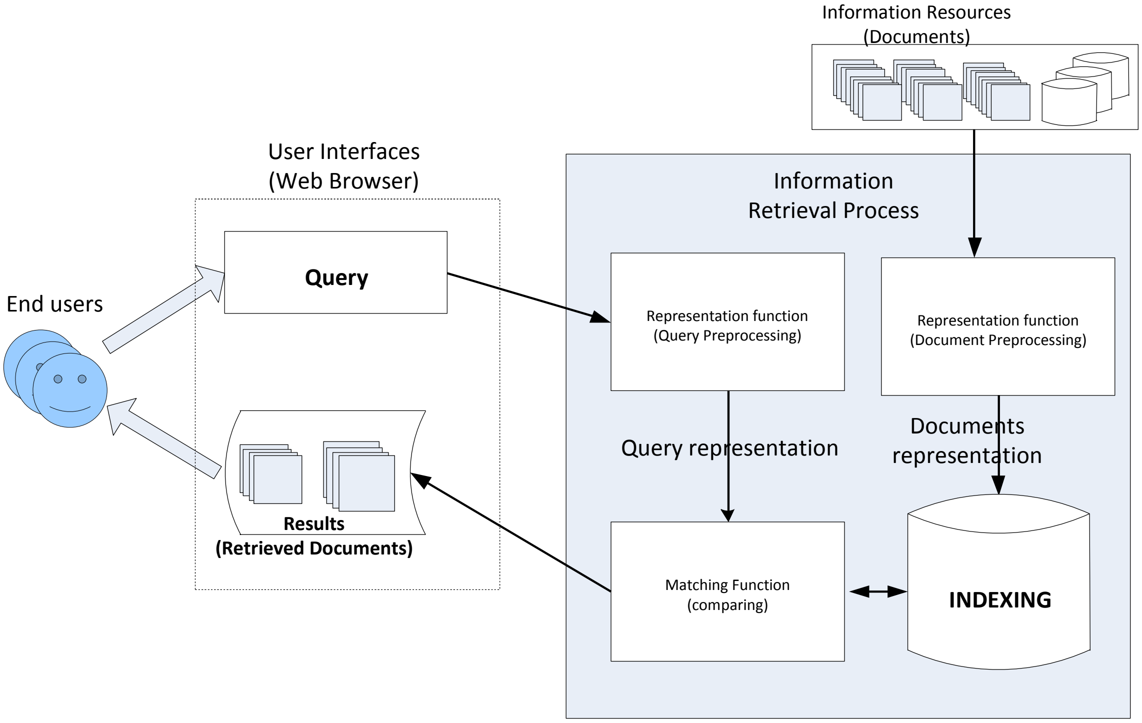


Figure 2.1: Basic information retrieval system Architecture.

2.1.2 Techniques Used in Information Retrieval

Some common model functions in current information retrieval techniques include the following:

- *Indexing*: Creating a full-text or keyword index.
- *Querying/Ranking*: Locating documents most relevant to the query.
- *Categorization*: Assigning documents to a given set of categories.
- *Clustering*: Mapping a data item into one of several clusters, where the clusters are inherent natural groupings of data items, based on similarity metrics or probability density models .
- *Classification*: Classifying the data into several predefined categorical classes.
- *Summarization*: A compact description for full-text documents.

It is evident that information retrieval techniques provide researchers and practitioners with a powerful means to extract and access useful information from the volume data resources.

2.1.3 Evaluating the Information Retrieval

The standard approach to evaluate IR systems targets the relevancy of retrieved documents to what the user is looking for. Evaluating the relevance of the retrieved document is a fundamental IR challenge. Judging whether a retrieved document is relevant or not is a subjective concept in that it depends entirely upon the end user's satisfaction with the retrieval results. A retrieved document may be relevant to the user's query, but not necessarily deemed relevant by the user if the document retrieved does not satisfy the user's need. Hence, measuring the relevance of a document retrieved by a given query depends on various factors: its relevance to the query; whether or not it satisfies the user's needs; the source of the document collection; the time when the user placed the query; and the priority position of the document in the list among the other documents. Various approaches have been proposed to address such subjectivity, including user profiling, personalization, and recommender systems based on the user's preferences.

Many measuring approaches of retrieval effectiveness have been proposed. Precision and recall are the most common IR success measurements and both are by far the most widely used [87, 98]. Both measurements are based on the concept of relevance to a given query and a user's needs. Furthermore, a benchmarking of information retrieval (IR) systems has been substantially adapted by the Text REtrieval Conference (TREC) [129]. TREC, an evaluation initiative for IR organized by the National Institute of Standards and Technology (NIST) and U.S. Department of Defense, was issued in 1992 as part of the TIPSTER Text program [129]. TREC includes a number of independent tracks, aimed at specific IR tasks which involve the design of appropriate test collections and evaluation measures.

As an example, a conventional TREC evaluation practice for an ad-hoc query track comprises of the following steps:

- Contributors are provided a data set, e.g., a collection of documents and a set of test topics. For each topic, they need to return to TREC organizers a list of retrieved documents.
- From each set of submitted retrieval results, the TREC organizers select the top N ranked documents to arrive at a pool of documents that will be manually judged.
- The collected relevance results are used to calculate the performance metrics for each system and topic pair. The most commonly used single-valued metrics are Average Precision (AP) and R-Precision [69].
- The overall system performance is typically characterized by the mean value of per-topic performance, i.e., the Mean AP value (MAP), which is then used to compare the systems.

Precision

Precision is defined as the ratio of relevant items retrieved to all items retrieved, or the probability that an item retrieved will be relevant [101]. *Precision (P)*, is expressed as an equation of the number of relevant documents retrieved divided by the total number of documents retrieved.

$$Precision = \frac{\#(relevant\ documents\ retrieved)}{\#(retrieved\ document)} = P(relevant|retrieved)$$

Recall

Recall is defined as the ratio of relevant items retrieved to all relevant items in a file [i.e., collection], or the probability given that a relevant item will be retrieved [101]. Other measures have been proposed [87, 98], but Precision and Recall are by far the most widely used.

Recall (R) is the ratio of relevant documents retrieved for a given query over the number of relevant documents. As an equation, Recall is expressed as follows:

$$Recall = \frac{\#(relevant\ documents\ retrieved)}{\#(relevant\ documents)} = R(Retrieved|relevant)$$

F-measure Hence, [87] derived Effectiveness measure, which allows users to specify the relative importance of Precision and Recall.

A single measure that trades off precision versus Recall is the *f-measure*. It is the weighted Harmonic Mean of Precision and Recall, calculated as follows:

$$F = \frac{1}{\alpha\frac{1}{p} + (1-\alpha)}\frac{1}{R} = \frac{(\beta^2+1)PR}{\beta^2 P + R} \quad where \quad \beta^2 = \frac{1-\alpha}{\alpha}$$

where $\alpha \in [0,1]$ and thus $\beta \in [0,\infty]$. The default balanced F-measure equally weights Precision and Recall, which means making $\alpha = 1/2$ or $\beta = 1$. It is commonly written as $F1$, which is short for $F_\beta = 1$, even though the formulation in terms of more transparently exhibits the F measure as a weighted harmonic mean. When using $\beta = 1 = 1$, the formula on the right simplifies to:

$$F_\beta = 1 = \frac{2PR}{P+R}$$

However, using even weighting is not the only choice. Values of $\beta < 1$ emphasize precision, while values of $\alpha > 1$ emphasize Recall. For example, a value of $\alpha = 3$ or $\alpha = 5$ might be used if the Recall is to be emphasized. Recall, Precision, and the F-measure take values between 0 and 1, but they are also commonly written as percentages, on a scale between 0 and 100. Precision, Recall and F-measure are set-based measures commonly computed for unranked documents (ordered).

Other Measures to Evaluate IR Systems

Other measures have emerged in recent years. These include:

- Average Precision - widely used in the research community, e.g., in Text REtrieval Conference (TREC). This type of measure expresses the inverse relation between Precision falls and Recall increases directly as a graph of Precision vs. Recall. Average precision is an attempt to summarize this of curve as a single value, e.g., for the purpose of comparing different IR algorithms, or even the same algorithm across different document collections.
- User-Oriented Measure [48] - proposed to address the variation of users' interpretations of which documents are relevant and which are not. User-oriented measure was introduced as coverage ratio, novelty ratio, relative Recall, and Recall effort [48]. The coverage ratio is defined as the fraction of the retrieved documents known to the user to be relevant. The novelty ratio is defined as the fraction of the relevant documents retrieved which are unknown to the user. A high coverage ratio indicates that the system is finding the majority of the relevant documents the user expected to see. A high novelty ratio indicates that the system is revealing to the user many new relevant documents which were previously unknown.
- R-precision - requires having a set of known relevant documents, from which the precision of the top relevant documents returned is calculated. R-precision adjusts for the size of the set of relevant documents .

2.2 General Information Retrieval Models

In general, IR technology and research consist of two major categories: semantic and statistical. Semantic approaches attempt to apply some degree of meaning and understanding through syntactic and semantic analyses of the natural language text relevant to the way human users would provide. On the other hand, statistical approaches apply some statistical measure (i.e., convert documents and queries into terms) to retrieve top-ranked documents that match the query. One common approach to document representation and indexing for statistical purposes is to represent each textual document as a set of terms. Most commonly, the terms are words extracted automatically from the documents themselves, although they may also be phrases, n-grams, or manually-assigned descriptor terms. Statistical approaches of IR can be categorized into the following models: boolean, vector space, probabilistic, and ranking model.

2.2.1 Boolean Model

In early retrieval systems, queries were represented as boolean combinations of terms, and the set of documents that satisfied the boolean expression was retrieved in response to the query [90]. The classical operators used in boolean queries are: AND, OR, and NOT. For example: The query *term1* AND *term2* is satisfied by a given document $D1$ if and only if $D1$ contains both terms *term1* and *term2*. Similarly, the query *term1* OR *term2* is satisfied by $D1$ if and only if it contains *term1* and/or *term2*. The query *term1* AND NOT *term2* satisfies D1 if and only if it contains *term1* and does not contain *term2*.

While the classical Boolean model is still in use today in some IR applications (i.e., search engines), it suffers from some drawbacks. Most significantly, it is difficult to manage the size of the retrieved set. The classical Boolean approach does not use term weights; the user is given no indication as to whether some documents are likely to be better than others in the retrieved set.

Boolean queries only express the appearance (true -"matches given documents")or non-appearance (false -"doesn't match given document") of some terms in a document. This model of queries is very limited, and cannot rank the results due to its all-or-nothing characteristic, that is to say, either the retrieved documents satisfy the query or they do not-there is no middle course. Several methods have been proposed and applied to refine the classical Boolean query model. Those methods include proximity operation and extended Boolean operations. In proximity operation , an additional Boolean operator is added to the classical set. For example, if a proximity operator is employed to the query *term1 AND term2*, the Boolean condition can be made to say that *term2* must immediately follow *term1* in the text. A proximity operator defines how closely in the text two terms must be to satisfy the query condition. The refinement of the classical Boolean model remains classical (either true or false) even with the addition a proximity operator. Extended Boolean Model [90, 96] is similar to the Boolean Retrieval Model, but with some additional operators included as term proximity operators. Extended Boolean operators make use of the weights assigned to the terms in each document. An extended Boolean operator evaluates its arguments to a number in the range from 0 to 1 , corresponding to the estimated degree to which the given logical expression matches the given document [96].

Although the extended Boolean model [96]enhanced IR performance in comparison to the classical Boolean model or the vector space model, it does have one drawback. Using extended Boolean queries demands an expertise in formulating the query domain versus Boolean queries, or a simple set of terms with or without weight, as in the vector space model.

2.2.2 Vector Space Model

The vector space model (VSM) or term vector mode, is another early retrieval model still used in information filtering, information retrieval, indexing, and relevancy ranking [73, 32]. VSM represents documents and queries by vector in multidimensional space, which has only positive axis intercepts. Smart [97], an information retrieval system developed by Cornell University, was the first IR system to apply the vector space model. For a given query, the process of VSM can be categorized into the following stages:

- Document indexing in which the content-bearing terms are extracted,
- Applying weight to the indexed terms,
- Computing the similarity between the input query and indexed documents.

As a result of the listed stages, the vector space produces a ranked list of documents. The document list is ranked based on the similarity of the document to the query. In the traditional vector space approach to IR, the collection of documents can be represented by a dimension of the space for each term occurred in the collection and as a vector for each document with coordinates for each term occurred in the document [73, 32]. The value of each coordinate is a weight assigned to the corresponding term, a weight intended to be a measure of how important the given term is in characterizing the given document and distinguishing it from the other documents in the collection.

Despite its effectiveness as the first approximation to the statistical properties of the collection, the VSM model's most significant limitation is that it assumes that the terms are independent, orthogonal dimensions of the document space. First, adding a new term to the space has no effect on the existing terms defining the space. Second, terms that co-occur in similar contexts in different documents are ignored. TF*IDF term frequency $*$ inverse document frequency, is the most widely used scheme to generate weight automatically to the term within the given document [32].

2.2.3 Ranked Retrieval Model

Ranked Retrieval Model is another statistical approach considered to be more complex than both the classical and extended Boolean models, yet is easy to use [131]. Model queries do not require Boolean operators, making them more user-friendly than Boolean queries. Furthermore, the documents retrieved are ranked by score, so the most representative documents to the query are listed at the top of the result. Nowadays, the ranked retrieval model is the most widely used within the IR systems and applications. Various search engines have adopted the Boolean operators' model as their "Advanced Search Option" in addition to the ranked retrieval model, to allow advanced users to be more selective in their results.

Other retrieval models exist, including several different probabilistic models and word proximity-based models. That is, retrieval effectiveness is not strongly influenced by the specifics of the model used as long as the model incorporates appropriate term weighting.

Term weighting, has been shown to have a primary effect on retrieval quality, with the best weights combining term frequency (tf), inverse document frequency (*idf*), and document length (*dl*) factors [3]. In this formulation, the tf factor weights a term proportionally to the number of times it occurs in the text, the *idf* factor weights a term inversely proportional to the number of documents in the collection that contain the term, and the *dl* factor compensates for widely varying document lengths. Another fundamental factor in the effectiveness of retrieval systems is good query formulation. Of course, the best way to guarantee a good query is to have the user provide one. Unfortunately, users tend not to provide sufficient context, usually offering only a few keywords as an initial question.

2.3 Machine Learning Approaches Used in IR

Machine-learning algorithms and approaches are used in IR systems and search engines to improve their functionality in various aspects. In addition to the automation of IR, processes such as document indexing, query refinement, word-relatedness, document classification, and document clustering are successfully applied in various IR systems and applications. However, knowledge representation and learning are considered areas that can be enhanced by applying some of the machine-learning techniques and algorithms. Learning is needed to improve the functionality of IR systems [18], since the objective of IR learning is to satisfy the end user by retrieving information that best matches their needs.

Both IR and AI (Artificial Intelligent) share the same objective of finding information. However, they archive this goal through different approaches: representation (AI) and anti-representation (IR).

The relation of IR and AI can be addressed in three aspects based information [119]:

- Knowledge representation. IR representation of entities is week: concept names are not normalized and descriptions of terms are instructed. The relationship between entities and terms in IR is only associated of co-occurrence. Various methods and techniques to model knowledge are currently existing in AI. Ontology can be considered as the generic term for generalizing these representation ideas.

- Reasoning. The strength in knowledge representation in AI provides the backbone for reasoning and also guarantees the reasoning.
- Learning. Though the relevant feedback of IR can be considered as a form of learning, learning is weak in IR.

Machine learning can be an ideal means to link IR and AI together to improve both approaches[30]. IR have demonstrated great success in finding information and allowing access to information despite the weakness of the model-based approach. They are facing problems handling the information overload and problems arising from knowledge management and electronic commerce. Nowadays, the manually generated ontologies cannot fulfil the increasing demands of ontologies, especially from the industrial side. Semi-automatically generating, mapping and evolving ontology have become interesting topics in AI, which some existing full-fledged techniques in IR could contribute. On the other hand, IR can further adopt ontology to refine and improve its search facilities.

There are three main categories of machine-learning approaches:

2.3.1 Supervised Learning

The goal of supervised learning, also called classification (pattern recognition), is to find a function mapping between the input and the output data. The learning agent aims to accurately predict the correct label on unseen data through a collection of labeled training data in which the agent knows the correct answer for each input. Some of the well-known supervised learning modes are: Naive Bayes classifier, Neural Networks, Decision Tress, and k-Nearest Neighbor classification

2.3.2 Unsupervised Learning

The learning process determines how the unlabeled collection of data is organized according to specific measures, such as similarity (as in data-clustering).

2.3.3 Semi-Supervised learning

Learning is achieved from a small collection of labeled training data. It combines both labeled and unlabeled examples to generate an appropriate function or classifier, e.g., Expectation Maximization (EM).

2.3.4 Reinforcement Learning

Every action impacts the environment, and the environment provides feedback in the form of rewards that guides the learning algorithm,(for example, Q-leaning, TD, and SARSA). The difference between these learning approaches is in how the learning process is conducted.

2.4 Intelligent Systems in IR

Advanced information technology has led to an increase in the complexity of information intensive systems, which require effective and efficient computation techniques to assist users in using the information for correct and rapid decision-making. Accordingly, AI and machine-learning research have addressed these challenges by designing systems and applications that simulate human intelligence to solve complex tasks accurately and efficiently. The measurement of intelligence depends on two prospective schools [135]: the *cognitive school*, where intelligence is measured by the level of cooperation of fairly complex agents, and the *reactive school*, where intelligence presupposes it is unnecessary for each agent to be individually intelligent to achieve intelligent behaviour. In the second school, agents are simpler and less intelligent, but more active, focusing on the cooperative working agents with low granularity [23, 80, 55]. Both information retrieval and artificial intelligence share the same task of finding information [63, 101] through different perspectives: representation (AI) and anti-representation (IR).

AI researchers model and represent knowledge in some logical forms due to their computational tractability, explanatory power, and inference function. While IR researchers attempt to retrieve information independent of any explicit data structure [7], AI has many characteristics that are suitable and can be employed to enhance information retrieval systems and applications, that include knowledge representation, reasoning, and learning.

2.4.1 Intelligent Agents

An intelligent agent (AI) is defined as a computer system that is situated in some environment that is capable of flexible and autonomous actions to meet its design objectives. Also, it has been defined as a computational entity that can perceive its environment through sensors, and acts in that environment through effectors [135, 5]. The various attributes of intelligent agents allow them to be used in differs system domains and applications. An intelligent agent is different from typical software applications in three fundamental ways: the agent reacts to and senses its environment at certain times (reactivity); it takes initiatives toward its goal (pro-activity), and socially interacts with other intelligent agents or users to reach their goals. Although there is no agreement on the definition of the term "agent," there is a consensus on the autonomy of the agent's core structure. If an agent can make its own local decisions, it is autonomous [135, 14, 71].

The following list classifies some agent attributes [14]:

- Adaptability: The ability to learn and improve with experience.
- Autonomy: Goal-directed, proactive, and self-starting behaviour.
- Collaborative Behavior: The ability to work with other agents to achieve a common goal.
- Inferential Capability: The ability to act on abstract task specifications.
- Knowledge Level Communications: The ability to communicate with other agents with a language that resembles more human-like "speech acts" than typical symbol-level program-to-program protocols.
- Mobility: The ability to migrate in a self-directed way from one host platform to another.

- Personality: The ability to manifest the attributes of credible users.
- Reactivity: The ability to selectively sense and act.
- Temporal Continuity: The persistence to identify states over time.

Overall, the definition of an agent can be classified according to the user perception of intelligent behaviour, design, and the intended application of the agent [14].

2.4.2 Multi-Agent Systems

A Multi-agent System (MAS) is an environment, where there is more than one autonomous agent. There are various definitions of MAS, depending on the application; including: "a federation of software agents interacting in a shared environment that cooperate and coordinate their actions, given their own goals and plans"[12].

There are several advantages to design an MAS instead of a single agent based system.

- To avoid the information and control overload of a single agent that needs to perform a domain specific task.
- To distribute various tasks and share the control load among the agents within a MAS.
- Agents can prospectively assist users discover information and interact with the users to reach desirable goals given the evolution of the environment.
- To achieve an adjustable autonomy, where agents not only achieve their own goals, but also adapt their autonomy according to the users' constraints.
- An agent's characteristics such as autonomy and sociability and the inherent distributed nature of multi-agent systems renders agents a promising tool in a dynamic environment.
- MASs are multi-threaded, where an agent controls one or several threads; intelligent agents observe the states of one or several threads for which they are designed.

- The data is decentralized and the computation is asynchronous, where a single agent is incapable of solving a problem on its own.

The environment plays an important role in how the MAS is designed, since researchers agree that agents are entities within an environment where they can act and interact with other agents or users. An MAS can be categorized as autonomous, adjustable, or mixed-initiative.

Autonomous Multi-Agent System

Traditionally, there are two approaches for designing a MAS:

- A system where all the tasks are delegated by humans to the intelligent agents;
- Intelligent agents that act autonomously on behalf of the user, after learning the users' interests over time.

When the agent has fulfilled its responsibility according to the designer's rules and specifications, the agent then faces a situation where the given rules and guidelines prompt it to make a decision based on the current situation. A reaction that can lead to an agent making an unacceptable decision [55, 29]. In some applications, autonomy is a suitable attribute, but the risk increases when a critical decision affects not only the decision but also the action of the agent. Some of the disadvantages of autonomous agents include:

- Increasing the degree of autonomous decision-making of the agent for complex and important tasks can lead to serious problems concerning the predictability of the system.
- The agent does not always have sufficient information to make a decision (i.e., dependency issue).
- The agents operate without the direct intervention of humans or others, and have some degree of control over their actions and internal state [40].

To avoid critical actions and states of such systems, the agent design should consider regulating the autonomous agent's behaviour by applying a mixed-initiative system, or an adjustable autonomy.

Mixed-Initiative Multi-Agent Systems

MASs in a dynamic environment tend to reduce the task load for single agents, as well as the users. However, the amount of complex information needed to perform such tasks can overwhelm to end users. Building and designing MASs based on mixed-initiative techniques addresses this problem by allowing and inserting the user into the MAS as an active participant. The user in mixed-initiative systems can provide guidance, whereas the system is able to perform data acquisition and management. The objective of mixed-initiative systems is to achieve efficient co-ordination and collaboration in an MAS. The coordination consists of the system's capability to allocate tasks, goals, and functions among the users and agents. Humans interact with the system through an interface agent which allows them to interface with the rest of the system. Accordingly, a mixed-initiative can be described a MAS that enables a team of agents (in which one or more of the agents is human and one or more is not) to collaborate to execute intelligent actions; (i.e., solving a problem). The term "mixed-initiative" emphasizes that neither a computer nor a user is solely responsible for taking the lead in the reasoning effort. Instead, the agent that has the most information seizes the initiative [117, 19].

There are three models of interaction among users, agents, and the associated MAS:

1. Client-server with the user as a client.
2. Client-server with the user as a server.
3. Peer-to-peer with both users and agents collaborating execute tasks, and where the human can participate in both client and server roles at any given time.

Some of the challenges in designing mixed-initiative systems in which agents and human work together in a seamless fashion are as follows:

1. Understanding the human and the agent's goals.

2. Employing the right amount of dialog between the user and the agents.

3. The timing of the actions and dialog.

4. Providing value-added automation.

5. Providing mechanisms for efficient collaboration.

6. Maintaining a working memory of recent interactions.

7. Employing socially appropriate behavior.

Multi-Agent Systems with Adjustable Autonomy

Here, the MAS environment is similar to the mixed-initiative system with respect to human involvement. However, the final decision is executed only after user confirmation. In adjustable-autonomy systems (AASs), an intelligent agent can be designed to autonomously perform controlled and predefined tasks without the need to consult a user before making a decision. For example, the information-collecting of specific domain data performing complex computations and processing tasks on which a final decision is not critical to the final outcome of the system [103, 102]. The continuous passing of control to and from the user is called Adjustable-Autonomy (AA). When an MAS is applied in a user environment, an interactive control is required. Some features, associated with the design of MASs with adjustable-autonomy, are as follows:

- The ability of agents to operate in a user organization.

- The ability to support humans through interaction and coordination.

- The ability to avoid mistakes.

- The possibility that the autonomy level of an agent can change dynamically.

- The ability to optimize the overall system performance.

There are three aspects of intelligent agents in designing an MAS: autonomy, mixed-initiative, and adjustable autonomy. However, these aspects are contingent upon the architecture and design specifications. There are two principal levels at which the MAS is designed, the domain level and the individual agent level. The domain level of the MAS design is the collaboration among the agents in the system. During the individual agent level design, each agent architecture is defined. Here, each agent has a specific set of functions, where specific information is stored or produced by the agent itself.

Agent-based technology, within the context of AI applications, offers a range of new architectures, techniques, and technologies that focus on the design and implementation of large-scale distributed intelligent systems. The use of multi-agent technology in distributed information retrieval, data mining and knowledge discovery tasks is an example of the new trend of information technology research. The combination of an MAS and information retrieval technologies is already established in various applications, such as Web intelligence (i.e., search engines). The background of Information Retrieval (IR) and its techniques are provided in the next section.

2.5 RL as a Machine Learning Approach to IR

Agent-based technology offers a range of new architectures, techniques, and technologies that focus on the design and implementation of large-scale distributed intelligent systems. To ensure the retrieved responses are relevant to what the user is searching for, a feedback mechanism is required from the user. The Reinforcement Learning algorithm is chosen to be embedded within the specialized agent for the purpose of learning the user's behaviours through feedback within an interactive environment.

RL appeals to many researchers because of its generality. The computer is viewed as an intelligent machine that, when given a problem to solve by trial and error, combines two disciplines (dynamic programming and supervised learning) to successfully solve problems that neither discipline can address individually. Such characteristics fit the attributes of IA of acting autonomously.

Definition: RL [123] is an intelligent approach with an autonomous entity which continually senses its inputs, takes actions by processing these inputs, and receives numeric rewards and punishments from its environment as a consequence of its action.

2.5.1 Key Features of RL

RL is an extension of dynamic programming, where RL is adopted to solve a set of problems [62, 140]. There are more significant differences between RL and other methods of machine learning, for example the supervised learning. RL is distinguished from supervised learning by the fact that the learner is not told the correct action for a particular state, but is instead informed how successful/unsuccessful the selected action is in producing the desired results. Unlike supervised learning, RL systems do not require explicit input-output pairs of training data. The traditional dynamic programming approach is limited by the size and complexity of the problems it can address, whereas the supervised learning approach requires sample input-output data pairs to be learned: a set of correct training data is required as a guide to solve the problem. Another way in which RL differs from supervised learning is that the evaluation of the system is often concurrent with on-line learning. Dynamic programming is defined as a field of mathematics that has traditionally been used to solve problems of optimization and control [49]. The agent's task is to learn by trial-and-error, and decide which action to take, whether to maximize the sum of the immediate rewards, or from a future reward. The objective of RL is to find a policy for selecting the actions that map states to actions so as to return a maximum number of rewards over time [123, 9].

RL characteristics are:

- Trial-and-Error search. The agent maximizes the reward if there is an immediate reward in the feedback, and minimizes the reward if there is no reward.
- Possibility of delayed reward. The RL agent considers the subsequent reward of the next state action.

- Little programming effort. Training and retraining can be done automatically and continuously on-line.
- No model of its environment is required. The RL agent's action must be observed during the interaction with the real environment.
- Incremental on-line learning. RL can interact directly with the users.
- Explore and exploit requirements.

2.6 Reinforcement Learning (RL) Model

RL interacts dynamically between an agent and its environment when the agent observes the environment in a certain state and chooses an action. There are four sub-elements to a RL system beyond the agent and environment. These include a policy, a reward function, a value function, and a model of the environment [123, 114]. The learning model responds either by reinforcing (also called rewarding) or punishing the agent's action.

On-Policy and Off-Policy

There are two types of policies in RL: on-policy and off-policy. *On-policy* is an algorithm method for updating the value function that uses the results from the executing actions determined by some policy. Value function updates are based strictly on experience [123]. *Off-policy* algorithms, on the other hand, can update the estimated value functions by employing actions which have not been attempted. Off-policy algorithms can separate exploration from control, whereas on-policy algorithms cannot. Consequently, the RL agent can end up learning tactics that it did not exhibit during the learning phase [123, 62].

Action Policy

A policy, π, is a description of the learning agent's behaviour and maps the transition from the perceived states, S, of the environment to the actions, A:

$\pi : S \to A$. The policy is the most important criteria in providing the RL with the ability to determine the agent's behaviour [123]. Three common polices are used for the action policy: ϵ-greedy, ϵ-soft, and *softmax*. *ϵ-greedy* is the method by which the action with the highest estimated reward is chosen most of the time; *ϵ-soft* is similar to ϵ-greedy in that the best action is selected with the probability 1-ϵ and, the rest of the time, a random action is chosen uniformly. *Softmax* action selection is different from ϵ-greedy and ϵ-soft methods in that it assigns a weight to each action according to the action-value estimate.

The goal of these policies is to balance the trade-off between exploitation and exploration [123].

Reward

A reward function indirectly defines the goal in a RL problem. A reward is a representation of a good or bad action. The reward function maps a perceived state (or state-action pair) of the environment to a single numeric number, usually $[0, 1]$, a *reward*, indicating the intrinsic desirability of that state.

Value Functions

Value functions are a state-action pair that estimates how good a particular action is in a given state, or what the expected return for the action is. The following notation is used for value functions: $V^{\pi}(s)$- the value of a state, s, is the expected return starting from that state S, depending on the agent's policy, π.

The Q-value function, represented as $Q^{\pi}(s, a)$, is the value of taking an action, a, in a state, s, under policy π is the expected return starting from that, s, taking that action, a, and thereafter following policy, π, is referred to as the *Q-Value function.*

Basic RL definitions:

Agent - A learning and decision making entity.

Environment - An entity the agent interacts with which cannot be changed arbitrarily by the agent.

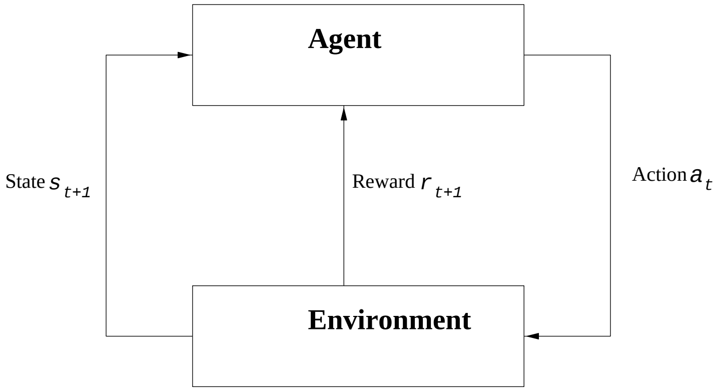


Figure 2.2: Reinforcement Learning (RL) base model.

State - The condition of the environment.

Action - A choice made by the agent, based on the state.

Rewards - The input upon which the actions are evaluated by the agent.

Figure 2.2 depicts an agent-environment interaction as a standard reinforcement-learning model. At each step, the agent and environment interact at each instant sequentially in discrete time ($t = 1, 2, 3, ...$). At each time step t, the agent receives some representation of the environment's *state*, $s_t \in S$, where s_t is a set of possible states. The agent chooses an *action*, $a_t \in A$, where a_t is a set of actions available in state s_t. One step later, the agent receives a numerical reward, $rt + 1$, and finds itself in a new state, as a consequence of the agent's action. The agent's behaviour will determine the actions that maximize the long-run sum of the reward values. Over time, the agent learns to do this by systematic trial-and-error.

