Belief Revision for Adaptive Information Agents

by

Raymond Yiu Keung Lau (M.Sc., M.AppSc.)

A dissertation submitted for the degree of

Doctor of Philosophy

Centre for Information Technology Innovation

Faculty of Information Technology

Queensland University of Technology

August 2003

Keywords

Adaptive Information Agents, Information Retrieval, Information Filtering, Logic-based IR, Belief Revision, Nonmonotonic Reasoning, Artificial Intelligence, Information Theory, Text Mining.

Abstract

As the richness and diversity of information available to us in our everyday lives has expanded, so the need to manage this information grows. The lack of effective information management tools has given rise to what is colloquially known as the *information overload* problem. Intelligent agent technologies have been explored to develop *personalised* tools for *autonomous* information retrieval (IR). However, these so-called adaptive information agents are still primitive in terms of their *learning autonomy*, *inference power,* and *explanatory capabilities.* For instance, users often need to provide large amounts of direct relevance feedback to train the agents before these agents can acquire the users' specific information requirements. Existing information agents are also weak in dealing with the *serendipity* issue in IR because they cannot infer document relevance with respect to the possibly related IR contexts.

This thesis exploits the theories and technologies from the fields of Information Retrieval (IR), Symbolic Artificial Intelligence and Intelligent Agents for the development of the next generation of adaptive information agents to alleviate the problem of information overload. In particular, the fundamental issues such as *representation, learning,* and *classjfication* (e.g., classifying documents as relevant or not) pertaining to these agents are examined. The design of the adaptive information agent model stems from a basic intuition in IR. By way of illustration, given the retrieval context involving a science student, and a query "Java", what information items should an intelligent information agent recommend to its user? The

agent should recommend documents about "Computer Programming" if it believes that its user is a computer science student and every computer science student needs to learn programming. However, if the agent later discovers that its user is studying "volcanology", and the agent also believes that volcanists are interested in the volcanos in Java, the agent may recommend documents about "Merapi" (a volcano in Java with a recent eruption in 1994). This scenario illustrates that a retrieval context is not only about a set of terms and their frequencies but also the relationships among terms (e.g., $java \wedge science \rightarrow computer$, $computer \rightarrow programming$, $java \wedge science \wedge volcanology \rightarrow merapi$, etc.) In addition, retrieval contexts represented in information agents should be revised in accordance with the changing information requirements of the users. Therefore, to enhance the *adaptive* and *proactive* IR behaviour of information agents, an expressive representation language is needed to represent complex retrieval contexts and an effective learning mechanism is required to revise the agents' beliefs about the changing retrieval contexts. Moreover, a sound reasoning mechanism is essential for information agents to infer document relevance with respect to some retrieval contexts to enhance their proactiveness and learning autonomy.

The theory of *belief revision* advocated by Alchourrón, Gärdenfors, and Makinson (AGM) provides a rigorous formal foundation to model evolving retrieval contexts in terms of changing *epistemic states* in adaptive information agents. The expressive power of the AGM framework allows sufficient details of retrieval contexts to be captured. Moreover, the AGM framework enforces the principles of *minimal* and *consistent* belief changes. These principles coincide with the requirements of modelling changing information retrieval contexts. The AGM belief revision logic has a close connection with the *Logical Uncertainty Principle* which describes the fundamental approach for logic-based IR models. Accordingly, the AGM belief functions are applied to develop the learning components of adaptive information agents. Expectation inference which is characterised by axioms leading to *conservatively monotonic* IR behaviour plays a significant role in developing the agents' classification components. Because of the direct connection between the AGM belief functions and the *expectation inference relations,* seamless integration of the information agents' learning and classification components is made possible. Essentially, the learning functions and the classification functions of adaptive information agents are conceptualised by K_q^* and $q \mid\!\sim_K d$ respectively. This conceptualisation can be interpreted as: (1) learning is the process of revising the representation K of a retrieval context with respect to a user's relevance feedback q which can be seen as a refined query; (2) classification is the process of determining the *degree of relevance* of a document d with respect to the refined query q given the agent's *expectation* (i.e., beliefs) K about the retrieval context.

At the computational level, how to induce epistemic entrenchment which defines the AGM belief functions, and how to implement the AGM belief functions by means of an effective and efficient computational algorithm are among the core re-

search issues addressed. Automated methods of discovering context sensitive term associations such as *(computer* $\rightarrow$ *programming)* and preclusion relations such as *(volcanology* $\not\rightarrow$ *programming)* are explored. In addition, an effective classification method which is underpinned by expectation inference is developed for adaptive information agents. Last but not least, quantitative evaluations, which are based on well-known IR bench-marking processes, are applied to examine the performance of the prototype agent system. The performance of the belief revision based information agent system is compared with that of a vector space based agent system and other adaptive information filtering systems participated in TREC-7. As a whole, encouraging results are obtained from our initial experiments.

Contents

List of Figures

List of Tables

Statement of Original Authorship

The work contained in this thesis has not been previously submitted for a degree or diploma at any other higher education institution. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made.

Signed:

Date:

Acknowledgements

A number of people have contributed to my success in this endeavour. My sincere gratitude goes first to my supervisors, Associate Professor Arthur H.M. ter Hofstede and Professor Peter D. Bruza for their generous support and comments to my research work. I would also like to express my many thanks to Professor Mary-Anne Williams for her advice about the Maxi-adjustment transmutation method in several occasions and her generosity of lending the source codes of the SATEN belief revision system. I would like to thank Associate Professor K.F. Wong for his support and comments on my work during my over one month's visit at the department of System Engineering and Engineering Management of the Chinese University of Hong Kong from December 1999 to January 2000. I am grateful to Dr. Dawei Song for his friendship, valuable and fruitful discussions particularly when I was still struggling in the area of logic-based IR at the beginning of my PhD candidature.

Last but not least, I would like to thank for every member in my family for their physical and emotional support over the past few years. There are two supremely honourable people in this world, to whom all my writing has always been dedicated, my great parents! This thesis is also dedicated to my lovely Wan Wan who is going to write her great thesis in the not so far away future.

Chapter 1

Introduction

1.1 An Overview

We are living in the so-called "information age". Enterprises need information to identify whom they should do business with and when the corresponding business transactions should be processed. On the other hand, individuals need information for their daily activities such as comparison shopping, financial management, education, and entertainment. Whether the agents are organisations or individuals, they need to seek and make use of information to survive in modern society. The term "information seeking" refers to the processes by which information seekers retrieve information objects from some information sources. Information objects can be of any kind such as video clips, audio files, traditional documents, electronic mail, HTML sources, etc. The research work reported in this thesis is about the development of intelligent information agents which autonomously process large streams of unstructured information objects on behalf of their users. We focus on information objects represented in text format or those converted to text format. Theories and techniques from the fields of intelligent agents [JSW98, WJ95], information retrieval (IR) [Rij86, Rij89, BH94, Lal98], and symbolic artificial intelligence (AI) [GM88, GM94] are explored for the development of intelligent information agents.

In general, information seeking processes involve the following elements: information seekers and their information needs, information objects, and a matching function which maps specific information needs to *relevant* information objects. Figure 1.1 provides an overview of the information seeking process. In particular, the characteristics of information seekers (e.g., background, tasks on hand, etc.) and their specific information needs induce the *retrieval contexts* in which the relevance of information objects is evaluated. In automated IR systems, the matching function can be expressed *quantitatively* [SM83] or *qualitatively* [Hun95, BSW00]. Nevertheless, the matching mechanisms in IR systems can only compare the *representations* of retrieval contexts (e.g., users' information needs) with the representations of information objects. As these representations are only incomplete descriptions of the underlying entities, the matching processes in IR systems involve high uncertainties (i.e., the partiality problem) [Rij86, Lal98]. In other words, there isn't a sharp boundary distinguishing relevant objects from non-relevant objects with respect to a retrieval context. The matching process in IR is also called *classification* (e.g., classifying objects as relevant or non-relevant) in the discipline of machine learning [BP98, Coh95, ZH00].

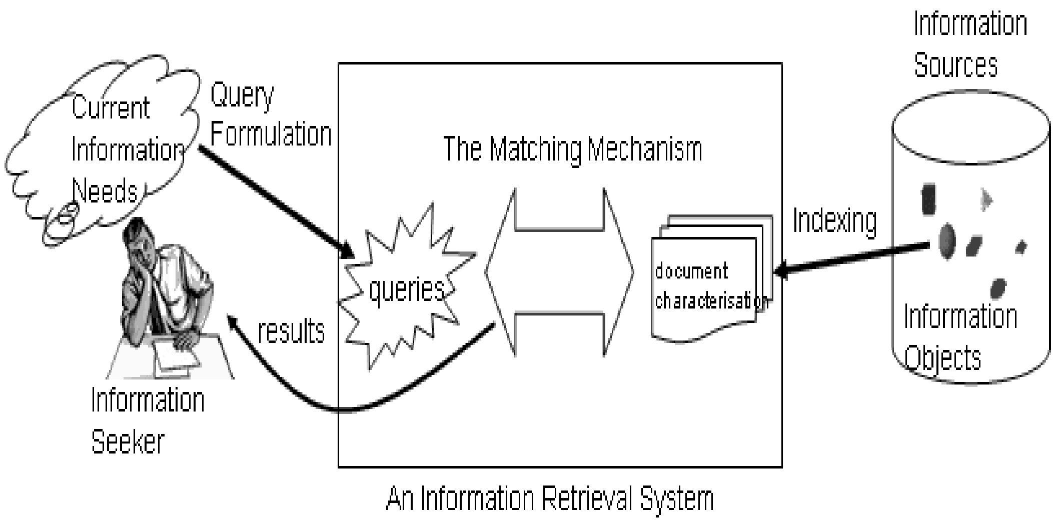

Figure 1.1: The Information Seeking Process

In this thesis, IR systems refer to any computer-based systems which automate the information seeking process. The representations of a users' specific information needs are often called "queries" by the IR research community [SM83]. On the other hand, the representations of information objects are called "document characterisations". If there is a close match between a query and a document characterisation, an IR system will infer that there may be a *semantic correspondence* between the underlying information need and the document. Documents refer to text-based information objects in this thesis. So, a Web page is called a document because it contains text corresponding to the HTML source. The most important function of an IR system is to estimate the *degree of match* between queries and documents (strictly speaking, document characterisations) with respect to specific *retrieval situations* (e.g., users' background, long-term IR goals, tasks on hand, etc.) [NBL95]. The degree of match between a document and a specific information need can be approximated by a distance metric. This requires transforming both documents and information needs to a common information space before applying a metric to quantify their distance. An information space in which both information needs and documents are expressed in terms of their semantics would be perfect. The corresponding distance metric could then exactly indicate their *semantic correspondence*. However, transforming information objects to a semantic information space requires extensive semantic analysis which is computationally expensive or even intractable [CBS90].

Another extreme is to choose a purely *syntactic* information space to represent both information needs and documents. This approach has been adopted by many existing IR models [SM83]. Nevertheless, there are problems with this approach. For instance, given a query term "Java", an IR system should retrieve any documents with indexing terms "Java" because there is a syntactic match between these two terms. However, if the information seeker just happens to be a tourist who is looking for resorts on the "Java" island, the documents about "Java programming" that may be returned by the IR system are totally irrelevant. The supposition that information is *intersubjective* [Dre81] can be applied to explain the above problem. The term "Java" probably carries some common (objective) information such as an island or a programming language. However, the intersubjective nature of information causes the mis-match in the information seeking process because the information seeker perceives that "Java" is about an island in Indonesia, whereas a human indexer may think that the term "Java" should be used to index document about a programming language. With the assumption that information is *intersubjective*, the design of the matching functions for IR should take into account the basic syntactical aspect (common meaning) of information as well as some considerations of the subjective interpretations of information seekers. To this end, *contextual information* should be used to refine a possibly ambiguous query term. The notion of *context* has been exploited in a variety of research disciplines which try to tackle the IR problems [DWR97, EM01, Hun95, LG98, Law00, NBL95]. Although there is no consensus about what constitutes a context, it is commonly believed that contextual information can be used to improve the effectiveness of IR [DWR97, Hun95, Law00, NBL95].

The proposed approach of developing the matching functions for IR lies in the middle of the two extremes of purely syntactic matching or purely semantic matching. An expressive language is used to capture users' specific information needs as well as the contextual information so that the intersubjective nature of information is taken into account by the IR model. From a pragmatic point of view, the work reported in this thesis exploits both the qualitative and the quantitative approaches for the development of adaptive information agent system, which is one kind of IR system. The expressive power of the AGM belief revision logic [AGM85] allows sufficient details of queries and query contexts to be captured. The learning and the classification (i.e., matching) functions in adaptive information agents are underpinned by the *AGM belief functions* and the corresponding *expectation inference relations* ($\mid\!\sim_K$) [GM94].

Expectation inference is a kind of *nonmonotonic inference* and its properties will be discussed in Chapter 3. Let q represent a user's query; d denotes the representation of a document; K represents a retrieval context; the learning and the classification functions in adaptive information agents can be conceptualised by:

1. Learning: Belief revision operations $*$ applied to K with respect to q (i.e., K^*_q);

2. Classification: Expectation inference relations such as $q \mid\!\sim_K d$.

An information agent's learning process can be interpreted in the way that the retrieval context K (i.e., an agent's knowledge base) is revised based on a user's relevance feedback q [SB90]. In the adaptive information agent model, the relevance feedback information q is considered as a refined query. Strictly speaking, a user's relevance feedback is used to generate the refined query q by *minimally transforming* the existing retrieval context captured in K using the AGM belief functions. Therefore, the learning processes of adaptive information agents are akin to the widely studied processes of query refinement based on relevance feedback information [BSA94, SB90]. The classification functions of adaptive information agents are underpinned by *expectation inference relations*. Conceptually, a document characterisation d is evaluated with respect to the refined query q given an agent's expectation K about a retrieval situation as background information. The seamless integration of the learning and the classification functions in adaptive information agents can be realised via the well-known connection between belief revision and expectation inference [GM94, MG91]:

$$\text{If } d \in K_q^* \text{ , then } q \mathrel{|\!\!\sim_K} d$$

The interpretation of the above logical connection in IR is that the refined query q nonmonotonically entails the document characterisation d given the set K of background information if d is in the agent's knowledge base after revising K with respect to the query q. Since an adaptive information agent manages a set of queries and the query context pertaining to a user, the focus is not on evaluating d with respect to an individual q but the set of queries containing in the agent's knowledge base K. At the conceptual level, $K \mathrel{|\!\!\sim_K} d$ can also be taken as the foundation of the agents' classification functions. In logic-based IR research, the usual formulation of the matching function is $d \mathrel{|\!\!\sim} q$, where d is the logical representation of an information object, q is a user query, and $\mathrel{|\!\!\sim}$ is a kind of inference relation [Rij86, CC92]. However, for knowledge-based or agent-based systems, it is a common approach to store a user's requirements in a knowledge base, and then apply formal reasoning to deduce if there are objects satisfying the user's particular requirements. For instance, when agent-based planning and scheduling is conducted, a user's requirements (also called constraints) are stored in a knowledge base K. Then, a particular plan or schedule d is evaluated with respect to the set of requirements stored in K [Kra97]. This knowledge-based view for general problem solving is adopted in the proposed adaptive information agent model. Accordingly, the matching function is characterised by $q \mathrel{|\!\!\sim_K} d$ based on an agent's expectation K about the particular retrieval situation.

In general, information seeking can be divided into two broad categories, namely *browsing* and *searching* [CHSS98, MSG97]. Sometimes an information seeker may not have a clear and specific search goal. They traverse the information sources such as the World Wide Web (Web) with the hope that interesting information objects will eventually appear. Such a process is referred to as *browsing*. The distinguishing feature of browsing is that the users' interests are assumed to be broader than those in information searching. For example, an information seeker trying to locate "the most touching stories around the world" is more likely to conduct browsing rather than searching. In other situations, an information seeker may have a more specific information need, for example, searching for information about "Mobile Agents". Information seeking of this kind is called *searching* [MSG97]. Information searching can be further divided into information retrieval (IR) and information filtering (IF). Information retrieval and information filtering are "two sides of the same coin" [BC92]. However, IR often refers to the situation that an information seeker takes an active role to specify their ad hoc queries, whereas IF is concerned with the removal of irrelevant information from a large incoming stream of dynamically generated information based on the user's long term and recurring retrieval goals stored in a persistent storage called *user profile*. This thesis focuses on information filtering where information agents take a *proactive* role of selecting relevant information objects for their users based on the users' long term information needs. Figure 1.2 depicts the adaptive information filtering process. The distinct features of an adaptive IF system are the deployment of a user profile to maintain a set of recurring information requirements (i.e., queries) and the application of users' relevance feedback to continuously revise the content of the user profile.

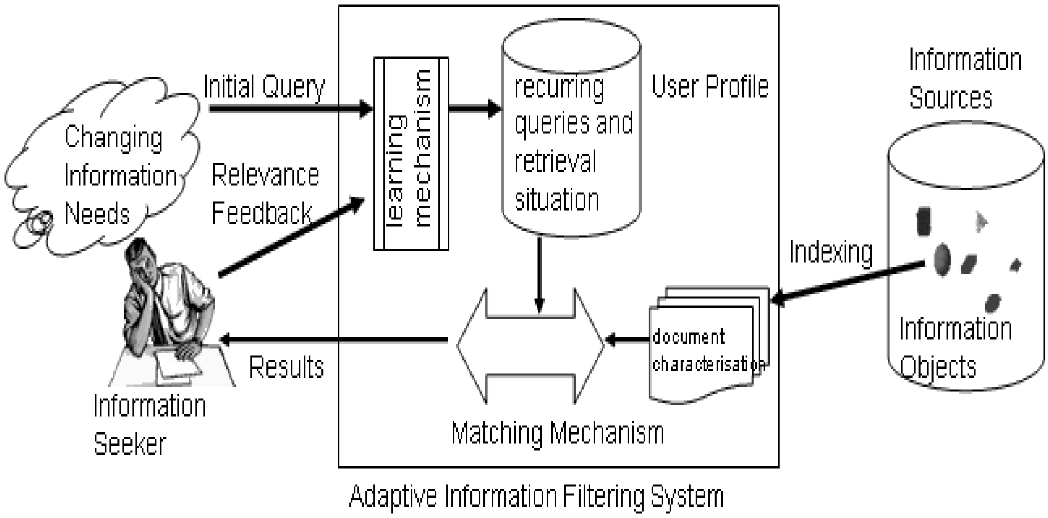

Figure 1.2: The Adaptive Information Filtering Process

1.2 Problem Statement

Distributed computer-based information systems such as the Internet have undergone exponential growth in recent years. This phenomenon has led to the growing availability of large, dynamic, heterogeneous, and distributed sources of information like the World Wide Web (Web). Information of this kind is normally unstructured when compared with the structured information stored in traditional database systems. As more information becomes available, it becomes increasingly more difficult to find relevant information from these ever-faster growing dynamic sources. Many information seekers may experience that information seeking on the Internet resembles "searching a needle in a haystack". This is the so-called problem of information overload [Mae94, TKS00]. Accordingly, there is a growing demand for the development of personalised and autonomous IR systems which can select relevant information objects on behalf of their users.

Existing general purpose search engines and browsers provide the basic assistance to information seekers for locating information objects. However, finding relevant information even for a narrow query (i.e., searching) in a specific domain has become more and more difficult with the growth of the Web. It is not uncommon for an information seeker to obtain thousands of hits which match their query while most of these hits are actually irrelevant with respect to their interests. The user then needs to traverse the list of retrieved documents in order to find the relevant ones. However, most users only have the patience to examine one result screen [BH98]. Therefore, relevant information may not be discovered via search engine IR. This difficulty is understood as the *low precision problem.* Precision is defined as the ratio of retrieved relevant document to retrieved documents [SM83]. There are many reasons for this low precision problem. Firstly, user queries are often short and not specific enough. In fact, a study conducted by Infoseek has showed that the average Internet query consists of only 2.2 terms [Kir98]. A more recent survey performed based on the Alta Vista's log files also confirms that the average number of terms in a query for the Alta Vista search engine is only 2.35 [Cor00]. Furthermore, natural language ambiguity often results in users describing concepts in their queries in a quite different manner than the authors describe the same concepts in the published information objects such as Web documents. With reference to the general information seeking process depicted in Figure 1.1, the problem can be understood in the way that information seekers have difficulties in translating their implicit information needs (e.g., the cloud in the diagram) into precise queries given an artificial query language. Even if they can partially express their needs by some query terms, these terms do not correspond to the indexing terms (i.e., document representations) used to characterise the required information objects because of the *intersubjective* nature of information. In other words, a query and its associated *context* is often not clear to an IR system.