2.6.1 Markov Decision Process

Markov Decision Processes (MDPs) [123, 28, 62] are the mathematical foundations for RL in a single agent environment [123, 62, 7]. MDP is a tuple $< S, A, T, R, \gamma >$, where S is a finite discrete set of environment states, A is a finite discrete set of actions available to the agent, T is a transition function given for each state and actions, R is a reward function of the agent, and γ is a discount factor ($0 \leq \gamma < 1$). In the MDP, the agent acts to maximize the long-run value it can expect to gain. Moreover, a stationary deterministic optimal policy exists for each MDP, and the RL algorithm must find an optimal policy by interacting with the MDP directly.

Typically, the learning task faced by an RL agent is assumed to be a MDP. Furthermore, in an MDP, the state transitions and rewards depend solely on the current state and the previous action. Agent-environment interaction becomes an MDP when the following criteria exist:

- Finite set of states, S.
- Finite set of actions, A: At each discrete time, an agent observes state, $s_t \in S$ and chooses action, $a_t \in A$, and then receives reward r_t, leading to a new state, s_{t+1}.
- Transition probabilities $P^a_{ss'}$:
  $P^a_{ss'} = Pr\{S_{t+1} = S^{'} | S_t = s, a_t = a\}$ for all $s, S^{'} \in S, a \in A(s)$.
- Reward probabilities $R^a_{ss'}$:
  $R^a_{ss'} = E\{r_{t+1} | S_t = S, a_t = a, s_{t+1} = s^{'}\}$ for all $s, S^{'} \in S, a \in A(s)$.

The probability of a transition from state s to state $s^{'}$ on action a is denoted as $P^a_{ss'}$, and the random reward associated with that transition is denoted as $r(s,a)$. Policy π maps each state to a probability distribution over actions. The probabilities are non-determining factors, in that the functions of both P and R are unknown to the agent. With policy π, the value function at each state is defined as

$$V^{\pi}(s) = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + ... = \sum_{t=0}^{\infty} \gamma^t r_t, \tag{2.1}$$

where r_t, is the reward received after the t the transition, the initial state s and following policy π. The discount factor, $0 \leq \gamma < 1$, renders the rewards in the future more valuable than the immediate reward. The task is to learn the optimal policy, π^*, that maximizes the value, $V^{\pi}(s)$, for each $s \in S$. This policy exists for each MDP [123, 28, 62]. The value function associated with π^* is denoted V^*:

$$V^*(s) = \max_{\pi} E\left(\sum_{t=0}^{\infty} \gamma^t r_t\right). \tag{2.2}$$

The value function is unique and can be defined as the solution to the simultaneous equation:

$$V^*(s) = \max_{a}\left(R(s,a) + \gamma \sum_{s^{'} \in S} T(s,a,s^{'})V^*(s^{'})\right), \forall s, \tag{2.3}$$

where $s^{'}$ is the random next state when executing action a in state s, and $R(s,a)$ is the expected value of $r(s,a)$. Given the optimal value function, the optimal policy can be defined as

$$\pi^*(s) = argmax_a\left(R(s,a) + \gamma \sum_{s^{'} \in S} T(s,a,s^{'})V^*(s^{'})\right). \tag{2.4}$$

The Q-function denoted by $Q^{\pi}(s,a)$, takes action a for starting state s for one step, and then follows policy π [123, 28, 62] such that:

$$Q^{\pi}(s,a) = R(s,a) + \gamma \sum_{s' \in S} T(s,a,s')x \sum_{a' \in A} \pi(s',a')Q^*(s',a'). \tag{2.5}$$

2.6.2 Q-Learning

Q-learning [123, 132, 7] is one RL algorithm that learns the utility values of the state and action pairs. The goal of Q-learning is to estimate the Q-values (state-action values) for an optimal policy. The Q-function Q^* for the deterministic stationary policy, π^*, is optimal for each starting state as defined by the following:

$$Q^*(s,a) = R(s,a) + \gamma \sum_{s' \in S} T(s,a,s')V^*(s), \tag{2.6}$$

where $V^{*}(s)$ is the value of s, assuming that the best action is taken initially, and

$$V^{*}(s) = argmax_{a' \in A} Q^{*}(s^{'}, a^{'}). \tag{2.7}$$

The optimal policy, or greedy policy, is defined according to the Q-function, Q^{*}, as $\pi^{*}(s) = rgmax_{a} Q^{*}(s^{'}, a^{'})$. In Q-learning, the agent uses its experience to improve its estimate by adding and combining the new information with the agent's previous experiences, which consist of a sequence of distinct episodes. These distinct episodes are described by a sequence of experience tuples, $< s_t, a_t, s_t^{'}, r_t >$. In Q^{*}, the values are the unique action that can be chosen by selecting the one with the maximum Q value for the current state. The Q-learning rule is defined as

$$Q(s, a) := Q(s, a) + \alpha \left[r + \gamma \max_{a' \in A} Q(s', a') - Q(s, a)) \right], \tag{2.8}$$

where $< s, a, r, s' >$ is an experience tuple.

This is the one step Q-learning (updates) equation, shown to converge for finite-state MDP problems, when a lookup table is used to store the values of the Q-function [28]. When the Q-function converges, the optimal policy, π, takes the action and predicts the highest reward at each state, s. The optimal policy, in terms of Q, can be defined by selecting the action with the highest expected future reward from each state: $\pi^{*}(s) = argmax_{a} Q^{*}(s, a)$. This is said to be greedy, as it consistently assigns the probability of "1" to an action in state s with the highest reward.

2.7 TD(λ) Temporal Difference Learning

The value function from the next state is used to estimate the current state value at each time step [112, 123, 62, 27, 49]. An RL algorithm learns by interactively reducing the discrepancy between the value function estimates and adjacent states. In (3.4), Q-learning, the reward is estimated in one step. With TD methods, an estimate of the final reward is calculated at each state, and the state-action value is updated. TD methods use an n-step reward estimation:

$$R_t^n = r_{t+1} + \gamma r_{r+2} + \gamma^2 r_{r+3} + ... + r^{n-1} r_{1+n} + \gamma^n V(s_{t+1}) \tag{2.9}$$

and

$$R_t^\lambda = (1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}R_t^n, \tag{2.10}$$

where $0 \leq \lambda \leq 1$.

2.8 Q-Learning and TD(λ)

Watkins [28] has suggested combining Q-Learning and TD-learning to accelerate training. The current update is designated to adjust the current estimated Q_t, as well as the previous states. Q-learning reduces the discrepancy among successive Q estimates. With (3.7), the equivalent expression of Q(λ)is

$$Q_t^\lambda = r_t + \gamma\left[(1-\lambda)\max_{a\in A} Q_{t+1} + \lambda Q_{t+1}^\lambda\right], \tag{2.11}$$

where $0 \leq \lambda \leq 1$.

2.9 SARSA

The SARSA algorithm [123, 56, 112, 7], an on-policy algorithm for TD-Learning, is an RL algorithm that depends on the actual learning policy that is being executed. The SARSA differs from Q-learning in that the Q-values are not updated according to the maximum reward of the next state. The SARSA algorithm uses the new action and, therefore, the reward is selected by the same policy that determines the original action. Updates are accomplished by using the original state and action, where reward r is observed and the next state-action is paired. In the SARSA algorithm, the action can be chosen randomly or by following some trajectory by a random policy. For the start state at each time step, the chosen action is greedy with the probability, $1-\epsilon$, and a random action with the probability, ϵ, for some small positive ϵ. The SARSA update algorithm is:

$$Q(s_t, a_t) = (1-\alpha)Q(s_t, a_t) + \alpha(r_t + \gamma Q(s_{t+1}, a_{t+1})). \tag{2.12}$$

SARSA is the same as the Q-learning algorithm, except that the value of the next state is not the maximum Q value. Instead, it is the Q value, associated with whatever action is chosen at the time $t + 1$. The action is the greedy action with the probability, $1 - \epsilon$. In this case, the update is identical to that of Q-learning. With the probability ϵ, the action is random, and the value that is backed up is lower.

2.10 Natural Language Processing in Text Information Retrieval

The unstructured, free-form, natural language text in which IR techniques are mostly used to retrieve information, does not have a well-defined syntactic or semantic in which the document is written or to which the document domain refers. Such IR technology and research are categorized as the semantic approach of IR. The semantic approach used in this section refers to NLP techniques that employ statistical techniques and linguistic processes. Many Natural Language Processing (NLP) techniques, including stemming, part-of-speech tagging, compound recognition, de-compounding, chunking, word sense disambiguation, and others, have been used in Information Retrieval (IR). NLP can be specified at both the query and document content levels. In recent years, NLP techniques have been used in conjunction with IR at the query level, including query enrichment, refinement, recommender, spell checking, and word tagging. There are two key approaches to NLP: a statistical approach as in latent semantic analysis and a linguistic approach as in WordNet. Both approaches differ considerably, even though in practice, NLP systems use a mixed approach, combine techniques from both.

2.10.1 WordNet

WordNet is a freely available online lexical database engine for English, utilized by various research fields such as the Natural Language Process (NLP), Information Retrieval and Artificial Intelligence communities. English words are organized into synonym sets (synsets), and each is represented by a lexical concept. A synset can have many words (synonyms) and one word can be a member of many synsets, one for each different sense [35]. WordNet represents both words and synsets, in which the relationships between words are lexical and those between synsets are semantic [35]. The goal of the WordNet approach is to support automated text analysis, sense disambiguation, term expansion in RL systems, and the structuring of representations of document contents. WordNet is also useful to determine semantic connections between sets of synonyms and for tracing morphological connections between words. WordNet provides a variety of semantic relations which are defined between concepts. The syntactic category of each word determines its semantic relationships. The following definitions clarify these relations:

Synonymy: Two concepts that have a similar meaning. Two expressions are said to be synonymous in a linguistic context C if the substitution of one for the other does not change the truth value. Synonymy is symmetric such that if x is similar to y, then y is similar to x.

Antonymy: Two concepts with opposite meanings. The antonym of a word x is sometimes not-x, but this does not held true in all cases.

Hyponymy/Hypernymy: X is a kind of y, where x is a more specific concept (hyponym) and y is a more generic concept (hypernym). H/H relationship is also called the subordination/subordination relationship, subset/superset relationship or ISA relationship. A concept represented by synset $x_1, x_2, ..., x_n$ is said to be a hypnoym of the concept represented by synset $y_1, y_2, .., y_n$ if there are sentences constructed from such frames as an x in a (kind of) y. On the other hand, hypernym is the opposite of hyponymy such that "tree" is the hypernym of "maple" and "plant" is the hypernym of "tree."

Meronymy/Holonymy: X is a part of y, where x is a concept that represents a part (meronym) of whole concept y (holonym). This relationship is also called the part-whole relationship, or HAS-A relationship. A concept represented by synset $x_1, x_2, ..., x_n$ is said to be a meronym of the concept represented by synset $y_1, y_2, ..., y_n$ if there are sentences constructed from such frames as a y has an x (as a part) or an x is a part of y. On the other hand, holonymy is the opposite of meronymy such that ”hand” is the holonym of ”finger.”

2.11 Search Engines

Search engines are IR resources that enable users to search information on the Internet [133]. There are many IR search engines available on the Web. These engines allow a user to submit queries and retrieve a ranked list of Web pages that match the user's query. Various research fields have contributed to the Web search existing today, including machine learning, NLP, AI, data mining, knowledge discovery, and many others. A search engine must create and maintain an index containing information about a set of Web pages. In general, most of the commercial, free, or specific search engines employ dynamic and automatic indexing document collections. Web search engines index and maintain partial (meta) keywords of the documents they find on the Web and present relevant onse to the user in the form of ranked search results [133]. General search engines consist of the following parts:

- A crawler that traverses the Web graph (in breath-first manner) and downloads Web documents,
- An indexer that processes and indexes the downloaded documents,
- A query manager that handles the user query and returns the search results of indexed documents to the user.

There are essentially four major types of search engines: Web crawler, Web portal, metasearch engine, and semantic Web. Each search engine type possesses different features and methods of providing its services, though they share the same IR objective, to find information relevant to the user's request.

2.11.1 Web crawlers

To index a Web page document, the search engine needs to find the document on the Web. That mechanism known as a Web crawler. Web crawlers, also known as robots or spiders, are almost as old as the Web itself [133]. The first crawler, Matthew Gray's Wanderer, was written in the spring of 1993, roughly coinciding with the first release of NCSA Mosaic. For the Web crawler to surf the Web and find documents, it required the Web addresses (URLs). In general, the Web crawler starts with a set of predefined Web addresses in order to find the documents and download them. Web crawlers are typically automatic, with the keywords stored in indexes, each of them associated with the documents where they were found within.

Web crawlers can be classified as either focused or unfocused. Unfocused Web crawlers create and maintain an index of pages, regardless of topic or site. Large-scale search engines usually apply the unfocused crawler mechanism. In contrast, focused crawlers create and maintain an index on a specific topic, type or some addresses (sites) of Web documents.

2.11.2 Web portals

In general, Web portals organize information on Web sites by topic to help navigate and locate that which the user is looking for [21]. In contrast to Web crawlers, users can define the search criteria and crawler search, and index the Web documents of those criteria. Portals are very efficient for finding common information, but they lack the ability to organize, as a result, specific information is not nearly as easy to find [21].

2.11.3 Meta-Search engines

Meta-search engines work as a user interface or an intermediary to a large number of search engines. The primary objective of using a meta-search engine is to take the user's query, employ several search engines, re-rank the documents identified, and present them to the user as if resulting from a single search engine. Such a scheme may increase recall and precision. Meta-search engines search the Web using three methods:

- Direct list of search engines: This kind of search engine sends the user query directly to a list of search engines and acquires their results for that query, as if the user directly posed his query in each of them individually. The benefit of this kind of search engine is that it saves the user time. This approach may also cover some search engines the user may never have sourced.
- Sequential searches: In this kind of search engine, a user selects search engines from a list and sends the user query to these selected search engines. Typically, the results are displayed just as they are returned from the search engines. These meta-search engines wait to receive all of the results and then display the result page, so speed corresponds to the slowest selected search engine.
- Concurrent search: This kind of meta-search engine is similar to the sequential search method, but it does not wait to receive the all results from every search engine before displaying. Rather, it receives the first search engine results it displays them, and new, received results are added gradually.

2.12 User Interaction with IR Systems

In general, IR systems consist of three major interdependent components: the end user, the IR engine, and the information resources. Users play an essential role in designing IR systems. In fact, IR systems are evaluated primarily on the user's satisfaction with the results returned by the engines. IR systems and applications, such search engines, are constructed and evolve around satisfying the end user's demands. Users interact with IR system in many ways, most commonly through browsing and searching.

2.12.1 Browsing

A user interacts with IR systems and search engines through a user interface. Web browsers are the typical methods for a user to interact with the IR systems. It enables end users to formulate and refine queries, review IR results, set up their profiles and preferences, and grants the ability to limit the number of document retrieved. Users interact with IR engine in many other ways. For advanced users, this can include building training sets, training the IR engines to classify documents, and guiding IR engines through a set of parameters. Most IR engines present the retrieval results in the form of listed, ranked documents. For the most part, search engines present the results as a list of document surrogates (title, author, source,..) to make it easy for the user to browse and select the desired document. In spite of its efficiency in retrieving information relevant to the user's query, a new set of problems has emerged. The most readily identified problem is that the number of retrieved documents is usually large. This, in many cases, leads to low precision, requiring a user to navigate through long lists in order to locate relevant information.

Various solutions have been proposed to enhance the way end users review and browse the results or information returned by the IR engine.

2.12.2 Direct and Interactive Searching

Efforts have been made to make users' interactions with the IR engines more convenient and effective. In contrast to reviewing and browsing, the end user engages with the IR system through formulating queries, refining original queries, and relevance feedback. Most IR systems support relevant feedback perform the reformulation and extension of the query automatically; the user is not aware the query is being refined. Aalbersberg [SIGIR '92] has suggested the automatic query reformulation approach using Rocchio [3], since each stage is modified in the query vector by either adding or subtracting a single document vector (if the document is judged by the user not to be relevant). Another approach suggested for interactive directed is by Roussinov et al. [89], where they suggest using unsupervised clustering to help the user refine and reformulate his query.

The idea is to automatically cluster the retrieval results of high-ranked documents from the list returned by the IR engine. Zamir et al. [34] later enhanced the on-line clustering of the retrieved document by developing an incremental clustering method, "Tree Clustering" (STC). The STC method is motivated by a problem that arises frequently when querying the Web with an IR engine, where a huge, ranked list of documents is retrieved of which only a very small number are relevant to the user's query. To make matters worse, the relevant documents are often far down the list of documents returned. In IR terminology, precision is often very low. Zamir et al. [34] proposed to alleviate this problem by clustering the documents returned to the user and labeling each cluster with phrases that intend to characterize its common topic(s).

The scheme is to submit a simple, natural language query to a Web IR engine (Roussinov et al. use Alta Vista). Roussinov and Zamir proposed methods fetches the 200 highest-ranking documents from the list returned by the IR engine. These documents are automatically clustered using an unsupervised clustering technique. The search engine, 'Clusty.com' is adopting a similar approach by categorizing the retrieved documents as a method to assist the end user in selecting the relevant information with respect to domain categories. In recent search engines, NLP linguistic techniques were employed to enhance and refine the end user's queries. For example, query and phrase auto spell, auto-finisher, and query recommender are applied in some of the search engines like Google.com.

2.13 A Brief Review of Some Related IR Systems Relevant to the Research Proposal

There are other related IR systems and applications designed to assist users to find specific information. This section addresses some of those IR systems that comprise techniques relevant to the proposed system, specifically, applications that include the user in the design, and those that employ other design techniques such as intelligent learning.

2.13.1 Web Spider Techniques

Jason Rennie and Andrew Kachites McCallum [86] have proposed a Web Spider technique that utilizes document classification and RL in a multi-agent form. In this approach, the focus is on how well the Spider retrieves relative information. The key feature of this approach is the search for information based on the document topic. Cora[23,24,25] is one of the search engine systems using Web Spider techniques and employs a machine-learning algorithm, such as the RL algorithm. The RL algorithm is structured to reward itself when it finds matches the related topic, and then follows the hyperlink of the found documents. The states are the sets of documents found about the same topic. This method is based on the training data found in those on-topic documents with their hyperlinks mapped into RL Q-learning algorithms by using naive Bayes text classifications. Web Spider is based on data sources found on the Web. In the proposed approach, however, the goal is to learn if the search results based on the document topics are related results, which can be considered as the RL reward instead of the users' feedback.

The hyperlink of the on-topic document is structured so that the tag of words reduces the state and action numbers. In the proposed approach, on-topic documents of varied data resources, including the Web data, are classified and categorized. Documents are mapped into the RL algorithm and the reward is based on the user's feedback. In the proposed approach, the RL is based on both on-topic related results and the user's feedback to process the learning. One strength of this approach is that it considers the future reward of links in their crawling priority, so the likelihood of crawling a link within an off-topic document that may lead to a reasonable amount of on-topic documents is high.

2.13.2 WAIR

Young-Woo Seo and Byoung-Tak Zhang [108] have suggested a method of applying RL to obtain relevant information by observing user behaviors during interaction. The proposed algorithm is called "WAIR" and operates by on-line evaluations of different agents: an interface agent, a Web-document retrieval agent, and a learning agent, derived from an RL approach. The task of the RL is to adapt to a user's profile and supply the Web document agent with relevant criteria, according to the user's modified profile. The user's feedback bookmarks the desired document, the time spent on each document, and revisits of the same document by the same user. Although the user's profile is constantly updating, the experiment environment depends on one data source. Moreover, the experiment duration is short to allow for user profile changes. A user's profile is a good place to start with the learning approach; however, the profile can affect the speed of learning if the RL checks the profile continuously in order to establish its learning. Thus, the program should be re-formulated, if the learning depends on many users at the same time, where each user has different profile and different behaviour. This novel approach employs the user's profile for the initial similar user's interest on during clustering , where the profile is not checked every time. Also, in this new, proposed approach, the user's profile is processed at the interface layer at the user's initial registration to use the system. In this case, the learning algorithm does not need each RL to dedicate its effort to only one user. Also, various data types that include unstructured Web data are used in this proposal.

There have been other related applications that intend to assist users in finding specific needed information. The focus is to review IR systems and approaches that employ machines-learning and intelligent-learning in their design. Table 2.1 depicts some of the related applications and their features, including the proposed learning model.

Application Name	Agent Structure	Learning Methodology	Data Resources	User
WBI	Multi-agent	None	Internet	Yes
SeTA & Intrigue	Multi-agent	None	Internet	Yes
MASPLANG	Multi-agent	None	Internet	Yes
WEB MINING	None	None	Internet	Yes
Amalthaea	Multi-agent	None	Internet	Yes
BASAR	Multi-agent	None	Internet	Yes
Electric Elves	Multi-agent	Yes (MDP); off-line	Specific data	Yes
Proposed Learning Model	Multi-agent	RL	Structured & unstructured data	Yes

Table 2.1: Related MAS applications.

2.14 Summary

In principle, the objective of designing and developing IR systems and applications is finding and retrieving the precise set of documents that pertains to the user's needs.

Traditional and current information retrieval systems and applications, i.e., Internet search engines, have addressed the problem of finding and retrieving information and made them available for end users to access. However, mapping the relevant retrieved information pertinent to the end user's tasks and specific needs is an ongoing and complex challenge for IR systems, especially with the exponential growth of different kinds of information. Information seekers must wade through large amounts of retrieved documents (mostly ranked documents) in order to find the desired information. To address such complexity, this research presents a novel approach for mapping users to the relevant information, through defining and constructing a specialized domain the user is interested in.

Various machine-learning, NLP, and agent-based techniques and algorithms are becoming essential components the design if IR systems and applications, to automate and enrich the efficiencies and effectiveness of information retrieval. An overview of those techniques used in IR systems and application were presented in this background. However, predominately, the focus of these techniques and algorithms have addressed the IR problems from the perspective of enhancing and refining the users' queries, ranking retrieved documents, automate d information indexing, and matching users' queries with the indexing information.

Current machine learning, NLP, and AI techniques and methodologies in IR systems can be scaled from impressive retrieval devices that respond reactively to the user's query into a specialized and intelligent system that learns and understands the user's behaviour and needs.

The proposed approach addresses the growing information by architecturing a framework of the IR system that specializes in domain knowledge pertaining to the user's needs and constructed through the user's feedback. The specialized knowledge domains are dynamically augmented by documents that are evaluated and selected by users through intelligent learning. Detailed descriptions of the proposed system framework and its components are presented in the next chapter.

Chapter 3

Specialized Multi-Agent System for IR: A Novel Framework and Subject Matter of Specializations

3.1 Introduction

Though traditional Information Retrieval (IR) techniques are useful in discovering and retrieving information that matches users' queries, the retrieved information does not always reflect what the user needs. Users still must review the retrieved information to determine which information is relevant to their needs. These continuous reviews are tedious and time-consuming, especially with the exponential growth in information.

IR systems and applications, *i.e.*, Internet search engines, have addressed the problems of locating information and finding information quickly regardless of location. However, mapping the relevant retrieved information to what the user actually desires and is looking for in an intelligent mode is an ongoing and complex challenge for IR systems. This research proposes an approach to IR that provides access to large amounts of information by organizing the search criteria based on the user's behavior, and establishes a specialized information agent in a multi-agent Reinforcement Learning (RL) paradigm. Specialized agents are established based on domains in which they provide their services through autonomous interactive learning with users in collaboration with other agents.

This chapter presents the proposed specialized multi-agent framework for IR as a collaborative learning environment. The framework consists of three layers: interface layer, multi-agent layer, and knowledge base layer. The interface layer is responsible for interacting with the user, pre-processing user queries and delegating tasks to the multi-agent layer. The multi-agent layer is comprised of specialized agents, where each agent contributes its own embedded intelligence technique to learn about a specific domain category. The specialized agents collaborate among each other by sharing relevant information. The knowledge domain space is related to the knowledge base and information access. Furthermore, this chapter analyzes what makes an intelligent agent a specialized agent, and addresses its structured components.

3.2 System Framework

Effective retrieval of relevant information is directly affected both by the user task and by the logical view of the documents adopted by the retrieval system. The challenges of designing Information Retrieval systems don't lie within the retrieving of desirable information from the repository data resources and presenting them to the user, but rather in the following problematic tasks:

- Identifying useful and desirable patterns of data that match the users' needs and requests;
- Predicting the users' expectations according to their previous patterns and behaviors;
- Involving users to evaluate new information and its presentation;
- Finding an efficient technique to categorize the learned knowledge;
- Combining relevant information under domain topics;

- Sharing the relevant information with other users searching for the same or similar information;
- Mapping the learned knowledge to the correct group of users seeking the same knowledge;
- Finding techniques that homogenize heterogeneous data.

To address these tasks, the proposed framework takes advantage of machine learning techniques, including the incorporation of an intelligent agent to perform complex operations on behalf of the user, and to apply learning algorithms to imitate human behavior. Figure 3.1 presents the proposed specialized multi-agent framework for Information Retrieval. A detailed description of each layer of the proposed Domain Specialized IR system is addressed in the following sections.

3.2.1 Interface Layer

Users are an essential part of designing an IR system. A user interface layer is needed as an intermediary between the system and users in order to interact with the information system, post queries, receive responses and evaluate domain concepts. An IR system bases its data search on the user's request and is evaluated based on the user's evaluation and satisfaction with the IR system performance, i.e., the relevance of retrieved information to the user's query. Therefore, the proposed specialized multi-agent system for IR is modeled around the user.

The interface layer comprises the following functions:

- User Interface: Interacts with the end user and the IR system. Captures user's request and feedback as well as presents the IR retrieved information using a web browser;
- Query Pre-processing: Reprocesses the user's query through stemming, removing stop words, and tokenizing;

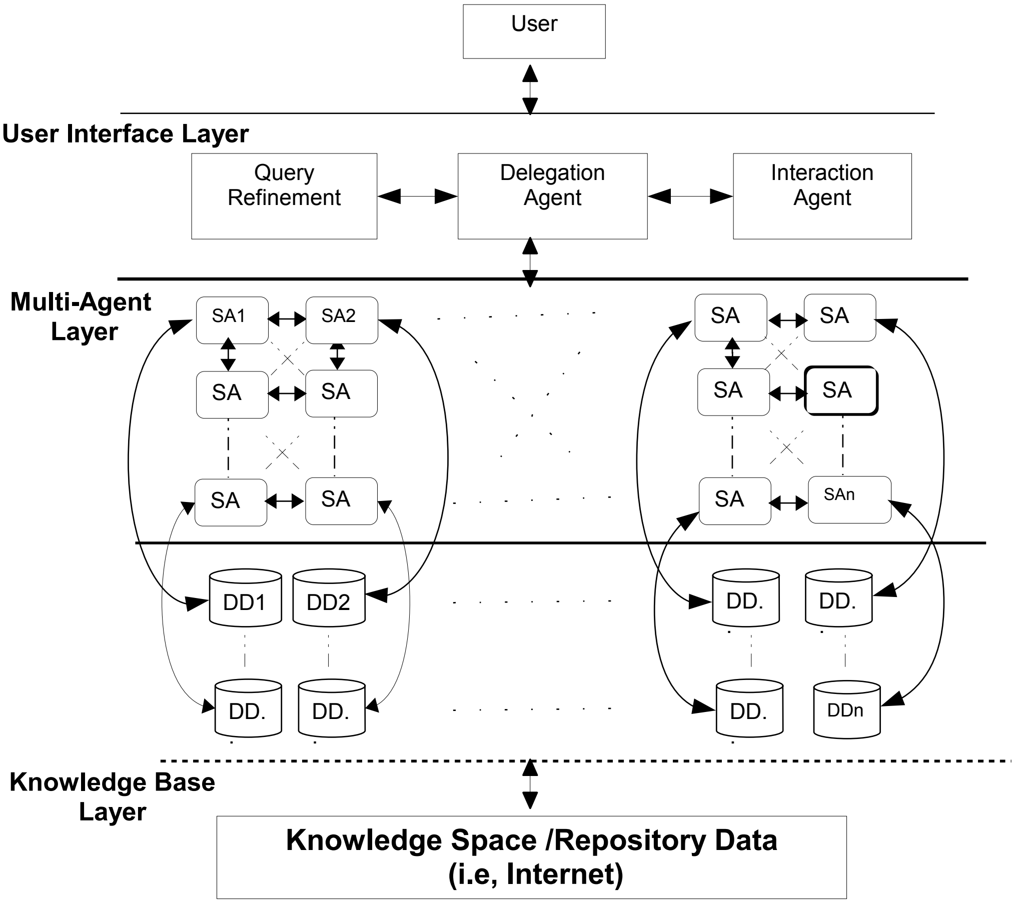


Figure 3.1: Specialized Multi-Agent Learning System for IR Framework.

- Delegations Agent:
  - Collaborates with the multi-agent layer agent through the bulletin-board to identify the existing specialized agents and their domains;
  - Delegates user's queries to the multi-agent layer and autonomously decide which specialized domain to choose based on the content of the query;
  - Trigger an action to construct a new domain if no domain is found to address the user's query.

User Interface

The user interface is established on a Web browser as an interactive means between users and IR systems. Users will be able to insert queries, and view the IR results through the Internet browser. Document filtering, query enrichment, learning process; user actions and feedbacks are processed autonomously within the proposed system.

The Internet Web browsers have become the vehicle to an increasing range of everyday activities. Web browsers have become the indirect management interface for interaction between the user and computer applications, so called "autonomous agents" in the AI field. The hypothesis is that users will use the Internet Web browsers as an interface medium to acquire (search) information, view, and evaluate, while the system process, learns, and presents the desirable information to the users. Also, users' feedback to evaluate the system's retrieved information results is presented through the interface.

Interface agents such as the Web Browser radically change the style of human-computer interaction and software application. Information can be exchanged locally (Intranet) or externally based (Internet), while considering the security of the information in regards to accessibility. Web browser such as Firefox, IE, and Opera are commonly used through personal computers as well as mobile media. The use of a Web Browser as an interface between the user and the system has the following features:

- Easy to learn;

- Platform independent, can be placed on any platform;
- A dynamic client interface that can be customized based on the user's preferences;
- Support various MIME types. This enables the Web browser to display or output files that are in various format;
- No need for the user to understand the interior interactions of the system

Interface design is part of the system that presents the information retrieved to the user and evaluates it through the web browser interface. The user's interface is focused on the user's behavior through interacting with the system. Aside from the Web Browser, the user's interactions with the system are monitored by managing the user's activities, what has been presented, and capturing the user's feedback and behavior. All these components are invisible to the user.

Query Preprocessing

The objective of introducing intelligent user queries pre-processing at the interface layer is to enhance the research result by retrieving relevant data, and to purify the query to identify what the user wants. The pre-processing of user queries occurs before they get to the multi-agent layer. User queries need to be evaluated to solve the problem of the query having too few useful terms or too many extraneous ones. The user's query is presented as keywords or phrases when searching for information. Extracting higher-level concepts from the user's keywords or phrases is one of AI and Machine Learning's ongoing research interests. Query terms are often too imprecise and studies have proven that the average query text consists of 1.8 words [127, 84, 107]. The user's query is considered to be the main factor in evaluating the precision of the information retrieval (IR) in various applications, such as search engines. Extracting the concept of the user's query is important to improving retrieval performance through mapping users to the relevant information.

A specialized multi-agent for the IR system resolves such problems in two stages: pre-processing the user queries before processing the search, and evaluating the information retrieved through the user's feedback. Furthermore, the proposed system design addresses users' queries individually. User queries quite often do not represent what the user intends to look for, because each user has a unique and different characteristic that drives their interests. Consequently, users would have unique interests in spite similarities in their search queries or users' characteristics such as age, gender, or interests. In order to eliminate the possibility of misrepresenting users' needs, the proposed system adopts each query of the end users.

Natural Language Process (NLP) pre-processing techniques such as stemming, removing stop-words and tokenizing are applied to purify the end user's queries before passing them to the delegation agent. Furthermore, to narrow the scope of mapping users to relevant information, this research presents an additional step to enrich the user's query by infusing semantic lexical terms to the query. This delegation approach has the effect of narrowing the scope of mapping users to relevant information.

Delegation Agent

A delegation agent is used as a communication method between the interface layer and specialized agents within the multi-agent layer. In particular, it delegates tasks to the specialized agents based on the query refinement produced by the pre-processing phase. Delegation is conducted through collaboration among agents across the system layers using Bulletin Board [74]. Determining whether a query can be mapped to an existing specialized agent or whether to establish a new specialized agent for a new domain is processed through the Bulletin Board, whereby each specialized agent is referenced through its unique specialities (keywords). Each user will be mapped to the relevant specialized agent to process the user's query, and evaluate the learning agent's results through the user's feedback.

3.2.2 Multi-Agent Layer

The Multi-Agent Aayer is comprised of several cooperative intelligent specialized agents, where each agent is constructed of an RL algorithm specialized to perform learning in specific categories and capture end user behaviour and interests. An intelligent specialized agent is triggered or initiated by the delegate agent at the interface layer to address the user's query and process learning at the multi-agent layer. The intelligent learning is performed by Reinforcement Learning (RL) to determine which information to present to the end user and build a knowledge base of a specific domain through end user feedback.

Some features associated with the design of this type of system of intelligent multi-agents are as follows:

- The ability to distribute various tasks and share the control load among agents within MAS. An RL learning process is initiated for each user's query;
- The ability to support users through interaction and co-ordination;
- The ability to facilitate collaboration among agents in the process of sharing knowledge.

Specialized agents at the multi-agent layer function can be summarized as having the following activities:

- facilitating collaboration among the agents in the process of sharing knowledge;
- building a knowledge base on a specific domain through interaction with the end user through the RL learning algorithm;
- constructing and updating a knowledge base repository of a specific domain;
- performing indexing and ranking of the agent domain knowledge base;
- constructing a summary of existing information of the agent domain knowledge base.

Learning Agents

Typical IR systems focus on retrieving information for the user based on the entered query, however, the critical part of this process is to identify whether the retrieved information is truly what the end user is looking for. Machine learning techniques embedded in various search engines have enhanced the way IR systems' design and increase their performances, but with new information continuously available to the end user, relying on just the end user's query to be satisfactory for IR results places the burden on the users. IR systems can be enhanced by introducing a learning agent to interact with the end user to establish satisfaction of IR retrieved results by introducing a learning algorithm that can interact with the user and build a knowledge domain base of what the user is looking for. The multi-agent layer in the proposed IR system is composed of a Reinforcement Learning (REL) agent- an intelligent agent that is capable of interacting with the end user autonomously and capturing the behavior of the user.

An intelligent learning agent is an agent that can adapt to the needs of different users, learn new concepts and techniques and anticipate the needs of the user. It can also take initiative and make suggestions to the user [74]. By utilizing AI inference engines and learning process components such as RL, intelligent agents gather and formalize knowledge from communications between users and systems to evaluate the IR results through interaction. To benefit from each learning experience, users' feedback (high score rewards) on information presented is utilized for future use by building a knowledge base of such learning experiences. Such learning can be considered as hyper-learning, in which the learning experiences of an agent can be used and shared by other specialized agents or users. The RL design within the system is a representation of multi-agent RL that links users and knowledge base resources.

Specialized Learning Agents (SLA) Information can be categorized into unique categories to enhance accessibility and organization of data. Data categorization has been applied to various search engines to organize data based on uniqueness and relevance; ie. Clusty.com search engine [141]. Specialization is done based on existing information and end user queries without evaluation by the end user. Specialization is performed in the proposed IR system through placing the user in the loop. The knowledge base of each domain can be established through end user's feedback of the IR retrieved information. A specialized learning agent(SLA) presents end users with relevant information retrieved based on their query and receives feedback on the information the end user is looking for. Relevant information that receives a high feedback score by the user is considered as a valuable knowledge base and is added into the repository of the SLA.

Each SLA is linked to the knowledge base repository of a specific domain. Information is ranked and dynamically updated based on the usage by the specific specialized domain agent and other domain agents within the multi-layer.

The SLA is a unique RL agent that learns only one specific category in which it builds the knowledge base through interaction with end users. The SLAs are distinguishable from each other, since each is specialized to perform learning on a specific domain. The SLA within this layer can access the other SLAs' entire knowledge base.

Reinforcement Learning Agents (RL) An SLA conducts the learning process by triggering the RL algorithm for each query. In this research, Reinforcement Learning Agents(RL) are used as part of a machine learning intelligent learning algorithm [121, 132, 86, 9]. RL agent algorithms have the advantage of being autonomous agents which are categorized under unsupervised learning [142, 6, 45, 122]. Furthermore, the IR system's objective is to present users with information relevant to what they are looking for, indicating that those end users are an essential part of the IR system. An RL agent algorithm component is based on an environment that obtains feedback as a reward of its action to move from one state to another, giving it Markov characteristics [49, 28]. The proposed IR system is structured around the end user to evaluate and present the system's action and rewards. The end user is structured to be the RLA environment, and RLA actions are evaluated by feedback (rewards). Also, there is no need to rely on precious training data, as in the supervised learning algorithm, to predict and reason with these intelligent agents.

Instead, user queries are pre-processed at the interface layer where the delegation agent can assign a specific agent to address the query. The specialized agent will trigger a learning process using the RLA algorithm. The RLA captures end user feedback as rewards and determine the goals (information that is selected and scored by end user) to be utilized and added into the knowledge base of the SLA for future users of the same SLA or shared by other RLAs within the multi-agent layer.

Collaborative Multi-agents

In collaborative agent systems, each agent contributes its own embedded intelligent technique to solve a complex problem. Collaborative agents emphasize autonomy and collaboration with other agents in order to perform tasks for their owners.

The proposed collaboration approach is derived from exploratory communication approaches, initiated by agents to locate relevant information for their peers. For agents to communicate and collaborate, they must speak a common language as well as follow a common protocol [36]. Collaboration among agents is achieved through using the bulletin board [74] in which each agent broadcasts its information. Instead of each agent re-learning what other agents have already learned through experience, the agents can simply search other agent domain knowledge bases for relevant knowledge. As a result, each agent has access to a vast body of learned knowledge that is based on the experience of the agents.

The collaboration levels are categorized into the following two groups:

- Agent-user collaboration;
- Agent-agent collaboration.

Collaboration and communication is facilitated by the Bulletin Board techniques [74]. The proposed collaboration protocol [74, 66] is presented as follows:

Collaboration

Collaboration among agents within the proposed approach occurs at two levels, specialized agents, and domain learning agents. Specialized agents collaborate through accessing and sharing learned knowledge of all domains, and the learning agents within each specialized domain collaborate among each other by sharing learned past experiences (learning policy). Each specialized learning agent has a knowledge base repository file which is used to store knowledge of a specific domain or category. Communication among agents occurs by directing an agent to another agent knowledge base to find and share relevant information. Utilizing the bulletin-board method, each specialized agent broadcasts its unique identity and specialty through the agent topic and keywords. Moreover, when an agent finds similar information broadcast on the bulletin board that describes the agent domain through unique keywords, the agent can communicate with others in the peer agent repository knowledge base to share learned knowledge. At time t_0, SA_1 begins to search for specific queries by searching the bulletin board registered agent's information for agents that may already have the learned knowledge about similar queries. If SA_1 query(ies) are found on the bulletin board, the SA_1 connects to the SA_{id} knowledge base repository that has the required learning knowledge. In addition to sharing the learned knowledge base among specialized agents, the proposed system has enhanced the collaboration at the learning agents level by sharing the learning policy of each agent when the queries are the same.

3.2.3 Knowledge Base Layer

IR systems are the means that map existing information to end users at the time a user looks for such information. This information exists in broader spaces such as the Internet, as well as in a constrained space such as medical, academic, and government institutes.

In a specialized MLA for an IR system, the knowledge base consists of two parts: specialized (learned) knowledge base domains and external knowledge space. The former includes a knowledge base that is gained through learning and interaction with end users; the latter consists of information available through the Internet (WWW) or through a specific entity such as an academic institute.

IR systems' standard objective is to retrieve information that a user seeks at a specific time and place. Various IR techniques and applications have contributed tremendously to that field as it becomes necessary for most people worldwide to use IR such as the Internet to access or present information. But given the increase in the demand and knowledge of end users of existing technologies such as the Internet, the need for an efficient IR system that will map users into the right information becomes urgent.

A Specialized MLA for IR systems presents a novel approach to this urgent problem: it will construct a knowledge and information repository built from user need and feedback(rewards), with a specificity and efficiency that does not yet exist.

Specialized Learning Knowledge Base Domains

By using specialized MLAs, various domains gain knowledge based on both user queries and user feedback. User queries are pre-processed to address unique knowledge domains, then mapped to the relative domain. Based on user feedback, the repository knowledge of each domain is ranked and weighted, after which retrieved information is presented to the end user. As a result of this process, each knowledge base is dynamically updated: information that has not been used for a measured period of time is purged, further improving the system efficiency.

3.2.4 Dynamic Domain knowledge Update

The Domain knowledge repository for each initiated specialized agent is dynamically updated and ranked based on the following criteria:

- Information in each repository receives a score every time it is used;
- Information is updated dynamically by checking the availability of the related information source;
- Information is ranked based on the usability score;
- Information can be purged if it has no scores or weights and if the threshold of repository storage has exceeded its the limit.

Specialization is established when an SLA utilizes its learning experiences through end user feedback in a specific category, and can be identified based on unique keywords that represent the information of the gained knowledge. Since, information is dynamically updated through matches to user queries that also match the specialized agent domain knowledge, learning knowledge is added into the existing agent's repository.

3.3 Agent: Subject Matter Specialization

3.3.1 Introduction

Both the number of information seekers and the ever-growing amount of information available to them demand an information retrieval system that is able to model the end user's search behavior and interest and to organize information into a specialized domain. Currently, users must search through vast amounts of information to satisfy their interests and needs: existing search engines and information retrieval applications can be considered merely a reactive approach to user's searches. While AI and machine learning have developed impressive techniques that allow users to create rapid, robust behaviours by replacing extensive planning and modeling with carefully engineered behaviours and a continuous sensing of the environment [63], what is needed is adopting AI intelligent strategies to develop an intelligent IR system. Intelligent systems that acknowledge and understand the user's state, learning, and behaviour in addition to finding and retrieving the relevant information efficiently and effectively.

Current search engine techniques and IR methodologies in IR systems can be scaled from impressive retrieval devices that respond reactively to users' queries into a user centric and an intelligent specialized domains' search engines.

This research addresses these challenges through a novel exploration of ways to expand the current reactive approach of intelligent agents in IR systems to focus on the use of specialized intelligent learning agents. Each intelligent agent is designed to learn about a specific domain (i.e., category) in direct and continuous interaction with the end user. Furthermore, the proposed approach organizes a specialized learning agent in a multi-agent learning paradigm that benefits from AI unsupervised learning algorithms and techniques, e.g., Reinforcement Learning (RL) algorithm.

Selecting an RL algorithm is motivated by the continually growing attention and potential uses of RL [6] in varied multi-agent systems and applications ranging from e-commerce, load balancing in networks, to space exploration by mobile robot [9, 145, 104]. The application of RL to multi-agent systems offers unique opportunities and challenges. RL is actively being studied as an effective means of learning in multi-agent environments. It allows an intelligent agent to learn how to reason and act by observing its environment. RL agents may coordinate their policies for mutual gain or share their experiences for optimal polices which offer advances for enhancing agent performance. Reinforcement Learning allows agents to evolve their knowledge of end user behaviour and interests as they function to serve the end user. Furthermore, RL allows each agent to adapt to changes in an end user's behaviour and interests. To build a specialized domain that pertained to end users needs, the proposed system design evolves around the end users though feedback. Such a system is scalable with the ongoing growth of information and users' needs. Many RL tasks require an extensive learning experience in order to achieve sound performance. Multi-agent systems are able to increase the speed of learning by collaborating/sharing information during the learning process. Advantages to the RL approach include its application in an on-line system, its learning environment can include real humans and its ability to receive continuous updates of its actions. In contrast to the supervised machine learning approach that offers no explicit feedback from the users. Specifically, SARSA RL algorithms is employed in this research.

A specialized multi-agent learning system using intelligent RL algorithms aims to improve the precision of information retrieval by mapping users' queries to the relevant information domains. This chapter analyzes what makes an intelligent agent a specialized agent, and addresses its structured components.

3.3.2 Specialized Agents

A specialized agent is defined as an agent assigned to learn about a specific topic (domain) and not only provide expertise but also facilitate easier access to the learned knowledge-base resources within that domain. As well, these specialized agents contain an ontology that represents the domain of interest of the agent, providing useful information for its domain of expertise. These specialized agents constitute the multi-agent system and are used to determine how to process information requests and share them with other agents.

The specialized agent's task is to map the user's query to a repository of specific domain in order to present users with relevant information. Mapping users' queries to only relevant information is one of the fundamental challenges in AI and machine learning research. To address such a challenge, this research examines the fundamental components that constitute the specialized agent: an intelligent machine learning system, user input queries, an intelligent agent, and information resources constructed through specialized domains. The following section will address what makes an intelligent agent a specialized agent.

Four essential elements constitute a specialized agent framework. First, input, which is a representation of the user's (U) query (Q). Second, the learning agent A, the tasks of which are to find the domain relevant to a user's query, learning user's behaviour through user feedback R(rewards) as well as finding the domain relevant to the user's query. Third, user feedback of information presented is measured as rewards *(R).* Fourth, output represents domain models *(D)* of varies concepts; the contents of each model consist of ranked (weighted) knowledge-base resources of specific domain. Figure 3.21 depicts the specialized agent framework.

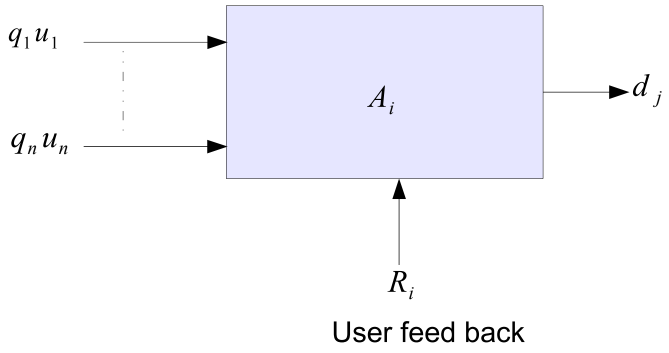


Figure 3.2: Specialized Agent Framework.

- Let $U : \{u_1, u_2, ..., u_i\}$; be the users' in a search,
- Let $Q : \{q_1, q_2, ..., q_n\}$; be the users' queries,
- Let $A : \{a_1, a_2, ..., a_m\}$; be the set of intelligent learning Agents,
- Let $R : \{R_1, R_2, ..., R_z\}$; be the user feedback noted as rewards R_z of the information presented by the agent.
- Let $D : \{d_1, d_2, ..., d_j\}$, be the set of specialized knowledge base domains.

3.3.3 Specialized Agents Learning Process

Given a query q_n, the learning system aims to map conceptual information related to the query q_n using the machine learning mechanisms by virtue of its reinforcement learning agents. Figure 3.3 depicts the mapping process of the specialized agent components.

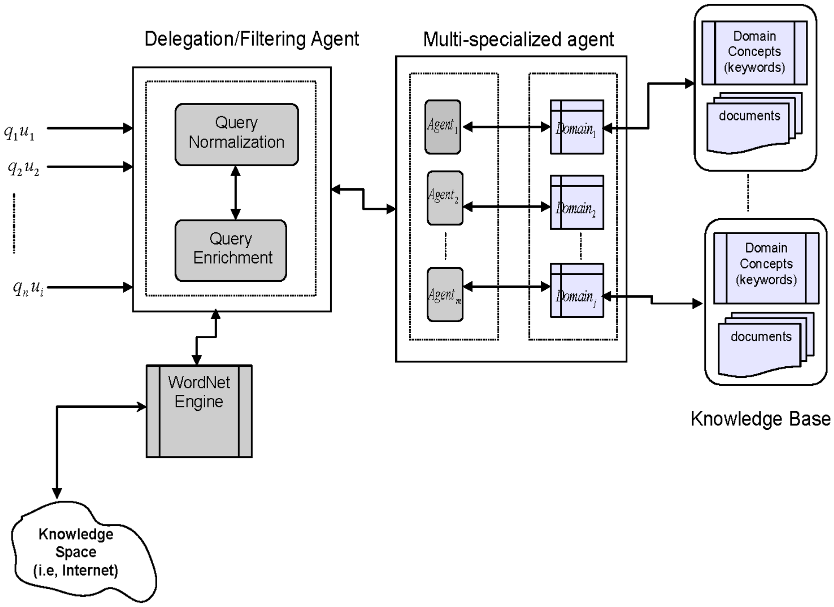


Figure 3.3: Specialized Agent Learning Process.

The following steps outline the general process by which an intelligent agent becomes a specialized learning agent:

1. Each user's query q_n is pre-processed to a match that exists within the relevant agent domain. Users' requests are compared with the existing specialized domain keywords (terms $t_1, t_2, ..., t_n$) to find a similarity and map the query to the specific agent.

2. If the user's query concept is not presently exist within the existing domains, the user's query is enriched with semantic lexical synonym terms to enhance the similarity search of relevant domain.

3. Learning agents will search information relevant to the user's query within a multi-agent domain based on the domain that is relevant to the query. If the domain of the user's query does not exist within the multi-agent specialized domains, the delegation agent will be acknowledged to initiate a new domain constriction. The knowledge base of the new domain will be constructed based on the end user's feedback on the information presented. The learning process occurs through Reinforcement Learning.

4. Information that receives a high score by the user will be added into the domain knowledge model. The domain model information resources will be ranked dynamically to place the most useful (weighted) information on the top so it can be presented to the similar query in the future instead of retrieving the same information from the knowledge space.

3.3.4 The Specialized Agent Algorithm

An agent is said to be specialized in a specific domain if it maps the user's query into a matching domain and retrieves information relevant to the user's query. The learning process of each specialized agent is conducted for each user query; however, more than one user can share the retrieved relevant information of the learning agent if their queries are the similar. As part of multi-agent collaboration, information relevant to a query q_n can be shared by more than one user u_i. Queries$\xrightarrow{F}$ Users $\equiv$ $q_n \xrightarrow{F} u_i$.
Queries are mapped to agents via the delegation agent. Each agent broadcasts its profile to

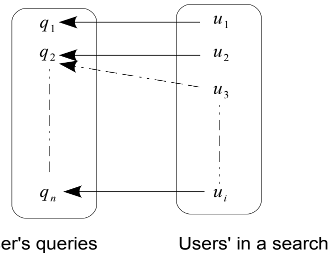


Figure 3.4: Mapping user U_i to queries Q_n.

other agents and the delegation agent through the bulletin board. User's query comprises a set of terms: $Q = \{t_1, t_2, ., ., ., t_n\}$, that can be enhanced and enriched with lexical synonyms extracted from WordNet.

Agents$\xrightarrow{F}$ Queries $\equiv$ $a_m \xrightarrow{F} q_i$. Every specialized agent for a specific domain is composed of a set of related concept keywords (ontologies) and a set of documents that relevant in their domain, $d = \{(c_1, c_2, ..., c_y), (doc_1, doc_2, ..., docx)\}$, where c_y denotes for the concept term and doc_x denote for obtained documents (information) within the domain for the specialized agent. Each document is ranked based on usability, importance and similarity function (bag of words). Usability is measured by the total number (frequency) of times the document is used by the user and other agents while the importance of the documents measured by the total rewards (feedback) received by end user. Documents are also measured based on the total number of terms similar to the user's query -measured by TF-IDF algorithm . When the agent retrieves relevant information from the specialized domain, the specialized agent will collect a set of high ranked relevant documents; $S = r_i^1, r_i^2, ..., r_i^N$. Each specialized domain obtains knowledge by adding the relevant R_i^N document that a user chooses and dynamically adds it to a domain to construct the knowledge base about a specific field. Figure 3.5 depicts mapping between agents and domains.

Domain$\xrightarrow{F}$ Agents $\equiv$ $d_j \xrightarrow{F} A_m$. Agent a_m is said to be specialized in domain d_j

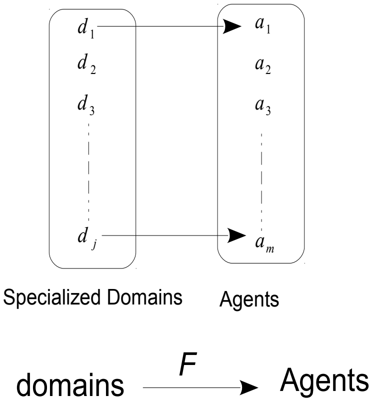


Figure 3.5: Mapping agents A_m to domains D_j.

if

$$q \in d_{\exists}\,,$$

$$\xi$$

$$r_i(q) = S = \{r_i^1,\, r_i^2, ..., r_i^N\}$$

$$where \quad r_i^K \in d_j \qquad \forall \;\; K = 1, ..., N.$$

$$DoR(r_i^k) \;\; \geqslant \;\; DoR(r_i^L), \quad \forall \;\; L > K \;,$$

where DoR denotes the degree of relevance of the extracted document(s)to the user's query q.

The TF-IDF algorithm is adopted to retrieve information relevant to the user's query. The specialized domain d_j carries a collection of ranked data: $SD_j = \{\ltimes_1, \ltimes_2, ...\ltimes_n\}$.

If a match is found for the user's query within the specialized agents' domains, the query is mapped to the specialized agent in which the learning process is initiated with the end users, $SF(q_n, SD_j) \leq \ltimes_{d_j} \Rrightarrow SF :: q_n \subseteq SD_j \Rrightarrow d_j \xrightarrow{F} A_m \Rrightarrow q_m \longrightarrow a_m \Rrightarrow u_i \longrightarrow d_j$. The agent will retrieve relevant r information from the agent domain based on it degree of relevancy to the user's query. Documents within the knowledge base are ranked and weighted based on usability by the user, other agents and also by the total weight of its similarity to the presented query.

3.3.5 Learning Process of a Specialized Agent Using RL

As described in the previous section, Reinforcement Learning (RL) agent learning is formalized in terms of reward signals passing from interaction with its environment. This interaction takes the form of the agent sensing the environment and based on this sensory input, choosing an action to perform in the environment. The chosen action changes the environment in some manner and this change is communicated to the agent through a scalar reinforcement signal. The use of a reward (feedback) signal to formalize the idea of a goal is one of the most distinctive features of RL. The specialized agent model consists of elements that match the RL model elements; which are:

- A dynamic learning environment represented in a matrix grid consisting of a set of relevant information to be checked by the agent.
- State S: current state of the agent at one of the grid cells represented by one of the specialized agents retrieved responses (information).
- Reward R (Feedback): represented by the user's feedback on the information presented.
- Action A: Agent's next move (state) within the matrix grid; agent can move up, down, left or right.
- Goal G: The document selected by the user matches the goal parameters: selecting the document (by clicking on the document), time spent viewing the document, and bookmarking the document.

RL is an ideal machine learning algorithm that suits the specialized learning process.

Learning Environment Model

In response to the user's query, a specialized agent retrieves the relevant information (responses) from the knowledge space and presents it to the user in the form of a learning environment matrix grid. Each cell within the matrix grid will contain a reference to the specialized agent that retrieved the relevant documents (responses). The retrieved responses are a set of relevant information to the user's query:
$r_i(q) = S_i = \left\{r_i^1, r_i^2, ..., r_i^N\right\}$
where $r_i q_i$ is the retrieved information per user's query for each specialized agent S_i, and r_1^N is one of the retrieved documents for query I of specialized agent S_i from the knowledge space (i.e., Internet).
The RL agent that is embedded with the specialized agent observes the learning environment's state S (matrix grid)and the agent can influence the change of the states by applying an action S to the environment. As a result, the RL agent receives an immediate reward R. The RL task is to optimize the interaction with the environment in which an agent performed action selection mechanism is based on the environment's feedback.

Suppose the specialized agent retrieves six responses per user's query: $r_i(q) = S_i = \left\{r_i^1, r_i^2, ..., r_i^N\right\}$ where $r_i q_i$ is the retrieved information per user's query for each specialized agent S_i. $S_1 = \{r_1^1, r_1^2, r_1^3, r_1^4, r_i^5, r_1^6\}$. The learning environment is represented in a matrix grid (world grid problem), where relevant documents retrieved by the specialized agent are populated within the grid cells. Each cell is a reference to the specialized agent relevant document. Each cell will be represented by a letter for explaining learning analysing $S_1 = \{A, B, C, D, E, F\}$ where each letter is a representation of relevant information: i.e., $r_1^1 \equiv A$, $r_1^2 \equiv B$, $r_1^3 \equiv C$, $r_1^4 \equiv D$, $r_1^5 \equiv E$, and $r_1^6 \equiv F$. Assuming the ultimate goal (documents) that the user selects and bookmarked are known, r_1^4 (D) is set to be the target goal. Initially, the agent can be in any state (documents cell) and can move from one state to another in four directions to find the goal (actions): up, down, left or right; In this simulation, the goal resides in (r_1^4) cell, which has an instant reward of 100. Other states that do not have the direct connection to the target room have zero rewards. The matrix grid can be depicted by the letters A, B, C, D, E, and F as in the following:

$$\begin{array}{|c|c|} \hline r_1^1 & r_1^2 \\ \hline r_1^3 & r_1^4 \\ \hline r_1^5 & r_1^6 \\ \hline \end{array} \equiv \begin{array}{|c|c|} \hline \mathrm{A} & \mathrm{B} \\ \hline \mathrm{C} & \mathbf{D} \\ \hline \mathrm{E} & \mathrm{F} \\ \hline \end{array}$$

The task of the learning agent is to follow the states that will lead to the ultimate goal. A state diagram graph representation of the retrieved responses documents is depicted in Figure 3, where each document is represented by a vertex (or node), and they are all linked with an edge. For example, the learning agent at state (cell) A would reach the goal from the following possible states:
$A \Longrightarrow B \Longrightarrow D$,
$A \Longrightarrow C \Longrightarrow D$ or
$A \Longrightarrow C \Longrightarrow E \Longrightarrow F \Longrightarrow D$.

The above grid can be represented by the graph as in the following:

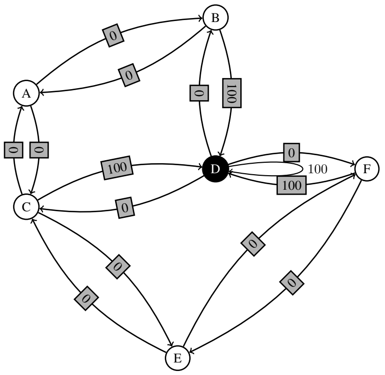


Figure 3.6: State diagram graph of the learning grid.

The learning model environment system of the state diagram and the instant reward R values can be structured into the following reward table, or matrix R. The minus sign in the table indicates that the row state has no action to go to the column state.

R=

State/Action	A	B	C	D	E	F
A	-	0	0	-	-	-
B	0	-	-	100	-	-
C	0	-	-	100	0	-
D	-	0	0	100	0	-
E	-	-	0	-	-	0
F	-	-	-	100	0	-

Simulation of The Learning Agent with SARSA RL Algorithm

This section presents in Table 4.3, an example of a simulation of the state graph of $S_1 = \{A, B, C, D, E, F\}$ and calculates the rewards with the SARSA learning agent at state D.

```
Initialize $Q(s, a)$
Repeat for each episode
  Initialize $s$
  Choose $a$ from $s$ using policy(e.g., $\epsilon - greedy$) derived from $Q$
    Repeat for each episode until $s$ terminal
      Take action $a$ observe reward $r$, state $s'$
      Choose $a'$ from $s'$ using policy(e.g., $\epsilon - greedy$) derived from $Q$
      $Q(s, a) = Q(s, a) + \alpha \left[r + \gamma Q(s', a') - Q(s, a)\right]$
      $S = s'$, $A = a'$
```

Table 3.1: SARSA learning algorithm[123].

The SARSA agent considers state-action at each state. It does not move to the next state based on the maximum reward it receives, but rather it checks all possibilities at each state (state-action), and considers the dynamic specialized agent environment in which the goal can be at any state since it depends on the user's action. SARSA is considered to be an ideal RL algorithm for such an environment because the algorithm will return the sequence of the current state from the initial state until it reaches the goal state. The

parameter of reward has a range value of 0 to $1(0 \leq \alpha \leq 1\ )$. If the range is closer to zero, the agent will tend to consider only the immediate reward. If it is closer to one, the agent will consider the future reward with greater weight and is willing to delay the reward. If the agent state is at D and $\alpha = 0.8$, the agent has six possible actions through which to go to state B, F, or C.
The agent uses the following algorithm to learn from the experience of training of the agent.

$$Q(state, action) = R(state, action) + \alpha[\textit{(next state,all actions)}]$$

$$Q(B, D) = R(B, D) + 0.8[Q(D, B), Q(D, C), Q(D, D), Q(D, F)] = 100 + 0.8.100 = 180$$

Each episode is equivalent to one training session. In each training session, the agent explores the environment (represented by Matrix R), and gets the reward (or none) until it reaches the goal state. The purpose of the training is to enhance the agent that is represented by the Q matrix. More training will provide a better Q matrix which can be used by the agent to move in the optimal direction. In this case, if the Q matrix has been enhanced, instead of exploring and going back and forth to the same node, the agent will find the fastest route to the goal state. Off-policy approach is what makes RL Sarsa different from Q-learning because in Q-learning, the agent action is based on the maximum reward value out of the available rewards. In contrast, an RL SARSA agent will pick the state-action of the next state. The learning process will continue until the agent has reached the goal. In the Specialized agent learning process, the goal is dictated by end user feedback. The parameters are set to time spent on each state, a selection of the state, and bookmarking of each state. The goal state which represents the information retrieved by the specialized agent is added into the knowledge base with its scoring weight. The RL agent goal is considered to be the relevant document (information) that the user selects and looks for based on the user's feedback. The selected document will be augmented into the agent domain repository with its reward scoring and the specialized domain will keep adding newer and more relevant information into its domain based on user feedback.

3.4 Summary

This chapter has described how a framework of specialized MLAs for IR will work to improve the efficiency of IR and the satisfaction of its users. The architectural aspects of the proposed system framework consist of three hierarchical layers: the interface layer, the multi-agent layer, and the knowledge space layer. Within the interface layer, an interaction medium is structured from user interface, query pre-processing and refinement, and delegation agent functions. The multi-agent layer consists of several agents, each of which is built from an RL algorithm that is specialized to perform learning in a specific domain; specialization, collaboration, and learning among agents comprise the main multi-agent layer function. The knowledge space layer consists of data resources used for access and retrieval of the domain knowledge base, and for extracting higher level information. A descriptive analysis of how an intelligent agent becomes a specialized agent in a multi-agent system has also introduced in this chapter. The proposed system offers a complete learning framework that is able to model the end user's search behavior and interests and to organize information into categorized domains so as to ensure maximum relevance of its responses as they pertain to the end user queries. Structure components of the specialized agent system are addressed in detail. In addition, agent learning process steps to outline the general process that allows intelligent agents to create new information based on the information seekers' feedback and their behaviours. This research maintains that, in this age of ever-accumulating supply and demand for information, such a system will increase both the efficiency of IR and its users' satisfaction.

Chapter 4

Knowledge Domains: Topic Extraction for Specialized Domains.

4.1 Introduction

In order for the delegation agent to map users' queries into the relevant domains, it has to search each existing knowledge domain to find the most relevant information before it decides that the knowledge domain does not exist. However, owing to the size and dynamic nature of the information resources, the delegation agent must sift through a large amount of retrieved information in order to find the desired information. Furthermore, specialized agents would require a similar process to search for relevant information other specialized agents might have learned or obtained.