For information filtering applications, the recurring information needs of a user are often stored in a persistent storage, also called user profile. However, as a user's information needs will change over time, the content of the user profile must be revised promptly and accurately to capture the user's latest interests; otherwise the filtering effectiveness of the IR system will drop. Unfortunately, most of the existing search tools such as Internet search engines or meta search engines do not support these functions. In summary, the general problem area examined by this thesis is:

> "*The development of an automated, personalised, and adaptive information management tool for the dissemination of relevant information from large, dynamic and unstructured information sources to information seekers.*"

The specific problems tackled by this thesis are:

- The Representation Problem: Capturing users' implicit and recurring information needs in terms of the corresponding queries and retrieval situations;
- The Learning Problem: Developing effective means for continuous refinement of the representations of retrieval contexts;
- The Classification Problem: Developing effective and efficient means of estimating the semantic correspondence between retrieval contexts and information objects.

1.3 The Requirements of Effective IR Systems

To tackle the information overload problem, effective IR systems should be:

- **Autonomous**: With the exponential growth of information encoded in electronic form, information seekers are faced with the problem of information overload. It is extremely difficult, if not totally impossible, for an information seeker to scan through all the available information items manually. Therefore, IR systems should be able to autonomously select relevant information for their users with a minimum amount of human intervention.
- **Proactive**: IR systems should not only work in a passive mode by taking users' instructions and responding accordingly, but also behave proactively in

the sense that they can retrieve relevant information without requiring the users to provide the low level instructions regarding what items should be retrieved. This kind of proactive behaviour can be achieved if IR systems can reason about the informational goals of their users given the appropriate contextual information.

- **Adaptive**: As a user's information needs will change over time, an IR system should be responsive to these changing needs and adapt its information retrieval behaviour accordingly. This requirement involves learning users' changing information needs promptly and accurately, and making use of the most current representation of a user's interests to determine relevant information objects.

- **Explanatory**: Because of the *intersubjective* nature of information, IR systems should be able to explain their decisions about document selection so that any difference in terms of the perceived document relevance between a system and its user can be reconciled. By explaining its decisions, an IR system can help its users understand their implicit information needs better and hence the users can develop more accurate queries at a later stage.

- **Scalable**: Because of the explosive growth of the amount of information available on computer-based networks, IR systems should be able to scale up (in terms of speed and capacity) to process large and dynamic streams of information in a timely fashion.

In addition, an effective IR system should optimise both precision and recall while retrieving information objects.

1.4 General Approaches for Adaptive IR Systems

The problem of developing effective and efficient IR systems has been examined by various research communities in computer science. This section provides an overview of the work conducted in various discipline areas such as intelligent agents, logic-based IR, and a specialised topic in logic-based IR - Belief Revision.

1.4.1 Intelligent Agent Technologies

Intelligent agents are a new paradigm for developing software applications. Currently, agent technologies are the focus of intense interest in many sub-fields of computer science and artificial intelligence. Intelligent Software Agents are being used in an increasingly wide variety of applications such as email filtering, Web page retrieval, comparison shopping, computer games, industrial process control, air traffic control, etc. An intelligent agent is a computer system situated in some environments, which is capable of *autonomous* action in these environments in order to meet its design objectives [JSW98, WJ95]. The concept of autonomy simply means that the agent system should be able to act with minimal human intervention, and should have control over its own actions and internal state. In addition, intelligent agents should be *responsive* in the sense that they can perceive their environment (e.g., a production line, a user, a collection of agents, the Web, etc.) and respond in a timely fashion to changes that occur in the environment. Intelligent agents are also *proactive*. They should not simply act in response to their environment, but also demonstrate opportunistic, goal-directed behaviour. In other words, intelligent agents take the initiative to perform tasks to fulfil their design objectives where appropriate. They can interact with other artificial agents and humans in order to solve their problems and help others complete their tasks.

One important contributing factor to the problem of information overload is that an information seeker is required to constantly direct the information seeking process. To alleviate this problem, intelligent information agents can search for relevant information on behalf of their users (e.g., agents acting autonomously to search the Web). The idea is so compelling that many research projects are directed to achieve this goal. Jasper is a distributed system of intelligent agents which perform information retrieval tasks over the Internet and the Web on behalf of a community of users [DWR97]. Jasper can summarise and extract keywords from the Web pages and can share information among users with similar interests automatically. A Jasper agent holds a profile of a user's interests and conducts autonomous information seeking based on such a profile. Moreover, by observing a user's interactive behaviour, the agent can learn more about the user's interests over time. SAIRE is another multi-agent information retrieval system operating in the space science domain [OKW+97].

It is divided to three layers. The top level contains interface agents responsible for accepting input from the users. The middle layer acts as a co-ordinator with the information retrieval engines working at the bottom level. Based on previous work from the User Modelling research community, the SAIRE agent system can assign its users to different stereotypical user groups. A user's initial information needs are inferred based on the corresponding stereotypical group and a user profile is then specialised based on the user's continuous interaction with the system.

1.4.2 Logic-based IR

The central issue in IR is to develop a matching function to determine if a document is relevant with respect to a user's information needs. For logic-based IR models, a document and a user's need are represented by the logical formulae d and q respectively. The matching function is underpinned by logical deduction of the form $d \mid\!\sim q$, where $\mid\!\sim$ is a non-classical inference relation. As both d and q are only partial representations of a document and a user's need, it is often the case that d cannot entail q based on the rigid classical derivability relation. The *logical uncertainty principle*, which is a generalisation of the Ramsey test for IR, states that [Rij86]:

> "Given any two sentences x and y, a measure of the uncertainty of $x \rightarrow y$ relative to a given data set is determined by the minimal extent to which we have to add information to the data set, to establish the truth of $x \rightarrow y$."

where x and y are often viewed as the logical representation of a document and a query respectively. Since Van Rijsbergen advocated the logical uncertainty principle for IR, many logic-based IR models have been proposed. Although these logic-based IR models employed different formalisms such as default logic [Hun95, Hun97], conditional logic [NBL95], logical imaging [CvR95, Cre98], situation theory [LR92], nonmonotonic inference [AG96], terminological logic [MSST93], modal logic [Nie89], preferential logic [BL98], etc., they all examined the idea of minimally revising the retrieval situation so as to evaluate the degree of match between d and q. The adaptive information agent system discussed in this thesis is built on top of a belief revision based IR model which adheres to the above principle. Through the close connection between the AGM belief revision and the Ramsey test [Gär88], the proposed logical information agent model can be seen as a direct implementation of the logical uncertainty principle.

Recent investigations into logic-based IR have attempted to formalize the notion of "aboutness" (i.e., information matching) by axiomatising its properties in terms of a neutral, theoretical framework [BH94, HW98, BSW00]. The motivation for this has been to study the aboutness relation from a theoretical stance in order to better understand what properties of this relation promote effective retrieval (as well as which properties do not). The neutral, underlying framework is important as it allows aboutness to be studied independent of the idiosyncrasies of a given information retrieval model. There is as yet no consensus regarding the property of aboutness except that it should be logic-based. The concept of aboutness in IR has been applied to examine some postulates of the AGM belief revision paradigm at the conceptual level to see if the belief revision framework is applicable in the context of IR.

In statistical analysis, the relationships among key phrases are established by frequency ratios, whereas in semantic analysis, the relationships are established by meaning. It is believed that semantic information is critical in matching a user's needs to information objects [Hun97]. For automating the use of semantic information, it is necessary to specify when a particular specialisation, generalisation, or synonym relationship should be used. Accordingly, an expressive formal framework is required to capture and reason about the semantic information. Hunter proposed to use nonmonotonic logics, particularly default logic, to process semantic information about terms, and hence to identify the semantic relationships between queries and documents [Hun95]. The notion of *term positioning* is proposed to conduct query re-formulation. Given a query q, it is possible to *strengthen* ($q^* \vdash q$ and $q \nvdash q^*$), *weaken* ($q^* \nvdash q$ and $q \vdash q^*$), or *substitute* ($q^* \nvdash q$ and $q \nvdash q^*$) q by the refined query q^* to improve the effectiveness of information retrieval. In particular, default logic provides the machinery to conduct term positioning. In default logic, a default theory $T = (W, D)$, which consists of a set of classical axioms W and a finite set of default rules D, is used to derive a new set of information called an extension E. This kind of reasoning can be characterised by a nonmonotonic consequence relation $\mid\!\sim$. A sentence $\gamma \in E$ is *defeasible* since the process of default reasoning is based on incomplete (e.g., W) and uncertain information (e.g., D). A default rule $\delta \in D$ has the following form: $\frac{\alpha:\beta}{\gamma}$, where α is called the prerequisite, β the justification, and γ the consequent of δ. The semantics of a default rule δ is that: If α is known, and if β is consistent with all the current knowledge E, then conclude γ. The current knowledge E (i.e., an extension) is obtained from the facts W and the consequents of some defaults that have been applied previously. Formally, a default δ is applicable to a deductively closed set of formulae E iff $\alpha \in E$ and $\neg\beta \notin E$.

When applying default logic to term positioning, the default rule set D consists of default rules capturing the semantic relationships such as synonym, specialisation, generalisation, and polysemy among terms. The classical theory W is used to describe the original query. Then, the notion of an *extension* E is used to describe the refined query. For example, given a default rule:

$$\frac{oil \wedge cooking : \neg petroleum}{\neg petroleum}$$

and a query $olive \wedge oil \wedge cooking$, the original query can be refined to exclude any information objects about $petroleum$. In other words, the new query $q^* = olive \wedge oil \wedge cooking \wedge \neg petroleum$ is positioned. Given a clearer retrieval context, it is anticipated that the precision and recall of the subsequent retrieval can be improved. It was proposed that the default rules between terms could be manually elicited from domain experts by asking them to illustrate the synonym, polysemy, generalisation, and specialisation relationships [Hun95, Hun97].

It has been proposed that IR processes should be underpinned by nonmonotonic reasoning [AG96, BL98, BH96]. Based on users' relevance feedback on information objects, preferential and rational orderings can be generated. Thereby, well-behaved nonmonotonic inference relations (e.g. preferential inference or rational inference) [Geo96] can be used to deduce the relevance of information objects with respect to a user's information needs. Essentially, given a set of relevant documents D^+ and a set of non-relevant documents D^- judged by a user, the *rational ordering* between two terms t_1 and t_2 is defined by: $t_1 \preceq_+ t_2$ iff $|D^+_{t1}| \leq |D^+_{t2}|$; $t_1 \preceq_- t_2$ iff $|D^-_{t2}| \leq |D^-_{t1}|$. In other words, a term t_2 is with a higher rank than another term t_1 with respect to $\preceq_+$ if the number of relevant documents containing t_2 is more than the number of documents containing t_1. In addition, t_2 is with a higher rank than t_1 with respect to $\preceq_-$ if the number of non-relevant documents containing t_2 is less than the number of documents containing t_1. Then the *preferential ordering* ($\preceq$) of these terms is: $t_1 \preceq t_2$ iff $t_1 \preceq_+ t_2$ and $t_1 \preceq_- t_2$. It is argued that the preferential ordering $\preceq$ will be changed with respect to the user's relevance feedback on D^+ and D^-. Therefore, the set of conclusions regarding document relevance grows nonmonotonically. However, the details of how to apply the proposed non-monotonic inference relations to information matching was not reported in their paper [AG96].

Huibers and van Linder [HvL96] attempted to formalise intelligent information retrieval agents based on modal logic. Modal operators were introduced to address the essential concepts such as *aboutness*, *non-aboutness*, and *information preclusion* in IR [BH94]. For example, one kind of retriever agents is defined based on the notion of aboutness $d \models_a q$ (i.e., a document d to be about a query q). Strictly speaking, $d \models_a q$ iff the agent knows that the query q is satisfied in at least one document model d. In addition, it is believed that such a satisfiability relation is non-classical. Moreover, the retriever agent considers a document d to be non-about q, denoted $d \not\models_a q$, iff it knows that d implies the negation of q. This is a step forward towards improving the expressive power and explanatory ability of information agents since the agents' behaviour can be justified based on formal reasoning.

Logical imaging has been applied to develop IR models [CvR95, Cre98]. The goal is to evaluate the probability of the *conditional* $d \rightarrow q$ based on the *kinematics of probability distributions* over terms. In conditional logic, a counterfactual such as $x \rightarrow y$ can be evaluated by first *imaging* x on the closest world w_x (i.e., the x-world) that satisfies x and then examining if y is satisfied in w_x or not [Sta81]. If y is satisfied in w_x, the counterfactual is true; otherwise it is false. The close world is determined by an accessibility relation $A \subseteq W \times W$, where W is the set of possible worlds based on the possible world semantics [Kri71]. When the probability $Pr(d \rightarrow q)$ of a conditional $d \rightarrow q$ is evaluated, the formula d will be imaged on the closest world(s) t, where t is a term (keyword) representing a world in the logical imaging IR model. Then, the formula q is evaluated in these closest world(s). To capture the uncertainty of an IR process, the worlds are characterised by a probability distribution. That is, a document d is satisfied in a world t with a prior probability. These prior probabilities are induced based on the Inverse Document Frequencies (IDF) of terms in a collection. The IR logical imaging paradigm consists of several methods to deal with the kinematics of probabilities associated with the worlds. Indeed, the transfer of probabilities among worlds rather than the inference relations examined in conditional logic is the key element of the imaging IR model. For instance, imaging on the d-world(s) is taken as transferring the priori probabilities from the non d-world(s) to the closest d-world(s) according to a distance measure derived from the mutual information MI between pairs of terms.

In the simplest form such as *standard imaging*, the probability associated with a non d-world is simply transferred to the closest d-world. Then, for each term appearing in a query, the posterior probability (with probability transferred from a non d term) of the term is summed to derive the Retrieval Status Value (RSV) of the document with regards to the query q. So, for standard imaging, the RSV is derived by: $Pr(d \to q) = \sum_t Pr(t) \times \tau(t_d, q)$, where $\tau(t_d, q) = 1$ if a query term appears in a d world (i.e., d and q have overlapping terms); otherwise $\tau(t_d, q) = 0$ is obtained. The probability distribution $Pr(t)$ represents the posterior probability assigned to each term (world) t. In *general imaging*, standard imaging is generalised in the sense that there could be more than one closest world where d is true. From the point of view of the kinematics of probability distributions, an *opinionated probability distribution* is defined for each term so that a set of probabilities can be transferred from some non d-term(s) to a d-term. In *proportional imaging*, the percentage of probability transferred from each non d-term to a d-term can be defined separately via another opinionated probability distribution. Implementations of standard imaging and general imaging were conducted using C programming and probabilistic datalog respectively. Rigorous evaluation based on the TREC-4 routing task was attempted [CRSR95]. The TREC-based evaluation approach applied to the belief revision based information agent system reported in this thesis is to a large extent motivated by the evaluation method adopted for the logical imaging IR model. However, the main difference between the logical imaging IR model and the belief revision IR model is that term weights representing a user's preferences are induced with respect to *epistemic entrenchment* which satisfies *possibilistic* rather than probabilistic axioms [DP91] in the belief revision based IR model. Above all, the entrenchment degrees of terms are derived according to a user's preferences over the underlying terms, and the kinematics of the entrenchment degrees are also driven by the changes of a user's preferences. In this sense, the assumption of the proposed IR model is quite different from the assumption of the *system perceived relevance* adopted in the logical imaging IR model.

It has been pointed out that evaluating logic-based IR models is a great challenge by itself [CRSR95]. One of the main contributions of our research work is to develop an operational logic-based information agent system and apply rigorous IR benchmarking processes to evaluate both the effectiveness and efficiency of the implemented system.

1.4.3 Belief Revision

The notion of *beliefs* has been used to represent users' information needs. In addition, belief functions have been applied to refine the representation (i.e., beliefs) of a user's information needs. The earliest and the most ambitious attempt of applying the belief revision formalism to IR was to use the notions of beliefs, desires, and intentions to characterise an information seeker's (e.g., a librarian) high level IR goals and to employ the belief revision framework to simulate the changes of mental states in an information seeker's memory [LRJ94]. Because of the changes of mental states of an information seeker, the corresponding IR system must revise its beliefs about the information seeker's interests in order to maintain accurate retrieval. A Natural Language Processing (NLP) technique was used to induce the system's beliefs about an information seeker's information needs based on the continuous interactions between the information seeker and the IR system. As the information retrieval process can be seen as comprising many low level sub-tasks, the corresponding IR system is designed as a multi-agent system with each autonomous agent performing a particular IR subtask. Accordingly, the belief revision process is not only applied to a single agent, but to a set of agents with inter-related interests and beliefs. The multi-agent belief revision model developed by Galliers [Gal92] was adopted to implement this function. In fact, such a multi-agent belief revision model is built based on the Assumption-based Truth Maintenance System (ATMS) computational apparatus [dK86].

The AGM belief revision framework was examined to develop a query reformulation logic in IR [AB99]. The correspondence between query refinement in IR and theory change in belief revision was analysed from a theoretical point of view. It is believed that query expansion and query refinement can be explained in terms of the revision of a user's beliefs in query terms [AB99]. For instance, if a user's original query is represented by a belief set K, and a new query term α is used to replace term β (i.e. query revision), the resulting query will be defined by: $(K^-_\beta)^*_\alpha$. It was assumed that query terms were represented as formulae in a chosen logic language. For query contraction (i.e. removing certain terms to broaden the search scope), the operation can be expressed by: K^+_α, where the query K is expanded by the originally rejected terms α. However, there may be difficulty in applying the belief revision paradigm to formalise the query reformulation logic. For instance, the interpretation of negation is different in these two settings. Given an information preclusion relation such as $dog \perp flying$, dog and $flying$ are considered inconsistent in most IR contexts. However, in general dog and $\neg dog$ are considered inconsistent, but dog and $flying$ are considered consistent in the AGM logic.

Dalal's belief revision operator [Dal88] was examined for document ranking in IR [LB99]. Essentially the construct of a total pre-ordering on interpretations, which is used to define belief revision operators for knowledge base changes, is applied to model a user's preferences over information objects. Dalal's revision makes use of the cardinality of the *symmetric difference* between two interpretations I and J as a measure of the distance $dist(I, J)$ between them. For example, the semantic distance between the set of models of ψ (i.e., $\mathcal{M}(\psi)$) and I is defined as: $dist(\mathcal{M}(\psi), I) = \mathrm{Min}_{J\in\mathcal{M}(\psi)}\, dist(J, I)$. Thereby, a faithful assignment of a total pre-order $\leq_\psi$ is defined: $I \leq_\psi J$ iff $dist(\mathcal{M}(\psi), I) \leq dist(\mathcal{M}(\psi), J)$. In IR, if a user's information needs N and a document D are represented as formulae q and d respectively, the *similarity* between N and D can be approximated by the symmetric distance of the corresponding models. For example, for each $m \in \mathcal{M}(d)$, $dist(\mathcal{M}(q), m) = \mathrm{Min}_{J\in\mathcal{M}(q)}\, dist(J, m)$ is computed. An average measure can then be applied to compute the symmetric distance between $\mathcal{M}(q)$ and $\mathcal{M}(d)$: $Sim(D, N) = \frac{\sum_{m\in\mathcal{M}(d)} dist(\mathcal{M}(q),m)}{|\mathcal{M}(d)|}$.

1.5 Justifications of the Proposed Approach

Justification of the proposed adaptive information agent model is provided with reference to the requirements of effective IR systems discussed in Section 1.3. Intelligent agents [WD00] provide the technological foundation to develop the next generation of information management tools. The idea of *autonomous* and self-motivated agents are appealing when it is applied to information retrieval in general and information filtering in particular. With the sheer volume of information available via computer-based networks such as the Internet, it is impossible for information seekers to traverse the huge information space by themselves given the limited time. For IF applications where information seekers are dealing with recurring IR tasks, the savings generated by employing autonomous and personalised information agents are even bigger. More importantly, intelligent information agents are able to infer users' information goals and *proactively* take actions to fulfil these goals with minimal direct instructions from their users [ACL$^+$00, BPR$^+$99, YKL00, O'L97, TNMH97]. Since information agents are *responsive*, they can continuously observe their users' on-line retrieval activities. Based on this information, information agents can learn users' changing information needs by revising the representations of the users' interests stored in the agents' persistent memories. Therefore, matching between users' interests and information objects becomes *adaptive*.

The expressive power of logic is believed to be able to model most of the fundamental aspects in information retrieval [CC92, LB98, Rij86, Seb98]. To empower intelligent information agents with the abilities to capture users' queries and the corresponding query contexts, an expressive representation language should be used. With such an expressive representation language, it is possible to generate appropriate explanations about an information agent's retrieval decisions. However, it is understood that classical logic is too rigid to deal with incomplete and uncertain information in IR processes [Lal98, Rij86]. Therefore, nonmonotonic logics have been explored to model the matching function of IR [AG96, Cre98, Hun95, Hun97, LB99]. One distinct characteristic of nonmonotonic logics is that the conclusions derived from the nonmonotonic reasoning processes are *defeasible*. This assumption is based on the observation that the information stored in an agent's knowledge base may be incomplete. Observing that the assumption of nonmonotonic reasoning closely resembles the characteristics of IR processes where the representation of a user's information needs is incomplete, and so the conclusion about the relevance of documents is in general defeasible. When more information about the user's interests is obtained at a later stage, the classification decision made by an agent at an earlier stage may turn out to be false. This gives rise to the requirement of continuously learning an information seeker's information needs. Learning and classification are orthogonal. However, these two functions are closely related and affect one another. It is not surprising to find that a formal logical framework is available to model this reality. The AGM belief revision logic has been proposed to formalise the changes of an agent's beliefs [AGM85], and it has also been proved that the basic information (e.g., epistemic entrenchment) that characterises the belief functions also induces the corresponding nonmonotonic consequence relations [MG91, GM94]. From a pragmatic point of view, by revising new information about a user's interests into an agent's persistent memory, it may lead to a new conclusion about document relevance drawn by the agent. This new conclusion may contradict the agent's previous conclusions about document relevance. Therefore, the AGM belief functions and the corresponding expectation inference relations are applied to develop the learning and the classification (i.e., matching) mechanisms of adaptive information agents. The idea of applying belief revision and nonmonotonic reasoning to practical applications has been explored [BGMS98]. The work reported in this thesis produces a concrete example of applying these closely related formal frameworks to real-life applications.

Logan reported that the multi-agent belief revision approach for modelling high level IR goals posed a serious computational problem [LRJ94]. Even the effectiveness of such a multi-agent IR system required further evaluation. The proposed belief revision framework for modelling adaptive information agents is quite different from Logan's approach [LRJ94]. Firstly, only propositional horn clauses are used to represent retrieval contexts. The computationally expensive model operators such as beliefs, desires, and intentions are not used. Secondly, belief revision and the corresponding expectation inference are applied to the belief set of a single agent which deals with a specific information topic. This approach substantially improves the computational efficiency over a multi-agent belief revision model. At the computational level, the AGM belief revision logic is implemented using the anytime transmutation algorithm which is shown to be computationally tractable [Wil97]. The AGM belief revision framework has been used for requirement analysis in software software engineering applications [Wil96a]. In addition, the AGM framework has been applied to model changes to consumer preferences with implementation based on relational database technologies [Wil96a].

Losada [LB99] applied another preference relation used for defining belief revision operators to develop a matching function which deals with partiality in IR. However, it is extremely costly to compute the symmetric difference between sets of models even with a moderate number of atoms. Whether such an approach can be implemented in practice is questionable. This is the reason why a formula-based rather than a model-based approach is preferred for the implementation of the AGM belief functions and the expectation inference relations [GM94]. The proposed learning and classification models for adaptive information agents are based on the formula-based approach since it is more computationally friendly. Therefore, there is a better chance for the proposed adaptive information agent model to satisfy the scalability requirement of intelligent IR systems.

1.6 Contributions of the Thesis

The work presented in this thesis applies theories and techniques from the fields of IR, and theoretical and applied AI to develop the next generation of information management tools to alleviate the information overload problem. The specific contributions made by this thesis are as follows:

1. The design and development of a novel adaptive information agent model. This includes:

    - Developing a formal framework to properly capture retrieval contexts;
    - Formalising the agents' learning functions by means of the AGM belief revision operators;
    - Formalising the agents' classification functions based on the expectation inference relations;
    - Seamless integration of the learning and the classification mechanisms;

- Enhancing information agents' learning autonomy by means of nonmonotonic reasoning;
- Improving the explanatory power of information agents based on enriched representations of retrieval contexts and nonmonotonic reasoning;
- Exploring effective IR in terms of balanced precision and recall in adaptive information agents;

2. Developing a new logical model for adaptive IR based on the AGM belief revision framework;
3. Developing a novel IR model which combines the strength of both quantitative and qualitative approaches;
4. Developing an efficient and effective transmutation algorithm to implement the AGM belief functions;
5. Developing context sensitive text mining methods to extract contextual information for adaptive IR;
6. Applying IR bench-marking processes to validate the proposed logic-based IR models and to compare the performance of the logic-based IR model with other quantitative IR models;
7. The formal connection between belief revision and nonmonotonic inference has been proposed a decade ago. Our work represents the first successful application of such a connection to large real-world applications;

8. The AGM belief revision framework has been studied in a purely theoretical context for more than a decade. A major contribution of this thesis is to perform a large scale empirical evaluation of the AGM framework in the context of adaptive information retrieval.

1.7 Research Methodology

To tackle the research problems raised in Section 1.2, the System Development Research Methodology [NCP91] is adopted as the framework to guide the entire research process. The iterative research process is depicted in Figure 1.3. The *Conceptual Framework* stage involves our extensive study of the chosen domain and the development of new theories and techniques to tackle the research challenges. For instance, how to represent retrieval contextual in information agents and how to empower adaptive information agents with learning and classification capabilities will be addressed at this stage.

At the *System Architecture* stage, an overall system architecture is developed to ensure that theories and models established at the Conceptual Framework stage can be implemented and subsequently tested. With reference to our research, the general architecture of the adaptive information agent system will be developed. The interfaces to other external systems (e.g., Internet search engines) will also be identified. Based on the overall system architecture, the *System Analysis and Design* stage

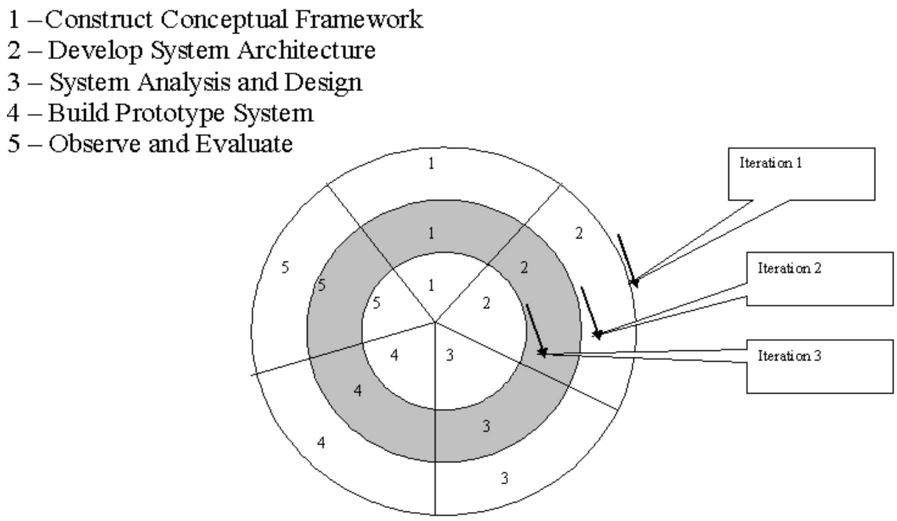

Figure 1.3: The System Development Research Methodology

involves a detailed specification of "what" the prototype system's functionalities will be and "how" these functionalities can be implemented on specific hardware/software platforms. Corresponding to the architect's model of a building, a prototype software system will be developed according to the specification produced at the design stage to demonstrate the feasibility and effectiveness of the proposed theories and models. At the prototype building stage, the necessary programming and testing work will be conducted based on the chosen development tools.

Above all, the resulting physical prototype system provides the basis to test and evaluate the proposed conceptual framework at the *Observe and Evaluate* stage. For example, with reference to the filtering function of the prototype system, a large collection of documents with predefined relevance judgement can be fed to the information agent to evaluate its effectiveness. The annual Text Retrieval Conference

(TREC) has developed a filtering track with predefined documents and relevance judgement to evaluate the effectiveness of IR and IF systems [Hul98]. Therefore, the bench-marking procedure developed by the TREC forum becomes an integral part of the evaluation procedure for our information agent model. Another text collections such as the Reuters-21578 collection will also be used to evaluate the prototype agent system to improve the *external validity* of the research work. Results from the evaluation stage may lead to the refinement of the original theoretical framework or the formulation of new research questions for further research.

1.8 Outline of the Thesis

The rest of this thesis is organised as follows. The following chapter is a critical review of existing adaptive information agent systems. It identifies the main paradigms of adaptive information agents and pinpoints the weaknesses of existing information agents. Chapter 3 gives the formal definitions of the AGM belief revision logic, and discusses the rationale of applying the AGM belief revision framework to IR in general and adaptive information agents in particular. Chapter 4 illustrates the proposed adaptive information agent model and its implementation with reference to the fundamental issues such as knowledge representation, learning, and classification. It describes the computational details (e.g., induction of epistemic entrenchment orderings, the transmutation algorithm that implements the AGM belief revision functions, the classification model, etc.) required to implement the belief-based information agent system. A thorough description of the experiments performed for the prototype adaptive information agent system and a detailed analysis of the initial experimental results are presented in Chapter 5. Chapter 6 contains the concluding remarks and some directions for future research work.

Chapter 2

A Review of Adaptive Information Agents

The materials presented in this chapter are largely based on those published in [Lau02b]. Contemporary models of adaptive information agents are developed with a view to alleviating the information overload problem [ACL+00, BP99, YKL00, MB00]. Some representative adaptive information agents are studied with reference to the various paradigms which underpin the development of these agents. Table 2.1 depicts a cross section of agent systems and their paradigms. Such a classification could be controversial. However, it serves the purpose of establishing a starting point for further investigation into the respective agents and the associated paradigms. The origins of these agent systems are highlighted, followed by their general functionalities such as on-line browsing, filtering, or direct Web traversal. Some adaptive information agents are hybrid systems which employ techniques from several paradigms. These agents are listed under the paradigm which best describes the dominating techniques. Although various paradigms of adaptive information agents have been explored, there is no general consensus of which paradigm or synergy of paradigms is the most effective one.

The main issues related to the development of adaptive information agents are examined. These issues include feature extraction (i.e., how to represent documents and users' interests), feature selection (i.e., the methods of removing noisy and irrelevant features), classification techniques, and learning and adaptation mechanisms. Each adaptive information agent paradigm addresses these issues in a different way, and leads to various IR behaviour. Although the experimental results of some surveyed agent systems are available, it is not practical to directly compare their performance (e.g., classification accuracy) since the experimental settings vary. Therefore, a qualitative analysis of the agents' performance is conducted. For example, the agents' learning autonomy (i.e., the extent of human intervention involved in an agent's learning and adaptation process), mode of learning (e.g., incremental versus batch mode learning), explanatory power (i.e., an agent's ability to justify its decisions), exploration capability (i.e., an agent's ability to explore novel information topics), and their capabilities of processing implicit feedback are examined to infer the advantages and disadvantages of these agent paradigms. This is not an exhaustive listing of adaptive information agent systems. The agent systems are surveyed based on the availability of their technical details (e.g., journal publications) or their origins (e.g., developed by well-known research groups in information agents).

Agent	Origin	Browsing	Filtering	Web Traversal	Agent Paradigm
WebWatcher	Carnegie Mellon University	Yes	No	Yes	Vector Space
Letizia	MIT	Yes	No	Yes	Vector Space
LIRA	Standford University	No	Yes	Yes	Vector Space
Fab	Standford University	No	Yes	Yes	Vector Space
Syskill & Webert	U. California Irvine	No	Yes	No	Naive Bayesian
News Dude	U. California Irvine	No	Yes	No	Naive Bayesian
INFOrmer	U. College Cork	No	Yes	No	Associative Network
Amalthaea	MIT	No	Yes	No	Evolutionary
GIRAF	Granada University	No	Yes	No	Fuzzy Sets
InfoSpiders	U. California San Diego	No	No	Yes	Connectionist
Colombo	U. Catania & U. Torino	No	Yes	Yes	Symbolic
SIGMA	U. Carleton & NRC	No	Yes	No	Computational Economy
Ringo	MIT	No	Yes	No	Collaborative

Table 2.1: A Summary of Adaptive Information Agents

2.1 The Vector Space Paradigm

The vector space paradigm refers to the information agents which make use of vectors of term frequencies to represent documents and user's interests (i.e., part of a retrieval context). The agents' learning and classification functions are implemented based on the algebraic operations on the vectors. The behaviour of most of the agents in this category can be understood with reference to the vector space model [SM83] and its variants. WebWatcher [AFJM95, JFM97] is an intelligent browsing agent which recommends hyperlinks to a user while the user is browsing the Web. When an agent is initialized, the user is asked to specify their information interests (i.e., a query q) via a set of keywords. Feature extraction is conducted by extracting words from a query or document and computing the root forms of the words based on a stemming algorithm [Por80]. Strictly speaking, keywords actually refer to stemmed keywords. A query vector $\overrightarrow{q}$ is used to hold the weights of the keywords appearing in a query. The Term Frequency Inverse Document Frequency (TFIDF) method is used to compute the weight $w_q(k)$ of a keyword k from a query q [Sal91]:

$$w_q(k) = \left(\alpha + \alpha \times \frac{rf(k)}{\max_{k' \in q} rf(k')}\right) \times \log_2 \frac{N}{N_k} \tag{2.1}$$

Based on empirical studies, the weight factor α is set to 0.5 to optimize retrieval performance [Sal91]. The normalized term frequency (TF) is expressed as the fraction: $\text{TF} = \frac{rf(k)}{\max_{k' \in q} rf(k')}$, where $rf(k)$ is the raw term frequency of a keyword k. The raw term frequency is defined as the number of times a keyword k appears in the query q. Inverse document frequency (IDF) is expressed as the fraction: $\log_2 \frac{N}{N_k}$, where N is the total number of documents of a collection and N_k is the number of documents which contain the keyword k in the same collection. For information retrieval on the Internet, N is often approximated by the number of locally cached documents in an agent system. A hyperlink or a Web page is also represented by a vector of TFIDF weights. Similarly, the weight $w_d(k)$ of a keyword k in a hyperlink or document d can also be computed according to Eq.(2.1). The discriminatory power of a keyword k in a document d is proportional to its occurrence frequency in d and is inversely proportional to its occurrence frequency in the entire document collection [SM83]. The set of keywords with their TFIDF weights greater than a system threshold is selected to represent the corresponding hyperlink or Web document. In fact, this is a widely used *feature selection* method in information agents [Bal97, BS95, KF95, MM98].

Given a user's information interests, a Web document, and a set of hyperlinks of the Web document, the objective of WebWatcher is to learn a target function: $LinkValue : page \times interest \times link \rightarrow [0, 1]$. The agent recommends a hyperlink with the highest LinkValue to the user. Two slightly different learning and classification models were used in WebWatcher. The first one is called ANNOTATE which predicts the relevance of a hyperlink based on its similarity with the user's information needs. The annotation of a hyperlink consists of its textual description and the queries of some users who followed that hyperlink before. All the keywords with the TFIDF weights greater than a pre-defined threshold are selected to represent the hyperlink. Basically, a hyperlink vector is created for each hyperlink of the current Web page. To predict if a user should choose a particular link, the *cosine angle* (i.e., the cosine similarity measure [SM83]) between the query vector and the hyperlink vector of the current page is computed:

$$sim(\overrightarrow{q}, \overrightarrow{d}) = \frac{\sum_{i=1}^{n} w_q(k_i) \times w_d(k_i)}{\sqrt{\sum_{i=1}^{n}(w_q(k_i))^2} \times \sqrt{\sum_{i=1}^{n}(w_d(k_i))^2}} \tag{2.2}$$

where $\overrightarrow{q}$ and $\overrightarrow{d}$ are the query vector and the hyperlink vector representing a query q and a hyperlink d respectively. The term $w_q(k_i)$ represents the weight of the ith keyword k_i in the query vector $\overrightarrow{q}$, and the term $w_d(k_i)$ represents the weight of the ith keyword k_i in the hyperlink vector $\overrightarrow{d}$. It is assumed that there are n elements in each vector. The value of n is an input to a feature selection process (e.g., the top n terms with the highest TFIDF weights). The hyperlink with the highest cosine similarity score $sim(\overrightarrow{q}, \overrightarrow{d})$ is recommended to the user. In fact, the ANNOTATE method integrates content-based filtering and collaborative filtering [Oar97] into a single framework. The ANNOTATE method can be viewed in the following way: If there is a *correlation* between the information needs of previous users, represented by the hyperlink vector $\overrightarrow{d}$, and that of a current user, represented by the query vector $\overrightarrow{q}$, the hyperlink explored by the previous visitors is recommended to the current user. This is also a content-based method since the content of a hyperlink represented by its textual description is compared with the content of a user's query.

Another classification method called (RL) is used in WebWatcher. With the RL method, the agent tries to find the most rewarding path, which comprises a sequence of hyperlinks and Web pages that these hyperlinks point to starting from the current page. A reward is measured in terms of the sum of the cosine similarity score between a user's query vector and a document (or hyperlink) vector. For instance, if the WebWatcher agent wants to recommend a browsing path to its user, it evaluates the total reward for each path originating from the current Web page. The evaluation encompasses a pre-defined distance measured from the current page. The path with the biggest reward indicates the closest match between a user's interests and the content of a particular segment of the Web measured in terms of cosine similarity. WebWatcher also supports other functionalities such as searching the Web using a variant of the Lycos search engine, and monitoring the changes of some Web pages specified by the users. However, as the agent is not endowed with a persistent memory (i.e., a user profile) to capture a user's information needs, personalized browsing is not supported across different sessions.

Evaluation of WebWatcher was based on the $5,822$ browsing sessions targeting at the site of the computer science department at CMU logged between August 1995 and March 1996. Some of these sessions, which consist of users' interests and their traversal paths, were used to develop the training and the test data to evaluate the classification performance of WebWatcher. It was found that the RL method was slightly better than the ANNOTATE method. However, the best classification accuracy was obtained by combining these two methods using logistic regression. In addition, to compare the performance between the ANNOTATE method and some human experts, eight subjects conversant with the CMU Web site were asked to recommend the hyperlinks based on the pre-defined users' interests. The result showed that the classification accuracy of the ANNOTATE method (42.9%) was comparable with that achieved by human experts (47.5%).

Letizia [Lie95] is another Web browsing agent with functionality similar to that of WebWatcher. It recommends promising hyperlinks while a user is browsing the Web. However, Letizia does not require a user to explicitly specify their interests (i.e., queries) at the beginning of a session; instead, it uses a set of pre-defined rules to infer a user's interests. A query vector is then used to represent these interests. For instance, if the user creates a bookmark or saves a Web document, Letizia will infer that they are interested in that particular document. When Letizia encounters new hyperlinks, it will evaluate the annotations associated with the hyperlinks and the Web documents pointed to by these hyperlinks. If there is a sufficiently close match between a query vector and a document vector representing both the hyperlink and the Web document, the hyperlink will be recommended. Letizia differs from WebWatcher in that only Boolean features representing presence or absence of keywords in documents or queries are used. Moreover, the matching function between any two feature vectors is implemented as the dot product of the corresponding vectors. In other words, if there are a large number of overlapping keywords between a user's query vector and a document vector, the corresponding hyperlink is likely to be recommended. Similar to WebWatcher, Letizia is not endowed with a persistent memory to capture a user's recurring information needs. Therefore, the agent needs to learn from scratch if the user returns to the system the second time.