To alleviate this difficulty, this thesis presents a novel approach for finding, which existing domain knowledge would be ideal for the given query, by constructing specialized domain topics for each existing domain. The domain topics of each domain would be the initial step for the delegation agent to conduct its search to determine whether the query has higher similarity to conduct further search for relevant information within that domain. Moreover, the domain topics of each domain will act as the specialized agents' identification. Specialized agents collaborate amongst each other through information sharing whereby each agent would have a representation of its specialty through domain knowledge metadata (keywords) of the domain topics.

4.2 An overview of Domain Ontologies Construction

This chapter presents a novel mechanism which uses an intelligent learning model to automatically construct specialized domain topics for knowledge domains. The domain topics can be modeled using two methods, utilizing users queries and an existing domain knowledge base. The first method is applied when a new knowledge domain is established. The second method is applied when the domain knowledge exists. In both cases, the domain topic of each domain is continuously updated as the knowledge base of each domain renews. This chapter will discuss how a new knowledge domain is established. The proposed approach combines three types of resources to automatically construct specialized domain ontologies (concepts): semantic lexical knowledge objects (dictionary based) and semantic statistical knowledge objects (from the Internet) that are evaluated by the end user through an intelligent learning system.

Constructing specialized domain ontologies (concepts) intelligently and automatically, involves enriching the user's query with related linguistic ontologies and statistical semantic-related concept terms. Natural Language Process (NLP) techniques, such as WordNet was employed to enrich the user's query with semantic, lexical, synonymous terms, and probabilistic topic models, such as Latent Dirichlet Allocation (LDA), to extract highly ranked topics from a query's retrieved information.

The proposed specialized multi-agent learning system aims to improve precision of information retrieval by mapping users' queries to the relevant information domains. The analysis of what makes an intelligent agent a specialized agent and its structured components was addressed in Chapter 4. Automatic domain modeling and knowledge creating without any prior knowledge such as a pre-defined domain name, category, or supervised training set of data is gaining momentum in Natural Language Process (NLP) and IR research areas.

Human created domain ontologies present strong semantic features, but require both time and consistency with which to grow large scale ontologies. Classification and clustering are among the traditional methods used to construct automatic domain ontologies. A hierarchical agglomerative clustering (HAC) algorithm is among these methods, but while such an approach from tree hierarchy structure is both consistent and scalable, it is usually a term-based technique and is not semantic-aware. In contrast, a specialized domain topic collects terms that are related semantically (conceptually) to a relevant domain. To automatically build a domain ontology that is relevant, semantically aware and scalable, this research proposes a novel approach that constructs knowledge in specialized domain ontologies through query enrichment, topic extraction, and user's feedback. The proposed approach employs WordNet to enrich the user's query with lexical synonymous terms, Internet to extracted topics form the information retrieved, and user's feedback to evaluate and label discovers domain concepts through learning agent(i.e., RL). Furthermore, this addresses the idea of enriching the suggested domain topics by involving the end user in tagging the suggested topic in addition to just selecting what the intelligent agent through RL proposes. This technique is known as Social Tagging. The user's suggested tags (keywords) are added into the mix (duplicates are removed). A tag is a non-hierarchical keyword or term assigned to a piece of information (such as an Internet bookmark, digital image, or computer file). This type of metadata helps describe an item and allows it to be found again by browsing or searching. Tags are generally chosen informally and personally by the item's creator or by its viewer, depending on the system.

Tagging was popularized by websites associated with Web 2.0 and is an important feature of many Web 2.0 services.

4.3 Architecture Design

The task of building a new domain topic consists of two parallel processes: topic extraction from the search engine and query refinement through the WordNet Engine. As Fig. 4.1 illustrates, the process of achieving query refinement takes seven steps:

- Query-related text documents are retrieved from Internet.

- At the same time, the query is enriched by infusing its terms with synset terms (synonyms), extracted from the WordNet database.
- Retrieved documents are filtered from the Web syntax format and converted into text format.
- Once the Web documents are filtered, text documents are normalized by stemming and removing stop-words.
- Using a Topic extraction algorithm, semantic topics are extracted from the text documents and clustered into relevant groups, through agglomerative clustering.
- Once the query is infused by WordNet synonyms, the similarity Function Process assesses similarities between the enriched query synonyms sets and each clustered topic group, and adds each query synonym into the topic group if no similarity has been found.
- The extracted and enriched topic sets are evaluated through end user feedback, using Reinforcement Learning.

These steps are discussed in detail in the following sections.

Used Annotations Descriptions

Word A a basic unit defined to be an item from a vocabulary of size W.

Document A sequence of N words denoted by $d = \{w_1, ..., w_n\}$ where w_n is the nth word in the sequence.

Corpus A collection of M documents denoted by $D = \{d_1, ..., d_m\}$.

Lexical The linguistic meaning and morpho-syntactic features of words and possibly one or more complex units such as idioms.

Ontology A set of classes and a set of relations that describe the properties of each class. Ontology formally defines relevant knowledge in a domain that describes and can be used to interpret data in this domain.

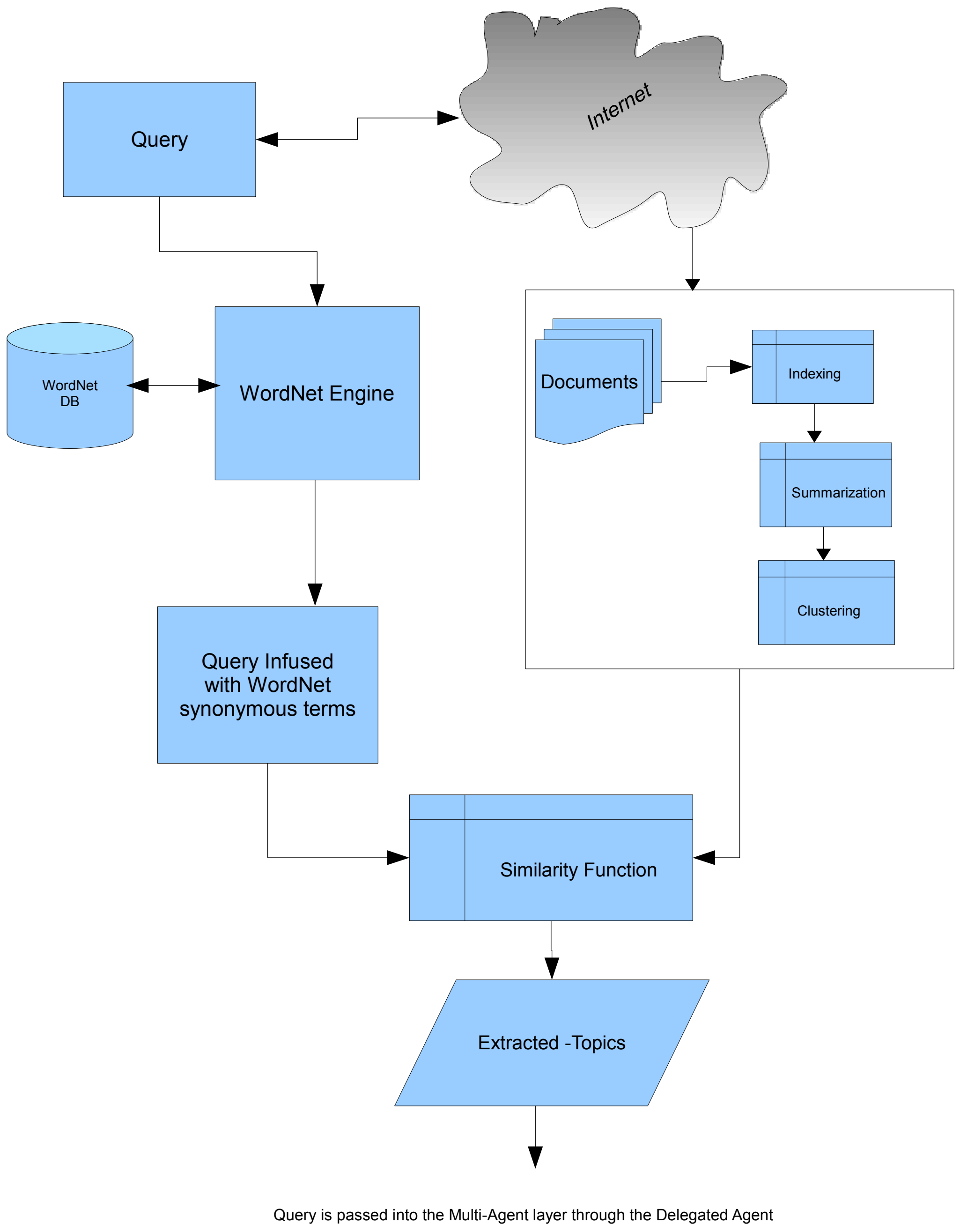

Query
Internet
WordNet DB
WordNet Engine
Documents
Indexing
Summarization
Clustering
Query Infused with WordNet synonymous terms
Similarity Function
Extracted -Topics
Query is passed into the Multi-Agent layer through the Delegated Agent

Figure 4.1: Query Refinement: Query-Topic extractions process.

Domain A set of conceptualized relevant ontologies.

4.4 Query Enrichment Via WordNet Ontology

Recently, ontologies have been used in the framework of the Semantic Web. Ontologies may be employed to associate meaning with data and documents found on the Internet, consequently increasing diversified applications of information retrieval systems. The adoption of ontologies in information retrieval systems is limited, due to their insufficient broad coverage and their need to be constantly updated, as evidenced by Guarino et al. [47]. Linguistic ontologies encompass both ontological and lexical information, thereby offering a way to partly overcome these limitations. The use of WordNet ontology in this thesis refers to linguistic ontologies. Linguistic ontologies are large scale lexical resources with an ontology structure, e.g., WordNet.

WordNet

WordNet is a freely available online lexical database engine for English utilized by various research fields such as the Natural Language Process (NLP), Information Retrieval and Artificial intelligent communities. English words are organized into synonym sets (sysnsets), and each one is represented by a lexical concept. A synset can have many words (synonyms) and one word can be a member of many synsets, one for each different sense [35]. WordNet represents both words and synsets, in which the relationships between words are lexical and those between synsets are semantic[35]. The goal of the WordNet project is to support automated text analysis, sens-disambiguation, term expansion in IR systems, and the structuring of representations of document contents. WordNet is also useful for determining semantic connections between sets of synonyms, and for tracing morphological connections between words.

WordNet provides a variety of semantic relations which are defined between concepts. The syntactic category of each word determines its semantic relationships. The following definitions clarify these relations:

Synonymy : Two concepts have a similar meaning.

Antonymy : Two concepts have an opposite meaning.

Hyponymy/Hypernymy : X is a kind of Y, where X is a more specific concept (hyponym) and Y is a more generic concept (hypernym).

Meronymy/Holonymy : X is a part of Y, where X is a concept that represents a part (meronym) of whole concept Y (holonym).

The use of ontologies for many natural language process applications (NLP) resulted in differentiating between two types of existing ontologies: formal and linguistic ontologies. Linguistic ontologies mainly differ from the formal ontologies is by size and degree of formalization. Linguistic ontologies are very large such as WordNet comprises several dozen thousand synsets, while formal ontologies are generally much smaller. By leveraging the WordNet lexical database [75], the end user's query is enriched with WordNet lexical terms. The WordNet ontology [20, 85] is a large, lexical English database whose structure makes it a useful tool for computational linguistics, data mining, information retrieval, and NLP [128]. The objective behind extracting lexical synonyms from WordNet is to enrich the overall domain ontologies with synset terms, in addition to concepts extracted through the Internet [85, 33, 111]. Using WordNet, synsets of each term are agglomerated into a set of ontology terms. Querying for the lexical enrichment of ontologies is described in the following four steps:

- The query is normalized, where all stop-words are removed and stemming applied, $q = \{t_1, ..., t_n\}$,where q is the initial query and t is the query term.
- Term synsets are extracted for all the query terms from the WordNet database.
- hypernyms of each term are obtained through WordNet.
- Synset hypernyms are added according to the corresponding term representation.

Synonyms of a term may be enriched further by checking the WordNet Hyponymy and/or Hypernymy hierarchical structure, in which the term would gain a more specific concept with hyponym, and a more generic concept with hypernym [75]. The new terms (words) are assigned for later semantic matching processes with the extracted topics. A given search query, q, is normalized (filtered and stemmed) before it passes to WordNet to extract semantic lexical hypernyms sense for the query terms t; $q = \{t_1, t_2, ..., t_n\}$, where q is the query consisting of normalized terms t.

$q(t1) = \{t_{11}, ..., t_{1n}\}$,

$q(t2) = \{t_{21}, ..., t_{2n}\}$, - - - - $q(tn) = \{t_{n1}, ..., t_{nn}\}$, where $q(t_n)$ is the query term and t_{nn} are the lexical synonyms terms extracted form WordNet sense hypernyms.

Example: if the user's search query is $Q = \{diabetes\ diets\}$, the query will be enriched with WordNet ontologies process by being first normalized, as shown in Table 4.1: $Q(t_1) = \{diabetes\}$, and $Q(t_2) = \{diets\}$, and then with an extract passed to the WordNet engine to extract the ontology concepts.

Table 4.1: An example of enriching a user's query with WordNet ontologies.

Query (q):”diabetes + diet”	
t_1:*diabetes* ⇓	t_2:*diet* ⇓
diabetes diabetes mellitus type I: diabetes insulin-dependent type II diabetes: non-insulin-dependent	diet,allergy diet balanced diet bland diet ulcer diet diabetic diet carbohydrate loading carbo loading .....

In spite of the vast lexical database of WordNet and other dictionary-based databases, important information may be overlooked within WordNet or similar dictionary-based sources (linguistic and semantic knowledge resources). Therefore, domain knowledge ontologies should be comprehensibly relevant to various types of concepts. Web data, in which the users search for information, can be utilized to extract such concepts.

4.5 Topic Extraction Using Web Data

At the same time in which the query is enriched with related concepts using WordNet, the same query is used to extract relevant topics through a retrieved set of documents from the Internet (or other information repository). The newly discovered specialized domain topics would comprise not only the computational linguistic knowledge objects (WordNet ontologies) but also related information extracted from a global and dynamic, evolved source of information, such as the Internet.

The limitation of WordNet is that it does not carry each word or concept within its database; as a result, it should ideally be enhanced to extend the extracted WordNet senses with concepts of dynamic and unstructured data such as the information on the Internet.

This thesis presents an unsupervised approach that would automatically construct a specialized domain topics for specialized knowledge domains, by leveraging the search engine applications to retrieve a set of documents related to the query. Retrieved documents will be pre-processed, using common NLP pre-processing techniques, including, text normalization and removing stop words. The Retrieved Web documents were converted into a text format; by removing the HTML tags and embedding web application codes. Through the Latent Dirichlet Allocations (LDA) [11], the retrieved and normalized documents are induced to discover topics from each document.

4.5.1 Topic Model

Extracting a topic that represents the document, or a set of documents, is one of the current challenges in the NLP, Data Mining and IR research areas [111, 143]. As a result, the latent topic extraction technique has emerged as a popular topic algorithm for identifying topics from text documents based on semantic concepts rather than on the bag of words. LDA [11] is a probabilistic topic model, originally used in natural language processing, but it has been applied to extract topics in various applications [134, 83, 125, 78, 77] and is also the ideal semantic analysis algorithm for the purpose of the domain construction this thesis proposes. Semantic analysis algorithms, like LDA, focus on topic detections in text data include Latent Semantic Analysis (LSA) and Probabilistic Latent Semantic Analysis (PLSA) [11, 125]. These algorithms have recently been an area of considerable interest in Machine Learning [11, 125, 46, 77].

Each document in the LDA model is assumed to be a random mixture over latent topics and each topic is specified by the distribution over words. LDA extends the structure of the PLSA model. LDA assumes that a document is represented as random variables denoting topic distribution, and a word occurs on the term probabilities of the topic corresponding to the word. The estimate parameters do not depend on the number of documents. Hence, LDA does not posses the problem of over-fitting where the PLSA model does. The probabilistic distribution of each document follows Dirichlet distribution.

Some of the LDA features superior to those of PLSA and cluster models can be listed as follows:

1. Compared to the PLSI model, LDA resolves the over-fitting problem, as well as the problem of generating new documents, by treating topic mixture distribution as a set of random hidden parameters instead of a large set of individual parameters [11],
2. LDA performs smoother topic range calculation then LSA and pLSA,
3. Compared to the cluster model, the LDA model allows a document to exhibit multiple topics to different degrees, which makes LSA more flexible than the cluster model assumption that each document is generated from only one topic [11, 125].

For these reasons, the LDA algorithm, a statistical model, has emerged as a popular topic algorithm that has been applied to extract text document classification and identify topics from text documents [11, 125, 83, 52]. Each document in the data is associated with a multinomial distribution other K latent topics [11]. For each word w_n in document d, LDA assumes that a topic z_n is sampled from the topic distribution for d, and that w_i is sampled from the unigram word distribution for that topic. In order to limit overfitting and handle unobserved words, the LDA model imposes a Dirichlet prior over the parameters of the topic and unigram distributions. The training process involves estimating both distributions given the observed documents, the fixed parameters of the Dirichlet priors, and the number of topics K. As an analytical solution is untractable, approximate inference methods such as Gibbs sampling or Variational Bayes are typically used [11]. A thorough and complete description of the LDA model can be found in [11].

4.5.2 Applying LDA to Extract Topics

LDA has been successfully applied in various applications, extracting semantic topics from text documents. The proposed system has extended its functionality by employing topic model algorithms such as LDA for building domain concepts (topics). The following steps describe the proposed technique of extracting domain concepts utilizing LDA:

1. Text document generation, Retrieved documents from the Internet are normalized by converting them into a text format. Each document is pre-processed, filtered, and represented as a text document, in a corpus, to be used as the input of LDA.

2. Latent topic extraction with LDA. A set of semantic latent topics is produced by extracting topics from each text document. Each document is associated with a topic vector which specifies the topic distribution of the document.

3. Clustering relevant topic groups. Once the topics are extracted for each text document, they are clustered into groups by comparing them to each other using the similarity function. This step is added to rank the extracted similar topics into a higher level of topic in a hierarchical manner. Cosine similarities with hierarchical agglomerative clustering (HAC) [143, 52] between each pair of topics is adopted to generate high level topics called "'super topics"'. If the similarity between two topics is greater than the threshold, the system clusters them into the same group. Note that a topic may belong to several different categories. The idea behind clustering around the topics is to rank groups with the most relative concepts [143]. The cosine similarity is calculated as follows: Let $t_u \{W_{u,1}, ..., W_{u,n}\}$ and $t_v = \{W_{v,1}, ..., W_{v,n}\}$ be two vectors of correlation valued for the topic t_u and t_v.

For example, Table 4.2 depicts some of the topic lists extracted from Web pages using the top ten retrieved pages of the Yahoo.com, BING.com, Google.com, Clusty.com, and

National Library of Medicine (NLM) search engines[1]. Query (q):"diabetes + diet" was used.

Table 4.2: An example of extracting Topic from the Internet using LDA

TOPIC extracted from Internet for the Query (q):"diabetes + diet"	
Topic 0th:	
type	0.023197
insulin	0.022811
carbohydrates	0.013933
blood	0.013933
sugar	0.013161
patients	0.012390
index	0.012390
Topic 1th:	
.. ..	
Topic 10th:	
.. ..	
.. ..	

4.6 Merging Statistic Model and Semantic Model to Refining the Discovered Topics

Mapping the semantic extracted ontology of each group to represent a specialized domain would require further enrichment with information based on linguistic terms relevant to the original query. The idea behind combining both terms based on raw data retrieved from the search engine and linguistic synonym is to fill the gap which exists in a domain based upon only one of these knowledge bases, which would exclude valuable information to be deemed irrelevant. For example, some technical and scientific terms are not yet listed within the WordNet database. Using the cosine similarity function to locate the similarity between the WordNet enhanced synonym terms and the search engine extracted topic concepts, terms which match the similarity threshold (i.e, 0.8) are retained or added into the topic groups, if no matches are found. The enhanced WordNet query hypernyms

[1] Right Reserved to the listed search Engines and http://vsearch.nlm.nih.gov.

are added into each semantic extracted topic (ontology class) groups if no similarity is found. The domain topic group would consist of the following information:

1. domain Identification (ID), where each domain would possess an ID that is unique,
2. a list of extracted semantic and linguistic ontology terms,
3. usability, i.e., the total number of times the domain selected is registered. It also indicates the number of times the domain has been dynamically update, since it would be updated every time it is utilized,
4. users' suggested tags (keywords) are added into the mix (duplicates are removed).

A constructed domain would consist of the following:

$$\mathrm{D}=([ID],[t_1,t_2,....,t_n],[Tag_1,Tag_2,......,Tag_i],[selected_j,weight_w]).\ (4.1)$$

For example, Table 4.3 shows the constructed domain of WordNet; user's search query is $Q = \{diabetes\ diets\}$ mixed terms (i.e, metadata of a specific domain).

Table 4.3: An example of combining WordNet ontologies to the Extracted Topic

Query (q):"diabetes + diet"
mellitus insipidus nephrogenic insipidus type I: insulin-dependent type II diabetes: non-insulin-dependent diabetes latent chemical diabetes allergy balanced bland ulcer diabetic loading carbo gluten-free high-vitamin vitamin-deficiency low-salt diet salt-free diet liquid diet reducing obesity vegetarianism *type may diabetes blood fat healthy cup fresh meals foods protein calories sugar people levels carbohydrate guidelines nutrition provider mg insulin patients*

Using social tagging techniques, the constructed domain concepts are presented to the end user in the form of terms with different colors. Size depends on the weight relevant to initial user's query. Thus, a user can select the suggested concept terms just by clicking them.

Furthermore, users can interact with the intelligent model by tagging additional terms above and beyond those proposed by the system.

Table 4.4: Term Extractions Using various Techniques and the Domain Construction Domain Model.

Query (q):"diabetes + diet"

Kea: Keyphrases	*Cluto: Vector Clustering*	*Bigram*	*GibbsLDA LDA*	*LDA+WNet*
Diabetes fat blood Diet sugar Glucose Food carbohydrate meal eat blood sugar calories levels healthy weight	sugar blood glucose insulin meal food eat people type complic exercises health control risk fat oil protein carbohydr loss weight	blood sugar blood glucose type diabetes glucose levels diabetes diet weight loss sugar levels people diabetes blood pressure people type ..	type may diabetes blood fat healthy cup fresh meals foods protein calories sugar people levels carbohydrate guidelines nutrition provider mg insulin patients	mellitus insipidus nephrogenic insipidus type I: insulin-dependent type II diabetes: non-insulin-dependent diabetes latent chemical diabetes allergy balanced bland ulcer diabetic loading carbo gluten-free high-vitamin vitamin-deficiency low-salt diet salt-free diet liquid diet reducing obesity vegetarianism *type may diabetes blood fat healthy cup fresh meals foods protein calories sugar people levels carbohydrate guidelines nutrition provider mg insulin patients.*

4.7 Evaluating Extracted Topics

To examine the effectiveness of the proposed specialized domain construction model, the quality of generated ontologies using the proposed domain topic extraction approach is compared with other well known terms, multi-term, and keyword generators such as WordNet engine, Kea to extract key-phrases [137], CLUTO clustering toolkit [65], and Ngram technique[16]. The proposed approached utilized GibbsLDA++ LDA[134] techniques to extract topics due to its speed and is designed to analyze hidden/latent topic structures of various datasets including text/Web documents. Using GibbsLDA++, 10 topics were estimated with $\alpha = 0.5$ and $\beta = 0.1$. Two hundred Gibbs sampling iterations were performed, saving a model at every 25 iterations. Each time a model is saved, the list of 7 most likely words for each topic are printed.

Table 4.4 shows the results of the domain topic extraction experiment of extracting relevant terms using the 5 mechanisms including our approach. It demonstrates that terms are rich and comprehensive in respect to a user's query and tagging. Acting as Metadata to enrich the user's query and filter the retrieved information of the query, only documents contents that contain the domain ontologies will be mapped to end users.

Using RL learning with the SARSA algorithm, as described in chapter 3, clustered topics are mapped to the RL learning environment, in which each topic group is mapped into a Learning grid cell. The higher level topic group is considered to be the initial goal of the learning process (RL). The feedback of the learning process will be added as a ranking feature by users. The ranking feature is updated as the domain is dynamically updated through its usability. The learning process adopted in this section is the same as the learning process described in Chapter 3. Since extracted topic concepts are based on the corpus contents, domain labeling of extracted topic groups automatically would not be meaningful and might be understandable by human. As RL is used to rank the topic groups that represent document contents relevant to the query, the learning process can be utilized to add a suggested label for each topic group. Such labeling is dynamically updated and similarity is adopted to find the most relevant label for the group contents.

4.8 Summary

This chapter has described how specialized domain concepts are constructed based on end users' queries. As part of the proposed specialized multi-agent learning system for IR, domain construction is an essential component aimed to improve precision of information retrieval by mapping users' queries to the relevant information domains. Not all domains can be constructed manually, or can be constructed based on dictionary or the Internet data. As a result, semantic topic extraction that consists of semantic lexical information, related Internet information that evaluated by the end user through intelligent learning are utilized to construct specialized domains.

This approach enhances the existing IR approaches and search engines by targeting the most relevant information. The general idea is to improve the precision of information retrieval by mapping user queries to relevant information domains.

The construction of knowledge domains - be it manually, using a dictionary, or using the Internet data - is becoming an increasingly difficult task, due to the rapid growth of available information. As a result, an automatic and intelligent approach is an ideal solution to construct specialized domain ontologies. The proposed solution combines semantic lexical information with semantic topics extracted from the Internet, which are further evaluated by the end user through reinforced learning and social tagging.

Chapter 5

Design and Implementation Considerations

5.1 Introduction

This chapter presents the Specialized Multi-Agent Learning System for an IR prototype implementation and design. It employs intelligent agents to provide the specialization and collaboration necessary to construct the specialized knowledge domain. This newly constructed domain enables an intelligent agent to adopt specific tasks, to build a knowledge base about the given task and specializes in performing these and future tasks through interactive learning with end users. The system is designed to be scalable with information growth and users requests. Specialized domains can be created and added in response to unique requests and into information domains. In addition, learning agents are not limited to information discovered individually, as they can also collaborate with each other by sharing learned knowledge. This proposed system is built around the users that specialized intelligent agents (SIA) serve, the IA that learns through interaction with the users, and knowledge-base repository of data.

5.2 System Architecture

The proposed system architecture, illustrated in Figure 5.1, is composed of three layers: the interface layer, the multi-agent layer, and the knowledge-base repository layer. These three layers interact in the following way:

- Interface Layer: A user may, at any given time during his/her active mode on the system, obtain a specific information using the system. Requested information either exists within the system knowledge domains or a new domain relative to the user's query is created utilizing the external information repository such as the Internet. Each query the user makes is dispatched through a user interface (i.e., a Web Browser) and filtered semantically through query enrichment.

- Multi-Agent Layer: This layer comprises of Specialized Agents (SA) that each serve a specific knowledge domain. Each SA consists of multiple intelligent learning agents that collaborate among each other at the learning agent level and among specialized agents. The design of each IA is based on an RL algorithm that can build a specific domain knowledge base that corresponds to the query of the user as it is dispatched and filtered.

- Knowledge Domain (Repository Data): The data repository contains the information users are interested in accessing. This data is categorized, constructed, and ranked (based on user query and feedback) into a specialized domain knowledge base.

This system architecture is schematized in Figure 5.1, below. It demonstrates the proposed system framework layers, and the essential components of each layer.

5.2.1 Implementation Tools

The platform and tools used in the architecture of the SMAL system are as follows:

The system is implemented using interdependent components that were built using various programming languages designed to interface with other tools, such as:

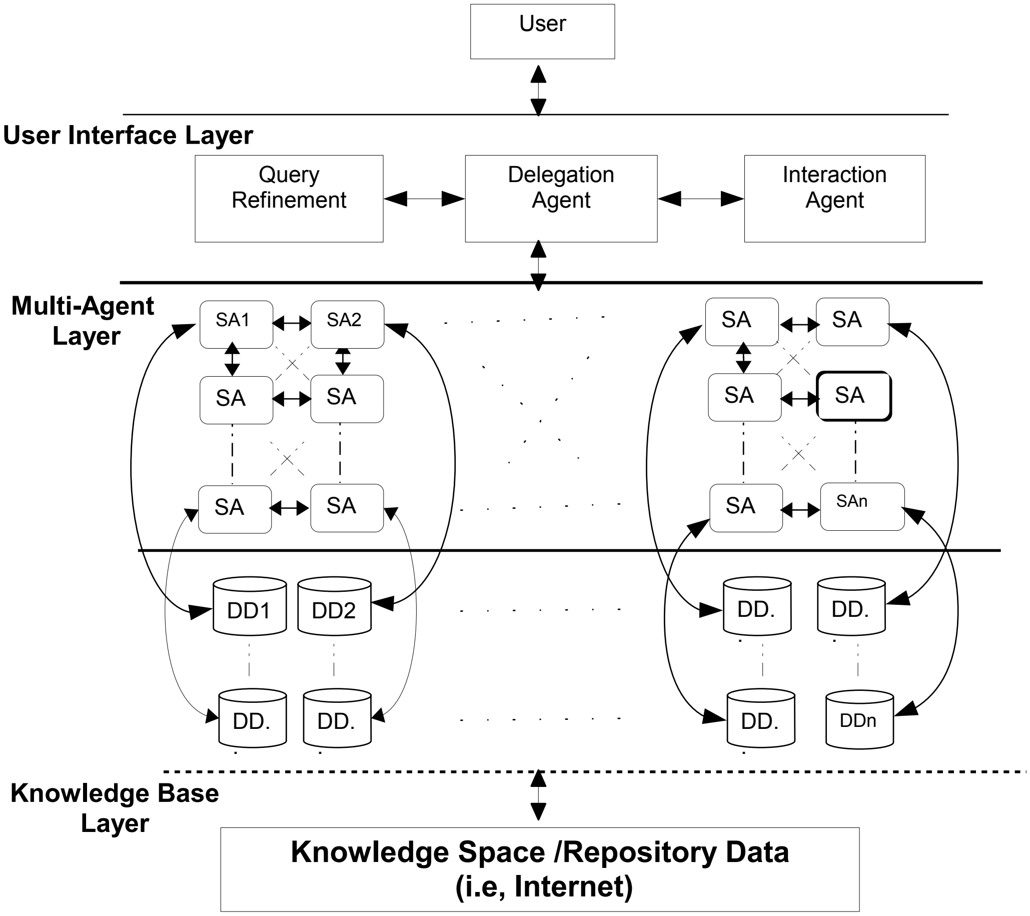


Figure 5.1: Architecture of the proposed SMAL System.

1. Platform;

   - The development based on Linux and computable with other Unix operating systems.

2. Research tools;

   - R Language.

3. Database Engine;

   - Mysql Engine.

4. Web based tools;

   - Apache was used as the Web Server,
   - Web Services based on SOAP,
   - PHP and CGI are used as the web applications.

5. Language(semantic) Engine;

   - WordNet is used to enriched users' queries with synonymous terms.

6. Development tools;

   - GCC 2.4, Perl and python.

7. Agent library;

   - RL-Glue (Reinforcement Learning Glue)

8. IR applications;

   - Lemur Toolkit. The Lemur Toolkit is designed to facilitate research in language modeling and information retrieval, where IR is broadly interpreted to include technologies such as ad hoc and distributed retrieval, cross-language IR, summarization, filtering, and classification. The Toolkit supports indexing of large-scale text databases, the construction of simple probabilistic language models for documents, queries, or sub-collections, and the implementation of retrieval systems based on language models as well as a variety of other retrieval models [81].