LIRA [BS95] and Fab [Bal97] are two adaptive information agent systems which employ the vector space model [SM83] for document representation and matching. In terms of the system architecture, these agents have persistent memories (i.e., user profiles) to capture each individual's information needs. Training these agents to learn and adapt to the users' changing information needs completely relies on the users' direct relevance feedback [SB90]. Unfortunately, this is a rather intrusive approach. Therefore, the agents' *learning autonomy* is relatively low. Once a set of documents is judged by the user, a variant of the Rocchio relevance feedback based learning method [Roc71] is used to revise the query vector. In particular, this variant considers positive examples only. The advantage of Fab over LIRA is that it employs both the content-based and the collaborative filtering strategies. An agent first learns a user profile for a particular topic. It is possible to share this profile with other users who are interested in the same topic. Apart from directly traversing the Web with a best-first search strategy, Fab is also equipped with interfaces to existing Internet search engines such as Alta Vista and Excite for information retrieval. As user profiles are used to capture users' long-term interests, personalized information delivery is supported by both LIRA and Fab.

2.2 The Naive Bayesian Paradigm

Syskill & Webert [PB97, PMB96] is an information agent designed to help users filter interesting Web pages of a particular topic. Essentially, each agent maintains a user profile corresponding to the user's topical information needs. A user is served by a set of Syskill & Webert agents with each one managing a particular topic. The Syskill & Webert agent develops queries based on the information stored in a user profile and then submits these queries to Internet search engines such as LYCOS. It ranks the relevance of the returned Web documents with respect to the user's topical information needs. Feature extraction is conducted by characterizing each Web document by a Boolean feature vector. This approach is similar to that employed in Letizia. A feature value corresponds to the presence or absence of a particular keyword in a Web document. The proposed *feature selection* method is based on the *expected information gain*, which picks features (i.e., keywords) with the best classification power from a set of training examples. A training example is a Web document together with a user's relevance judgement. The information content or Entropy $I(D)$ of a set of training examples D is derived from:

$$I(D) = -\sum_{c \in C} Pr(c) \times \log_2 Pr(c) \tag{2.3}$$

where $C = \{\text{relevant}, \text{non-relevant}\}$ is the set of classes. $Pr(c)$ represents the estimated probability that an arbitrary document d is with a class label c based on the observations from the training set D. *Expected information gain* is a method used in the ID3 algorithm to generate minimal decision trees [Qui86]. In the context of feature selection for IR, expected information gain $E(k, D)$ can be seen as a measure of the reduction of the uncertainty involved in classifying an arbitrary object $d \in D$ to class $c \in C$ based on the presence or absence of a keyword $k \in d$. Expected information gain is defined by:

$$E(k, D) = I(D) - [Pr(k) \times I(D^k) + Pr(\neg k) \times I(D^{\neg k})] \tag{2.4}$$

where $Pr(k)$ is the estimated probability that a keyword k appears in a document d, and $Pr(\neg k)$ is the estimated probability that an arbitrary document d does not contain the keyword k. The term $I(D^k)$ represents the information content of the set D^k of documents. Each document $d \in D^k$ contains the keyword k. The term $I(D^{\neg k})$ is the information content of the set $D^{\neg k}$ of documents with each $d \in D^{\neg k}$ not containing k.

The prediction model of Syskill & Webert is based on the naive Bayesian classifier. The advantage of this paradigm is its computational efficiency when compared with that of other more sophisticated paradigms. The objective is to predict if a Web document d is relevant given the fact that certain keywords are present in the document: $Pr(relevant|k_1 \wedge k_2 \wedge \ldots \wedge k_n)$. In general, the conditional probability estimates the chance that a document d is of class $c \in \{\text{relevant}, \text{non-relevant}\}$ given the fact that the set of features $(k_1 \wedge k_2 \wedge \ldots \wedge k_n)$ is found in d. If the features are independent, the posterior conditional probability $Pr(c|k_1 \wedge k_2 \wedge \ldots \wedge k_n)$ is *proportional* to the following probability function [DH73]:

$$Pr(relevant) \prod_{j=1}^{n} Pr(k_j|relevant) \tag{2.5}$$

where $Pr(k_j|relevant)$ is the conditional probability that a document d with a class label *relevant* contains the keyword k_j; this prior conditional probability can be estimated from D. In fact, Eq.(2.5) does not compute a conditional probability because the denominator $Pr(k_1 \wedge k_2 \wedge \ldots \wedge k_n)$ is not included. However, as the objective is to compare $Pr(\text{relevant}|k_1 \wedge k_2 \wedge \ldots \wedge k_n)$ with $Pr(\text{non-relevant}|k_1 \wedge k_2 \wedge \ldots \wedge k_n)$, using the numerators alone yields the same result as that of comparing the true conditional probabilities. The possible efficiency gain is important for real-time applications. The posterior probability $Pr(c|k_1 \wedge k_2 \wedge \ldots \wedge k_n)$ is approximated for each class $c \in \{\text{relevant}, \text{non-relevant}\}$. Then a document d is assigned to the class c with the highest conditional probability.

To evaluate the performance of the agent, four human experts were asked to judge Web documents over nine topics. The largest topic contained 154 Web documents with users' judgement and the smallest topic contained 26 documents with users' judgement. For each topic, the set of documents was divided into a training set and a test set. After training the agent with examples from the training set, the agent predicted the relevance of unseen documents from the test set. The classification accuracy of the naive Bayesian classifier was compared with other techniques such as the nearest neighbour algorithm, PEBLS, decision tree, the Rocchio method, and neural networks. The results showed that the naive Bayesian classifier outperformed some of the more sophisticated models across all nine topics. The best classification accuracy achieved in one of the nine topics was 81.5%. Attempts were made to employ semantic relationships among keywords to improve the agent's classification performance. The general lexical knowledge base WordNet [MRF+90] was used to remove irrelevant features from the training examples. The result confirmed that using the lexical knowledge could improve classification performance. Such an improvement is more obvious if only a small training set is available. Moreover, it was found that employing domain knowledge (e.g., lexical knowledge) and effective feature selection methods produced more significant performance improvement than that achieved by using an effective classification algorithm alone.

News Dude [BP99] is an adaptive news filtering agent on the Web. It employs a multi-strategy machine learning approach to filter Internet news. The agent's classification model is divided into a short-term model and a long-term model. The purpose of the short-term model is to classify incoming news stories into one of the recently retrieved news threads. The vector space model [SM83] is used for news representation and matching in the short-term model. On the other hand, the long-term model is used to represent a user's general preferences and to predict news which could not be classified by the short-term model. The long-term model is developed based on the naive Bayesian classifier. As a result, a news story may have two representations.

Firstly, it is represented by a TFIDF vector, and is compared with other TFIDF vectors which represent those recently seen stories using the cosine similarity measure. If there is a sufficiently close match, the incoming story will be classified to the corresponding news thread. On the other hand, if all the similarity scores are below the minimum threshold, the naive Bayesian classifier will be activated. Under such a circumstance, the incoming story is represented by a Boolean feature vector. The conditional probability $Pr(c|k_1 \wedge k_2 \wedge \ldots \wedge k_n)$ is computed to determine the category (i.e., class) $c = \{1, 2, \ldots, n\}$ representing one of the user's preferences. Apart from using a multi-strategy classification approach, the agent is able to explain and justify its decisions based on three pre-defined explanation templates. This is a distinct advantage of News Dude when compared with other adaptive information agents. The evaluation of News Dude is similar to that of Syskill & Webert. Ten users were asked to train the system over a period of four to eight days. About $3,000$ labelled news stories were obtained during this period. These stories were divided into a training set and a test set to evaluate the performance of News Dude in terms of classification accuracy and the F_1 measure comprising both the precision and the recall elements. It was confirmed that the multi-strategy classification model outperformed each individual classification method. With their particular experimental setting, the average classification accuracy of 72.5% and an average F_1 measure of 60.1% were achieved.

2.3 The Evolutionary Paradigm

Amalthaea [MM98] is a multi-agent ecosystem for information discovery, filtering, and monitoring on the Web. The agent gradually learns and adapts to a user's changing information needs based on the users' relevance feedback [SB90] and the agent's evolutionary mechanism. The evolution process is based on the principle of "natural selection". For instance, only the effective agents can survive and produce offspring in the system. Those agents which cannot produce relevant information to the users will be eliminated gradually. In Amalthaea, there is a clear distinction between the *discovery agents* which interact with external information sources such as Internet search engines, and the *filtering agents* which select and present the relevant documents to the users. Each user is in fact served by a group of discovery agents and a group of filtering agents respectively. The current implementation as published is a centralized server-based system [MM98]. Web documents are represented by keyword vectors with TFIDF weights. In other words, the procedure of feature extraction and selection is similar to that employed in WebWatcher [AFJM95, JFM97]. To estimate the inverse document frequency (IDF) factor, the set of locally cached documents is used to approximate the entire Web document collection. The weight of a keyword is adjusted based on whether it comes from the particular sections (e.g., header) of an HTML document. It is believed that keywords from the header section are better indicators about the content of the Web document, and so should be assigned higher weights. Web documents extracted from a user's bookmark file are used to represent the user's initial information needs. The user's information needs are represented by a TFIDF vector.

In Amalthaea, a filtering agent consists of two components, namely the *genotype* and the *phenotype*. Genotype is the element which will be modified by the evolutionary mechanism. Its main component is a TFIDF vector which represents one of the user's information needs. The phenotype of a filtering agent contains the non-evolvable elements such as the agent's fitness, date of creation, type of agent (e.g., user created or system generated), and executable codes. It should be noted that the meanings of the terms "Genotype" and "Phenotype" as adopted in Amalthaea are quite different from that normally being referred to in evolutionary computing. At the time of initialization, the set of TFIDF vectors representing a user's initial information needs is clustered into different topics. Within each cluster, a filtering agent is randomly assigned a TFIDF vector. Agent evolution is then performed on a cluster by cluster basis. If a filtering agent of a particular cluster presents a Web document to the user, a reward or penalty will be given dependent on whether the document is judged relevant or not by the user. The amount of the reward $+\delta$ or the penalty $-\delta$ is proportional to the agent's confidence $c(d)$ in its recommendation for a document d. The δ value is used to update the filtering agent's fitness f:

$$f_i = f_{i-1} + \delta_{i-1} - cost_{i-1} \tag{2.6}$$

where f_i is the filtering agent's fitness pertaining to the ith generation, and $cost_{i-1}$

is its living cost during the $(i-1)$ period. It is assumed that each agent has to pay for its survival in each evolution cycle. So, if a filtering agent does not recommend any document, it will die eventually. Each filtering agent employs the cosine similarity measure [SM83] $sim(\overrightarrow{q}, \overrightarrow{d})$ to estimate the correspondence between the user's information needs (i.e., a query q) and the content of an incoming document d. The confidence level of a recommendation is derived from:

$$c(d) = sim(\overrightarrow{q}, \overrightarrow{d}) \times f \tag{2.7}$$

where $c(d)$ is the agent's confidence level of recommending the document d, and f is the agent's current fitness. If the filtering agent recommends a Web document with a high confidence and the user's feedback is positive, it will receive a high reward δ. On the other hand, if $c(d)$ is high and the relevance feedback is negative, a large $-\delta$ will be generated. Consequently, filtering agents which consistently present relevant documents to the user will accumulate high fitness. Only a variable number of highly fit agents are chosen for reproduction in each evolution cycle. The number of agents allowed to go into the reproduction process is linearly related to the number of unfit agents to be eliminated from the system. The reproduction process involves three possible operations:

1. Cloning: creating multiple copies of the same agent in the new generation.

2. Two point crossover: randomly selecting two points from each keyword vector

and exchanges all fields within the chosen boundaries of the parents' vectors to generate two new keyword vectors.

3. Mutation: creating a randomly modified individual; the new mutated keywords are randomly selected from an agent belonging to another cluster or from a document recently judged as relevant by the user; the existing pairs of keywords and weights are randomly selected and replaced by the new pairs of keywords and weights.

There are two levels of agent evolution. Firstly, each individual's evolution is controlled by its fitness level. Secondly, the rate of evolution of the whole population is determined by the overall fitness measured by the average fitness of the entire population. In a particular evolution cycle, if the average document rating from the user is low, the number of agents going into the reproduction process will increase. In other words, the rate of adaptation is increased so that the agents can converge to the user's information needs quicker. The structure of a discovery agent is similar to that of a filtering agent. A discovery agent's genotype contains a search engine's URL, parameters for query construction, minimum hits, and maximum hits. If a discovery agent retrieves a document from a search engine, and this document is subsequently judged relevant by the user, a reward $+\delta$ is received from the filtering agent which recommends this document to the user. The same evolution process applies to the discovery agents. Those discovery agents which often retrieve relevant documents from the search engines are reproduced. Therefore, only the useful information sources with respect to the user's specific information needs are explored. Coordination among the filtering agents and the discovery agents is based on a shared buffer. The filtering agents place their requests (i.e., queries) in the buffer (i.e., a queue). A discovery agent selects a request to fulfil based on its work history with a particular filtering agent. If a filtering agent's queries often lead to an increase of the discovery agent's fitness, the filtering agent's query has a better chance to be served by the discovery agent.

Several experiments were conducted to evaluate the performance of Amalthaea. Some user profiles comprising rated documents were manually constructed. The documents were placed under different directories of a local machine to resemble the different search engines. At the beginning of an experiment, a set of users' information needs was randomly assigned to the filtering agents. A constant changing rate of 5% was applied to each profile. The result showed that Amalthaea could converge to the virtual users' information needs. However, on average it took around 200 agent evolution cycles to reach such an equilibrium because the agents were initially assigned random interests. Other experiments were developed to test if the system could adapt to abrupt changes or evolve based on less amount of direct relevance feedback. In both cases, Amalthaea could pick up a user's information needs after dozens of evolution cycles. Finally, seven users were organized to test the effectiveness and the usability of the system. During the testing period, the system's recommendations and the users' feedback were logged. In general, the users gave more positive feedback rather than negative feedback to the system. The mean absolute error of 22% was recorded under this particular experimental setting. According to their usability study, the majority of users felt that the agents could recommend relevant information. However, a mixed feeling about the adaptation capabilities of the agents was obtained.

2.4 The Computational Economy Paradigm

SIGMA [FK96, KF95, KF98] is a multi-agent system for filtering Usenet news on the Internet. The design objective of SIGMA is to integrate reinforcement learning, relevance feedback, and market equilibrium into a framework of *computational economy* which allows the agents to learn and adapt to both changes in the information sources and the changes in users' information needs. The problem of allocating limited resources among competing agents has been extensively studied in the field of economics. The metaphors of markets and pricing have been used to reach an equilibrium (i.e., optimal solutions) of resource distribution among the producers and the consumers. In the context of SIGMA, the resources are documents (e.g., news articles), and the consumers are the profile selector (PS) agents representing users' queries; the producers are the profile generator (PG) agents and the profile extractor (PE) agents. The PG agents purchase documents from the PE agents and then sell these documents to the PS agents. Document representation (i.e., feature extraction and selection) is conducted by the PE agents, and this is done based on the traditional IR techniques [SM83]. Each PE agent is responsible for characterizing documents from a particular news group.

Both the PS agents and the PG agents are endowed with user profiles which capture the users' queries. In addition, each PS agent keeps a history of documents purchased and the corresponding rewards received from its user. There are two levels of learning in SIGMA. At the local level, the profiles of the PS agents and the PG agents are updated based on the users' relevance feedback. In particular, the profile (i.e., a keyword vector with TFIDF weights) of a PG agent is revised by a variant of the Rocchio learning method [Roc71]:

$$\overrightarrow{Q}_{t+1} = \begin{cases} \overrightarrow{Q}_t + \beta \overrightarrow{d} & \text{if } d \text{ relevant} \\ \overrightarrow{Q}_t - \gamma \overrightarrow{d} & \text{if } d \text{ non-relevant} \end{cases} \tag{2.8}$$

where $\overrightarrow{Q}_t$ is the TFIDF vector representing a user's query at time t, and β and γ are the learning factors for the relevant and the non-relevant documents respectively. They are set to 0.9 and 0.1 in the TREC-7 experiments [KF98]. The term $\overrightarrow{d}$ is a TFIDF vector representing a document d judged by the user. Global learning of the entire system is based on the *pricing mechanism*. The main pricing mechanism is implemented through the PG agents. Each PG agent sets a standard price for all the documents that it sells at time $t+1$ based on the following function:

$$price_{t+1} = \mathcal{F}(e_t,\ b_t) \tag{2.9}$$

e_t is the average precision of the PG agent's output up to time t, and b_t is the ratio of the number of times the PG agent has been selected in a bid over the total number of bids quoted up to time t. Unfortunately, the details of the function $\mathcal{F}$ were not published [KF95]. However, the basic idea is that the PG agents purchase documents from the PE agents with a cost c. This cost could be fixed or floating (e.g. the cost can vary dependent on the information content or popularity of a news group). A PG agent's profit equals $\sum_{i=1}^{n} price_i - c_i$, where i represents each document bought and sold by the PG agent. If a PG agent continuously sells interesting documents to the PS agents and the users eventually provide positive feedback to these documents, the PG agent's precision increases. Accordingly, the PG agent can raise the prices of its items at a later stage. Assuming that the PS agents have sufficient budget to purchase items from this PG agent because it tends to produce interesting items, the cumulated profit of the PG agent increases.

On the other hand, if the PG agent does not purchase the right items from the PE agents, a loss is incurred because the PG agent has to pay for the cost of purchasing each item no matter if there is any PS agent to buy the item or not. Those PG agents which cannot produce relevant documents will eventually go *bankrupt*, and vanish in the market. Consequently, the SIGMA system can gradually converge to the users' information needs. Basically, the classification function of SIGMA is implemented based on the cosine similarity measure [SM83]. For instance, the PG agents purchase documents from the PE agents based on the cosine similarities between its profile vector and the document vectors supplied by the PE agents. To support exploratory learning, a probability value p is defined to control the PG agents to purchase documents with high cosine similarity scores from the PE agents, and another probability value $1 - p$ to allow the PG agents to purchase documents with low similarity scores. These documents represent the novel topics which were not explicitly requested by the users before. Moreover, each PS to PG market is formed based on the cosine similarity between the profile vectors of the respective agents. Evaluation of SIGMA was performed based on the adaptive information filtering task of the seventh annual TREC conference [KF98]. Unfortunately, the performance of SIGMA, in terms of the precision oriented F1 measure, is below the average as achieved by the majority participants in the TREC-7 adaptive filtering task. However, SIGMA is among the very few agent systems with a large scale and rigorous evaluation.

2.5 The Fuzzy Set Paradigm

GIRAF [MBVL99] is a fuzzy information filtering agent on the Internet. It utilizes fuzzy sets and genetic algorithms for classification and learning. Document representation, which comprises feature extraction and selection, is based on the traditional IR techniques [SM83]. For example, term frequency (TF) is used as a measure to select significant keywords from a document. At system initialization time, a user first chooses some relevant documents. The Parser module of the system extracts the average term frequency $tf(k)$ of each keyword k from the set of relevant documents. The set of documents judged by a user is stored in the form of document vectors in a local database. This database is continuously updated based on the user's relevance feedback [SB90]. A user's information needs are eventually represented by a population of chromosomes. Each chromosome comprises a set of genes, and each gene characterizes a *fuzzy information requirement* in terms of a keyword k and its average frequency c derived from the set of documents judged by the user. Basically, there are four types of genes. Each type of gene is characterized by a fuzzy membership function μ. Gene type g_1 represents an information item that the user requires (i.e., positive keyword); gene type g_2 represents an information item that should not be in a relevant document (i.e., negative keyword); g_3 is similar to g_1 but with a different membership distribution:

$$\mu_1(x) = \frac{\min(x,c)}{\max(x,c)} \qquad \mu_2(x) = \begin{cases} \frac{c-x}{c} & x < c \\ 0 & x \geq c \end{cases} \qquad \mu_3(x) = \begin{cases} \frac{x}{c} & x < c \\ 1 & x \geq c \end{cases}$$

where x is the term frequency of a keyword k in a document d. For example, the membership value $\mu_3(x)$ of a term k is 1 (i.e., a very positive keyword) if its term frequency in a document d is greater than or equal to the average frequency c that characterises a user's positive interest in the term k. Gene type g_4 takes into account the fact that a token's significance varies dependent on its location in a document (e.g., the first 10% of text, the last 10% of text, and the 80% of text in the middle). The membership value $\mu_4(x)$ is defined as the weighted OR-aggregation of the membership values $\mu_1(x)$ of a keyword k which appears in various sections of a document. This fuzzy genetic approach differs from the other genetic approaches in that the same keyword k may be applied to more than one gene type in the same chromosome or in different chromosomes because the concept of positive/negative keyword is fuzzy.