5.3 Interface Layer

The interface layer is comprised of the Internet Web Browser that acts as the user interface with the system, and contains the delegation/filtering agent of the user's query. Figure 5.2 illustrates the interface layer process within the proposed system. Apache Web server and Web applications are used to establish a prototype for the user interface and the intermediate engine between the system and the user which consists of delegation and filtering management process.

5.3.1 User Interface

Since users are modeled as part of the Reinforcement Learning (RL) design (referred to as RL environment), a user interface layer is needed as an intermediary to interact with the information system, post queries, receive answers, and define and evaluate knowledge domains. The interface layer of the proposed system for IR is built around Internet WWW web clients such as FireFox and Internet Explorer. WWW web browsers are chosen because they are platform independent, accessible, and easy to use and learn. Moreover, they are the common methods used by end users to access information in various IR applications.

5.3.2 Delegations and Filtering Agent

In addition to facilitating the end users' communication and interactivity with the system (inner layers), the delegation and filtering management performs the following essential tasks:

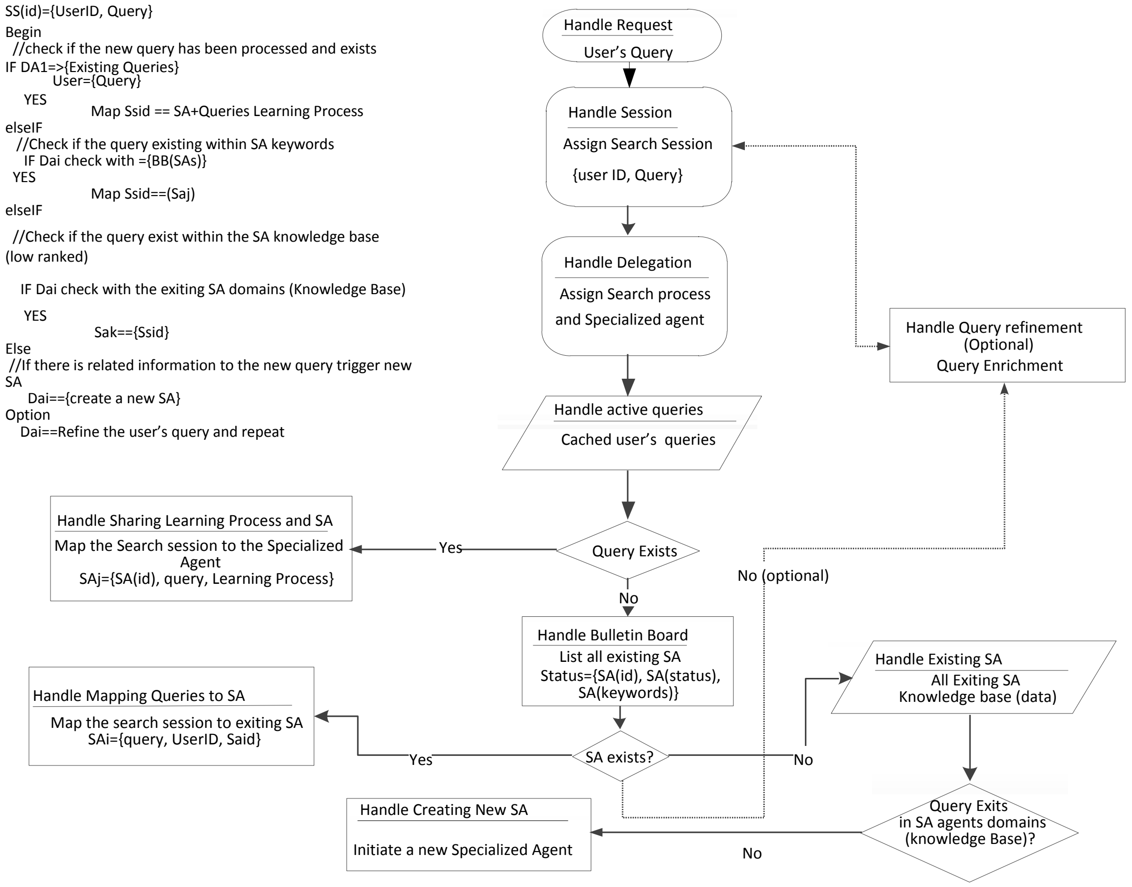
- It determines whether or not a specialized domain exists in the multi-agent layer in order to process the user's query accurately. This process is accomplished via collaboration with the Bulletin-board [74], in which all specialized agents post their profiles as one means of collaboration both with each other and also with the interface layer.

- If the specialized domain does not exist and no relevant documents are found (returns null), the query will be assigned to the domain construction process phase (presented in Chapter 4).
- If the query has a specialized domain, the delegation/filtering agent dictates which specialized agent the user's query is assigned. The specialized domain that has the highest weight of relatedness to the query selected. Other specialized domains that have a lower weight will be also passed to the chosen specialized agent for sharing the domains information. The user's query will be normalized at this level before it is assigned to the specialized agent.
- The delegation/filtering agent initiates in an independent session (through the Web server and applications, and managed by database engine) for each query that it assigns to the multi-agent layer (where the specialized learning agent resides) for learning that includes the user's invisible activities such as feedback, time spent on each document, and the document selected.

Query Enhancement

Semantic search is added as an option for the end user using the system. Leveraging WordNet database, the end user's query is enriched with WordNet lexical semantic synonymous terms (as discussed in chapter 4 for detailed information). WordNet 3.0 is configured on the Linux platform. $WN(Q) = (wnt_1, wnt_2, ..., wnt_i)$, where WN denotes the query extended with WordNet, and wnt_i is the WordNet lexical synonym for the query Q.

The search process of the Semantic search is similar to the regular search except that the users' queries are enhanced semantically.



```
SS(id)={UserID, Query}
Begin
  //check if the new query has been processed and exists
IF DA1=>{Existing Queries}
        User={Query}
   YES
              Map Ssid == SA+Queries Learning Process
elseIF
  //Check if the query existing within SA keywords
   IF Dai check with ={BB(SAs)}
  YES
              Map Ssid==(Saj)
elseIF
  //Check if the query exist within the SA knowledge base
(low ranked)
   IF Dai check with the exiting SA domains (Knowledge Base)
   YES
              Sak=={Ssid}
Else
 //If there is related information to the new query trigger new
SA
    Dai=={create a new SA}
Option
   Dai==Refine the user's query and repeat
```

Figure 5.2: Users' query process flow.

5.3.3 Document Filtering (conversion)

Within this same interface layer, information used within the system based on the text format. Documents retrieved from the Internet are usually tagged with XML, HTML and other media type, (for example; audio, video, and scripting codes). Web pages also contain diverse textual fragments such as bullets or short phrases that carry no information (e.g., date page last revised copyright note), so being able to identify narrative text from non-narrative text is very important when moving from traditional coherent text such as news stories to Web documents. The first step in processing retrieved Web pages is to identify non-text formats. Among non-text formats are, HTML, XML tags, Web application scripts codes are removed, leading to smaller set of Web pages. Plain text is then extracted from all Web pages by utilizing developed filtering program in the proposed system that coincide with Web Spider technique, which is found to outperform several alternative text extraction tools such as html2text.

5.4 Multi-Agent Layer: Learning and Collaboration Process

The Multi-agent layer incorporates specialized agents that facilitate interactive learning, information sharing, and collaboration. An SMAS specialized agent provides tools and utilities for users to access and retrieve relevant information and build repository knowledge based information from a specific domain. An intelligent agent, RL agent, that is embodied within each specialized agent, conducts the learning process, allowing the specialized agent to instantaneously trigger more than one RL learning agent. For example, the specialized agent can receive similar queries from the interface layer. Though queries addressed by the same specialized domain share similar retrieved information, user behaviour is unique and distinguishable. As a result, a new learning agent (RL) is triggered for each query passed to the Multi-agent layer by the interface layer. Through a data caching mechanism, temporary repository-retrieved information is adopted during the agent learning process to facilitate learning. The cache-retrieved information generated by the query during the learning process is shared with and accessed by the same SA. Collaborative information sharing and learning among RL agents is further supported by automated information exchange. RL agents share the same information with one another through caching for similar queries of different users. A detailed description of Multi-Agent Layer components and their learning process is depicted in figure 5.1 and addressed in the coming subsections. Figure 5.3 illustrates the framework design of specialized agent process.

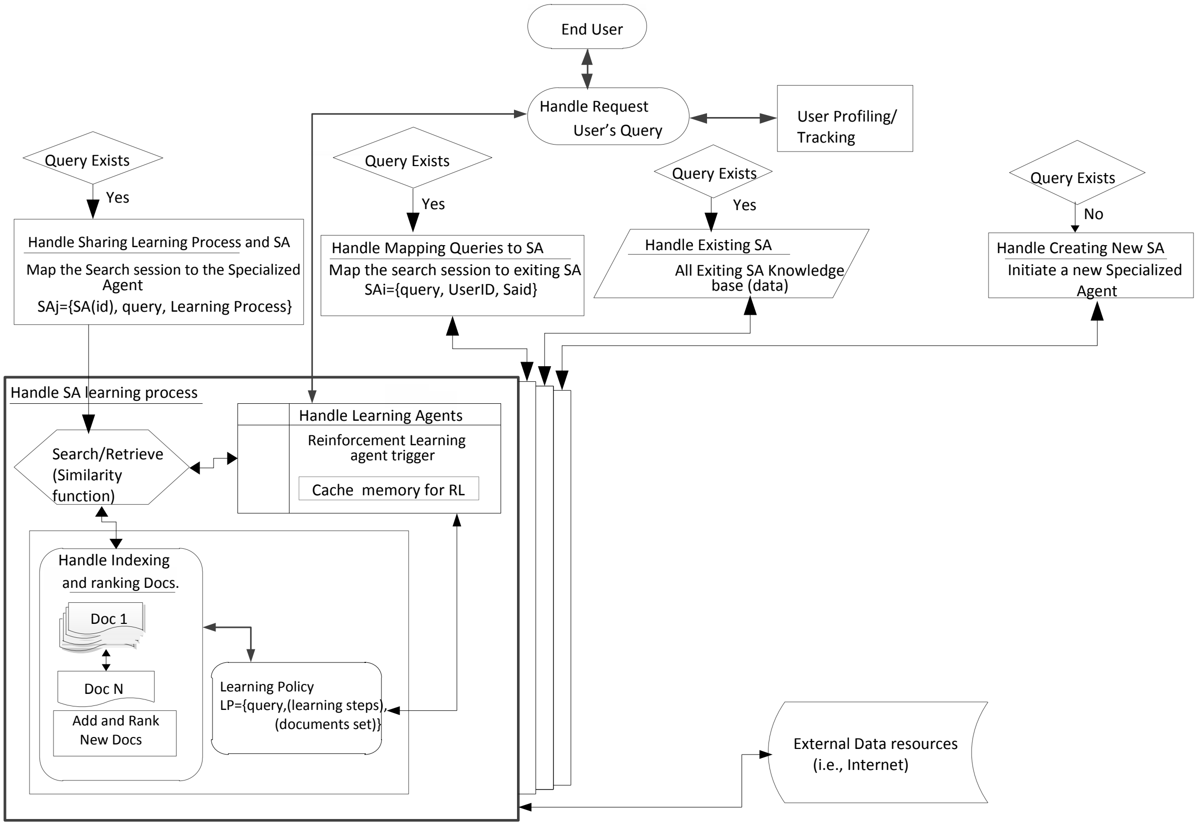


Figure 5.3: Framework design of specialized agent process.

5.4.1 Learning Process

The learning process within the SA is triggered upon receiving the information (command) from the delegation management. This information consists of the specialized agent ID, the query, and links to related information that exists in other specialized agents. Figure 5.4 depicts the learning process of a specialized agent using RL. The learning is conducted through one or two phases, depends on the user's feedback. First, the retrieved information for the learning process is solely based on the existing knowledge base of the domain and other domains. The second phase is processed when the end user is not satisfied with the information presented from the existing knowledge base. The knowledge base information is enriched by information retrieved from the Internet.

Learning through Learned Knowledge Base

Specialized agents can carry expertise in various domains. Each specialized agent maintains its own knowledge base to organize and manage its collection dynamically. Specialized agents can also, in their own collection, amass relevant information from external specialized domain information represented as links. When a user query is directed to a specialized agent through the delegation agent, the delegation agent sends not only the user's query information, but also a set of other specialized domains that might be relevant to the current query. Finding the relevant information within the specialized agent knowledge domain is processed through finding the similarity of the user's query and the indexed information in the knowledge base repository. The standard Cose similarity function is adopted in this research to find information similar to the query within the knowledge-base indexing system. For the semantic option process, Lesk (similarity of two concepts algorithm proposed by Lesk [97])similarity function of two concepts is adopted to find relevant information semantically.

Figures 5.5 and 5.6 illustrate the learning process using RL agent and knowledge base. The retrieved information relevant to the query is stored in the SA cache data file and mapped to the RL agent. The RL learning environment is structured of a set grid matrices in which each retrieved document is mapped to each cell of the RL agent grid. The indexing system of each specialized agent indexes the information based on its word frequency. In addition, user feedback is weighted for each index data set and dynamically updated. Document information that receives a higher weighting through the user's feedback (reward), is ranked the highest in the indexing system. The number of times the document has been used by the SA or other SA is also recorded.

The RL agent process initiates its state at the CELL that host (map) information ranks based on end users' feedback. This is set as the initial goal. The RL agent learning process sets the initial goal as the documents that have the highest rank of feedback: if a user does not select such documents, the RL agent goal will be based on the highest weight of similar words (IDF).

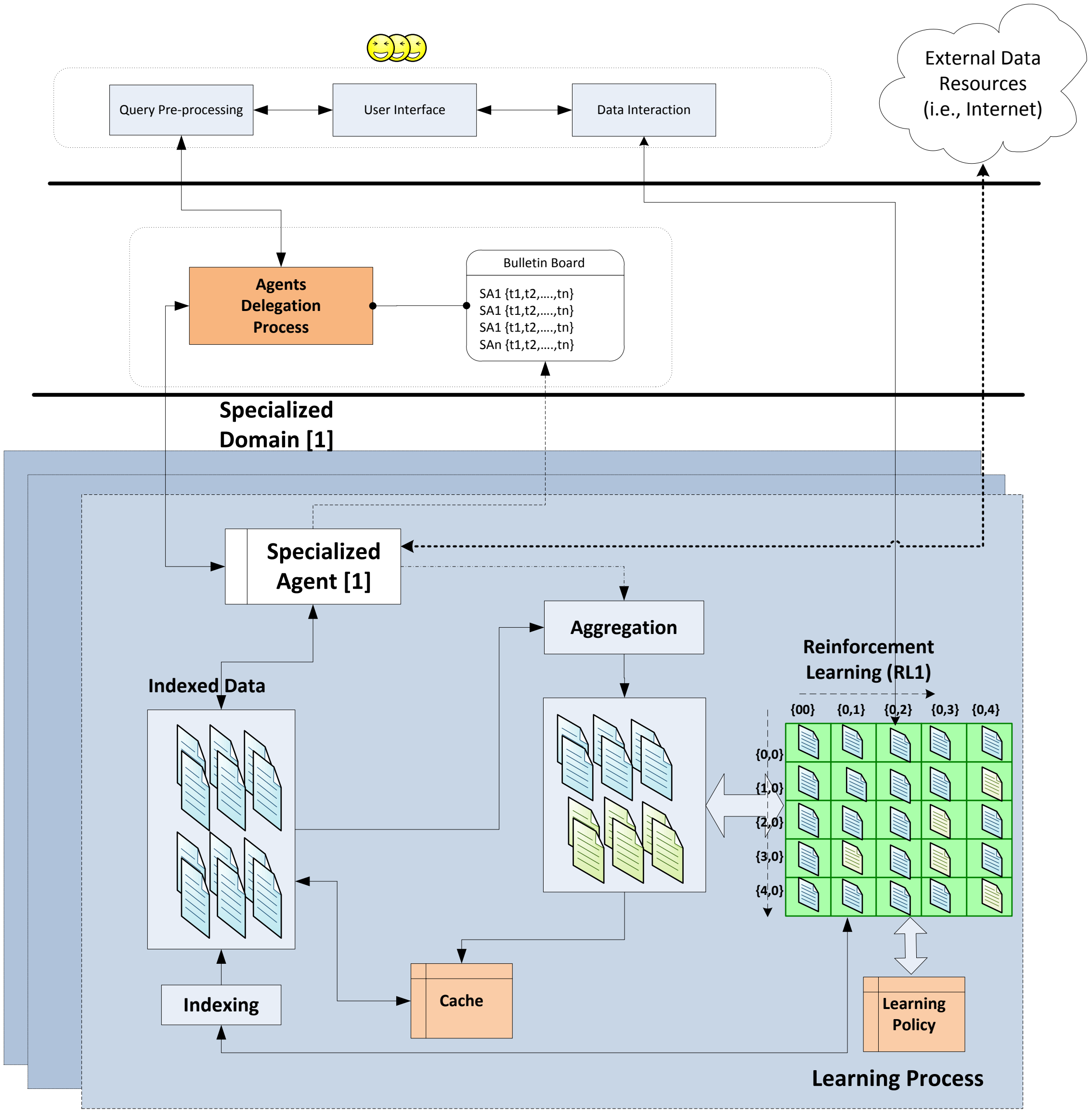
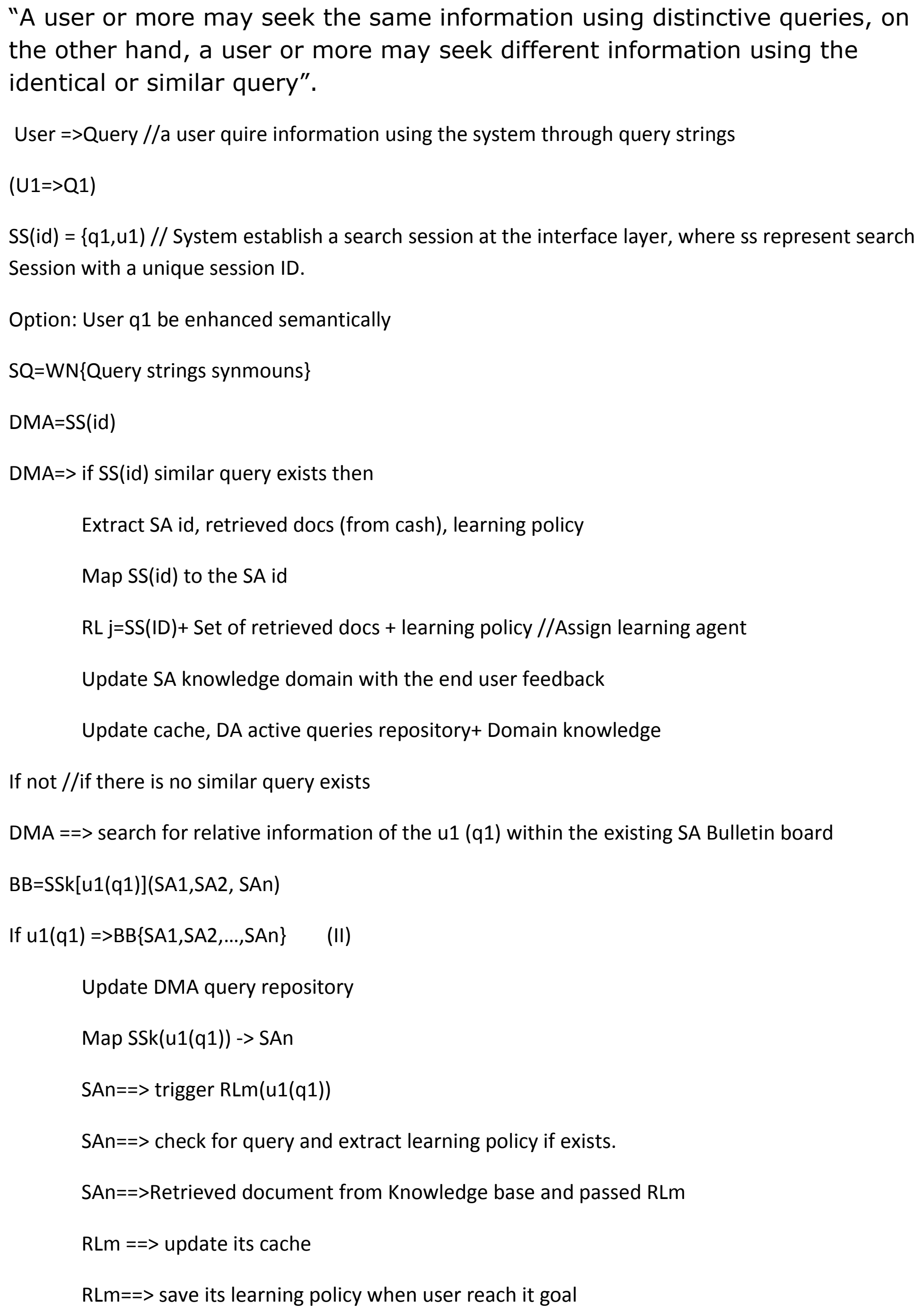


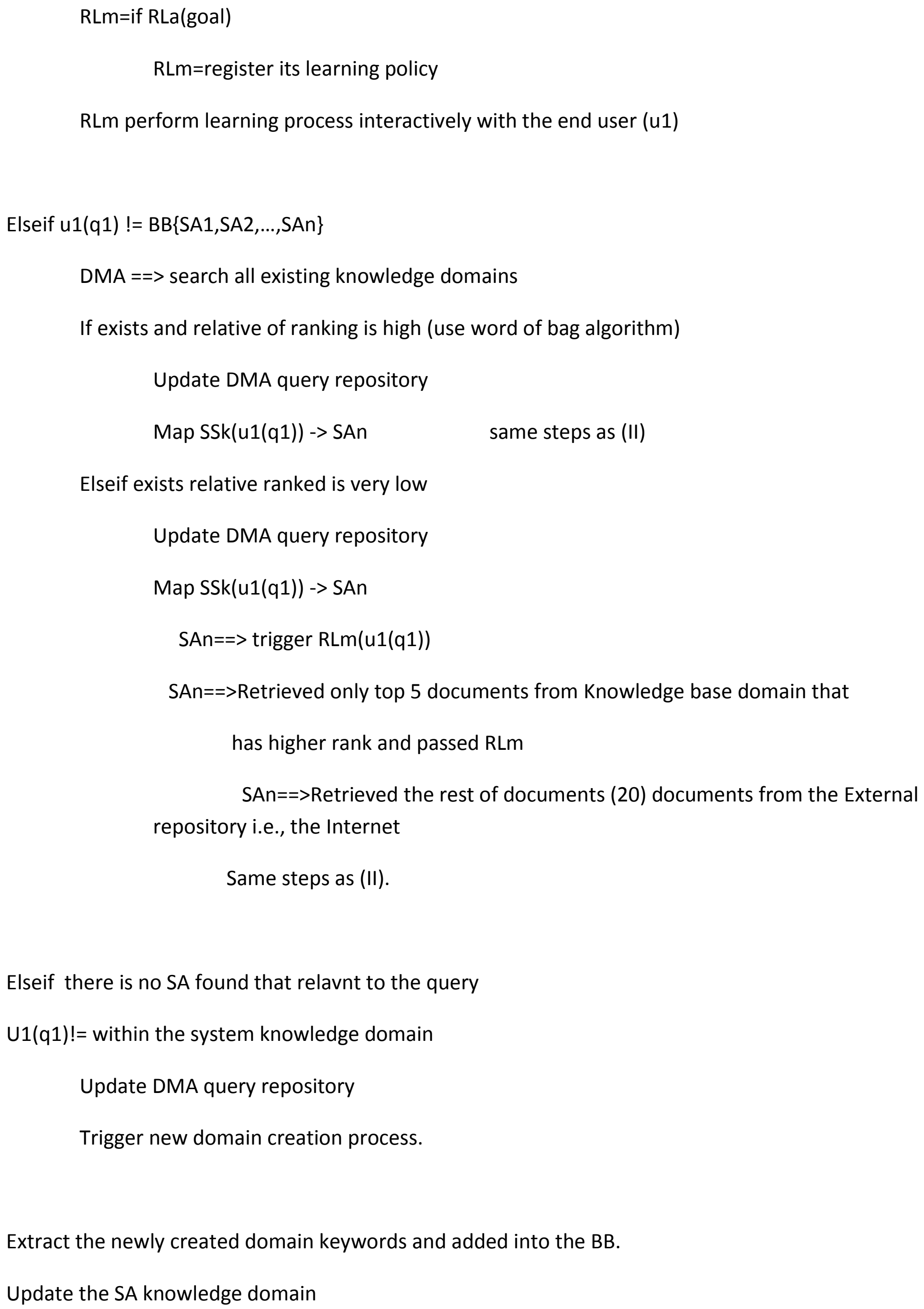
Figure 5.4: Learning process in a specialized domain.



“A user or more may seek the same information using distinctive queries, on the other hand, a user or more may seek different information using the identical or similar query”.

```
User =>Query //a user quire information using the system through query strings
(U1=>Q1)
SS(id) = {q1,u1) // System establish a search session at the interface layer, where ss represent search Session with a unique session ID.
Option: User q1 be enhanced semantically
SQ=WN{Query strings synmouns}
DMA=SS(id)
DMA=> if SS(id) similar query exists then
        Extract SA id, retrieved docs (from cash), learning policy
        Map SS(id) to the SA id
        RL j=SS(ID)+ Set of retrieved docs + learning policy //Assign learning agent
        Update SA knowledge domain with the end user feedback
        Update cache, DA active queries repository+ Domain knowledge
If not //if there is no similar query exists
DMA ==> search for relative information of the u1 (q1) within the existing SA Bulletin board
BB=SSk[u1(q1)](SA1,SA2, SAn)
If u1(q1) =>BB{SA1,SA2,…,SAn}        (II)
        Update DMA query repository
        Map SSk(u1(q1)) -> SAn
        SAn==> trigger RLm(u1(q1))
        SAn==> check for query and extract learning policy if exists.
        SAn==>Retrieved document from Knowledge base and passed RLm
        RLm ==> update its cache
        RLm==> save its learning policy when user reach it goal
```

Figure 5.5: Learning process of SA: Algorithm steps of the system.



```
          RLm=if RLa(goal)
                    RLm=register its learning policy
          RLm perform learning process interactively with the end user (u1)


Elseif u1(q1) != BB{SA1,SA2,…,SAn}
          DMA ==> search all existing knowledge domains
          If exists and relative of ranking is high (use word of bag algorithm)
                    Update DMA query repository
                    Map SSk(u1(q1)) -> SAn                    same steps as (II)
          Elseif exists relative ranked is very low
                    Update DMA query repository
                    Map SSk(u1(q1)) -> SAn
                       SAn==> trigger RLm(u1(q1))
                      SAn==>Retrieved only top 5 documents from Knowledge base domain that
                              has higher rank and passed RLm
                                SAn==>Retrieved the rest of documents (20) documents from the External
                    repository i.e., the Internet
                              Same steps as (II).


Elseif  there is no SA found that relavnt to the query
U1(q1)!= within the system knowledge domain
          Update DMA query repository
          Trigger new domain creation process.


Extract the newly created domain keywords and added into the BB.
Update the SA knowledge domain
```

Figure 5.6: (Continue) Learning process of SA: Algorithm steps of the system.

The RL agent actions and states are guided by the total weight of each document mapped in the RL agent cells to reach the goal, otherwise, by the total number of similar word frequency weight. As discussed in Chapter 4, the document weight that receives the highest rewards from the end user will be added as a new document into the SA knowledge base. The goal parameters of RL agents consist of the following: documents that have the highest rewards in the past, documents that have the highest weight based on word similarity frequency, user activity on each document in terms of selection of documents and time spent on each document. User activities on the system during the learning are conducted through the learning sessions, time elapsed, and the monitoring of documents. The cache data files that deposit the information during learning will be removed once the goal is reached. Documents are chosen and are now in use by other RL agents that might share the same query and cache data file.

The learning process is represented by the weight each document receives relevant to the query within the knowledge base of SA. Initiating the RL agent goal is based on the documents that are ranked by the end user and not by the total number of similar words matching the query. This difference distinguishes the proposed system from other IR systems in which documents are ranked based on word similarity frequency.

Hyper-Learning

"Hyper-learning" describes the interaction that occurs between the proposed system and the Internet. Hyper-learning is triggered when the query result combines information (documents) retrieved from existing knowledge domain and other source such as the Internet. As a result, the specialized agent will retrieve information from the Internet (based on the query) to mix with the existing information retrieved from the knowledge domain. Figure 5.4 illustrates the hyper-learning process diagram. The RL agent learning action starts when the information is retrieved from the specialized agent domain(SAD) knowledge-base and combined with the information retrieved from the Internet. Learning process actions and states are guided by the user's feedback in the form of rewards. The learning process of the RL agent is similar to that of learning through an existing knowledge base section. But the goal setting of the RL agent in this case is based on both the highest weight of related information found in the knowledge base and on the highest frequency of words in the documents. These documents were selected because they received the highest reward by the end user. Their weighting increased or they were added into the knowledge base if they have also been retrieved from the Internet.

Cache Data Files

This ability for the specialized RL agent to gather and store retrieved information for the learning process (e.g., documents link) considered pertinent to the users' query is known "caching". The term "cache data file" denotes the space in which information retrieved by the specialized agent is temporarily stored during the SA learning process. The cache data expires when the RL agents end or terminate the learning process. Cache data files within the SMAS carry the following characteristics:

- They are domain specific;
- They are temporary deposits for information retrieved by the SA;
- They are used to reduce network latency by reducing the search process for the same query already conducted and retrieved by another SA;
- They can be shared by more than one learning agent for similar queries. Each user's query generates a learning agent, since users behave differently although they may search for the same information. This retrieved information is stored in cached data files and can be shared by other learning agents for the same queries.

In addition to the above, they also facilitate collaboration among SAs by allowing SAs to share the same retrieved information.

5.4.2 Collaboration Among Agents

SMAS is designed to facilitate collaboration among SAs. When an agent is specialized to learn about a unique domain, its learned knowledge base is shared with other agents. This forms a set of agents that are each specialized to build a knowledge base about a specific domain through learning. Each SAs knowledge base collects information related to its domain. To avoid redundancy and overlapping among agent tasks and learning, a bulletin board [74], collaborative communication mechanism is adopted. Similar to matchmaker techniques, the bulletin board supports automated information exchange among agents. Each established SA broadcasts its profile on the bulletin board. However, some information can also be shared by different domains to avoid redundancies and increase network latency. For example, in Cancer and Diabetes domains, information can be shared and exchanged among agents. Relevant information is added to SA knowledge bases by links to avoid redundancy of storing similar information. Such communications are conducted within the local system, and collaboration across systems globally is conducted through access validation and identification.

The proposed collaboration protocol [74, 66] is presented as follows:

1. *Bulletin Board Registration*: Each agent registers with the bulletin board agent in order to collaborate with other agents. The registrar is an agent providing information about its *unique identification*, *task (specialty)*, *status*, and *keywords.* Each specialized learning agent within the system is denoted as $SA_{id}(T, s, k_i)$, where SA is the notation of the Specialized Agent, id is the Specialized Agent ID, T is the specialized agent task (speciality), s is the current status of the agent (0 is not-active and 1 is active), and k_i are the key words of the associated learning.

2. *Collaboration*:

   Collaboration among agents within the proposed approach occurs at two levels, specialized agents, and the domain learning agents. Specialized agents collaborate through accessing and sharing learned knowledge of all domains, and the learning agents within each specialized domain collaborate among each other by sharing learned past experiences (learning policy).

   (a) Collaboration among specialized agents: Knowledge base domains are shared among specialized agents. Information are interrelated despite of the domain topic categorized under. Information of a specific topic might interest one user

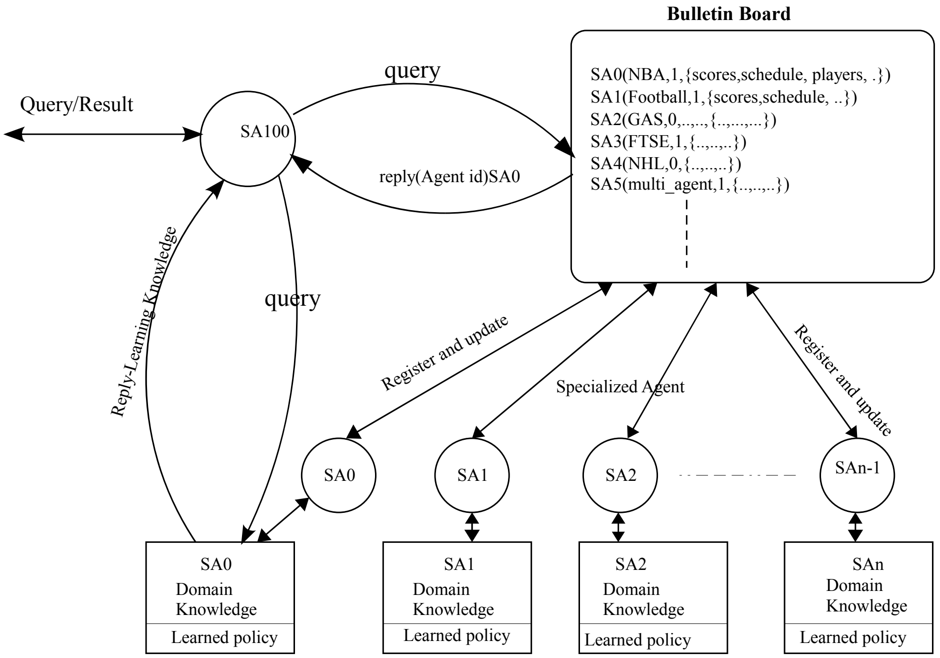


Figure 5.7: Collaboration among specialized agents..

but not another user. Therefore, related information that selected by a user are added into the knowledge domain as a link to the information source. Each agent can access to any SA domain knowledge base as well as temporarily held information stored in the learning agent cache data files. No request nor reply messages (i.e., authentications) are exchanged among the agents, which is advantageous in allowing multiple agents access to the knowledge base without affecting the principal goal. There is no direct agent-to-agent communication [66, 36], as the communication between agents-to-agent repository knowledge

Query Entered	Date Processed	Total Unique Users	Total unique users within one hour
google	8/9/2002	565	32
google.com	9/9/2002	138	45
diabetes	08-09/08/2002	49	49

Table 5.1: Multiple unique Users searching Altra-Vista Search Engine of 2002 Using the same query.

bases and cache data files. Another feature of this approach is that an agent is not required to monitor requests and messages from other agents. Moreover, when an agent finds similar SA keywords residing on the bulletin board, the agent can communicate with others in the peer agent knowledge base to share learned knowledge. Figure 5.7 illustrates the agents' collaboration process. At time t_0, SA_1 begins to search for specific queries on the bulletin board register of agent information for agents that may have already learned knowledge about similar queries. If SA_1 query(ies) are found on the bulletin board, the SA_1 connects to the SA_{id} knowledge base containing the required learning knowledge.

(b) Collaboration among Learning agents; RL agents share their experiences . The proposed approach is designed around the end user mainly because IR essentially serves and depends upon the end user to retrieve and evaluate its process. Naturally, end users do share many similar and related information inquiries that are based on behaviour, gender, locations, and current events(e.g; students learning about the same subject, or users interested in knowing about particular news events). IR depends on the end users' queries for its process. Altra-vista actual user' query datasets have shown that different users search for similar information within the same time or in very close time windows [113].

Table 6.5 depicts patterns of various users performing the same query in searching for the same or similar information using the Altra-vista search engine [113]. In addition to sharing the learned knowledge base, the proposed system has enhanced the collaboration at the learning agent level by sharing the learning

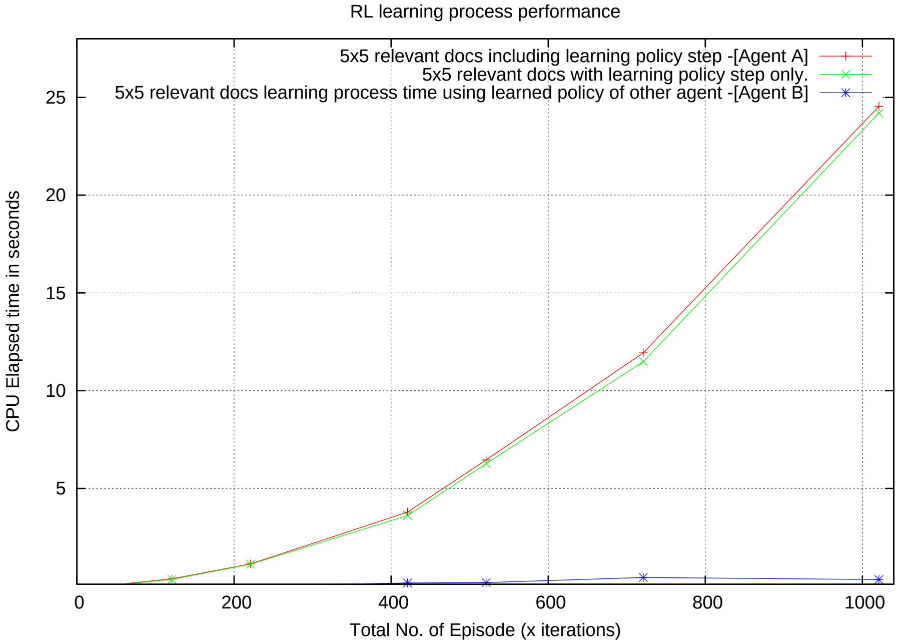


Figure 5.8: Learning performance through collaborating –sharing learned policy

policy of each agent when the queries are the same. Similarity function was used to confirm the similarity between queries; the $TD - IDF$ weight (term frequency inverse document frequency) algorithm was used on value 0 no similarity of 1 when the two queries of two different users are the same. Such an approach has led to better enhance the learning process of the agent. Figure 5.8 illustrates the learning performance which the CPU time has enhanced significantly using the same RL SARSA algorithm and its learning parameters with 25 documents in the learning environment. The end user might prefer to select a different document.

Collaborative information sharing and learning among agents is further supported by automated information exchange. Specialized agents exchange information with one another through the construction of a Specialized Domain Data Base.

5.5 Knowledge-Base Layer:Building Learned Knowledge Base of Specialized Agent

The knowledge space layer within the SMAS system is composed of two mechanisms for collecting and storing data: first, information constructed by SA through learning from the end user's feedback, which is described as the knowledge base belonging to a unique specialized agent; and second, a repository data set that is unstructured and not organized into domains. These data types are known as a knowledge space (for example, Internet information). The knowledge base repository data of each SA is formed of indexed text document. Each of these documents within the knowledge base is tagged with the information about the documents, such as the document ID, total rewards, total number of times used, and total word frequency to the domain concepts. Documents are also dynamically ranked so that document with a higher weighting is placed at the top of the list. The retrieved query information is temporarily added into the specialized agent knowledge base repository temporarily. If the specialized agent receives a similar query, such information is used and shared. Each specialized domain knowledge is dynamically updated and indexed utilizing the Lemur IR Toolkit [81]. Information with low scores and no usage records are eventually purged.

5.6 Summary

SMAS implementation and design for information retrieval incorporate three hierarchy layers: interface, multi-agent, and knowledge base layers. The interface layer forms the interaction medium between the system and users. User interface, delegation, and information filtering functions are processed in this layer. The multi-agent layer consists of several specialized agents, where each agent is built from an RL algorithm specialized to perform learning in the specific domain. The proposed system not only facilitates learning by SAs, but also encourages collaboration among agents to exchange information and learned experiences. The knowledge-base layer consists of the data repository that specialized agents build, learn, and dynamically updated through interdependent activity, to form specialized domain knowledge base. The SMAS provides greater interactivity and efficiency than traditional systems, and thus produces greater satisfaction for the user.

Chapter 6

Experiments and Results

6.1 Introduction

To demonstrate the viability of the proposed system hypothesis, proof-of-concept experiments and case studies are detailed in this chapter with their results and analyses. Mapping users into the relevant retrieved information, namely an intelligent learning process of constructing specialized domain knowledge based on users' feedback and behavior, is investigated and constructed. The specialized domains are evaluated qualitatively, by means of precision and recall. The experiments and study cases presented in this chapter aim at aligning the proposed framework of a specialized multi-agent system for information retrieval tasks. The first task involves constructing specialized knowledge domains, and the second task evaluates the proposed approach of building such specialized domains. The task of constructing the knowledge base of specialized domains is based on two distinct information resources: dynamic data such as information resources available on the Internet, and static data such as the MEDLINE journal abstracts data set (OHSUMED) [51]. Constructing specialized domains involves crawling, intelligent learning process, building specialized domain topics models, indexing, and building knowledge domains. The proposed system was evaluated using established information retrieval measuring mechanisms such as precision and recall in comparison, and compared with other IR applications such as search engines. Furthermore, the proposed system was evaluated with one of the IR techniques, TREC-9 [129], a well known evaluation technique the IR research community. TREC-9 evaluation techniques (tfidf, okapi, kljm, klabs, and twostage) [4] were also used for consistences, and the IR application (Lemur) as well as the knowledge domain, in this case Diabetes.

This chapter first describes the experimental setup and the data sets used to build the knowledge base of specialized domains. The chapter then introduces the measures used to evaluate the performance of the specialized agents, RL, and learning and collaboration. Three case studies are used: Case Study I to demonstrate that the proposed approach can be embedded and integrated with existing IR applications and search engines, and Case Study II to compare the proposed approach with commonly used search engines. Case III highlights the advantages of enriching users' queries semantically to further enhance IR processing of relevant information; static and dynamic data were used.

6.2 Experiments and Case Studies Datasets

The data set used to demonstrate the hypothesis of the proposed approach of constructing specialized domain knowledge and domain topics consists of two sets of data: dynamic data and static data. For the domain construction using dynamic data, Web documents were crawled from Internet resources using search engines and actual end users queries. Crawling was conducted using real user' queries over Internet search engines such as Google, Yahoo, BING, and a specific search engine, the National Library of Medicine (NLM). Document collection consists of 348,566 abstracts collected from 270 medical journals over a period of five-years (1987-91) by the National Library of Medicine on-line medical information database (MEDLINE)-known as OHSUMED TREC-9 [25, 129]- represents the static data set. Specialized domains were constructed using both data sets independently but with the same domain topics.

6.2.1 Dynamic Data

As part of constructing the specialized knowledge domain bases, two subsets of the dynamic data set were used: actual users' queries, and web documents retrieved from the Internet. The query data sets simulate exact end user' queries that have been used in a real search engine. In this research, Excite– a major Internet media company offering Web searching and personalization portal– user' query data sets were used for searching information from the search engines. The Excite query data sets were culled from greater than one million queries submitted by more than 200,000 users of the Excite Web search engine and AltaVista, collected in September 1997, December 1999, May 2001, and March 2003 consecutively [113, 58, 57, 120].

Characteristic of Excite Queries Data Sets

Each Excite query log record contained four fields [57, 120, 59, 88]:

- Sessions–entire query sequence by a user.
- Identification–anonymous code assigned by the Excite server to a user machine.
- Time of day in hours, minutes, and seconds.
- Queries–one or more terms as entered by users.
  - Terms–any string of characters bounded by white space.

For the purpose of constructing specialized knowledge domains, queries that are related to the unique domains were considered categories and extracted from the data set to crawl information from the Web. The WordNet query term ”synonymous” was used in this process.

In addition to the Excite query data set, the Alta-Vista query data set is also used. Specifically, the Excite data set is used for crawling the Internet, while the Alta-Vista query data set is utilized to evaluate the multi-agent collaboration of the proposed approach. The

Collection	**Corpus Source**	**Total No. Queries**	**Data Type**
Data Set-1	Excite User Query Altra-Vista User Query	1 million queries 1 million queries	Users Queries (2001) Users Queries (2002)
Data Set-2	Internet	20,000	Web Documents
Data Set-3	OHSUMED-87-91	348,566	Text

Table 6.1: Description of Data Sets Used in the Experiments

Web document retrieved from the Internet data set is a collection of 10,000 Web pages for each unique domain, crawled using query sets of the Excite Query data set of 2003 and four well known search engines: Google, Yahoo, Bing, and the National Library of Medicine (NLM). All retrieved Internet documents were pre-processed, they were converted from Web format into text document format. Web content (i.e. HTML tags, embedded images, links, and Web codes) were removed.

6.2.2 Static Dataset

The OHSUMED (87-91) [25, 129] dataset from the TREC-9 filtering track is used in this thesis to evaluate the proposed approach study case experiment. The OHSUMED document collection is a set of 348,566 abstracts collected from 270 medical journals over a period of five-years (1987-91) by the National Library of Medicine on-line medical information database (MEDLINE). In addition, the TREC-9 data set consists of OHSUMED and MeSH topic files (files: query.*) and relevance judgment documents (files: qrels.*).

The common similarity of all data sets is they are unstructured text data. However, the resources of the data as well as the length of documents differ. Internet data on average are one-page long articles, while OHSUMED's data are short consisting of two sections; the document description and the document abstracts. The Internet documents are unstructured and noisy with Web tags and attributes (called noisy tags and attributes), while OHSUMED's is a text in TREC format.

Table 6.1 depicts the data set resources and attributes that are used in this research.

6.2.3 Domains Topics: Specialized Knowledge Base

In order to carry out a set of experiments to evaluate the novel specialized knowledge domains in mapping the user to the relevant information, a prototype applying the multi-agent learning system for IR over the Web information resources was developed. In these experiments and study cases, three knowledge domains were constructed; Diabetes Health, Eye Health, and Healthy Diet. Interrelated domains such were chosen to allow the specialized agent of each domain to collaborate and share related information. Domain knowledge-building is based on finding, first whether a domain already exists; second, mapping the users' query to the most relevant knowledge domain; third, retrieving the relevant documents; fourth, recommending the most likely relevant documents to the user through learning, and last, augmenting the relevant document into the knowledge base domain. This process is conducted by specialized agents and learning agents. Agent learning, collaboration, and performance (converging) are presented and discussed in the next sections.

6.2.4 Challenges and Resolutions

Building a knowledge base on specific domains through user' queries and feedback requires large-scale query logs and many user' queries over a period of time. To meet such requirements of building a specialized knowledge base, regression testing and batching process methodologies were adopted in these experiments. Obtaining large-scale query log data sets from actual users was made possible through the Excite Web Search Engine query data set of 2003. Excite Web search consists of more than one million queries submitted by more than 200,000 users of the Excite web search engine [57, 120, 59, 58]. Using such large data sets of queries led to modifying the learning algorithm to adopt a batching process instead of individual base learning. The objective of the learning agent is to intelligently recommend the best possible set of documents to the end user based on sets relevance to the user's query. The ultimate document that would be augmented into the domain knowledge base is selected according to, first, its weight (relevance and/or its ranking within the domain knowledge base), and second, the number of documents found in the agent learning path from the initial state into the goal. Such documents usually have higher relevant weights. To mimic and automate end users' behavior for regression testing, users' actions within the application were captured and recorded during testing. A regression application called System Load Test (SLT) that is used by Xerox, known as ValueQuix, was adopted [139]. SLT is an automation program used by ValueQuix quality assurance teams to perform regression testing. Utilizing Excite search engine user' actions records, Xerox's regression application scripts were integrated into thesis experiments. The regression testing actions are listed as followings:

1. User entered a query,
2. User re-entered a query,
3. User selected one document from the results,

Whether the learning agent (SARSA) exploits the learning environment depends on the initial state (from where the agent starts) and the weight (rewards) of the documents that would lead the agent to its goal. The learning process would take different paths to reach its goal. In real-time learning, a user would select any of the documents recommended (presented) by the learning agent, and the user might also select a document in spite of its relevant weights. In other words, the selected document (among the set recommended by the agent) can be any of the documents within the learning domain. Also, a user can enhance or change his/her query at any give time without selecting what the agent recommends. Therefore, in this experiment, the selection of a document would be done randomly by the learning agent from among the set of documents suggested by the intelligent agent. Since the document selected by the real user would be predicted to be among those recommended by the learning agent, the learning process of RL agents was redesigned to process with large numbers of queries and the learning process in batches (regression process).

6.3 Agents Performance: Preliminary experimentation

6.3.1 Specialized Agent performance using RL

The learning environment size and structure have an impact on the RL algorithm performance. Information retrieval applications such as search engines are critical in respect to the efficiency of getting the retrieved information to the user. RL agents explore and exploit the learning environment to reach their goal. The number of episodes and iteration of learning, as well as the total number of documents presented to the agent, are factors in such learning. For the proposed approach, the number of episodes and size of the learning environment (total number of documents presented to the agent to explore) were determined based on a collective experiment. Figure 6.1 illustrates the RL SARSA agent performance in respect to conversion as well as to the total number of documents.

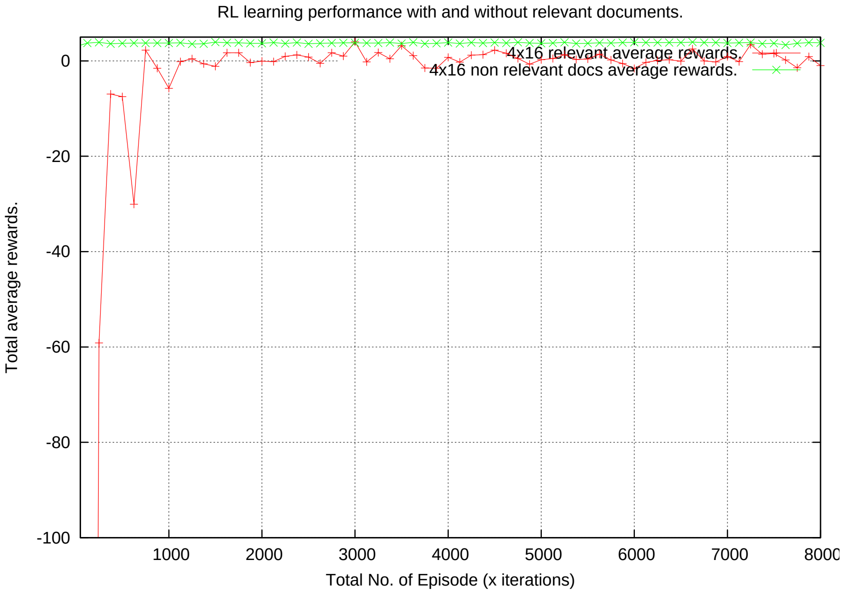


Figure 6.1: SARSA learning converges using relevant and non-relevant documents.

The objective of the first experiment is to prove that RL was the correct choice in respect to other machine learning algorithms. RL did converge during the learning process.

The total number of documents that RL processed and converged was between 25 and 64 documents per learning environment, which is an ideal number for end user to choose from.

One of the critical parts of IR applications and search engines (besides finding the relevant information for the user), is the efficiency of retrieving and presenting the relevant information to the end user in an online, real-time process. This is one of the factors why RL SARSA was chosen from among other machine learning algorithms. The experiment consisted of two parts: SARSA converged and performed through a set of crawled documents from the Internet and retrieved documents from the learned knowledge domain base. Documents crawled from the Internet showed a low weight of relevance to the query, while documents retrieved from the learning knowledge base repository has a higher weight of relevance. Taking into consideration the total number of retrieved documents ranged between 24 and 64 in each learning process environment. The SARSA algorithm parameters are the same during experiments; $\gamma = 1.0, \lambda = 0.1, and \epsilon = 0.1$. In the second part of the experiment, the user varied a total number of episodes during learning. The total number of episodes per learning ranged from 100 to 1000 episodes, each doubled with a number of iterations.

Table 6.2, 6.3, and 6.4 illustrate summary results of collective experiments to evaluate and conclude the total number of documents needed per learning process in our proposed system learning environment in respect to both the total number of episode and the CPU performance per learning cycle.

Table 6.2: Collective experiments to evaluate the total number of documents used in Learning: 5x5 documents (grid) per learning environment.

Total No. Episodes	**CPU Latency:**	**CPU latency:**
Learnign Grid	**Relevant Docs**	**Non Relevant Docs**
100	0230	0.330
400	2.500	0.760
500	5.210	3.900
700	7.40000	4.47
1000	14.250	5.050

Table 6.3: Collective experiments to evaluate the total number of documents used in Learning: 6x6 documents (grid) per learning environment.

Total No. Episodes Learnign Grid	**CPU Latency: Relevant Docs**	**CPU latency: Non Relevant Docs**
100	0.140	0.580
400	5.700	1.420
500	9.280	2.150
700	16.450	4.210
1000	33.040	8.260

Table 6.4: Collective experiments to evaluate the total number of documents used in Learning: 7x7 documents (grid) per learning environment.

Total No. Episodes Learnign Grid	**CPU Latency: Relevant Docs**	**CPU latency: Non Relevant Docs**
200	1.670	0.520
400	5.930	1.360
500	8.70	1.940
700	17.800	3.740
1000	32.910	7.370

Analysis

As Figure 6.1 illustrates, RL SARSA converges throughout the learning process with higher or lower relevant weights. However, the total number of episodes per learning has a significant affect on the learning performance. As shown in figure 6.2 and table 6.2, 6.3, and 6.4, the CPU latency rises when the total number of episodes increases. It was also observed that documents with less or zero relevant weights would use less CPU time learning performance for the agent to reach its goal.

The learning environment size, that is,the total number of documents used in learning and the number of iterations, are essential factors for the RL agent to converge and perform efficiently. Fig. 6.2 depicts the correlation between the total number of documents and

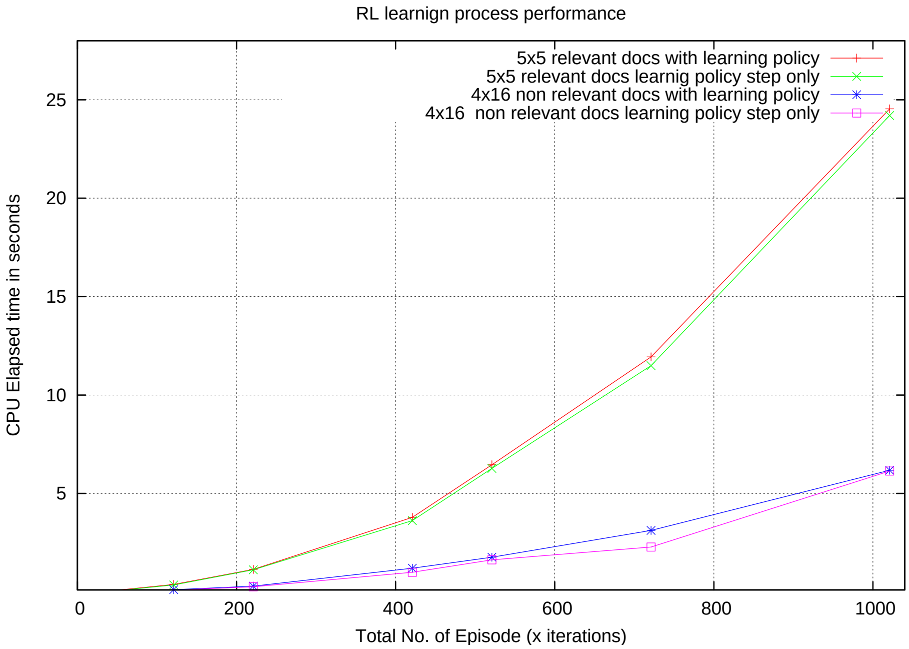


Figure 6.2: SARSA learning converges using relevant and non-relevant documents.

total number of iterations for the RL agent to reach the goal. Though the SARSA converges to reach its goal, the main factor in the approach is the weight (reward values of each document as well as the total number of episodes and iterations per episode) for the learning RL agent (SARSA) to build its learning policy.

As shown in table 6.2, 6.3, and 6.4, CPU latency increases with the increase of the episodes as well as the set of documents' weights. Based on collective experiments, the total number of ideal documents and episodes for the SARSA agent to perform under two seconds would be 25 documents and 200 episodes, with 100 iterations per learning episode. It can be conclude that the learning process using the RL SARSA algorithm with IR in a real-time yield converge efficiently. Two factors need to be considered using RL SARSA within the IR application and search engines; one, the total number of episodes that the agent needed to learn in real-time, and second, the total number of relevant documents used in the learning environment.

In addition to the RL algorithm's converging efficiently in its application in the proposed approach, the learning policy of each agent is utilized to be shared and used as the initial

Query Entered	**Date Processed**	**Total Unique Users**	**Total unique users within one hour**
google	8/9/2002	565	32
google.com	9/9/2002	138	45
diabetes	08-09/08/2002	49	49

Table 6.5: Multiple unique Users searching Alta-Vista Search Engine of 2002 Using the same query.

step in the agent learning process. The RL algorithm has also evolved into a new way to enhance the multi-agent process of the RL algorithm within IR applications. In addition to allowing collaboration among the specialized agent to share and access each other's knowledge base, the specialized sub-learn agents also collaborate among each other. The next subsection presents how the RL SARSA agents collaborate with each other.

6.3.2 Collaboration performance Among Learning Agents.

This section presents an experiment that illustrates the advantages of sharing learned policy among learning agents.

The proposed approach is designed around the end user mainly because IR essentially serves and depends upon the end user to retrieve and evaluate its process. Naturally, end users do share many similar and related information inquiries that are based on behavior, gender, locations, and current events, e.g; students learning about the same subject, or users interested in knowing about particular news events. IR depends on the end users' queries for its process. Alta-vista actual user query datasets have shown that different users search for similar information within the same time or in very close time windows [113].

Table 6.5 depicts a patterns of various users making the same query in searching for the same or similar information using the Alta-vista search engine [113]. In addition to sharing the learned knowledge base, the proposed system has enhanced the collaboration at the learning agents level by sharing the learning policy of each agent when the queries are the same. Similarity function was used to confirm the similarity between queries;

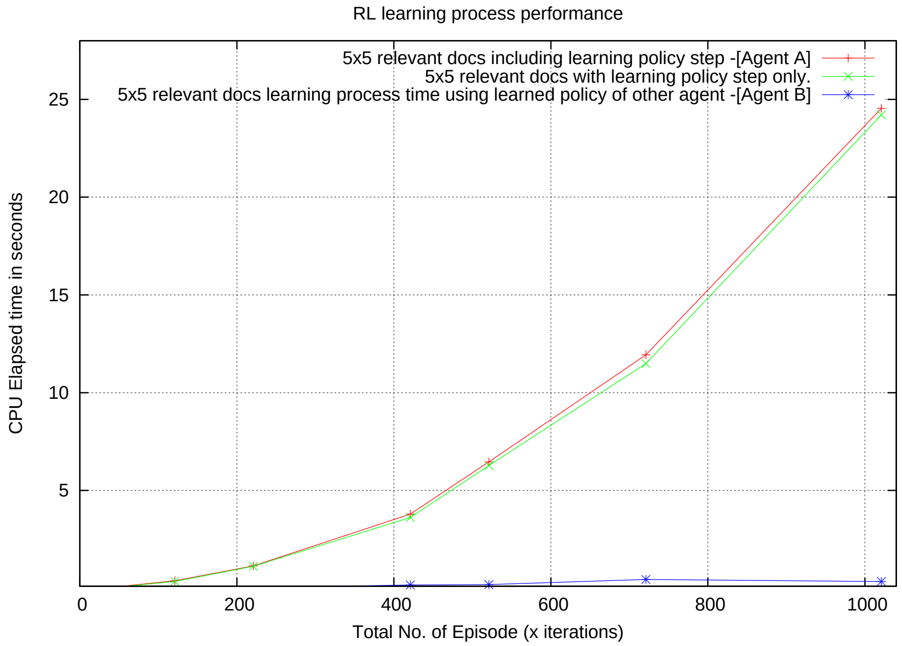


Figure 6.3: Learning performance through collaborating -sharing learned policy

the $TD - IDF$ weight (term frequency inverse document frequency) algorithm was used on value 0 no similarity of 1 when the two queries of two different users are the same. Such as approach has led to better enhance the learning process of the agent. Figure 6.3 illustrate the learning performance whereby CPU time has enhanced significantly using the same RL SARSA algorithm and its learning parameters with 25 document in the learning environment. The end user might prefer to select a different document.

6.4 Study Case I: Integrating the Proposed System with existing IR Applications

As a proof of the concept that the proposed system can be embedded and integrated with existing IR applications, a case study is presented in this section. The case study aim to serves three purposes: first, to demonstrate that the proposed approach can be embedded and integrated with other (existing) search engines and/or IR applications to enhance their performance; second, to apply topic model ontology to build specialized domain topics to be used for refining the domain queries. Query refinement and filtering is intended to enhance mapping queries into the relevant information within the specialized knowledge base of the domain, and third, to evaluate the proposed approach using a standard IR evaluation mechanism (in this case, the TREC evaluation approach was used). Furthermore, using the TREC approach has allowed the proposed system to evaluate the experiment with a large data-set including large sets of queries and to compare results with the provided relevant judgment documents (i.e, *qrel.∗* files) as presented by TREC for the OHSUMA-87-91 data-set. Lemur Toolkit, one of the IR applications, was used in this case study. Lemur Toolkit [81] supports various features in addition to supporting a simple text processing IR system that allows integration with our proposed system. Among these featuers are:

- Lemur opensource license and API (Application Performance Interface) allow the possibility for the proposed approach to interface and integrate seamlessly,
- Lemur Toolkit accepts text documents in TREC format,
- Lemure Toolkit allows indexing of large-scale Ad hoc and query-structured data-sets, and
- The retrieving mechanism implementation is based on a simple language model as well as on a variety of other retrieval models.

The indexing and retrieving of the Lemur Toolkit was enhanced by augmenting the specialized agent learning document selection (ranked by learning agent by adding a weight value during indexing). In addition to Lemur indexing, each indexed document has a weight value in the hash-table.

6.4.1 Case-Study Setup

The case study was conducted over the OHSUMED ad hoc document collection used for the TREC-9 Filtering Track (including documents, topics, and relevance judgments). The OHSUMED data collections are relatively large and represent various topic domains, with

348,566 documents and 4904 topics (queries). The case study consists of the following steps:

- build domain knowledge using a real user's query as found in the Excite query dataset;
- enrich each domain knowledge with documents retrieved using existing search engines;
- build a query set for each domain base to evaluate each domain IR performance;
- build domain topics for each domain;
- index each domain knowledge base using Lemur Toolkit;
- Apply TREC tools to evaluate the constructed domain based on the evaluation judgment document of the OHSUMED dataset:
  - process the TREC for each domain using each domain queries set,
  - process the TREC for each domain using the domain topic to enhance mapping,
  - process the TREC of each domain query against the general OHSUMED dataset.

Applying the learning process of the proposed approach (RL SARSA algorithm), three specialized knowledge domains were constructed from the OHSUMED data set, search engines, and query datasets within Diabetes and Eye domains. The domains were built using real user queries extracted from Excite-2003 query datasets. Fifty queries for each domain were used to retrieve information from the OHSUMED datasets and the Internet (using various search engines), indexed using Lemur Toolkit. The extracted queries were pre-processed to be used for the Lemur application indexing format. Each query set was processed 20 times; in each, one document was selected by the agent. The learning agent selects document from among those that have a higher co-occurrence in respect to the query and as part of the document sets that leads to the agent goal. The $TDIDF$ similarity function was used in ranking the documents with higher word relevancy to the query. Redundancy documents that were augmented by the learning agent into the specialized constructed domain were purged. Lemur Toolkit (an IR application) was used to build the index of the OHSUMED document and to retrieve the information based on the Excite query sets for each domain. There were 875 documents in the *Diabetes* domain, and 874 documents augmented to each domain after the duplicate documents were removed. The document sets represent the domain knowledge base for each domain. Using the TREC tools set for the OHSUMED dataset, three query sets were extracted from the 4905 topics (queries) based on keywords relevance to the three domains constructed. In this case, WordNet was used to build synonymes of the two domain topics, "Diabetes" and "Eye"; duplicate queries were purged. The Lemur index-building tool was used to index each newly constructed domain. The last stage was to use the TREC tools including the *trec_eval* which are the standard tools used by the TREC community for evaluating the given results file and a standard set of judged results.