Matching incoming documents with respect to a user's information needs is conducted by computing the *population score* $Sim(d)$ of the current generation:

$$C_l(d) = \frac{1}{|G|} \times \sum_{i \in T, j \in G, k \in d} \mu_i^j(tf(k)) \tag{2.10}$$

$$Sim(d) = \frac{\sum_{l \in F} C_l(d)}{|F|} \tag{2.11}$$

where $C_l(d)$ is the *chromosome score* of a document d computed with respect to the chromosome l. Each keyword $k \in d$ is matched with the keyword of each gene from the set of genes G in the chromosome l. If a match is found, the term frequency *tf(k)* of the keyword k is used to compute the fuzzy membership value $\mu_i^j(tf(k))$ with respect to the corresponding concepts (e.g., positive keyword or negative keyword). The set T defines the allowable types of genes in the system. The set F contains the chromosomes with high fitness of a particular generation. A parameter ξ, which is defined in terms of a percentage of the current population, controls the cardinality of F. So, the population score $Sim(d)$ of a document d is defined as the arithmetic mean of the chromosome scores $C_l(d)$ derived from the best fit chromosomes $l \in F$.

The ability of a chromosome in classifying documents is called a payoff (pay) in GIRAF. In each evolution cycle, the current fitness f_i of a chromosome of a generation i is computed by adding the payoff to its previous fitness f_{i-1}, and subtracting the cost of living incurred during that period: $f_i = f_{i-1} + pay - cost$. The *cost* of living is a constant applying to the whole population so that those poor performers are gradually eliminated. Three methods are proposed to compute the payoff of a chromosome. The basic method is to compute the difference between the chromosome score of a document d and the user's relevance feedback for d: $pay_1 = 1 - |C_l(d) - U(d)|$, where $U(d)$ is the user's rating on d. To assign extra credits to high payoff values, another method is used: $pay_2 = 1 - [C_l(d) - U(d)]^2$. The last method is a weighted combination of pay_1 and pay_2: $pay_3^i = (pay_2^i)^\alpha \times [(Sim(d)_{i-1} - U(d)_i)^2]^\beta$, where pay_3^i and pay_2^i are the payoffs with respect to the current evolution cycle i; the difference between the population score $Sim(d)_{i-1}$ obtained from the $(i-1)$ evolution cycle and the user's rating $U(d)_i$ obtained from the current evolution cycle i represents the classification power of the chromosome at the $(i-1)$'th generation. The control parameters $\alpha \in [0,1], \beta \in [0,1]$ specify the significance of respective elements in computing the final payoff. Once the fitness of each chromosome is determined, the standard genetic operators such as *crossover* and *mutation* are applied to the individuals with fitness greater than a threshold so that both the *exploitation* and the *exploration* learning [Bal98] can take place. GIRAF differs from Amalthaea in that the size of its population is maintained constant. Whenever a new chromosome is born, another chromosome with the lowest fitness will be purged from the current population.

The agent was evaluated based on a virtual user profile. There were 13 test sets with each one containing five documents. The purpose was to examine the agent's capability of adapting to the changes, both smooth and abrupt, of the virtual profile. In particular, the correspondence between a set of system parameters and the agent's classification accuracy and rate of adaptation was studied. It was found that increasing the occurrence of gene types 3 and 4, lowering the crossover probability, and using the payoff function pay_1 improve classification accuracy. On the other hand, increasing the occurrence of gene type 1, raising the mutation probability, and using the payoff function pay_3 improve the agent's learning and adaptation ability.

2.6 The Connectionist Paradigm

InfoSpiders [MB00] and EVA [YKL00] are adaptive information agents endowed with neural networks and genetic algorithms for intelligent information retrieval. The discussion in this section focuses on InfoSpiders. The InfoSpiders agents directly traverse the Web to collect relevant information on behalf of their users. The design philosophy of InfoSpiders is that information agents can make use of the link topology on the Web to predict the (local) optimal traversal paths so that as many relevant Web pages are visited as possible. This assumption can be expressed as follows: $Pr[rel(d_2)|rel(d_1) \wedge link(d_1, d_2)] > Pr[rel(d_2)]$, where $Pr[rel(d_2)|rel(d_1) \wedge link(d_1, d_2)]$ is the probability that a Web document d_2 is relevant given that the agent is currently visiting a relevant document d_1 and there is a hyperlink from d_1 to d_2; $Pr[rel(d_2)]$ represents the probability that an arbitrary document d_2 visited by the agent is relevant. It is believed that the probability of retrieving a relevant document by following the hyperlinks from relevant documents is higher than that of performing a random walk over the Web. In InfoSpiders, Web documents are represented by keyword vectors. There are both local and global representations of a user's information needs. When the system is initialized, a user is asked to specify a set of significant keywords to characterize their information needs. Moreover, the user can also submit a bookmark file to the system. Based on the traditional IR techniques [SM83], a set of keywords are extracted to represent the user's initial interests. These keywords are weighted in the interval $[0, 1]$, and stored in the centralized *keyword table* of the system (i.e., the global representation). Each InfoSpiders agent can then access this table to determine a user's most current information needs.

In addition, each agent's *genotype* contains a Boolean feature vector $\overrightarrow{q}$, a neural weight vector, and a control parameter β. The parameter β specifies the significance of using the link topology to predict traversal paths. The Boolean feature vector is a local representation of the user's information needs. When an agent is initialized, keywords from the user's initial query are assigned to the agent. Furthermore, the agent is sent to one of the bookmarked pages as the starting point for Web traversal. The non-evolving component of the agent contains the configuration of a feed-forward neural network and other parameters which control the agent's evolution (e.g., energy level, mutation rate). In the simplest form, the feed-forward neural network is a single layer network (i.e., a perceptron). Basically, predicting which hyperlink to follow is based on the agent's neural network. A hyperlink is represented by a feature vector. Each feature value is the distance between the hyperlink and a surrounding keyword. In particular, the agent computes the distance values for each keyword defined in its genotype. The assumption is that a hyperlink is often surrounded by some words (i.e., annotations) which describe the nature of the document pointed to by the hyperlink. Based on the distance function $dist(k, l)$, the agent can estimate how closely a hyperlink l corresponds to its local representation of the user's interests stored in the query vector $\overrightarrow{q}$. For each keyword k in the agent's query vector $\overrightarrow{q}$, an input value in_{kl} for the corresponding input unit of the neural network is computed:

$$in_{kl} = \sum_{i:dist(k_i,l)\leq\rho} \frac{1}{dist(k_i, l)} \tag{2.12}$$

where k_i is the ith occurrence of k surrounding the hyperlink l in the current Web document d, and $dist(k_i, l)$ is a simple count of the intervening links from l up to a maximum window size of $\pm\rho$ links away. Each in_{kl} is then fed to the corresponding input unit of the neural network. The initial output of the jth unit is computed according to the integrator:

$$o_j = \tanh\left(b_j + \sum_{k\in q} w_{jk} \times in_{kl}\right) \tag{2.13}$$

where b_j is the jth unit's bias term; w_{jk} and in_{kl} are the jth unit's incoming weight and input respectively for each $k \in q$. Basically, the function of the integrator is to compute the weighted sum of the set of inputs in_{kl}. The final output of a unit is the activation value λ_j derived according to a logistic function f_j. This process is repeated for each hyperlink l contained in a document d. Finally, the retrieval agent employs a stochastic selector to select a link with the GIBBS probability distribution:

$$Pr(l) = \frac{e^{\beta\lambda_l}}{\sum_{l' \in d} e^{\beta\lambda_{l'}}} \tag{2.14}$$

where λ_l is the neural network's activation value for a link l, and $l' \in d$ represents one of the links in the current document. $\lambda_{l'}$ is the activation value of a link l' in the current document d. If the user provides relevance feedback $\phi(d) \in [-1, +1]$ for the document d pointed to by l, the feedback value can be used to update the agent's energy (i.e., fitness). The user's relevance feedback will also be used to update the centralized keyword table. For example, new keywords are added or the weights of existing keywords are updated. It is claimed that an InfoSpiders agent can perform local learning without the user's direct relevance feedback. Based on the keyword table, the relevance of a new document d pointed to by the chosen link l can be estimated by: $\phi(d) = \tanh\left(\sum_{k \in d} tf(k) \times w(k)\right)$, where $tf(k)$ is the term frequency (TF) of a keyword k in the document d normalized by document size; $w(k) \in [0, 1]$ is the weight of the keyword k recorded in the system's keyword table. If k is not found from the keyword table, its weight is zero. To prevent an agent from travelling the same path several times, the agent will not gain any energy from a visited document.

Moreover, there is a constant cost (i.e., energy deduction) of an agent's actions such as following a link, or reading the keyword table. Those agents with the energy level (i.e., fitness) below a threshold will be purged by the evolution process. Local learning in an agent takes place in the form of adjusting the input weights of its neural network. Essentially, a neural network is trained on-line based on the local context characterized by the hyperlinks and the documents surrounding the retrieval agent. After visiting a new document, the relevance estimation $\phi(d)$, generated by the system or provided by the user, is taken as a reinforcement signal to compute the teaching error:

$$\delta(d) = \phi(d) + \mu \times \max_{l \in L}\{\lambda_l\} - \lambda_d \tag{2.15}$$

where μ is a discount factor; L is the set of links of the document where the agent originally resides, and λ_d is the activation value of the hyperlink leading to a new document d. Based on $\delta(d)$, the neural network's weights are modified by using the standard back-propagation method. After this local learning, the agent can improve its prediction in the following moves.

Global learning in InfoSpiders has a significant impact on the agents' adaptive behaviour. The evolutionary approach is used to reproduce effective agents that traverse relevant Web documents, and eliminate those that perform poorly. The retrieval agents with high energy level (i.e., fitness) have a better chance to be selected for reproduction. Two-point crossover is used to generate new keyword vectors in the offspring. The parents' energies are then evenly distributed to their children. In addition, mutation is applied to an agent's keyword vector and neural weight vector. The neural vector is mutated by adding a random noise to a fraction of the neural weights. On the other hand, the keyword vector is mutated with a probability. In particular, the probability that a candidate keyword is selected to replace the least significant keyword in the keyword vector is proportional to its term frequency in a relevant document (e.g., the starting page of the agent) and its weight in the system's keyword table. Because of mutation, InfoSpiders can explore potential topics even though it might not be explicitly requested by the user before. The evolutionary mechanism ensures that the entire population of the retrieval agents will gradually converge to the user's interests.

Controlled experiments based on the Encyclopaedia Britannica (EB) were conducted in a closed environment. The purpose was to study if the retrieval agents could adapt to both the *spatial context*, an agent's ability to select significant features based on the surrounding linkage topology, and the *temporal context*, an agent's ability to absorb important features with respect to the user's information needs exhibited at different points of time. Furthermore, a small scale Web case study was conducted. Four Web pages were selected to represent a user's information needs. Moreover, a pre-defined query was submitted to the Excite search engine to establish the starting traversal points. Since the user's information needs were assumed constant, this case study only served to evaluate the agents' adaptation capability in a spatial context. A population of ten InfoSpiders agents was initialized and sent to the top ten Web pages returned from Excite. These agents autonomously traversed the Web and adapted to the surrounding linkage topology via automated reinforcement learning and agent evolution. There was no relevance feedback provided by the users. The result was that 66 Web pages had been visited and all the four relevant Web pages were found in 9 minutes. Although these experiments had a limited scale, they shed some light on the potential effectiveness of the InfoSpiders agents.

2.7 The Symbolic Paradigm

Quantitative approaches such as the vector space paradigm and the naive Bayesian paradigm have been applied to develop adaptive information agents. However, the weakness of these paradigms is that an agent's decision, based purely on a relevance score or a probability, is not sufficient to generate human comprehensible explanation of the agent's decision. Moreover, because of the deficiency in knowledge representation and reasoning (e.g., cannot reason about term associations), the agents' *learning autonomy* is also weakened. The symbolic paradigm has been explored for developing intelligent information agents [BPR+99, LRJ94]. Colombo [BPR+99] is a mobile agent for distributed information retrieval over the Internet. A user specifies their queries in terms of a set of weighted (e.g., in the interval $[0,1]$) keywords. These keywords are used to personalized the knowledge base of Colombo. For example, the following Prolog facts represent a user's interests about "Shakespeare" and "Hamlet":

$$keyword(shakespeare).$$

$$keyword(hamlet).$$

Web documents are characterized by a set of attributes (e.g., keywords) with the corresponding TFIDF weights [SM83]. Those keywords with TFIDF weights greater than a pre-defined threshold are selected to represent the Web documents. The Colombo agents represent knowledge about a user's preferences in terms of the weighted links between a set of query terms (i.e., keywords) and a set of attributes characterizing the document collection. This preference knowledge is represented as Prolog facts and rules:

$$link(K,\ [K, 1.0]).$$

$$link(shakespeare,\ [british\_drama, 0.8]).$$

$$link(football,\ [match, 0.6]).$$

The first Prolog clause states that if a user's query term (the first K) is the same as the attribute (the second K) characterizing a document, this link contributes a weight of 1.0 to the overall document score. The second clause represents the fact that the query term "shakespeare" is associated with a document characterized by an attribute "british-drama", and this association contributes a weight of 0.8 to the document score. The last Prolog clause says that the query term "football" is associated with the document attribute "match" and the association contributes a weight of 0.6 to the document score. It should be noted that the weights associated with a set of query terms and a set of document attributes may vary according to an individual's information preferences. At the time of initialization, the association weight of a link is set to 1.0 if a query term matches a document attribute; otherwise it is set to 0.5. Retrieving a document also requires the knowledge of the database agents. A database agent holds the TFIDF vectors of all the documents pertaining to a particular Web site. These vectors are represented as Prolog facts in the database agent's knowledge base:

$$good\_file(\text{“classic.html”},\ [british\_drama, 0.8]).$$
$$good\_file(\text{“classic.html”},\ [italian\_paintings, 0.8]).$$
$$good\_file(\text{“football.html”},\ [ball, 0.9]).$$
$$good\_file(\text{“football.html”},\ [match, 1.0]).$$

The first two Prolog clauses represent the document "classic.html" by the attributes "british-drama" and "italian-paintings". Both of these attributes (i.e., tokens) have the TFIDF weight of 0.8. The classification method of the mobile information agent system is essentially based on the *overlapping model* [BSW00]. For example, if a user is interested in "shakespeare", this interest matches the keyword element of the keyword to attribute link $link(shakespeare, [british\_drama, 0.8])$ in the Colombo agent's knowledge base. Moreover, as the attribute "british-drama" matches the attribute element of the document to attribute link $good\_file$("classic.html", $[british\_drama,\ 0.8]$) in the database agent's knowledge base, the document "classic.html" is retrieved with a document score computed based on the weights of the associated links. The more overlapping between the query terms and the attributes of a document, the better chance that the document is retrieved. This reasoning process is implemented as the formal deduction of the Prolog inference engine.

Learning and adaptation of the system is based on the users' relevance feedback. There are two possible learning models. The basic learning model is to directly modify the association weights between the query terms and the document attributes in a Colombo agent's knowledge base. For example, if the user's query term is "shakespeare" and the Web document characterized by the attribute "british-drama" is judged as relevant by the user, the weight of the link $link(shakespeare, [british\_drama, 0.8])$ will increase; otherwise its weight will decrease. In the advanced learning model, users' relevance feedback is used to generate a set of background knowledge (i.e., Prolog clauses) comprising links, keywords and attributes. By means of the techniques of Inductive Logic Programming (ILP) [BGNR96], a set of first-order rules are induced. This rule set can then be used to update the Colombo agents' knowledge bases. The same method can be applied to learn new knowledge for the database agents. It is believed that Prolog rules such as the following can be learnt using ILP [BPR+99]:

$$
\begin{aligned}
good\_file(File, User, Weight1 * Weight2) \quad &:- \quad keyword(User,\ K),\\
&link(K, Attribute, User, Weight1),\\
&relevance(Attribute, File, Weight2).\\
link(Keyword, Attribute, User, Weight) \quad &:- \quad domain(User,\ \text{“}uk''),\\
&equals(Keyword, \text{“}hamlet''),\\
&member(Attribute, [\text{“}theater'', \text{“}literature'']),\\
&equals(Weight, 0.8).\\
link(Keyword, Attribute, User, Weight) \quad &:- \quad equals(Keyword,\ \text{“}hamlet''),\\
&equals(Attribute, \text{“}music''),\\
&equals(Weight, 0.2).
\end{aligned}
$$

Unfortunately, neither the evaluation of the agent system nor the details of the ILP-based learning process was reported in the publication. It seems that further work is required to assess the effectiveness and the efficiency of the mobile information agent system.

2.8 The Associative Network Paradigm

INFOrmer [OS95, SOO97] is an adaptive information agent for filtering Usenet news. Feature extraction involves using a lexical analyser to tokenize the documents (i.e., news articles), extracting words, dealing with punctuation, and expanding acronyms. Then, sentence boundary disambiguation is performed to isolate individual sentences.

Feature selection is performed by using a pre-defined stop word list to remove high frequency words. Finally, a stemming algorithm is applied to strip inflectional and derivational word endings. After document pre-processing, significant phrases are extracted from the news articles. Associative networks are then used to represent the phrases extracted. An associative network is a special kind of semantic network; its edges represent the term association relationships only. There is no generalization nor specialization relationship in an associative network. The nodes in an associative network represent keywords, and the edges with attached weights connect keywords into phrases. The weights indicate the significance of the term associations. The advantage of the associative networks is that not only keywords and their frequencies are considered but also their context (e.g., a sentence) is captured.

A user's information needs are also represented by an associative network. Matching between a user's information needs and the incoming messages is conducted by comparing the structural similarities between the corresponding networks. Four types of graph comparison algorithms are used in INFOrmer [SDG+85]. Essentially, they are all based on the overlapping of neighbourhoods to measure the similarity between a pair of graphs. These algorithms only differ in the normalization methods used. For instance, the index of similarity for a common node in two graphs is computed as the cardinality of the intersection of the nodes' neighbourhoods divided by the cardinality of the union of the neighbourhoods. Let $A(V, E_1)$ and $B(V, E_2)$ be two graphs with a common node set V of cardinality n, their similarity is derived by:

$$Sim(A,B) = \frac{1}{n}\sum_{v \in V} \frac{|A_v \cap B_v|}{|A_v \cup B_v|} \tag{2.16}$$

where v is a node in the common node set V. The terms A_v and B_v are the sets of neighbourhoods identified from graph A and graph B respectively. An incoming document B is considered relevant with respect to the user's information needs A if the corresponding graphs demonstrate high structural similarity. In other words, the $Sim(A,B)$ value is greater than a pre-defined threshold. Learning in INFOrmer heavily relies on the user's relevance feedback. The news articles with the user's feedback are used to update the prototype associative network. A set of phrases representing a news item is first extracted. The weight of each word from the judged document is then used to modify the weight of a matching node in the prototype network. A variant of the Rocchio method is used for this purpose. The weight of each edge in the prototype network is updated by computing the arithmetic mean of the associated nodes. The system was formally evaluated based on a large document collection and the procedure of the routing task pertaining to the second TREC conference. It was claimed that the performance of INFOrmer, in terms of precision and recall, was comparable with other filtering systems participating in TREC-2. Nevertheless, no specific details of the computational efficiency of the agent system have been reported.

2.9 The Collaborative Filtering Paradigm

The adaptive information agents discussed so far are mainly based on the content-based IR approach. Basically, the agents characterize the content of documents and the users' queries by means of observable features (e.g., keywords). If these representations are similar, measured by a matching function, the incoming documents are deemed relevant by the agents. There is an alternative way for information retrieval. Ringo [SM95] is an adaptive information agent on the Web. It makes personalized music recommendations for its users. Instead of characterizing the description of an album or artist (i.e., a document) by means of its content, the agent represents and recommends items (e.g., albums, books, Web pages, etc.) via the "word of mouth" mechanism called *automated collaborative filtering*. The basic principle of the collaborative paradigm is that a user's interests are correlated with others based on their feedback pertaining to some items. Groups of *like-minded consumers* are then formed based on a similarity metric. To predict if a user will be interested in an item, the agent makes use of the preferences of other members in the same group to compute the preference index for the user. Users with similar interests are identified via the Pearson correlation coefficient $r(u_x, u_y)$:

$$r(u_x, u_y) = \frac{\sum_{i \in D}(u_{xi} - \bar{u}_x) \times (u_{yi} - \bar{u}_y)}{\sqrt{\sum_{i \in D}(u_{xi} - \bar{u}_x)^2} \times \sqrt{\sum_{i \in D}(u_{yi} - \bar{u}_y)^2}} \tag{2.17}$$

where $r(u_x, u_y)$ is the Pearson correlation coefficient between user u_x and user u_y.