As a result, 63 queries set for the *Diabetes* domain and 42 queries for the *Eye* domain were constructed using various search engines. Statistic ontology using LDA was utilized to create a domain topic for each of the created specialized domains.

Furthermore, the proposed system was integrated with common search engines such as Google, Yahoo, Bing, and NLM to build specialized knowledge base. Figures 6.4, 6.5, 6.6 and 6.7 depict the precision and recall of the proposed system using the specialized knowledge base that was retrieved using various search engines with the OHSUMED dataset. The knowledge domain was specifically related to *Diabetes* information that was built separately using OHSUMED data-set and search engines. The task of the proposed approach is to build specialized knowledge domains in which the task of Lemur Toolkit (including modified API codes) is to index, search and retrieve information.

6.4.2 Analysis of the Study Case experiments

The results presented show that specialized domains as well as the domain topic have enhanced precision and recall (the performance of the Lemur search engines) when the query and the dataset are static.

As shown by the collective experiments and study case, the growing information available for the end user can be addressed through segmenting large data into specialized

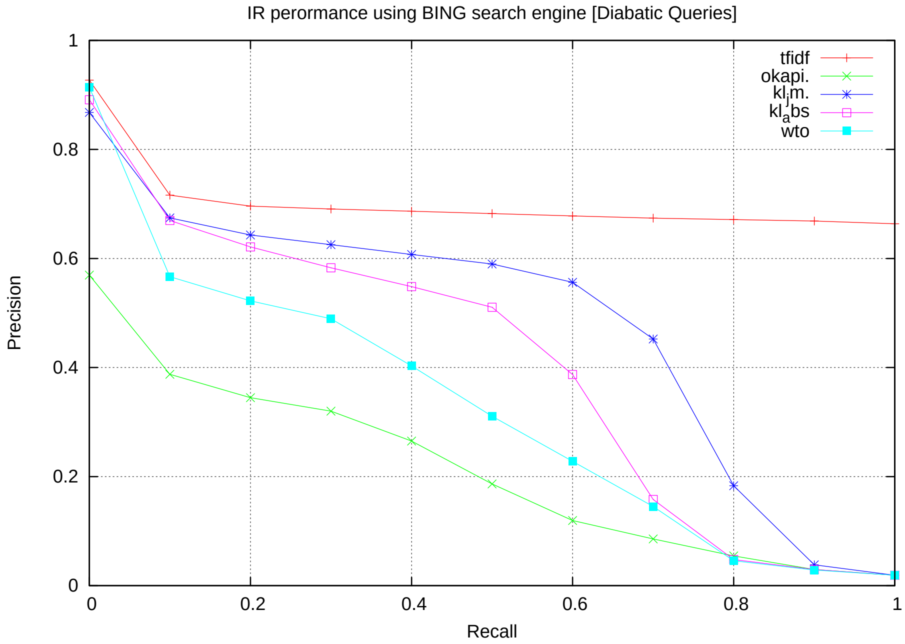


Figure 6.4: Precision and Recall of Specialized Domain Construction with BING Search Engines.

domains. Using a specialized learning agent is an ideal machine learning approach that addresses user's needs. Furthermore, domain topic modeling has further enhanced mapping queries into the relevant documents.

The purpose of these experiments was to build specialized domains from the OHSUMED data set and evaluate these domains using the OHSUMED TREC provided queries. The queries were pre-processed based on the domain.

The knowledge domain was constructed using 50 Excite queries for each domain. As a result, 875 document were augmented into the Eye domain, and 874 document with the Diabetes domain. Since only one document among the retrieved documents can be augmented into each domain by the learning agent, each query is processed 20 times to simulate 20 possible documents being selected from the retrieved data. The batch process was conducted to simulate 20 users, and each query was processed 20 times to retrieve

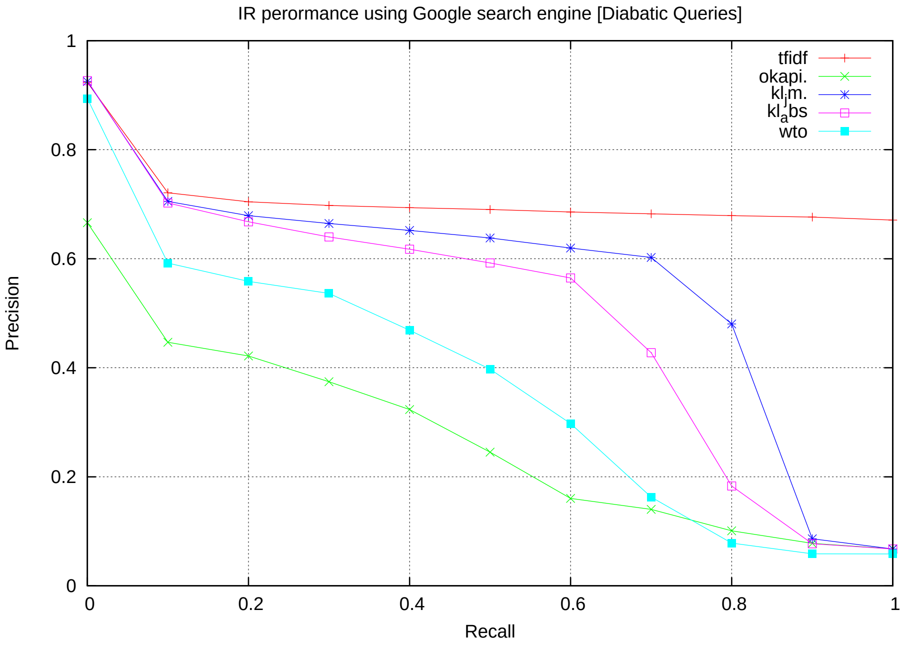


Figure 6.5: Precision and Recall of Specialized Domain Construction with GOOGLE Search Engines.

the most relevant document from the OHSUMED data set. Redundant documents were purged from the created domains. The final total of documents augmented into the *Eye* domain was 870 documents and 870 for the *Diabetes* domain.

6.5 Study Case II:Evaluation the proposed system performance in comparison to other search engines

A precision and recall performance study was conducted to compare the performance of the proposed system (SMAS) with that of existing major search engines such as Google, Yahoo, Bing, and NLM.

1. Two specialize domains related to medical data set (Diabetes and Healty Diet) were created using the SMAS;

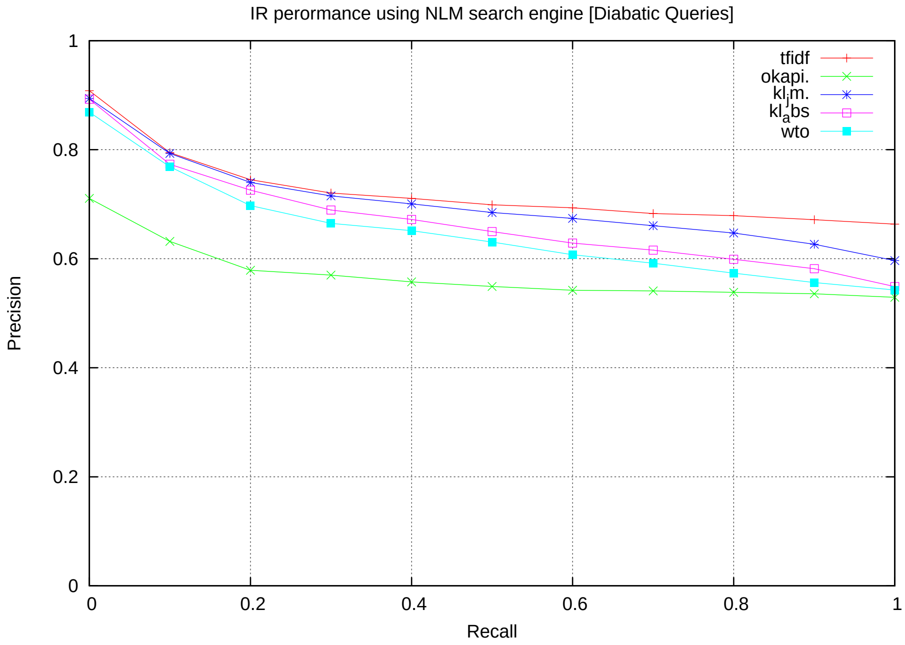


Figure 6.6: Precision and Recall of Specialized Domain Construction with NLM Search Engines.

2. TREC IR measuring techniques were used;

3. Five term matching techniques within the TREC measuring tool were utilized to compare the proposed system IR performance with that of the major search engines;

4. Lemur toolkit was embedded with SMAS for indexing, retrieving and searching;

5. Alta vista domain queries sets were equally used to evaluate each system, based on the same total number of queries, terms, sequences, and trials;

6. Searches were conducted automatically through regression testing with the objective of avoiding bias in the experiment;

7. TREC evaluation process steps were also conducted equally across each system to eliminate any bias

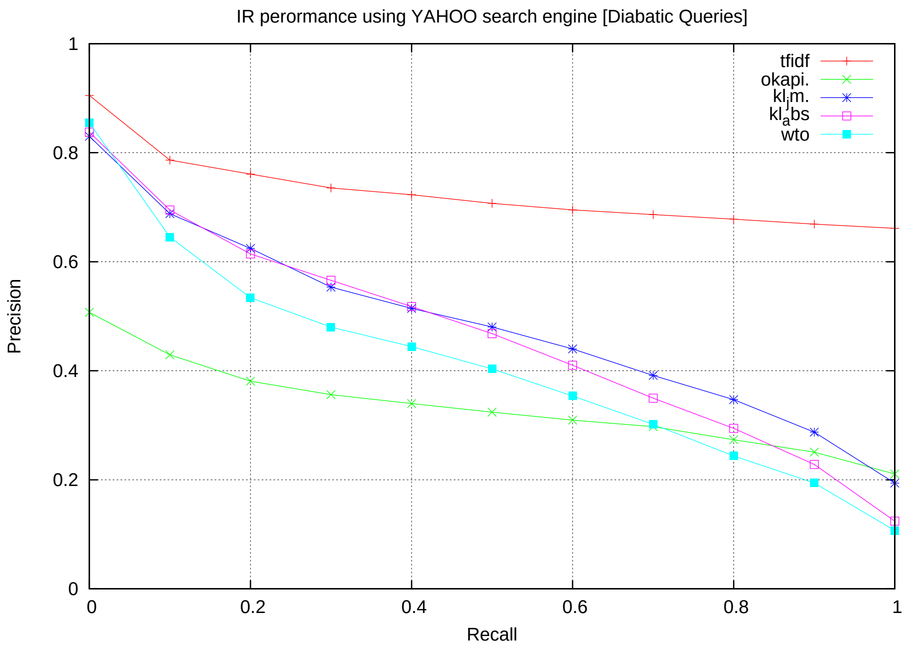


Figure 6.7: Precision and Recall of Specialized Domain Construction with YAHOO Search Engines.

This experiment measured the precision and recall (IR measuring standard) of information retrieved using Google, Yahoo, Bing, and Medical Search Engines in addition to the proposed system (SMAS). The Lemur Toolkit [81] and IR application toolkit were embedded into the proposed system (SMAS) as an intermediate to indexing and retrieving relevant information and mapping it to the end users. The specialized domain ontology $Diabetes\&HealthyDiet$ that was constructed in experiment [1] for building domain knowledge base and enriching uses' queries was adopted. Thirty five top pages were retrieved from each search engine using the tested approach; the duplicate and irrelevant pages were removed once from each search engine result. Two hundred and ten pages (35 pages per search engine) were used for the precision and recall experiment. After the first 15-25 web pages, the degrees of relevance to users and web pages becomes very slow. As shown in Fig.6.8, the precision and recall results of IR are higher than those of Google, Yahoo, Bing, and NLM. These higher results stem from the experimental approach and the NLM

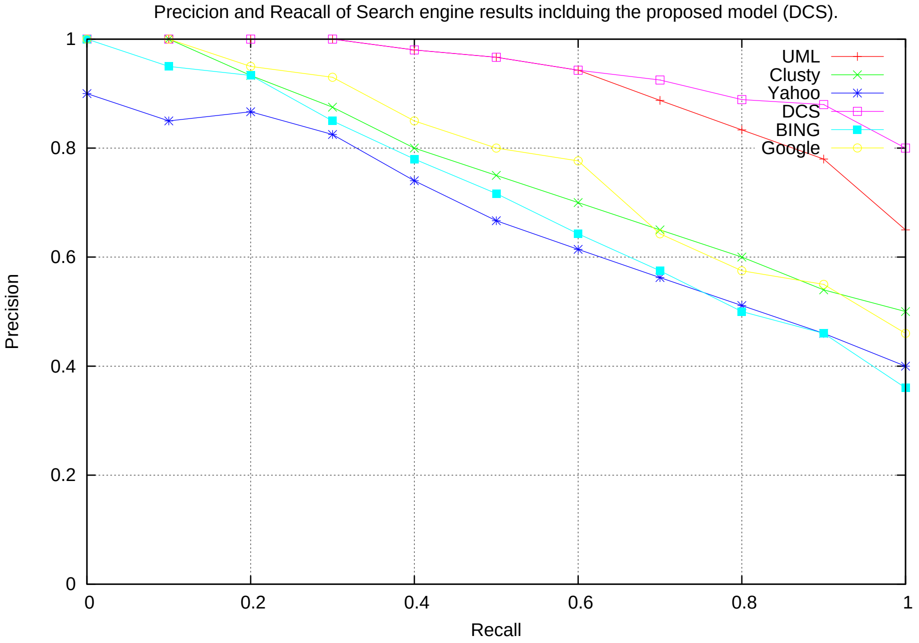


Figure 6.8: Precision and Recall of Specialized Domain Construction with Other Search Engines..

search engine because both domains are specialized in retrieving information related only to medicine about diabetes diets.

Fig.6.8 illustrates that the precision of mapping users to the relevant information can be achieved through enhancing the existing IR system and application by focusing on domain specialization. Specialized domain construction is the step of constructing a specialized knowledge domain to address the growing available information and the demand of such information.

The experiment shows that information that is categorized based on actual end users' needs and queries within a common domain has out-performed IR applications based on open-ended information. In spite of the robust search engines used as well, allocating a specialized agent for each user and user's query performs as a one-on-one search engine during the request. This process has a direct affect on the performance of SMAS over other methods. Furthermore, domains that are based on actual human needs such as queries, selection of relevance and the quality of the information has also improved the SMAS performance.

6.6 Case Study III: Semantic enrichment of search queries

In this case study, a set of TREC-87 queries were semantically and automatically enriched to evaluate the advantages of query enrichment with terms that are related to the end user specialized domains. Users are given the choice to select the system recommended query or to re-enter their queries. Extracted concepts from the uses exiting personalized and specialized domains have shown improvement of mapping end users needs to the desirable and relevant information. Though the TREC-87 query sets are complex and consist of large number of terms, semantic enrichment was able to perform better, as illustrated in Figure 6.9. This experiment used a set of TREC-query with Lemur tools to search for relevant information of TREC data sets. In this study, the data set used is only the actual TREC data, not the specialized domains.

The test criteria were:

1. Total number of queries used: 22 queries from 66 queries;

2. Total number of TREC documents used were 873 documents;

3. WordNet and LDA in addition to the Internet were used to semantically enrich the TREC queries;

4. TREC evaluation mechanism was used to measure precision and recall.

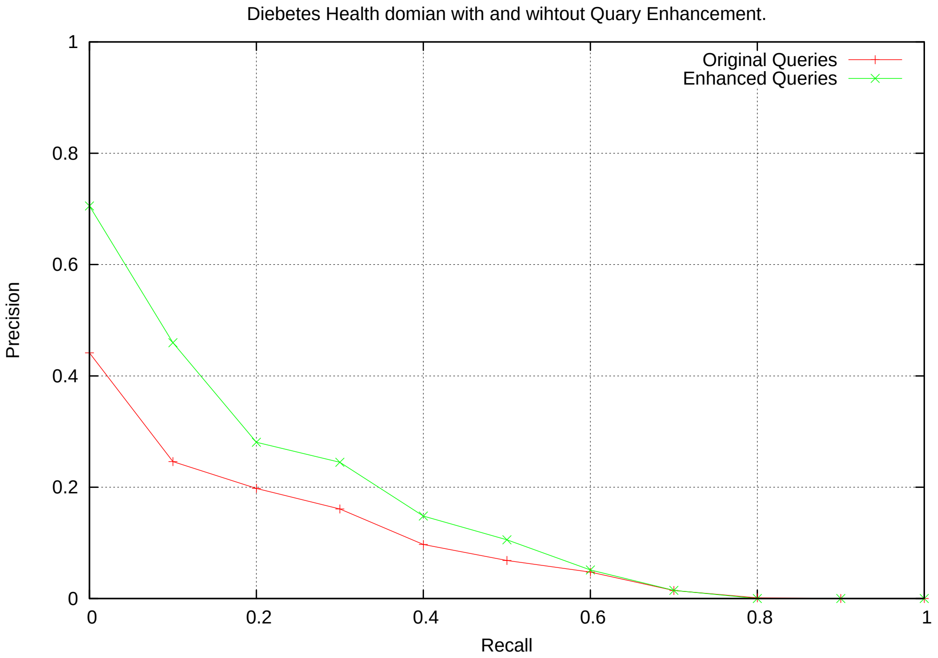


Figure 6.9: Query enrichment process for Diabetes Health domain.

6.6.1 Results Analysis

Although the system was able to enrich only a subset of the TREC queries, the set that was semantically enhanced yielded better precision and recall results in comparison to the original non enriched queries. Figure 6.10 illustrates the difference between queries that were enriched and queries that were not Due to the queries' complexity, only 22 queries of the original were able to be enriched.

6.6.2 Evaluating IR performance using Static Data

This study provides a comparable evaluation of our approach to construct specialized ontology domains using a static dataset. The precision and recall of the IR was evaluated using once standard TREC-9 (OHSUMED) [4, 129, 130] original data, and again, when the OHSUMED dataset was semantically enriched using the proposed approach of building a specialized domain ontology. In both cases, the queries were set to be interrelated

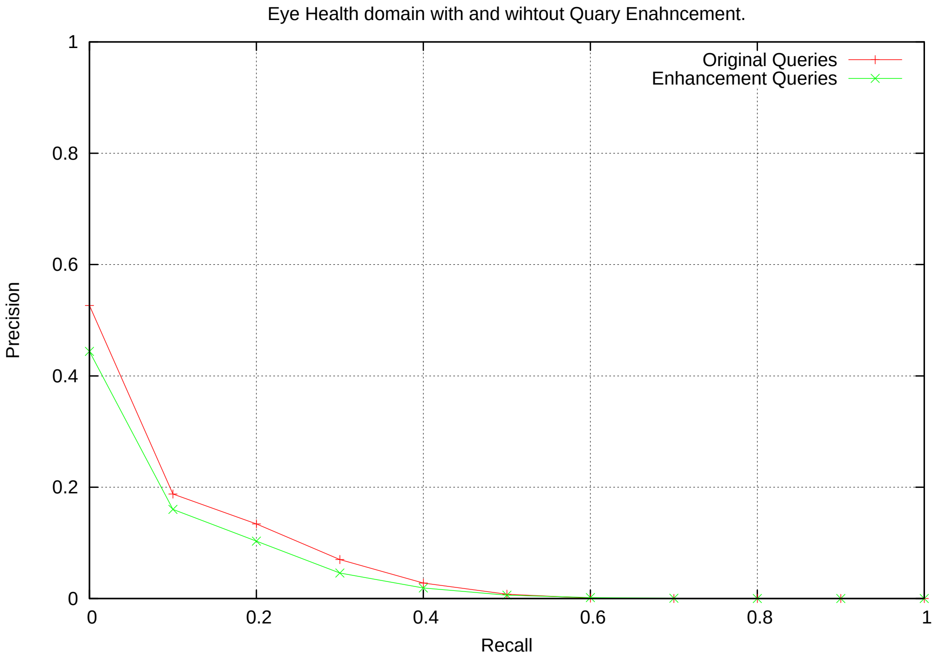


Figure 6.10: Query enrichment process for Diab domain.

Query Enrichment process for Eye Health domain.

to a particular and unique domain. Since the OHSUMED dataset consists of 348,566 OHSUMED documents collected [51] (medical information –titles and/ or abstracts– from MEDLINE) [51], queries related to one of the medical domains, in this case, the Diabetes domain, were selected. TREC-1987-1991 dataset was used along with topics, queries, and TREC evaluation techniques. This study adopted the evaluation metrics used in TREC-9 and integrated this approach with Lemur4.1 Toolkit [81] in order to index and retrieve documents from the OHSUMED dataset. The queries were extracted from the OHSUMED query set using the cosine similarity function –TFIDF algorithm– to be applied later for evaluating the ontology domains and retrieving related information. Terms and key words that related to the Diabetes domain were used to extract queries of the domain that was intended to the built. As a result, 44 queries were collected relating to the Diabetes domain information.

Using TREC evaluation techniques and Lemur Toolkit, the constructed ontology domains were evaluated in this study case using the following methods:

1. IR was evaluated using the original data set with unenriched queries, and the precision and recall of information retrieved was measured using typical OHSUMED 44 unenriched and unique queries with the OHSUMED original dataset [69]. As illustrated in Fig. 6.11, the precision and recall using unenriched queries with the OHSUMED data set were low.

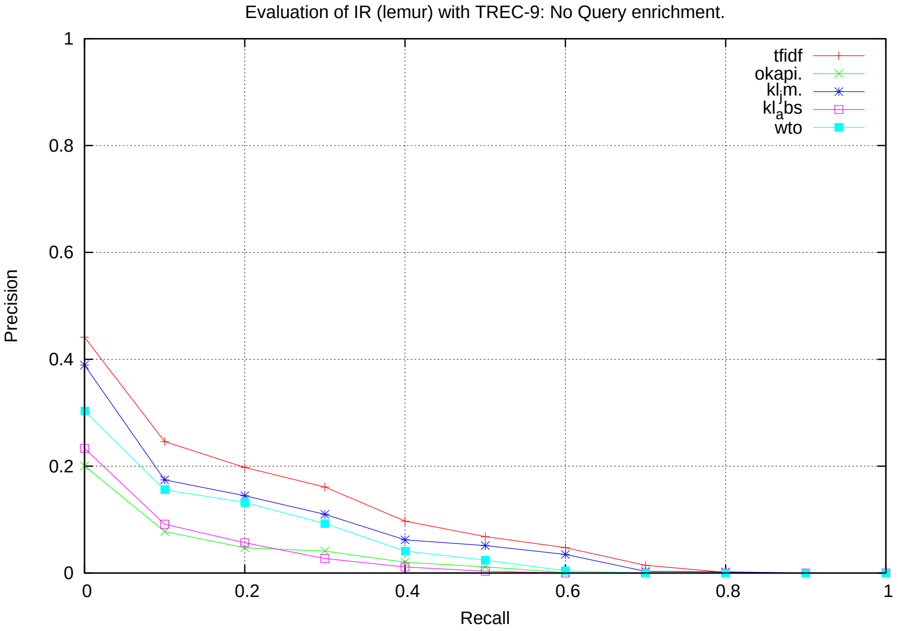


Figure 6.11: Precision and Recall of TREC-9(OHSUMED) dataset using unenriched queries.

2. IR was evaluated using semantically enriched queries with TEREC-9 dataset. The 44 queries were semantically enriched using WordNet only. The TREC-9 dataset, including 44 enriched queries of the Diabetes domain, was used to measure precision and the recall of information retrieved. As illustrated in Fig.6.12, the precision of using enriched queries with TREC-9 (OHSUMED) dataset improved slightly but the total number of related documents recalled gradually decreases.

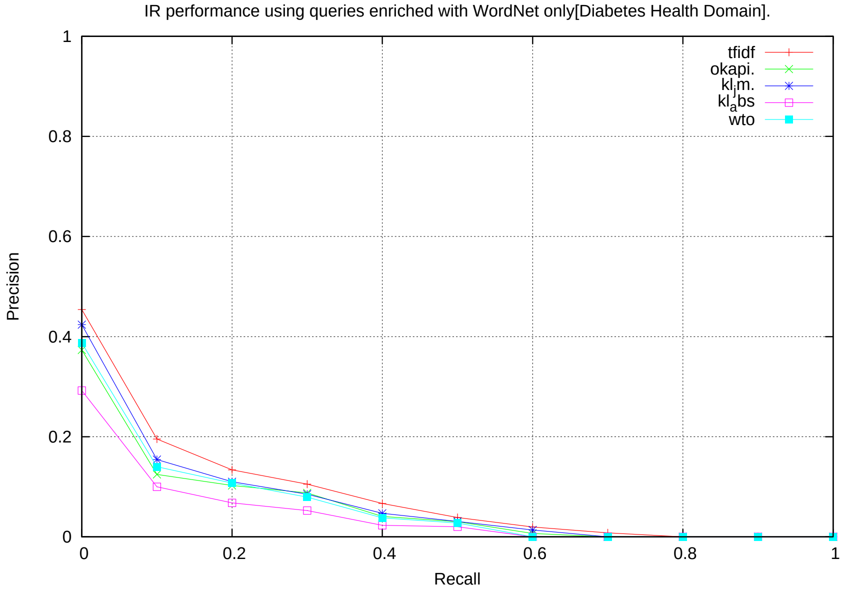


Figure 6.12: Precision and Recall of TREC-9 (OHSUMED) dataset using enriched queries with WordNet only.

3. IR was evaluated using specialized ontology domains. The ontology domain was created using the static data, in this case the Diabetes domain was chosen. In addition to constructing a specialized ontology domain, the query set used in the previous experiments was further enriched by external data sources. In this experiment, queries were enriched using our proposed approach which combines both WordNet with Topic model using LDA, in which the queries were enhanced with external information extracted from the Internet.

   From Fig.6.13, it is easy to observe that mixed query enrichment using the proposed approach with static dataset has improved the IR precision and recall comparing to the previous methods.

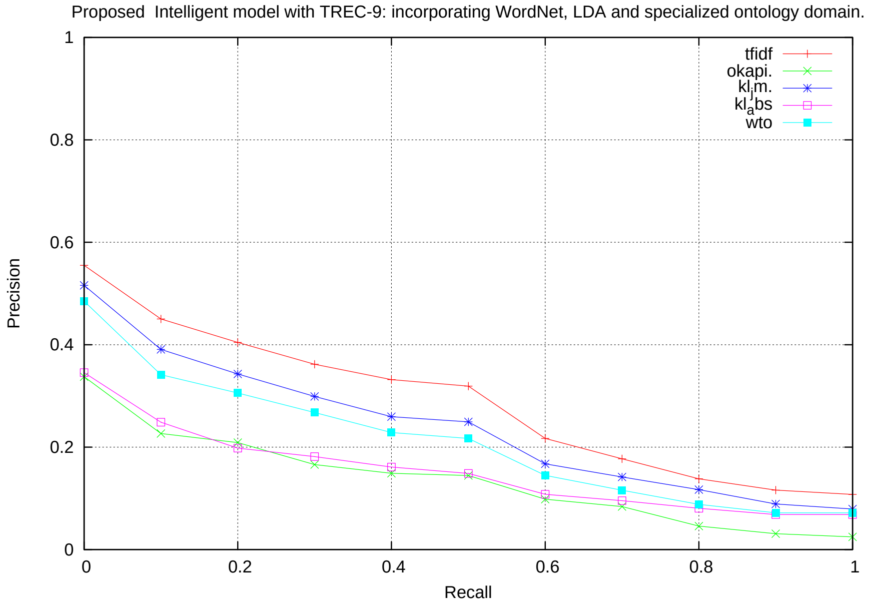


Figure 6.13: Precision and Recall of TREC-9 (OHSUMED) dataset using Intelligent domain model.

6.6.3 Evaluating IR performance using dynamic data

To further evaluate the proposed approach, this case study used dynamic data (i.e., the Internet) to construct specialized domains, and TREC to evaluate the precision and recall of IR. The evaluation of the IR precision and recall is based on building the specialized ontology domains on dynamic data, from the Internet. TREC-9 evaluation techniques such as tfidf, okapi, kljm, klabs, and twostage [4] were used for consistency, and the IR application (Lemur) as well as the knowledge domain, in this case Diabetes. The main difference in this case study from ”case study I” is that both the query and data set were constructed completely from the Internet. The specialized domain ontology and knowledge domain of the domain diabetes were constructed using the proposed intelligent domain ontology. The query set of this study was based on real time end user queries extracted from a general-purpose search engine, AltaVista [113]. The domain-related documents that are crawled from the Internet were augmented into the original OHSUMED data set where they were indexed and ranked based on these most relevant to the Diabetes synonymous terms. The data set consists of more than 80,000 documents; 44 unique related (Diabetes) queries were selected. Lemur Toolkit was used for indexing and retrieving the related information.

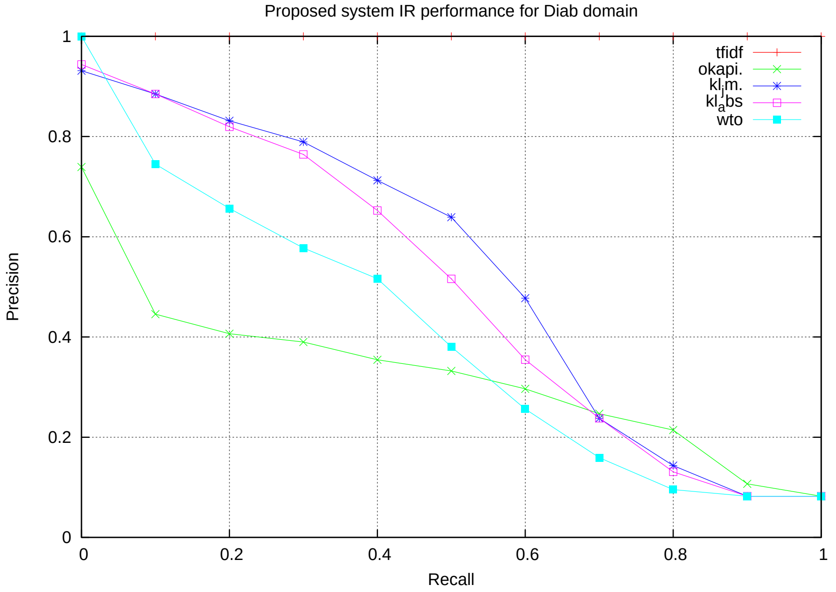


Figure 6.14: Precision and Recall of Dynamic dataset using Intelligent domain model and TREC-9 evaluation technique.

As illustrated in Fig.6.14, the precision of using dynamic data (Internet) to enrich queries and documents has significantly improved the precision of retrieval. Although these experiments have shown that query enrichment using ontology domains approach significantly improves the IR performance, there is still plenty of room for improvement if both the information and query are enriched as shown in the second study. This case demonstrates that the idea behind ontology-based information retrieval will increase the precision of retrieval results taking into account the semantic information contained in queries and documents, lifting keywords to ontological concepts and relations.

6.7 Conclusion

In order to map users to a relevant and desirable information, a multi-learning agents approach was adopted to construct specialized domains of knowledge that pertain to the user's needs.

The proposed system presents a novel approach for enhancing IR that pertain to end user needs and desire. This enhancement was achieved through the deployment of specialized multi-agents that learn and collaborate to guide the end user to find relevant information by constructing specialized knowledge domains based on end user' behaviours. Various experiments and case studies have demonstrated the advantages of building IR application based on end user' feedback, and knowledge bases segmented into specialized and unique domains. As the experimental results have shown, the proposed system introduces a novel idea that does not exist in typical search engines or information retrieval applications. Several experiments and case studies were conducted to validate the effectiveness of the proposed system: (1) specialized knowledge domain construction, (2) specialized domain ontologies construction, (3) intelligent learning and collaboration among specialized agents, and (4) ability to adopt the proposed approach with other IR applications and search engines. Furthermore, the experiments have shown that there is always room for improvement to enhance IR by mapping the end user to the relevant information. In addition to IR engine, the enhancement can be applied the end user side by enhancing his/her query semantically, giving the end users extra options to enhance their queries. The retrieval precision of the proposed system was evaluated, and the specialized domains that were constructed based on previous experiences and collaboration among agents achieved the best precision. This approach allows specialized intelligent agents to construct a more appropriate knowledge base domain and perform a search that is better adapted to each user's needs.

Chapter 7

Conclusions and Further Research

In conclusion of the thesis, this chapter provides an overview of the work performed herein, highlighting its contributions and findings, and discussing potential future extensions. Section 8.1 points out the novelties of the research direction, and briefly presents the approach taken towards an enhanced information retrieval process that adopts machine learning techniques - specifically the multi-agent system approach. Section 8.2 cites experimental results and refers to conclusions drawn from these results, justifying the advantages of a specialized knowledge domain approach. Section 8.3 summarizes contributions of the work to advance the next generation of research in the area of information retrieval by introducing and developing a framework based on specialized knowledge domains. Finally, section 8.4 concludes the thesis by detailing directions for further research, which stem directly from this work.

7.1 The Proposed Approach

The premise of this thesis was to construct specialized knowledge domains that, based upon user feedback, provide greater precision when retrieving relevant information. For existing IR applications, mapping the relevant, retrieved information to what the user actually desires, in an intelligent mode, is an ongoing and complex challenge. Furthermore, handling large numbers of users, and organizing information into related domains, would require an efficient and accurate distributed system. To help make this possible, a multi-agent approach, one of the machine-learning techniques, was adopted to construct specialized knowledge domains through intelligent learning of the user's behaviour.

By specializing in knowledge domains, and by employing intelligent learning agents that collaborate amongst each other and interact with the end user, this approach presents a novel solution. It maps users to the desired information, and segments the ever-increasing information available via the Internet by utilizing the specialized knowledge domain obtained through user feedback. Each agent is specialized in one knowledge domain, constructed by learning agents that are triggered for each user by the specialized agent. Adopting machine-learning through a multi-agent approach requires collaboration among agents in two ways: sharing the challenging tasks; and preventing redundancies and duplications. User interaction and feedback were quantified through an intelligent learning agent, RL, which reflects what document a user selects among the set of presented documents.

During the initial domain construction, documents were presented to the user based mainly on the frequency of the terms within the document with respect to the user's query. Through learning, documents selected by the users are augmented into the knowledge base of a specific domain and become part of that knowledge domain. As the learned knowledge base of a domain grows, it becomes the primary resource of knowledge about a specific domain. Consequently, hyper-learning is derived by utilizing past learning experiences, because the knowledge base incorporates documents added by users throughout the learning process. Nevertheless, for each search query, the retrieved information is based first on of a learned knowledge domain reinforced with an external information (i.e., the Internet).

This thesis has further enhanced the IR process by proposing a new technique to enhance the user queries and build domain topics as presented in Chapter 4. In addition to how to construct the knowledge domain, the system needs to know which knowledge domain should be augmented. As the knowledge domains grow, delegating the query through the delegation agents would require time to map the user's query into the most relevant domains. Such a process requires the delegation agent to search through the knowledge domains with respect to the user's query. This challenge is addressed in the thesis by illustrating that the constructing of knowledge domain topics (keywords) for each knowledge domain as a critical means to efficiently map the user's query to the correct knowledge domain. Furthermore, this domain topic process would also be applied to refine and enrich the user's query as well as the retrieving process by double-indexing the domain knowledge. Research results have shown that domain topics have enhanced the IR process throughout the experiments conducted.

The research work focused on two areas:

1. theoretical and modeling; and
2. experimental and testing.

This thesis has dealt mainly with the following challenges in information retrieval:

- adopting machine-learning techniques alongside efficiency to learn the user's behavior and map users to the most relevant information;
- constructing specialized knowledge domains;
- making an accurate representation of the knowledge domain as well as the user query through domain topics;
- mapping users to the most relevant information that matches their needs.
- evaluating information relevant beyond the conventional IR techniques, such as term frequency, co-occurrence, or through links, and not through user profiling and personalization techniques.

This thesis addressed these challenges through building a specialized multi-agent learning model to construct specialized knowledge domains and topics through end users feedback, providing an instantiation of the context hypothesis, towards which this thesis provided some evidence.

7.2 Findings

This research portrays specialized knowledge domains for IR systems to address the ever-growing information available for end users. It proposes a framework for mapping end users to the most relevant information by constructing specialized knowledge domains through intelligent learning agents, instead of simply retrieving relevant information, as is the case in most current IR systems and search engines. In the specialized knowledge domains approach, relevant information is augmented into the knowledge base, based on end user feedback. An intelligent learning agent technique is employed to learn the end user's behavior towards selected information.

The specialized domains constructed were based on previous and collaborative experiences among agents which achieved the best precision. Hyper-learning, a process in which an agent shares its learning policy with another agent searching for the same or similar information, as described in Experiment 3, yielded efficiency in respect to searching, retrieving and learning. Reinforcement learning has been proven in this thesis to be the right candidate to generate a real-time and online learning algorithm for such a task. Experimental results have shown that the proposed system demonstrated efficiency and higher precision in comparison to the conventional search engines. A case study constructing a specialized knowledge domain information retrieval task (i.e., embedding the proposed system with an existing IR application) was considered. The study shows how the proposed approach is workable, and how it can be applied to information retrieval systems and search engines. Experimentally, significant improvements in all precision retrieval are achieved for the information retrieval process. Moreover, retrieval precision tends to improve as more learning processes are performed and new information is augmented into the knowledge base domains.

The third important part of the system is constructing specialized ontology topics for each specialized domain, which has further enhanced the information retrieval process. Specialized ontology topics of existing domains affect the performance of IR by mapping the user's query to the more relevant existing knowledge domains instead of searching for all existing domains. The most accurate mapping can be achieved when each specialized domain is represented semantically to be ontology keywords. Consequently, the quality of information retrieval is improved when specialization of domain is constructed, and when users are included to construct those knowledge domains. This is considered to be approach that surpasses traditional search engine and IR techniques, i.e., the vector space model.

Finally, not all domains can be constructed manually, or constructed based on dictionary or Internet data. As a result, semantic domain topic extraction method was proposed in this thesis. It consists of three stages to construct a new, specialized domain: semantic lexical information, related Internet information that is evaluated by the end user through intelligent learning, and tagging the result of the previous two steps with terms defined by the users.

7.3 Contributions

This thesis is the introduced the novel idea of enhancing IR through the construction of specialized knowledge domains. By using the most efficient and effective way to map users to the most relevant information, new findings and facts were discovered, including adapting multi-intelligent agent systems to specialized domains, as well as assisting and learning the users' behaviours, and proof of effectiveness of the approaches on real world application.

The main contributions of this thesis lie in the following:

- A novel technique to segment, in real time large amounts of information into domain specifics, based upon human involvement through intelligent learning, instead of techniques similar to those which occur in other conventional method such as, categorization, clustering, classification, and rank-based. Humans play an essential role in building the knowledge domain that matches their needs.
- The introduction of an intelligent agent to serve as a specialized agent. A machine-learning intelligent agent can fall into any one of three categories of common intelligent agent types: autonomous, adjustable, or mixed initiative agents. All three, however, share the same intelligent characteristic which is reasoning abut a task. This

thesis presented a novel aspect to this agent, where any one of these three common intelligent agents may serve as the specialized agent.

- A mechanism to construct specialized domain knowledge and topics.
- A multi-agent learning system to assist and guide users to select relevant information, where the RL algorithm can be enhanced to make both a hyper-learning agent and a directed learning agent. Hyper-Learning agents are created through sharing past experiences (i.e., sharing initial learning policy), based on past rewards. In the IR process, documents are ranked according to three features: the frequency of terms occurring within the document with respect to the query; a total rewards weighting assigned when the end users select it, and the total number of times the document is used by the same specialized learning agent domain or by other specialized agents.
- A collaboration technique for multi-agent systems. Collaboration among agents is a challenged for multi-agent systems. This research utilized two techniques of collaboration among agents: a bulletin board to share knowledge, and agent-learned policies.
- The establishment of novel techniques to build domain keywords (i.e., metadata). By combining the intelligent agent, semantic (statistical) ontology, and social tagging [146, 15, 118], the keywords of a document or a set of documents (i.e., Website) can be constructed automatically.
- User query enrichment, a technique which enriches the user's query by accurately and efficiently mapping the user to the relevant information, while adopting social tagging to refine the final user queries and validate the system recommendations.

7.4 Future Extensions

In the future, real users are expected to form various specialized knowledge bases, and the expected information access will be improved with the organization of the information into a greater number of different domains.

In this thesis, the information format has focused mainly on text data, such as Web pages. In the future however, specialized knowledge domains based on information content other than text data, such as images, voices, and videos) is another interesting point to explore. Furthermore, text data itself became revolutionized through the growing interest in social media. Therefore, the proposed approach can be extended to construct specialized domain knowledge for such information.

There are many ways to extend this work in its framework layers, beginning with data representation, user query representation, constructing domain topics, and learning agents, and ending with applying and implementing constructing specialized knowledge domain processes that are effective and efficient in finding relevant information. In all of these areas, contributions could be furthered and refined. The areas that lend themselves most to future research include:

- Upgrading the proposed system to be portable and stand-alone application.
- Applying different formats of information, such as audio, movies, social media data, etc.
- Extending the system to address the specialization of knowledge domains for other areas of machine-learning, such as data mining, information extraction, and knowledge discovery.
- Extending the system to address the structure and semi-structure information of documents.
- Extending the system to applications other than documents, considering the specialized multi-agent framework for games, robotics, vehicles, biotechnologies applications, and mobile phones.

In conclusion, with the continuous growth of information available to end users, it is critical to construct specialized knowledge domains based on the user's interest. Further research to achieve this purpose should continue to be explored.

References

[1] A., B.-Y. R., and Berthier, R.-N. *Modern Information Retrieval.* Addison-Wesley Longman Publishing Co., Inc., 1999.

[2] A., M., O., B., and A., Y. An intelligent model to construct specialized domain ontologies. *3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), 2010 5* (2010), 696–702.

[3] Aalbersberg, I. J. Incremental relevance feedback.

[4] Allan, M. C., and Croft, B. Inquery and trec-9. 551–563.

[5] Ardissono, L., Goy, A., Petrone, G., Segnan, M., Console, L., Lesmo, L., Simone, C., and Torasso, P. Agent technologies for the development of adaptive web stores. *In Agent Mediated Electronic Commerce, The European AgentLink Perspective* (2001), 194–213.

[6] Bab, A., and Brafman, R. I. Multi-agent reinforcement learning in common interest and fixed sum stochastic games: An experimental study. *Journal of Machine Learning Research 9* (2008), 2635–2675.

[7] Banerjee, B., Sen, S., and Peng, J. Fast concurrent reinforcement learners. *IJCAI 2001* (2001), 825–832.

[8] Barrett, R., Maglio, P. P., and Kellem, D. C. How to personalize the web. 75–82.

[9] Barto, A. G., and Mahadevan, S. Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems 13(4)* (2003), 341 – 379.

[10] Bhide, M., Deolasee, P., Katkar, A., Panchbudhe, A., Ramamritham, K., and Shenoy, P. Adaptive push-pull: Disseminating dynamic web data. 652–668.

[11] Blei, D., Ng, A., and Jordan, M. Latent dirichlet allocation. *Journal of Machine Learning Research 3* (2003), 993–1022.

[12] Boissier, C. C. O. Multi-agent platforms on smart devices: Dream or reality? *Ecole Nationale Supï¿rieure des Mines de Saint Etienne, proceedings SOC* (2003).

[13] Bond, A. H., and Gasser, L. Readings in distributed artificial intelligence.

[14] Bradshaw, J. M. Introduction to software agents. *Software Agents, Jeffery M. Bradshaw* (1997), 4–66.

[15] Cattuto, C., Benz, D., Hotho, A., and Stumme, G. Semantic analysis of tag similarity measures in collaborative tagging systems. *CoRR abs/0805.2045* (2008).

[16] Cavnar, W. B., and Trenkle, J. M. N-gram-based text categorization. 161–175.

[17] Cesta, A., and 'Aloisi, D. D. Agent-based meeting scheduling with users in the loop. *Working Notes of the IJCAI Workshop on Adjustable Autonomy Systems* (August 1998), 7–15.

[18] Chalupsky, H., G. Y.-K. C. L.-K. O. J. P. D. R. T. e. M. Electric elves: Applying agent technology to support human organizations. *Proceedings of Innovative Applications of Artificial Intelligence Conference* (2001).

[19] C.Hartrum, T., and A.DeLoach, S. Design issues for mixed-initiative agent systems.

[20] Chuang, S.-L., and Chien, L.-F. Enriching web taxonomies through subject categorization of query terms from search engine logs. *Decis. Support Syst., Elsevier Science Publishers B. V. 35* (2003).

[21] Chung, W. Web searching in a multilingual world. *Commun. ACM 51* (May 2008), 32–40.

[22] Clara-Ines Pena, Jose-L Marzo, J.-L. d. l. R. *ICALT2002* (2002).

[23] Cohen, P.R., L. H. R., and Smith, I. On team formation. *Contemporary Action Theory. Synthesis* (1997).

[24] Cosmin Carabelea, O. B., and Florea, A. Autonomy in multi-agent systems: A classification attempt. *Agents and Computational Autonomy* (2003), 103–113.

[25] Craswell, N., Fetterly, D., Najork, M., Robertson, S., and Yilmaz, E. Research at trec 2009: Web and relevance feedback tracks. 500–278.

[26] Cucchiarelli, A., and Velardi, P. Finding a domain-appropriate sense inventory for semantically tagging a corpus. *Natural Language Engineering* (1999), 325–344.

[27] Dayan, P., and J.Sejnowski, T. Td (λ) converges with probability 1. *Machine Learning 14* (1994), 295–301.

[28] Dayan, P., and Watkins. Reinforcement learning. *CJCH: Encyclopedia of Cognitive Science* (2001).

[29] Deters, R. Scalability and multi-agent systems.

[30] Ding, Y. Ir and ai: The role of ontology. In *Proceedings of 4th International Conference of Asian Digital Libraries* (2001).

[31] Dixon, M. An overview of document mining technology.

[32] Dolamic, L., and Savoy, J. Comparative study of indexing and search strategies for the hindi, marathi, and bengali languages. 11:1–11:24.

[33] Endrullis, S., Thor, A., and Rahm, E. Tevaluation of query generators for entity search engines. *Proc. of Intl, Workshop on Using Search Engine Technology for Information Management (USETIM)* (2009-08).

[34] ET. AL, O. Z. Web document clustering: A feasibility demonstration.

[35] FELLBAUM, C. Wordnet: An electronic lexical database. *MIT Press* (1998).

[36] FININ, T., LABROU, Y., AND MAYFIELD, J. Kqml as an agent communication language in: Software agents. 291–316.

[37] FININ, T., LABROUR, Y., AND MAYFIELD, J. *KQLM An Agent Communication Language*. Jeffery M.Bradshaw, 1997.

[38] FININ, T., R.FRITZSON, D.MCKAY, AND MCENTIRE, R. Kqml as an agent communication language. *CIKM-94: The proceedings of the third international conference on information and knowledge management (CIKM 94)* (November 1994).

[39] FLORES-MENDEZ, R. A. Towards a standardization of multi-agent system frameworks. *An article by Roberto A. Flores-Mendez in the ACM Crossroads magazine' s special issue on agents* (2000), 28.

[40] FRANKLIN, S., AND GRAESSER, A. Is it an agent, or just a program : A taxonomy for autonomous agents.

[41] FUNG, G. Comprehensive overview of basic clustering algorithms.

[42] GAUCH, D. S. H. . S. Intelligent information agents: Review and challenges for distributed information sources. *Journal of the American Society for Information Science 49(4)* (April 1998), 304–311.

[43] GREENGRASS, E. Information retrieval : A survey by ed greengrass. *Information Retrieval 163*, November (2000), 141–163.

[44] GREENGRASS, E. Information retrieval: A survey. *DOD Technical Report TRR52-008-001* (2001).

[45] GREENWALD, A. Correlated-q learning. *ICML- 2003, Proceeding of the Twentieth International Conference on Machine Learning* (2003).

[46] Griffiths, T., Steyvers, M., and Tenenbaum, J. Topics in semantic representation. *Psychological Review 114* (2007), 211–2.

[47] Guarino, N., Masolo, C., and Vetere, G. Ontoseek: Content-based access to the web. *IEEE Intelligent Systems 14(3)* (1999), 70–80.

[48] Haddad, M. R., and Baazaoui, H. A user-oriented accessibility measure for spatial web personalization. *Exaleadcom* (2009).

[49] Harman, M. E., and Harman, S. S. Reinforcement learning: A tutorial.

[50] Hernandez, F., Gaudioso, E., and Boticario, J. G. A multiagent approach to obtain open and flexible user models in adaptive learning communities. *User Modeling* (2003), 203–207.

[51] Hersh WR, H. D. Use of a multi-application computer workstation in a clinical setting. 382–389.

[52] hua Yeh, J., and Yang, N. Ontology construction based on latent topic extraction in a digital library. *Book Series: Lecture Notes in Computer Science, Springer Berlin / Heidelberg 5362* (2008), 93–103.

[53] Huang, L. A survey on web information retrieval technologies. *http://citeseer.nj.nec.com/559043.html*.

[54] Ingwersen, P. Information retrieval interaction.

[55] Ishida, T. Real-timesearch for learning autonomous agents.

[56] J., R. S., and Norvig, P. Artificial intelligence:a modern approach.

[57] Jansen, B. J., Spink, A., Pederson, J., and Saracevic, T. Searchers, the subjects they search, and sufficiency: a study of a large sample of excite searches. *Proceedings of WebNet 98 Conference* (1999).

[58] Jasen, B. J., and Spink, A. The excite research project: a study of searching characteristics by web users. *Invited paper - Bulletin of the American Society for Information Science*.

[59] Jasen, B. J., S. A., and Pederson, J. A temporal comparison of altavista web searching. *Journal of the American Society for Information Science and Technology 56(6)* (2005), 559–570.

[60] Jian-Yun, J.-R. W., and Zhang, H.-J. Query clustering using logs. *ACM TOIS: ACM Transactions on Information Systems 20(1)* (January, 2002), 59–81.

[61] Joshi, K. P. Analysis of data mining algorithms.

[62] Kaelbling, L., Littman, M., and Moore, A. Reinforcement learning: A survey. journal of artificial intelligence research 4. 237–285.

[63] Karen, and Jones, S. Information retrieval and artificial intelligence. *Artificial Intelligence 114*, 1-2 (1999), 257 – 281.

[64] Kargupta, H., Hamxaoglu, I., and Stafford, B. Scalable, distributed data mining using an agent based architecture. *Processing of High Performance Computing 97 Knowledge Discovery and Data Mining 97* (1997).

[65] Karypis, G. Cluto a clustering toolkit. *Technical Report, Dept. of Computer Science, University of Minnesota* (2002).

[66] Kay, J., etzl, J., Rao, G., and Thies, J. The atl postmaster: A system for agent collaboration and information dissemination. 338 – 342.

[67] Knoblock, C. A., and Ambite, L. Agent for information gathering. *Software Agents, Jeffery M. Bradshaw* (1997), 348–373.

[68] Laurel, B. *Interface Agents: Metaphor with Character.* 1997.

[69] Lee, C. T., Vinay, V., Mendes Rodrigues, E., and Kazai. Measuring system performance and topic discernment using generalized adaptive-weight mean. 2033–2036.

[70] Lin, K.-I., and Kondadadi, R. A similarity-based soft clustering algorithm for documents.

[71] Lind, J. Pattern in agent-oriented software engineering. *AOSE* (2002), 47–58.

[72] Magnini, B., and Speranza, M. Merging global and specialized linguistic ontologies. 43–48.

[73] Melucci, M. A basis for information retrieval in context. *ACM Trans. Inf. Syst. 26* (June 2008), 14:1–14:41.

[74] Metral, M., Maes, P., and Lashkari, Y. Collaborative interface agents.

[75] Miller, G. A. Wordnet: A lexical database for english. *Princeton University, Princeton NJ* (1993).

[76] Mobasher, B., Dai, H., Luo, T., and Nakagawa, M. Discovery and evaluation of aggregate usage profiles for web personalization. *Data Mining and Knowledge Discovery 6, No. 1* (January 2002), 61–82.

[77] Mooman, A., Basir, O., and Younes, A. Architecture design: Building specialized domain ontologies using an intelligent model and topic mode. *In The Eleventh IASTED International Conference on Artificial Intelligence and Applications (AIA 2011)* (Februray 2011).

[78] Mooman, A., Basir, O., and Younes, A. An intelligent model to construct specialized domain ontologies. *I4th IEEE International Conference on Computer Science and Information Technology* (June 2011).

[79] Moukas, A., and Maes, P. Amalthaea: An evolving multi-agent information filtering and discovery system for the www. *Autonomous Agents and Multi-agent Systems*, 1(1) (1998), 59–88.

[80] Muller, H. Negotiation principles. *in G.M.P.O Hare and N.R.Jennings, eds, Foundations of Distributed Artificial Intelligence* (1996), 211 – 229.

[81] of Massachusetts, U., (CIIR), A., and the Language Technologies Institute (LTI) at Carnegie Mellon University. *Lemur Project.* Center for

Intelligent Information Retrieval (CIIR) at the University of Massachusetts, Amherst, and the Language Technologies Institute (LTI) at Carnegie Mellon University.

[82] Piatetsky-Shapiro, G., and Fayyad, U. M., and Smyth, P. From data mining to knowledge discovery: An overview. *AAAI: In U.M. Fayyad, et al. (eds.), Advances in Knowledge Discovery and Data Mining* (1996), 1–35.

[83] Rama, G. M., Sarkar, S., and Heafield, K. Mining business topics in source code using latent dirichlet allocation. *ISEC 2008* (2008), 113–120.

[84] Ray, E., Seltzer, R., and Ray, D. *The AltaVista Search Revolution.* Osborne/McGraw-Hill, New York, 1997.

[85] Reiter, N., and Buitelaar, P. Lexical enrichment of a human anatomy ontology using wordnet. *In Proceedings of the 4th Global WordNet Conference, Szeged* (2008).

[86] Rennie, J., and McCallum, A. K. Using reinforcement learning to spider the web efficiently. *ICML-99: Workshop, Machine Learning in Text Data Analysis* (1999).

[87] Rijsbergen, C. J. V. Information retrieval.

[88] Ross, N. C. M., and Wolfram, D. End user searching on the internet: an analysis of term pair topics submitted to the excite search engine. *Journal of the American Society for Information Science* (June 2000).

[89] Roussinov, D., K. T. e. a. Visualizing internet search results with adaptive self-organizing maps. *SIGIR 99: ACM SIGIR Conference on Research and Development in Information Retrieval* (1999).

[90] Salton, G. A simple blueprint for automatic boolean query processing. *Inf. Process. Manage. 24* (May 1988), 269–280.

[91] Salton, G. *Automatic text processing: the transformation, analysis, and retrieval of information by computer.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[92] Salton, G., Allan, J., and Buckley, C. Approaches to passage retrieval in full text information systems. 49–58.

[93] Salton, G., and Buckley, C. Term weighting approaches in automatic text retrieval. Tech. rep., Ithaca, NY, USA, 1987.

[94] Salton, G., and Buckley, C. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science 41* (1990), 288–297.

[95] Salton, G., and Buckley, C. Improving retrieval performance by relevance feedback. *Readings in information retrieval* (1997), 355–364.

[96] Salton, G., Fox, E. A., and Wu, H. Extended boolean information retrieval. Tech. rep., Ithaca, NY, USA, 1982.

[97] Salton, G., and Lesk, M. E. The smart automatic document retrieval systems¿an illustration. *Commun. ACM 8* (June 1965), 391–398.

[98] Salton, G., and McGill, M. J. *Introduction to Modern Information Retrieval.* McGraw Hill, Inc., New York, NY, USA, 1986.

[99] Salton, G., and McGill, M. J. *Introduction to Modern Information Retrieval.* McGraw-Hill, Inc., New York, NY, USA, 1986.

[100] Sanderson, M. Word sense disambiguation and information retrieval. 142–151.

[101] Saracevic, T. Evaluation of evaluation in information retrieval. 137–146.

[102] Scerri, P. Designing agent for systems with adjustable autonomy. *PhD Dissertation, Department of Computer and Information Science* (December, 2001).

[103] Scerri, P., Pynadath, D., and Tambe, M. Adjustable autonomy in real-world multi-agent environment. *JAIR i17: J.Artif.Intell.Res.* (2002), 171–228.

[104] Schaerf, A., Shoham, Y., and Tennenholtz, M. Adaptive load balancing: A study in multi-agent learning. *Journal of Artificial Intelligence Research 2* (1995), 475–500.

[105] SChiaffino, S., and Amandi, A. User-interface agent interaction: personalization issues. *International Journal of Human - Computer Studies 60* (2004), 129–148.

[106] Selberg, E. W. Information retrieval advances using relevance feedback.

[107] Seltzer, R., Ray, D. S., and Ray, E. J. *The AltaVista Revolution: How to Find Anything on the Internet*. Osborne/McGraw-Hill, Berkeley, CA, USA, 1996.

[108] Seo, Y. W., and Zhang, B. T. A reinforcement learning agent for personalized information filtering. *IUI-2000: In Proceedings of International Conference on Intelligent User Interface* (2000), 248–251.

[109] Shafiei, M. M., and Milios, E. E. Latent dirichlet co-clustering. 542–551.

[110] Shafiei, M. M., and Milios, E. E. Latent dirichlet co-clustering. 542–551.

[111] Shen, D., Pan, R., Sun, J.-T., Pan, J. J., Wu, K., Yin, J., and Yang, Q. Query enrichment for web-query classification. *ACM Trans. Inf. Syst. 24* (2006), 320–352.

[112] Shoham, Y., and Powers, R. Multi-agent reinforcement learning:a critical survey. *AAAI Fall Symposium on Artificial Multi-Agent Learning* (2004).

[113] Silverstein, C., Henzinger, M., Marais, H., and Moricz, M. Analysis of a very large altavista query log.

[114] Singh, S., Jaakkola, T., L.Littman, M., and Vari, C. S. Convergence results for single - step on - policy reinforcement - learning algorithm. *Kluwer Academic Publishers, Machine Learning 38 Issue 3* (March 2000).

[115] Singhal, A. Modern information retrieval: A brief overview. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering 24*, 4 (November 2001), 35–42.

[116] S.K., P., V., T., and P., M. Web mining in soft computing framework: Relevance, state of the art and future directions. *IEEE Transactions on Neural Networks 13, Issue: 5* (2002), 1163–1177.

[117] Skinner, J. M. Multi-agent systems and mixed-initiative intelligence.

[118] Smith, G., T. People-powered metadata for the social web (voices that matter). *New Riders Press* (2007).

[119] Sparck Jones, K. Information retrieval and artificial intelligence. *Artifical Intelligence 114* (1999), 257–281.

[120] Spink, A., B. J., and Jansen, B. J. Searching the web: survey of excite users. *Internet Research: Electronic Networking Applications and Policy 9(2)* (1999), 117–128.

[121] Sridharan, M., and Tesauro, G. Multi-agent q-learning and regression trees for automated pricing decisions. *In Proc. 17th International Conf. on Machine Learning* (2000), 927–934.

[122] Stone, P., and Veloso, M. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots 8(3)* (July 2000), 345–383.

[123] Sutton, R., and Barto, A. Reinforcement learning. *MIT Press* (1998).

[124] Thomas, C. Basar: A framework for integrating agents in the worldwide web. *IEEE Computer 28(5)* (1995), 84–86.

[125] Tian, K., Revelle, M., and Poshyvanyk, D. Using latent dirichlet allocation for automatic categorization of software. *MSR '09: Proceedings of the 2009 6th IEEE International Working Conference on Mining Software Repositories, IEEE Computer Society* (2009).

[126] Tierney, B. L., Lee, J., Crowley, B., and Holding, M. A network-aware distributed storage cache for data intensive environments. *In Proceedings of the Eighth IEEE International Symposium on High Performance Distributed Computing* (August 1999), 185–193.

[127] Vakkari, P., and Taneli, M. Comparing google to ask-a-librarian service for answering factual and topical questions. In *Proceedings of the 13th European conference on Research and advanced technology for digital libraries* (Berlin, Heidelberg, 2009), ECDL'09, Springer-Verlag, pp. 352–363.

[128] Voorhees, E. M. Natural language processing and information retrieval. 32–48.

[129] Voorhees, E. M., and Harman. *TREC Experiment and Evaluation in Information Retrieval.* MIT Press, 2008.

[130] Voorhees, E. M., and Harman, D. K. *TREC: Experiment and Evaluation in Information Retrieval.* MIT Press, 2005.

[131] Wang, L., Lin, J., and Metzler, D. A cascade ranking model for efficient ranked retrieval. 105–114.

[132] Watkins, C., and Dayan, P. Q-learning. *Machine Learning 8* (1992), 279– 292.

[133] Wegrzyn-Wolska, K. Statistical classification of search engines interrogated by the meta-search system. 317–322.

[134] Wei, X., and Croft, W. Lda-based document models for ad-hoc retrieval. *In Proc. of ACM SIGIR* (2006).

[135] Weiss, G. Multiagent system. *MIT Press Ed.* (1999).

[136] Wen, J.-R., Nie*, J.-Y., and Zhang, H.-J. Clustering user queries of a search engine. *Proceeding of the Tenth World Wide Web conference (WWW10)* (May, 2001), 162–168.

[137] Witten, I. H., Paynter, G. W., Frank, E., Gutwin, C., and Nevill-Manning, C. G. Kea: practical automatic keyphrase extraction. *Proceedings of the fourth ACM conference on Digital libraries* (November 1999), 254–255.

[138] Wolfram, D. A query-level examination of end-user searching behaviour on the excite search engine. *CAIS 2000: Proceedings of the 28th Annual Conference of the Canadian Association for Information Science* (September 2000).

[139] Xerox, V. Valuequix - noth america information management, xerox corporation. *http://electronics.zibb.com/trademark/valuequix/29679369*.

[140] Yang, E., and Gu, D. Multiagent reinforcement learning for multi-robot systems: A survey. *A Survey, Technical Report CSM 404* (2004).

[141] Yang, Y. An evaluation of statistical approaches to text categorization. *Inf. Retr. 1* (May 1999), 69–90.

[142] Y.Yang, and J.O.Pedersen. A comparative study on feature selection in text categorization. *ICML '97: Proceeding of the 14th International Conference on Machine Learning* (1997), 412–420.

[143] Zeng, J., Wua, C., and Wanga, W. Multi-grain hierarchical topic extraction algorithm for text mining. *Expert Systems with Applications 37*, 4 (April 2010), 3202–3208.

[144] Zhang, H.-J., and Wen, J.-R. Query clustering in the web context. *Information Retrieval and Clustering* (December, 2003), 195–226.

[145] Zheng, Z., Shu-gen, M., Bing-gang, C., Li-ping, Z., and Bi, L. Multiagent reinforcement learning for a planetary exploration multirobot system. *In PRIMA* (2006), 339–350.

[146] Zubiaga, A. Enhancing navigation on wikipedia with social tags. *Wikimania 2009, the 5th International Conference of the Wikimedia Foundation* (August 26-28 Buenos Aires, Argentina, 2009).