The set D is commonly rated documents or products by both u_x and u_y. The term u_{xi} represents the user u_x's rating for an item $i \in D$. $\bar{u}_x$ is u_x's average rating for all the items in D, whereas $\bar{u}_y$ is u_y's average rating for all the items in D. This kind of pair-wise comparison is conducted for each pair of users. For a pair of users u_x and u_y, if $r(u_x, u_y)$ is above a pre-defined threshold, they will be considered in the same group. To predict if a particular user u_x is interested in an item d, the agent refers to the ratings of that item given by other members in the like-minded user group:

$$pred(u_x, d) = \bar{u}_x + \frac{\sum_{u_y \in U}(u_{yd} - \bar{u}_y) \times r(u_x, u_y)}{\sum_{u_y \in U} |r(u_x, u_y)|} \tag{2.18}$$

where $pred(u_x, d)$ is the agent's prediction for user u_x's rating of an item d. In other words, it is the agent's prediction of how much the user will like or dislike the item d. The set U contains all the nearest neighbours of the user u_x; the term u_{yd} represents u_y's rating for an item d, and $\bar{u}_y$ is u_y's average rating for all the items. The term $|r(u_x, u_y)|$ is the absolute value of the Pearson correlation coefficient between a pair u_x and u_y. According to $pred(u_x, d)$, the information agent can rank all the items which have not been seen by the user u_x before. Moreover, if the predicted rating of an item d is above a system threshold, the agent can recommend this item to the user.

With the collaborative paradigm, a document is represented in terms of the preference values (i.e., ratings) of a group of users. A user's information needs are represented by their own preference values (ratings) for some items. Classifying an item into one of the classes (i.e., one of the preference values) is based on the user's average rating, other nearest neighbours' ratings of the same item, and the *correlation* between the ratings of the user and that of their nearest neighbours. The information agent is adaptive by taking into account a user's changing ratings for some items and the correlation between the user's ratings and others ratings for the items. The changes of the correlation values trigger the agent to generate different recommendations. In general, this kind of leaning is not incremental because each correlation value between a user and another member in a group needs to be computed again if the user changes their ratings for an item. One advantage of the collaborative paradigm is its simplicity in terms of representing items and users' preferences.

Evaluation of Ringo was performed based on the rating data of the 1876 artists from 1000 users. The data is divided into a training set (80%) and a test set (20%). Several variants of the Pearson algorithm were compared with the mean squared differences algorithm in terms of the mean absolute error and the standard deviation of error. To produce recommendations to a user, each algorithm is used to compute the correlation between the user and another member in a group (i.e., the like-minded group). All users whose correlation coefficient greater than a threshold were identified, and the weighted average of their ratings were used to generate the agent's predictions. It was shown that the constrained Pearson algorithm, which used a chosen value instead of the mean rating value of a user to distinguish positive correlation from negative correlation between a pair of users, performed best. When the similarity threshold was set to 0.6 to train the agent, 94% of the ratings in the test set could be predicted. Feedback from the 2,000 users who used Ringo during the usability study period was collected. It was found that some users were initially disappointed by the recommendations of the agent. However, as the number of ratings grew, more positive feedback was received from the users. There are other collaborative information agents which recommend Internet news [GSK+99, RNM+94], research papers [DIU98], or Web pages [GCS98, LDV99] to individuals.

2.10 Analysis of the State of the Art

Tables 2.2 and 2.3 summarize the information pertaining to the adaptive information agents discussed in this chapter. It aims at a systematic analysis of the pros and cons of the various adaptive information agent paradigms. The characteristics of document representation, profile representation, feature selection, classification methods, and the impact of these features on the agents' explanatory power are tabulated in table 2.2. Moreover, issues such as the agents' learning methods, the agents' capabilities of processing implicit feedback, and the impact of these issues on the agents' exploratory capabilities, learning autonomy, and the modes of learning (e.g., incremental vs. non-incremental) are tabulated in table 2.3. TF stands for term frequencies, and TFIDF stands for term frequency inverse document frequency. In some systems, different granularity of representation are used. For example, both the TFIDF vectors and the abstraction of chromosomes are used to represent users' information needs in Amalthaea. Only the high level abstraction such as chromosomes are shown in Tables 2.2.

Agent	Paradigm	Document Rep.	Context Rep.	Feature Selection	Classifying Method	Explanatory Power
WebWatcher	Vector Space	TFIDF vectors	TFIDF vectors	TFIDF	Cosine similarity	Low
Letizia	Vector Space	Boolean vectors	Boolean vectors	TF	Dot product	Fair
LIRA	Vector Space	TFIDF vectors	TFIDF vectors	TFIDF	Cosine similarity	Low
Fab	Vector Space	TFIDF vectors	TFIDF vectors	TFIDF	Cosine similarity	Low
Syskill & Webert	Naive Bayesian	Boolean vectors	Boolean vectors	Information gain	Naive Bayesian	Low
News Dude	Naive Bayesian	TFIDF vectors+ Boolean vectors	TFIDF vectors+ Boolean vectors	TFIDF + Information gain	Cosine similarity + Naive Bayesian	Fair
INFOrmer	Associative Network	Associative networks	Associative networks	Stop word list	Graph comparison	Low
Amalthaea	Evolutionary	TFIDF vectors	Chromosomes	TFIDF	Cosine similarity	Low
GIRAF	Fuzzy Sets	TF vectors	Fuzzy chromosomes	TF	Membership functions	Low
InfoSpiders	Connectionist	Weighted TF vectors	Neural networks	TF	Neural networks	Low
Colombo	Symbolic	Prolog clauses	Prolog clauses	TFIDF	Formal deduction	Fair
SIGMA	Computational Economy	TFIDF vectors	TFIDF vectors	TFIDF	Cosine similarity	Low
Ringo	Collaborative	Users' ratings	Correlation matrices	not applicable	Correlated mean ratings	Low

Table 2.2: Analysis of Adaptive Information Agents (representation & classification)

First generation adaptive information agents such as WebWatcher [AFJM95, JFM97], LIRA [BS95], Fab [Bal97], Letizia [Lie95] utilize weighted (e.g., TFIDF or

Agents	Implicit Feedback	Learning Methods	Exploratory Learning	Incremental Learning	Learning Autonomy
WebWatcher	No	Linear correlation	No	No	Low
Letizia	Yes	Inference rules	No	No	Moderate
LIRA	No	Rocchio variant	No	Yes	Low
Fab	No	Rocchio variant	No	Yes	Low
Syskill & Webert	No	Bayesian learning	No	No	Low
News Dude	No	Concept feedback	No	Yes	Moderate
INFOrmer	No	Rocchio variant	No	Yes	Low
Amalthaea	No	Genetic algorithms	Yes	Yes	Moderate
GIRAF	No	Genetic algorithms	Yes	Yes	Moderate
InFoSpiders	No	Back propagation	Yes	Yes	Moderate
Colombo	No	ILP	Yes	No	Moderate
SIGMA	No	Market equilibrium	Yes	Yes	Moderate
Ringo	No	Linear correlation	Yes	No	Low

Table 2.3: Analysis of Adaptive Information Agents (Learning)

Boolean) vectors to represent documents and user's information needs. Classification is conducted by computing the cosine angles or the dot products of these vectors. These techniques have been extensively studied in the field of IR and are generally considered efficient and effective [SM83]. However, the implicit assumption of term (e.g., keyword) independence in these models is not able to capture the realities in IR where information items are often related to each other. For instance, if one is interested in documents about "automobile", it is desirable that an information agent can automatically infer that a document about "car" is relevant because "automotive" and "car" are related by the synonym relationship [Hun95]. The term independence assumption not only affects the agents' *classification effectiveness* but also their *learning autonomy* since the users need to provide direct relevance feedback to train the agents. In terms of the learning autonomy among the first generation adaptive information agents, Letizia is prominent because it can utilize pre-defined rules to infer the users' information needs rather than asking them to provide direct feedback.

The advantage of the Rocchio learning method is that it is an incremental learning mechanism. Nevertheless, it lacks the power of *exploring* new information topics as the learned prototypical vectors only describe the documents previously viewed by the users. This is the so-called *serendipity* problem [MM98]. In general, it is more desirable to have a balance between exploitation oriented and exploration oriented learning [Bal98]. Moreover, many agents in this category are weak in terms of their *explanatory power*. Justification of an agent's information retrieval decision is purely based on a similarity score or probability value. This weakness is an obstacle of developing trust between the agents and their users because the users cannot fully understand the decision making behaviour of the agents. It has been pointed out that the issue of users' trust on information agents has a significant impact on the practical applications of these agents [MM98]. Moreover, some of the first generation adaptive information agents such as WebWatcher and Letizia are not endowed with persistent memories to hold users' recurring interests. Consequently, proactive and personalized recurring IR is not supported by these agents.

The naive Bayesian paradigm suffers from problems similar to that of the vector space paradigm since it is also based on the naive assumption of feature independence. Moreover, the conditional probabilities alone may not be sufficient to generate comprehensible explanations of the agents' decisions. Since the conditional probability that a document is relevant given the presence of certain features is computed solely based on the previously seen documents, the information agents are not learning to *explore* novel information topics. In addition, the mode of learning is not incremental because all the conditional probabilities need to be computed again if new training examples are added to or deleted from the user profiles.

The computational economy paradigm found in SIGMA is one of the early attempts to address the issue of multi-agent learning and co-ordination in the context of IR. The intuition behind this paradigm is that there are uncertainties about a user's information needs. Through a *computational market*, these uncertainties are represented (e.g., by the diversity of agents with each one capturing a possible information need) and processed (e.g., via the pricing mechanism). This paradigm may be an alternative to the evolutionary paradigm which is based on genetic algorithms.

The evolutionary paradigm has been explored in many contemporary models of adaptive information agents, whereas the computational economy paradigm is yet to be further developed and evaluated. The price mechanism as reported in the literature is incomplete [FK96, KF95, KF98]. For example, the PE agents' marginal profit thresholds, bankruptcy threshold, the algorithmic details of the pricing function $\mathcal{F}$, the consumers' budgets, etc. are not illustrated thoroughly. In addition, the interaction between the price mechanism and the vector space model probably requires further refinement. More recent work pertaining to the computational economy paradigm demonstrates the continuous development of this paradigm for IR [WFG01].

The collaborative paradigm [GCS98, RNM$^+$94, SM95] offers the advantages of a handy document representation, a better balance between exploitation and exploration oriented learning, and efficient classification. However, this paradigm alone has not been widely used to build adaptive information agents. One of the reasons is the *sparse rating* problem [BP98]. For a highly dynamic domain such as the Web, it is difficult, if not totally impossible, to collect sufficient ratings from the users for a significant number of items such as Web documents. Some empirical studies have shown that the collaborative paradigm alone is not as effective as combining the content-based and the collaborative paradigms for information retrieval [DIU98, SSH99]. Fab [Bal97] and RAAP [DIU98] are among the information agents which employ a hybrid model of the collaborative and the content-based approaches to improve the agents' effectiveness. In general, the collaborative paradigm demonstrates non-incremental learning behaviour since the correlation data between a user and each member in a group must be recomputed if the user's rating for a single item is changed. The *learning autonomy* is low because this paradigm heavily relies on the users' direct feedback. Moreover, one practical issue of applying this paradigm to information agents is that users may not want to share their preferences with other people because of the privacy concern.

It has been a trend to apply genetic algorithms to develop the learning mechanisms of adaptive information agents [YKL00, MBVL99, MB00, MM98]. Some of these agents such as InfoSpiders [MB00], EVA [YKL00], and GIRAF [MBVL99] actually demonstrate a synergy between different paradigms. This paper describes GIRAF under the heading of the Fuzzy set paradigm and InfoSpiders under the heading of the Connectionist paradigm because the corresponding paradigms seem to best describe the dominating techniques in these agents. In general, the notion of "chromosome" is used to represent a user's distinct information need. A gene on a chromosome represents the presence or absence of a particular keyword. Based on the genetic operators such as cloning, crossover, and mutation, a better balance between *exploitation-oriented* and *exploration-oriented* learning in the high dimensional information space is achieved. It is a kind of *incremental learning* because a new population of information agents is gradually evolved from previous generations. The principle of *natural selection* ensures that effective agents measured by a fitness function will gradually dominate the entire population. Therefore, retrieval performance of the agents is improved over time. The evolutionary paradigm and the computational economy paradigm share some common properties. On the one hand, they both rely on an evolution mechanism. For instance, the pricing policy in the computational economy paradigm and the principle of natural selection in the evolutionary paradigm are enforced in the adaptive information agents so that they gradually converge to the user's information needs. On the other hand, both of these paradigms are also faced with the challenge of *responsive learning*. It may take a while (e.g., dozens of evolution cycles) for the agents to completely absorb users' new interests into the corresponding user profiles. However, with the help of the genetic operators, it seems that the evolutionary paradigm is stronger, in terms of the exploratory power, than the computational economy paradigm.

Various fitness functions have been used in adaptive information agents. These fitness functions heavily rely on user's relevance judgements. Accordingly, a large amount of direct human intervention is still required to train the agents. Therefore, the *learning autonomy* of the evolutionary paradigm is only moderate. Both InfoSpiders [MB00] and EVA [YKL00] distinguish local learning from global learning, and support *automated relevance feedback*. The basic idea is that the results of a local classification are compared with a global representation of a user's interests. Then, relevance feedback is automatically generated based on these comparisons. For example, if there is a sufficiently close match between the local classification result and the global information needs, a positive relevance feedback is generated; otherwise negative feedback is produced. The problem is that the global representation of a user's interests still heavily relies on the user's relevance feedback to bring it up-to-date; otherwise the automated feedback mechanism will fail.

Genetic operations such as *mutation* enhance exploratory learning. However, it may have a negative impact on the agent's classification effectiveness and explanatory power because irrelevant or strange information needs could be composed during the mutation process. Finally, the evolutionary paradigm requires the development of a set of evolution parameters such as fitness threshold, fitness function, crossover rate, mutation rate, population size, etc. A thorough methodology is not available to guide the development of the genetic parameters. Consequently, two different agents employing similar evolutionary operators may demonstrate quite different learning and adaptation behaviour. One of the challenges of applying the evolutionary paradigm to adaptive information agents is to develop a more disciplined way of establishing the evolutionary parameters.

It is intuitively appealing to apply the concept of fuzzy sets to develop the classification models of information agents because the concept of relevance is vague. The focus of this paradigm is on improving the *classification effectiveness* based on the fuzzy membership functions. With the GIRAF agents [MBVL99], three basic types of *membership functions* are pre-defined and they are assumed valid in all retrieval situations. However, the concept of relevance is more likely dependent on a local context [Law00, XC96]. Therefore, the challenge of applying the fuzzy set paradigm to information agents is to develop an automated means of dynamically learning the fuzzy membership functions based on the local document collections and users' relevance feedback. Another issue is how to generate human comprehensible explanations of the agents' decisions based on the underlying fuzzy membership functions.

The associative network paradigm allows primitive semantic relationships among information items to be represented in information agents. In principle, this approach may improve the agent's learning autonomy, exploration power, and classification effectiveness. However, representing documents and retrieval contexts by graphs, and computing their similarities based on the structural characteristics of the graphs is computationally expensive. Even though a graph can help visualize the semantic relationships between tokens (e.g., keywords), it may not be easy for novice users to understand the agent's decisions based on the structural similarities of graphs. In terms of the learning autonomy of INFOrmer [OS95], the users still need to provide a considerable amount of direct relevance feedback to revise the associative networks. Moreover, more empirical studies are required to prove the scalability of the associative network paradigm.

The connectionist paradigm has been successfully applied to many real life applications. It offers the advantage of automatically learning non-linear classification functions [YKL00]. Representing IR matching functions by the non-linear relationships between features (e.g., keywords) and document classes is a sound approach. Therefore, the connectionist paradigm is a viable alternative for improving the agents' *classification effectiveness* when compared with the fuzzy set paradigm. Although only supervised learning (e.g., back propagation) is explored in InfoSpiders [MB00], unsupervised training algorithms for artificial neural networks are available [Bar89a].

So, the connectionist paradigm has the potential of enhancing the *learning autonomy* of adaptive information agents. In addition, learning in neural networks is *incremental*. It should be noted that only the local learning method of InfoSpiders is depicted in table 2.3 because this method is relevant to the connectionist paradigm. However, in general, the computational complexities associated with artificial neural networks are high. More empirical studies are required to test the scalability of this paradigm for on-line information agents. On the other hand, it is difficult, if not completely impossible, to generate human comprehensible explanations of the agent's decisions solely based on the network configurations and the weights of neurons. Research in knowledge extraction from neural networks sheds light on generating high level rules to explain the agent's decisions [Tsu00].

Contemporary models of adaptive information agents focus on the agent's *knowledge representation*, *classification effectiveness*, *learning autonomy*, *explanatory capability*, and the balance between *exploitation oriented* and *exploration oriented* learning. It has been observed that employing domain knowledge such as lexical knowledge and contextual information can substantially improve the agent's classification effectiveness [ACL$^+$00, Law00, PB97]. The agents' abilities to represent and reason about complex retrieval contexts are particularly important because it is unrealistic to assume that the users will spend a lot of time and effort to train these agents before the agents are expected to retrieve relevant information autonomously. A rich representation of a retrieval context can also enhance an agent's explanatory power, and hence improve the user's trust in using the agents. It is believed that the explanation mechanisms of information agents can actually speed up the agent's learning [BP99].

The symbolic agent paradigm seems promising for the development of the next generation of adaptive information agent systems. The expressive power of logic allows complex retrieval contexts to be captured in information agents. Based on enriched representations of retrieval contexts, information agents can use sound and robust inference mechanisms to enhance their learning autonomy and proactive IR behaviour. Above all, the agents can justify their decisions based on the formal reasoning frameworks. However, logic-based system is in general computationally expensive. This is one of the major obstacles for applying sound logical frameworks to build practical applications. Therefore, apart from the development of a sound and robust logic-based information agent model, it is essential to implement and evaluate such a model to examine if the model can scale up for IR applications with a realistic scale.

Existing symbolic information agents such as Colombo is weak in demonstrating its ability to deal with realistic IR requirements since rigorous evaluations of these agents are missing. Moreover, for the Colombo agent system, it seems that its classification model is mainly based on the *overlapping* IR model which is known to be ineffective [BSW00, Rij86]. The symbolic inference power seems not fully utilised. Moreover, how to learn a retrieval context in general and a user's information needs in particular is not illustrated with sufficient details. Unfortunately, this issue is crucial to the success of a symbolic information agent model. The following chapter will discuss a rigorous symbolic framework, which is based on the sound AGM belief revision logic, for the development of an effective IR model. Such an IR model underpins the learning and the classification functions of adaptive information agents. The computational aspects of the proposed belief revision based adaptive information agents will then be illustrated in Chapter 4.

Chapter 3

Belief Revision and Expectation Inference

This chapter explains the intuition behind the AGM belief revision paradigm and illustrates the implementation of the AGM belief functions. A new transmutation-based strategy for implementing the AGM change functions is proposed. Moreover, the interconnection between belief revision and nonmonotonic inference is discussed. Finally, how the AGM belief functions and the related expectation inference relations are applied to adaptive information agents is examined at the conceptual level.

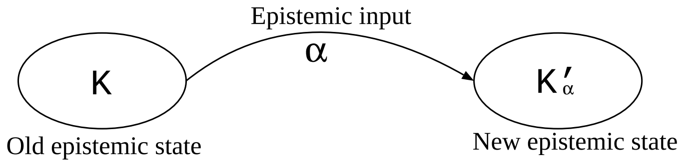

Figure 3.1: Transition Between Epistemic States

3.1 The AGM Belief Revision Paradigm

The AGM belief revision framework is coined after its founders Alchourrón, Gärdenfors, and Makinson [AGM85]. It is one of the most influential works in the theory of belief revision. The AGM framework provides a rigorous formal foundation for modelling the changes of beliefs in rational agents. A belief change in an agent is viewed as a transition from an epistemic state K to a new epistemic state $K^{'}_{\alpha}$ with respect to the new epistemic input α as depicted in Figure 3.1. The AGM principle ensures that the new epistemic state remains *consistent* and modified in a *minimal* way after an epistemic state transition (revision). Whether a *foundational* approach such as the Assumption-Based Truth Maintenance System (ATMS) belief revision [dK86] or a *coherent* approach such as the AGM belief revision [AGM85] should be used to model belief changes in rational agents has undergone a long debate [Gär90]. However, it has been shown that it is possible to simulate the behaviour of the ATMS using the AGM approach by encoding the foundational beliefs as an epistemic entrenchment ordering [DF93]. It has also been proven that these two main paradigms of belief revision (i.e., foundational or coherent) are mathematically equivalent [dV97]. For instance, every belief revision operator that can be defined by the foundational approach can also be defined by the coherent approach and vice versa. Nevertheless, the AGM paradigm is a pure logical approach. Therefore, formal reasoning can be conducted within the same system. For the ATMS based system, logical reasoning needs to be carried out by a separate problem solver. To achieve a seamless integration between the learning and the matching components of the proposed adaptive information agents, the AGM approach seems intuitively more attractive. Moreover, the AGM belief functions can also be used to revise the IR contextual information (e.g., association and preclusion rules) into an agent's knowledge base, whereas the assumptions maintained by an ATMS system must be literals. In fact, belief revision has been taken as a learning paradigm and the *learning power* of various belief revision formalisms has been formally studied [Kel98]. With all these reasons, the AGM belief revision framework is exploited to develop the learning components of adaptive information agents.

In the AGM belief revision framework, the notion of *belief sets* was introduced to represent epistemic states in rational agents [Gär88].

Definition 1 *A set of sentences K is a non-absurd belief set iff:*

(1) $K \nvdash \perp$, and

(2) $K \vdash \alpha$ implies $\alpha \in K$.

The consequence relation $\vdash$ is defined with respect to an object language. In general, $K \vdash \alpha$ means that a set of sentences K of the object language logically entails a sentence α of the same language. Usually this object language refers to a propositional language $\mathcal{L}$ closed under the usual Boolean connectives $\neg$, $\rightarrow$, $\leftrightarrow$, $\wedge$, and $\vee$ [AGM85, GM88, GM94]. The two sentential constants $\top$ (truth) and $\bot$ (falsity) of $\mathcal{L}$ are also used. The background logic is defined by its consequence operation Cn which satisfies the following conditions:

`Inclusion:`	$\Gamma \subseteq Cn(\Gamma)$
`Iteration:`	$Cn(Cn(\Gamma)) = Cn(\Gamma)$
`Monotonicity:`	$Cn(\Gamma) \subseteq Cn(\Gamma^{'})$ whenever $\Gamma \subseteq \Gamma^{'}$
`Supraclassicality:`	$\alpha \in Cn(\Gamma)$ if Γ classically implies α
`Deduction:`	$\beta \in Cn(\Gamma \cup \{\alpha\})$ iff $(\alpha \rightarrow \beta) \in Cn(\Gamma)$
`Compactness:`	If $\alpha \in Cn(\Gamma)$ then $\alpha \in Cn(\Gamma^{'})$ for some finite $\Gamma^{'} \subseteq \Gamma$

where Γ and $\Gamma^{'}$ are sets of sentences of $\mathcal{L}$ and α and β are sentences of $\mathcal{L}$. Therefore, the consequence relation $\Gamma \vdash \alpha$ means $\alpha \in Cn(\Gamma)$. The set of logical consequences of Γ is $Cn(\Gamma) = \{\alpha : \ \Gamma \vdash \alpha\}$. With reference to the definition of non-absurd belief sets (or simply refers to as belief sets in this thesis), the first property states that a belief set must be consistent. The second property specifies that a belief set is closed under logical consequence. Therefore, a belief set K is essentially a *theory* of $\mathcal{L}$. The set of sentences K represents the information (e.g., propositions) that a rational agent believes. The notation $K_\perp$ denotes an absurd belief set. According to the AGM principle, every effort should be made to prevent the transformation from K to $K_\perp$ because rational agents do not entertain absurd epistemic states. The transition between any two epistemic states K and K'_α as depicted in Figure 3.1 can be modelled by a change function from $\Gamma \times \mathcal{L}$ to Γ. In other words, the processes of belief revision are modelled by some change functions which transform a theory Γ of $\mathcal{L}$ with respect to a formula α to another theory Γ^*_α. In the AGM framework, three types of belief state transitions are identified and they are modelled by the corresponding *belief functions* $F : K \times \mathcal{L} \mapsto K$:

Expansion (K^+_α) is the process of accepting a new belief α that does not contradict existing beliefs in a belief set K (i.e., $\alpha \notin K,\ \neg\alpha \notin K,\ \alpha \in K^+_\alpha$). This is a straightforward operation of incorporating the new information α and its logical consequences into the belief set K;

Contraction (K^-_α) is the removal of a belief α and all other beliefs that logically imply α from a belief set K (i.e., $\alpha \in K,\ \alpha \notin K^-_\alpha$);

Revision (K^*_α) is the incorporation of a belief α that may contradict existing beliefs in a belief set K (i.e., $\alpha \notin K,\ \neg\alpha \in K,\ \alpha \in K^*_\alpha$).

Unlike the expansion functions, both the contraction functions and the revision functions cannot be uniquely defined purely based on set oriented operations. There are close relationships among the belief functions. For instance, the processes of belief revision can be derived from the processes of belief contraction via the *Levi Identity* [Lev77]:

$$K^*_\alpha = (K^-_{\neg\alpha})^+_\alpha$$

The Levi identity states that a belief revision operation K^*_α is equivalent to first contracting the negation of α from K (i.e., a contraction operation) followed by an expansion operation of adding α to the belief set K. Moreover, a belief contraction function can also be defined in terms of a belief revision function via the *Harper Identity* [Har77]:

$$K^-_\alpha = K \cap K^*_{\neg\alpha}$$

The Harper identity says that a belief contraction operation K^-_α is equivalent to the set intersection of the original belief set K and the result of the belief revision operation $K^*_{\neg\alpha}$ which revises K with respect to $\neg\alpha$. Essentially, the AGM framework includes sets of postulates to characterise well-behaved belief functions and various methods such as *epistemic entrenchment orderings*, *selection functions* on belief sets, *systems of Spheres*, etc. to construct the change functions [Gär88]. The AGM postulates for expansion, contraction, and revision attempt to identify classes of change functions for modelling the manner in which a rational agent should alter its beliefs in face of changes. Let $\mathcal{K}$ represent the set of all non-absurd belief sets. The AGM postulates for belief contraction are defined as follows:

(K^-1) $K^-_\alpha \in \mathcal{K}$ (Closure)

(K^-2) $K^-_\alpha \subseteq K$ (Inclusion)

(K^-3) If $\alpha \notin K$, then $K^-_\alpha = K$ (Vacuity)

(K^-4) If $\nvdash \alpha$, then $\alpha \notin K^-_\alpha$ (Success)

(K^-5) If $\alpha \in K$, then $K \subseteq (K^-_\alpha)^+_\alpha$ (Recovery)

(K^-6) If $\vdash \alpha \leftrightarrow \beta$, then $K^-_\alpha = K^-_\beta$ (Preservation)

(K^-7) $K^-_\alpha \cap K^-_\beta \subseteq K^-_{\alpha \wedge \beta}$ (Conjunction)

(K^-8) If $\alpha \notin K^-_{\alpha \wedge \beta}$, then $K^-_{\alpha \wedge \beta} \subseteq K^-_\alpha$ (Selection)

The first contraction postulate simply states that a contraction operation maintains the property of non-absurd belief set for the belief set involved in the change. The second postulate indicates that no new belief should be included into a belief set K after a contraction operation. The third postulate implicitly applies the *informational economy* principle to the belief contraction processes. For instance, if the information to be contracted is not contained in a belief set (i.e., $\alpha \notin K$), the information content of the belief set should remain the same after a contraction operation. The fourth postulate defines the successful criterion of a belief contraction operation. After a contraction operation, the contracted belief α will not be a logical consequence of the resulting belief set if α is not valid (i.e., $\alpha \notin K^-_\alpha$). The fifth postulates states that all the beliefs in K can be recovered after contracting a belief α if the contracted belief set K^-_α is expanded with respect to the same belief afterwards. (K^-6) says that the results of two belief contraction operations will be the same if the same belief set is contracted with respect to two logically equivalent sentences. (K^-7) and (K^-8) explain the nature of contraction with respect to a conjunction of sentences. The result of contracting a belief set K with respect to the conjunction of two sentences α and β contains all the beliefs that are in both K^-_α and K^-_β. Moreover, the contraction of K with respect to α and β results in either α or β (or both) being removed. This postulate actually reinforces the principle of informational economy. For instance, the minimal change to $K^-_{\alpha\wedge\beta}$ may be achieved by just removing either α or β dependent on which belief is more important to an agent.

The AGM postulates for belief expansion, contraction, and revision functions define the classes of change functions which adhere to the rationales of *consistent* and *minimal* belief changes. However, these postulates do not provide the necessary information to develop the corresponding functions. Extra information is required to uniquely define a contraction or a revision function. One of the ways to construct the AGM change functions is by *epistemic entrenchment* ($\leqslant$) [GM88]. The epistemic entrenchment relation is defined over the sentences of $\mathcal{L}$, and is relative to a belief set K. For instance, if α, β are beliefs in a belief set K (i.e., sentences of $\mathcal{L}$), $\alpha \leqslant \beta$ means that β is at least as entrenched as α. Intuitively, epistemic entrenchment relations induce *preference orderings* of beliefs according to the importance of these beliefs in the face of change. When inconsistency arises during a belief change operation, beliefs with the lowest degree of epistemic entrenchment are given up in order to maintain the properties of minimal and consistent belief changes in rational agents. The concept of epistemic entrenchment captures the notions of *firmness*, *significance*, or *defeasibility* of beliefs as perceived by some agents. This approach is considered more appropriate than measuring the *magnitude* of belief changes in terms of the cardinality of the modified information. By way of illustration, an intelligent information agent (human or software) strongly believes that understanding the paper about "common sense aboutness" and/or the paper about "the logical uncertainty principle" will help her develop an insight about logic-based IR. Now, she reads some IR papers perhaps about the above topics (the agent is not really sure since she is only a novice in this field), but finds that she has no idea about logic-based IR at all. Should the agent contract the beliefs α, β, or $\alpha \vee \beta \rightarrow \gamma$ because of the new information $\neg\gamma$? The propositions are used to represent these events: α : "*understanding common sense aboutness*", β : "*understanding the logical uncertainty principle*", and γ : "*understanding logic-based IR*". For a cardinality-based measure of minimal belief change, the agent should contract $\alpha \vee \beta \rightarrow \gamma$. After such a contraction, a new consistent belief state such as $\{\alpha, \beta, \neg\gamma\}$ is reached. However, is this a rational approach? The agent is almost certain that $\alpha \vee \beta \rightarrow \gamma$, but not sure if α and β are true. The agent may not read papers really about the chosen topics, or she may read relevant papers, but she still does not understand the content of these papers. The reliability or the firmness of the beliefs α and β is low. In face of a strong belief of $\neg\gamma$, a more rational attitude of the agent is to contract the beliefs α and β since she is less certain (firmly believing) about this information. Therefore, measuring the magnitude of belief changes in terms of the underlying epistemic entrenchment orderings is a better solution. Formally, an epistemic entrenchment ordering is a total pre-order of the sentences (e.g., α, β, γ) in $\mathcal{L}$, and is characterised by the following postulates [GM88, Gär92]:

(EE1) If $\alpha \leqslant \beta$ and $\beta \leqslant \gamma$, then $\alpha \leqslant \gamma$ (Transitivity)

(EE2) If $\alpha \vdash \beta$, then $\alpha \leqslant \beta$ (Dominance)

(EE3) For any α and β, $\alpha \leqslant \alpha \wedge \beta$ or $\beta \leqslant \alpha \wedge \beta$ (Conjunctiveness)

(EE4) When $K \neq K_{\perp}$; $\alpha \notin K$ iff $\alpha \leqslant \beta$ for all β (Minimality)

(EE5) If $\beta \leqslant \alpha$ for all β, then $\vdash \alpha$ (Maximality)

(EE1) simply states that an epistemic entrenchment ordering is transitive. (EE2) indicates that a logically weaker sentence is at least as entrenched as a logically stronger sentence. (EE3) tells us that a conjunction is at least as entrenched as one of its conjuncts. (EE4) says that sentences not in a consistent belief set are minimal with respect to an in epistemic entrenchment ordering. (EE5) defines that valid sentences are maximal in epistemic entrenchment orderings. Gärdenfors has indicated that epistemic entrenchment has its roots in information theory [Gär88]. The basic idea is that different sentences have different information content (e.g., measured in terms of entropy). Because information is valuable, it is rational to minimise the loss of information when giving up sentences in a contraction of a state of belief. Gärdenfors and Makinson have established the (C-) condition for the construction of belief contraction functions directly from the underlying epistemic entrenchment orderings [GM88]. The contraction condition (C-) is defined in Theorem 1. They proved that if an ordering satisfies (EE1) - (EE5), the contraction function uniquely determined by (C-) satisfies all the contraction postulates (K^-1) to (K^-8) [GM88]:

Theorem 1 *Let K be a belief set represented by a set of sentences of $\mathcal{L}$. For every contraction function K^- for K, there exists an epistemic entrenchment $\leqslant$ related to K such that the (C-) condition holds for every sentence $\alpha \in \mathcal{L}$. Conversely, for every epistemic entrenchment $\leqslant$ related to K, there exists a contraction function K^- such that (C-) is true for every $\alpha \in \mathcal{L}$.*

$$\text{(C-)} \qquad K^-_\alpha = \begin{cases} \{\beta \in K : \alpha < \alpha \vee \beta\} & \text{if } \nvdash \alpha \\ K & \text{otherwise} \end{cases}$$

where $<$ is the strict part of epistemic entrenchment defined above. This condition states that the contraction of K with respect to α is the set of sentences β such that the epistemic entrenchment of $\alpha \vee \beta$ is strictly greater than that of α. Since a belief revision function can be defined based on a contraction function and an expansion function, the above theorem is sufficient to uniquely define a revision function as well.

Independently, Peppas and Williams have later proved that the (C*) condition holds for belief revision functions [PW95]:

Theorem 2 *Let K be a belief set represented by a set of sentences of $\mathcal{L}$. For every revision function K^* for K, there exists an epistemic entrenchment $\leqslant$ related to K such that the (C*) condition holds for every sentence $\alpha \in \mathcal{L}$. Conversely, for every epistemic entrenchment $\leqslant$ related to K, there exists a revision function K^* such that (C*) is true for every $\alpha \in \mathcal{L}$.*

$$\text{(C*)} \qquad K^*_\alpha = \begin{cases} \{\beta \in \mathcal{L} : \neg\alpha < \alpha \rightarrow \beta\} & \textit{if } \nvdash \neg\alpha \\ \perp & \textit{otherwise} \end{cases}$$

For the convenience of representing a subset of sentences (e.g., a theory) with respect to an epistemic entrenchment ordering, the $cut_\leqslant$ operator is introduced. Essentially, a *cut* operation such as $cut_\leqslant(\alpha)$ extracts the set of sentences which is at least as entrenched as α from a belief set K. Similar to the AGM belief revision operators, the *cut* operation can be generalised to apply to any sentence $\alpha \in \mathcal{L}$ rather than a belief in a belief set. It has been shown that for an epistemic entrenchment $\leqslant$ and a sentence $\alpha \in \mathcal{L}$, $cut_\leqslant(\alpha)$ always returns a theory [Wil96a].

Definition 2 *For an epistemic entrenchment ordering $\leqslant$ and a belief $\alpha \in K$, the cut operation $cut_\leqslant(\alpha)$ returns a set of beliefs defined by:*

$$cut_\leqslant(\alpha) = \{\beta \in K : \alpha \leqslant \beta\}$$

A $Cut()$ function is introduced in Chapter 4 when the Rapid Anytime Maxi-adjustment transmutation algorithm (RAM) is illustrated. The $Cut()$ function can be seen as the implementation of the above *cut* operation. However, the $Cut()$ function assumes that the *entrenchment rank* of a belief α or the ranks of two delimiting beliefs are known. A formal definition of *cut* was also introduced with respect to a finite partial entrenchment ranking [Wil95]. However, the above definition is more general in the sense that it applies to both epistemic entrenchment orderings and finite partial entrenchment rankings. Moreover, the above definition which is based on [Wil96a] is more concise and precise than the one presented in [Wil95].

The AGM belief revision framework provides a rigorous foundation for modelling the changes of belief states in rational agents. As a belief set K is a theory of a logical language and a theory could be infinite even for a finite language, there could be a representation problem for epistemic entrenchment orderings when they are implemented on computer-based systems which store finite data structures. Moreover, the AGM change functions take a belief set and a sentence as inputs and produce a modified belief set such as $K \times \mathcal{L} \mapsto K$. The change functions do not produce a revised epistemic entrenchment ordering as output. This makes it difficult to perform iterated belief revision which is often a compulsory feature for many real-life applications. For example, in the context of adaptive information retrieval, the information agents' beliefs about users' information needs require continuous revision because the users' interests change over time. As a whole, for a computer-based implementation of the AGM belief functions, a finite representation of epistemic entrenchment and an iterated belief revision mechanism are needed. Williams has proposed the *finite partial entrenchment ranking* ($\mathbf{B}$) that ranked the sentences of a theory in $\mathcal{L}$ with the minimum possible degree of entrenchment ($\leqslant_{\mathbf{B}}$) [Wil95]. Moreover, implementing the AGM change functions based on a *transmutation* mechanism was also explored [Wil94]. The *Adjustment* transmutation algorithm [Wil95] which exactly implements the AGM change functions, and the *Maxi-adjustment* algorithm [Wil96b, Wil97] which is based on the rationale of absolute minimal change under maximal information inertia have also been developed. In a transmutation-based approach, belief revision is not just taken as adding or removing sentences to or from belief sets but also the *transmutation* of the underlying epistemic entrenchment ranking. A finite partial entrenchment ranking $\mathbf{B}$ assigns the minimal degree of entrenchment (in terms of a real number) to each sentence, and hence induces the underlying epistemic entrenchment ranking. The following definitions are based on Williams' work [Wil95, Wil96a, Wil96b, Wil97]:

Definition 3 *A finite partial entrenchment ranking is a function* $\mathbf{B}$ *that maps a finite subset of sentences of* $\mathcal{L}$ *to the real interval [0, 1] such that the following conditions are satisfied for all* $\alpha \in dom(\mathbf{B})$*:*

(PER1) $\{\beta \in dom(\mathbf{B}) : \mathbf{B}(\alpha) < \mathbf{B}(\beta)\} \nvdash \alpha$.

(PER2) If $\vdash \neg\alpha$ *then* $\mathbf{B}(\alpha) = 0$.

(PER3) $\mathbf{B}(\alpha) = 1$ *if and only if* $\vdash \alpha$.

(PER1) states that the set of sentences ranked strictly higher than a sentence α cannot entail α. In other words, a logically stronger sentence should have a lower *entrenchment degree* represented by $\mathbf{B}(\alpha)$. This property of finite partial entrenchment ranking corresponds to the postulate of epistemic entrenchment (EE2). The meaning of (PER2) is that inconsistent sentences have the lowest entrenchment degree or should be ranked at the highest position. (PER3) says that valid sentences are assigned the maximal entrenchment degree or should be ranked the lowest. The set of all possible finite partial entrenchment rankings is denoted $\mathcal{B}$. $\mathbf{B}(\alpha)$ is referred to as the *degree of acceptance* of an explicit belief α. The explicit information content of $\mathbf{B} \in \mathcal{B}$ is $\{\alpha \in \mathrm{dom}(\mathbf{B}) : \mathbf{B}(\alpha) > 0\}$, and is denoted $exp(\mathbf{B})$. In other words, $exp(\mathbf{B})$ defines a finite *theory base* which captures a rational agent's explicit beliefs. In addition, the implicit information content of $\mathbf{B} \in \mathcal{B}$ is derived by $Cn(exp(\mathbf{B}))$, and is denoted $content(\mathbf{B})$. The operator Cn is the classical consequence operator as defined before. Therefore, $content(\mathbf{B})$ corresponds to the belief set K, which is the information content of an agent's knowledge base characterising an IR context. For a set Γ of explicit beliefs, the degree of acceptance of Γ is defined by $\mathbf{B}(\Gamma) = min(\{\mathbf{B}(\alpha) : \alpha \in \Gamma\})$. In order to describe the epistemic entrenchment ordering ($\leqslant_{\mathbf{B}}$) generated from a finite partial entrenchment ranking $\mathbf{B}$, it is necessary to compute the degrees of acceptance (i.e., entrenchment degress) of implicit beliefs. The following definition is equivalent to the one presented in [Wil97] but our refined definition is based on the *cut* operation defined in Definition 2:

Definition 4 *Let $\alpha \in \mathcal{L}$ be a contingent sentence. Let $\mathbf{B}$ be a finite partial entrenchment ranking and $\beta \in exp(\mathbf{B})$. The degree of acceptance of α is defined by:*

$$degree(\mathbf{B}, \alpha) = \begin{cases} sup(\{\mathbf{B}(\beta) \in ran(\mathbf{B}) : cut_{\leqslant_{\mathbf{B}}}(\beta) \vdash \alpha\}) & \text{if } \alpha \in content(\mathbf{B}) \\ 0 & otherwise \end{cases}$$

The *sup* operator returns the maximal degree of acceptance from a set of ordinals in the *range* of $\mathbf{B}$. The $cut_{\leqslant_{\mathbf{B}}}(\beta)$ operation extracts a set of explicit beliefs which is at least as entrenched as β from an epistemic entrenchment ordering $\leqslant_{\mathbf{B}}$ generated based on a finite partial entrenchment ranking $\mathbf{B}$. Therefore, the above definition states that the degree of acceptance of an implicit belief α equals the maximal degree of acceptance of a cut (in accordance with $\leqslant_{\mathbf{B}}$) of explicit beliefs that classically entail α.

The Maxi-adjustment method [Wil96b, Wil97, Wil96a] transmutes (e.g., raising or lowering) the degrees of the explicit sentences in a theory base to simulate the processes of incorporating beliefs into (or removing beliefs from) a belief set. In order to implement the AGM change operations which are applied to a set of logically closed sentences, the Maxi-adjustment algorithm needs a classical theorem prover to evaluate the implicit sentences captured in *content*($\mathbf{B}$). The Maxi-Adjustment method differs from the Adjustment method which exactly implement the standard AGM change functions in that it transmutes a partial entrenchment ranking $\mathbf{B}$ according to the rationale of absolute minimal change under maximal information inertia [Wil96b]. In other words, it may retain even more sentences than the standard AGM contraction function while preserving the AGM principles of minimal and consistent belief change. In addition, Williams also tried to introduce the notion of *reasons* as advocated by Spohn [Spo87] and reason maintenance in her Maxi-adjustment method [Wil96a]. With reference to finite partial entrenchment rankings, a sentence α is a *reason* of β if and only if $degree(\mathbf{B}, \alpha \rightarrow \beta) > \mathbf{B}(\beta)$. However, the main difference between the Maxi-adjustment transmutation method and the Adjustment method which directly implements the standard AGM belief functions is that the sentences in the theory base $exp(\mathbf{B})$ are assumed independent unless logical dependences can be derived via $\vdash$. This assumption behind the Maxi-adjustment method makes it a better candidate for modelling belief changes in many real-life applications. The assumptions behind the Maxi-adjustment method correspond to the characteristics as demonstrated in IR applications. For example, when modelling the IR requirements of information seekers, term independency is often assumed unless the inter-dependencies are explicitly specified. This approach has been adopted in existing quantitative IR models [SM83] as well as logic-based IR models [LB98]. The following is a re-production of the definition of the Maxi-adjustment method based on [Wil96a]:

Definition 5 *Let $\mathbf{B} \in \mathcal{B}$ be finite. The range of $\mathbf{B}$ is enumerated in ascending order such as $j_0, j_1, j_2, \ldots, j_{max}$. Let α be a contingent sentence, $j_m = degree(\mathbf{B}, \alpha)$ and $0 \leq i < 1$. Then the (α, i) Maxi-adjustment of $\mathbf{B}$ is $\mathbf{B}^\star(\alpha, i)$ defined by:*

$$\mathbf{B}^\star(\alpha, i) = \begin{cases} \mathbf{B}^-(\alpha, i) & \text{if } i \leq j_m \\ (\mathbf{B}^-(\neg\alpha, 0))^+(\alpha, i) & \text{otherwise} \end{cases}$$

where for all $\beta \in dom(\mathbf{B})$, $\mathbf{B}^-(\alpha, i)$ *is defined as follows:*

1. For β *with* $\mathbf{B}(\beta) > j_m$, $\mathbf{B}^-(\alpha, i)(\beta) = \mathbf{B}(\beta)$.

2. For β *with* $i < \mathbf{B}(\beta) \leq j_m$, *suppose* $\mathbf{B}^-(\alpha, i)(\beta)$ *for* β *is defined with* $\mathbf{B}(\beta) \geq j_{m-k}$ *for* $k = 0, 1, 2, \ldots, n-1$, *then for* β *with* $\mathbf{B}(\beta) = j_{m-n}$,

$$\mathbf{B}^-(\alpha, i)(\beta) = \begin{cases} i & \textit{if } \alpha \vdash \beta \textit{ or} \\ & \alpha \nvdash \beta \textit{ and } \beta \in \Gamma \\ & \textit{where } \Gamma \textit{ is a minimal subset of} \\ & \{\gamma : \mathbf{B}(\gamma) = j_{m-n}\} \textit{ such that} \\ & \{\gamma : \mathbf{B}^-(\alpha, i)(\gamma) > j_{m-n}\} \cup \Gamma \vdash \alpha \\ \mathbf{B}(\beta) & \textit{otherwise} \end{cases}$$

3. For β *with* $\mathbf{B}(\beta) \leq i$, $\mathbf{B}^-(\alpha, i)(\beta) = \mathbf{B}(\beta)$.

For all $\beta \in dom(\mathbf{B}) \cup \{\alpha\}$, $\mathbf{B}^+(\alpha, i)$ *is defined as follows:*

$$\mathbf{B}^+(\alpha, i)(\beta) = \begin{cases} \mathbf{B}(\beta) & \textit{if } \mathbf{B}(\beta) > i \\ i & \textit{if } \alpha \equiv \beta \textit{ or} \\ & \mathbf{B}(\beta) \leq i < \textit{degree}(\mathbf{B}, \alpha \to \beta) \\ \textit{degree}(\mathbf{B}, \alpha \to \beta) & \textit{otherwise} \end{cases}$$

The algorithm deals with contingent sentences because they are the principle cases. For a valid sentence, a transmutation operation can easily be defined and implemented by assigning the sentence with the maximal degree. On the other hand, for an inconsistent belief as input, a transmutation operation can trivially be done by returning the existing entrenchment ranking. The intuition of the above definition is that if the new entrenchment degree i of a sentence α is less than its existing degree j_m, it is equivalent to a contraction operation (i.e., lowering its degree). A contraction operation is implemented by $\mathbf{B}^-(\alpha, i)$ in the algorithm. If the new degree of α is higher than its existing degree, it is considered a revision operation. Hence, $\neg\alpha$ must first be assigned the lowest degree of acceptance (i.e., contracting it from the belief set). The contraction process could be very time consuming because $\neg\alpha$ may not be in the theory base $exp(\mathbf{B})$, but implied by other explicit beliefs in the theory base. So, a theorem prover must be invoked to perform the satisfiability check. After contracting $\neg\alpha$ and all the beliefs that entail $\neg\alpha$, the degree of α is raised to the new degree i. This process corresponds to belief expansion and is implemented by $\mathbf{B}^+(\alpha, i)$ in the algorithm. Therefore, the Maxi-adjustment method ensures that the AGM principle of consistent belief revision is enforced (i.e., $content(\mathbf{B}) \nvdash \bot$). During raising or lowering of the degree of α, the degrees of other sentences in the theory base are adjusted in a minimal way such that the (PER1) property (i.e., the *dominance* property of epistemic entrenchment) is maintained. This is a very time consuming process since it invokes the theorem prover to prove certain logical conditions for each sentence being affected by the belief change process. It should be noted that with reference to the postulates (K^-1) to (K^-8), the part $\mathbf{B}^-(\alpha, i)(\beta) = i$ if $\alpha \vdash \beta$ in the Maxi-adjustment method is not an element of a standard AGM contraction operation. It was introduced as a kind of reason maintenance called *subsumption* *removal* [Wil97]. For instance, if α is the only reason for β to be included in a belief set, β should not exist after α is contracted. It has been shown that if $i > 0$ then $content(\mathbf{B}^{\star}(\alpha, i)) = (content(\mathbf{B}))^{*}_{\alpha}$ [Wil96b]. In other words, maxi-adjustment with $i > 0$ is an AGM revision. On the other hand, $content(\mathbf{B}^{\star}(\alpha, 0))$ satisfies all but the *recovery* postulates for AGM contraction [Wil96b].

The advantage of the Maxi-adjustment method for belief revision can be illustrated with an example. Assuming that an information seeker is looking for documents about "apple", "banana", and "cat". Her preferences can be characterised by an epistemic entrenchment ordering which is finitely represented by a finite partial entrenchment ranking $\mathbf{B}$:

$\mathbf{B}(apple) = 0.8$

$\mathbf{B}(banana) = 0.7$

$\mathbf{B}(cat) = 0.6$

If the information seeker is no longer interested in documents about "apple", a contraction function can be defined to model the change of her beliefs. By using the standard AGM contraction function K^{-} as defined by the (C-) condition in Theorem 1, $K^{-}_{apple} = \{\}$ is derived because $apple \not< (banana \vee apple)$ and $apple \not< (cat \vee apple)$ are true. The degree of acceptance $degree(\mathbf{B}, banana \vee apple) = 0.8$ for the belief $(banana \vee apple)$ is computed according to Definition 4. Therefore, $apple \not< (banana \vee apple)$ is derived. Similarly, $apple \not< (cat \vee apple)$ is also derived.

In other words, if an information agent is told that the information seeker gives up her interest about "apple", the agent will believe that she is no longer interested in "banana" nor "cat". Is it true that a person who does not like "apple" will always reject "banana" or "cat"? The problem is not caused by the AGM rationale of belief change but the implicit assumption of the (C-) condition where information in the belief set is inter-related by default. On the other hand, by applying the Maxi-adjustment method to model the same situation, the result of $\mathbf{B}^{\star}(apple, 0) = \{(banana, 0.7), (cat, 0.6)\}$ is obtained. With reference to the contraction part of the Maxi-adjustment method, it is easy to see that the minimal subsets in the two entrenchment ranks are $\{banana\}$ and $\{cat\}$ respectively. In either case, the minimal subset Γ together with any strictly more entrenched beliefs does not entail *apple* (i.e., $\{\gamma : \mathbf{B}^{-}(\alpha, i)(\gamma) > j_{m-n}\} \cup \Gamma \vdash \alpha$ is not true). Therefore, the entrenchment degrees are not changed $\mathbf{B}^{-}(\alpha, i)(\beta) = \mathbf{B}(\beta)$. In other words, the beliefs $(banana, 0.7)$ and $(cat, 0.6)$ are retained in the belief set. The Maxi-adjustment method seems to appropriately model the changes of beliefs in adaptive information agents.

The most costly procedure of the Maxi-adjustment method is to evaluate if the minimal subsets Γ together with other strictly more entrenched beliefs in the entrenchment ranking will entail α, the sentence to be assigned a lower degree in the contraction operation $\mathbf{B}^{-}(\alpha, i)$. If there are many sentences in the same rank, the computational complexity grows exponentially $O(2^N)$ in the worst case, where N is the number of sentences with the same rank. In general, $O(2^N)$ is required to enumerate all the possible subsets of a base set with size N. The proof of whether each subset can logically entail α is costly as well although a polynomial time algorithm exists if the representation language is a classical propositional Horn language $\mathcal{L}_{Horn}$ [Hod93]. Lang has proven that if $\mathbf{B}$ has x natural partitions then it requires $\log_2 x$ satisfiability checks [Lan97]. So, the computational cost of the Maxi-adjustment method decreases as the number of ranks increases. When the ideal case occurs where each rank in $\mathbf{B}$ contains only one sentence, the computational complexity of the Maxi-adjustment algorithm is polynomial since $\log_2 x$ plus the polynomial time for the satisfiability check of x sentences of $\mathcal{L}_{Horn}$ is till characterised by a polynomial time complexity. Williams has proposed the *anytime* version of the Maxi-adjustment method which can approximate $\mathbf{B}^{\star}(\alpha, i)$ based on a time parameter that defines the maximum time allowed for each $\mathbf{B}^{-}(\alpha, i)(\beta)$ or $\mathbf{B}^{+}(\alpha, i)(\beta)$ operation [Wil97]. The *anytime* approach allows a trade-off between computational cost and the quality of the belief revision processes. Basically, the anytime algorithm copies all the un-changed beliefs to a new theory base first. For each belief β from the problematic segment of the theory base, transmutes the degree of β as defined in the Maxi-adjustment method and copies it to the new theory base if the elapsed time is within the time limit. Therefore, the anytime algorithm can revise as many beliefs as possible while ensuring that all the properties of epistemic entrenchment are fulfilled. However, whether this approach is feasible for large real-life applications still requires empirical evaluation. One of the contributions of this thesis is to provide an answer for such a research question.

An alternative of implementing the AGM change functions is to develop another more efficient transmutation method which avoids the computational bottle-neck of generating and evaluating the minimal subsets Γ in an entrenchment ranking (thereby circumventing the $O(2^N)$ computational cost) and yet adheres to the AGM principle of minimal and consistent belief revision. To this end, the Rapid Maxi-adjustment method is proposed in this thesis. In particular, the anytime version of this method called Rapid Anytime Maxi-adjustment (RAM) is the standard transmutation method for implementing the AGM change functions in adaptive information agents. The computational algorithm of RAM will be illustrated in Chapter 4.

3.2 The Rapid Maxi-adjustment Method

The Rapid Maxi-adjustment method is developed based on the Maxi-adjustment method. The major improvement is the removal of the minimal subset generation procedure during belief contraction. Moreover, the reason maintenance mechanism is also removed because causal reasoning is less applicable to IR processes. Finally, some corrections to the Maxi-adjustment method are done so that the segment of a finite partial entrenchment ranking under revision is clearly identified to facilitate performance tuning. The anytime feature is not included in the following logical definition because it is more an implementation oriented feature.

Definition 6 *Let* $\mathbf{B} \in \mathcal{B}$ *be finite. Let* α *be a contingent sentence,* $j = degree(\mathbf{B}, \alpha)$

and $0 \leq i < 1$. *Then the* (α, i) *Rapid Maxi-adjustment of* $\mathbf{B}$ *is* $\mathbf{B}^{\star}(\alpha, i)$ *defined by:*

$$\mathbf{B}^{\star}(\alpha, i) = \begin{cases} \mathbf{B}^{-}(\alpha, i) & \textit{if } i < j \\ (\mathbf{B}^{-}(\neg\alpha, 0))^{+}(\alpha, i) & \textit{if } i > j \\ \mathbf{B}^{+}(\alpha, i) & \textit{if } i = j > 0 \textit{ and } \alpha \notin exp(\mathbf{B}) \\ \mathbf{B} & \textit{otherwise} \end{cases}$$

where for all $\beta \in dom(\mathbf{B})$, $\mathbf{B}^{-}(\alpha, i)$ *is defined as follows:*

1. *For* β *with* $\mathbf{B}(\beta) > j$, $\mathbf{B}^{-}(\alpha, i)(\beta) = \mathbf{B}(\beta)$.

2. *For* β *with* $i < \mathbf{B}(\beta) \leq j$,

$$\mathbf{B}^{-}(\alpha, i)(\beta) = \begin{cases} i & \textit{if } \{\gamma : \mathbf{B}^{-}(\alpha, i)(\gamma) > \mathbf{B}(\beta)\} \cup \\ & \{\delta : \mathbf{B}^{-}(\alpha, i)(\delta) = \mathbf{B}(\beta) \wedge Seq(\delta) \leq Seq(\beta)\} \vdash \alpha \\ \mathbf{B}(\beta) & \textit{otherwise} \end{cases}$$

3. *For* β *with* $\mathbf{B}(\beta) \leq i$, $\mathbf{B}^{-}(\alpha, i)(\beta) = \mathbf{B}(\beta)$.

For all $\beta \in dom(\mathbf{B}) \cup \{\alpha\}$, $\mathbf{B}^{+}(\alpha, i)$ *is defined as follows:*

1. *For* β *with* $\mathbf{B}(\beta) \geq i$, $\mathbf{B}^{+}(\alpha, i)(\beta) = \mathbf{B}(\beta)$.

2. *For* β *with* $j \leq \mathbf{B}(\beta) < i$,

$$\mathbf{B}^{+}(\alpha, i)(\beta) = \begin{cases} i & \text{if } i < degree(\mathbf{B}, \alpha \rightarrow \beta) \\ degree(\mathbf{B}, \alpha \rightarrow \beta) & \text{otherwise} \end{cases}$$

3. For β with $\mathbf{B}(\beta) < j$, $\mathbf{B}^{+}(\alpha, i)(\beta) = \mathbf{B}(\beta)$.

When a transmutation process begins, the algorithm will not invoke a contraction process $\mathbf{B}^{-}(\alpha, i)$ if the entrenchment degree of the belief $\alpha \in exp(\mathbf{B})$ under question does not change. The Rapid Maxi-adjustment method eliminates the computational bottle-neck of evaluating the minimal subsets in a rank when a contraction operation is performed. The $\mathbf{B}^{-}(\alpha, i)(\beta)$ part works by sequentially processing each affected belief β. When a belief β from the *problematic segment* of $\mathbf{B}$ (e.g., $i < \mathbf{B}(\beta) \leq j$) is evaluated, β together with other beliefs with the same entrenchment rank but assigned lower sequence numbers such as $\{\delta : \mathbf{B}^{-}(\alpha, i)(\delta) = \mathbf{B}(\beta) \wedge Seq(\delta) \leq Seq(\beta)\}$ and the strictly more entrenched beliefs such as $\{\gamma : \mathbf{B}^{-}(\alpha, i)(\gamma) > \mathbf{B}(\beta)\}$ are added to the theorem prover to test if they can logically entail α. If it is true, the degree of β will be lowered to i. Moreover, the sentence β will be removed from the theorem prover before evaluating the remaining sentences in the problematic segment. Thereby, the properties of finite partial entrenchment ranking are maintained. The Seq function simply assigns unique numbers to the beliefs residing in the same rank in ascending order, the Seq construct is not part of an epistemic entrenchment ranking. It is introduced to handle beliefs with the same epistemic entrenchment degrees. If $\mathcal{L}_{Horn}$ is chosen as the representation language, the Rapid Maxi-adjustment algorithm will only involve polynomial time complexity (i.e., $\log_2 x$ plus the polynomial time for proving x horn clauses). Another distinct advantage of the Rapid Maxi-adjustment method is that it may retain more beliefs than the Maxi-adjustment does in some cases. This shows the operational characteristic of the proposed method in terms of fulfilling the AGM principle of minimal changes. The following example shows the advantage of the Rapid Maxi-adjustment method. An agent's theory base is described by a finite partial entrenchment ranking. The sequence numbers on the left are not part of the finite partial entrenchment ranking. These numbers help uniquely identify each belief, and they can be seen as the sequence numbers returned by the Seq function when there are several beliefs in the same entrenchment rank. ranking $\mathbf{B}$. An information agent perceives an IR context in terms of some beliefs such as "Australia", "Brazil", "Canada", "Denmark", "Egypt" because its user is interested in retrieving information about these countries. These beliefs are represented by the corresponding propositions of $\mathcal{L}$. Initially, the agent's theory base $exp(\mathbf{B})$ comprises a set of beliefs characterising the user's preferences of information pertaining to different countries. By relevance feedback, the user informs the agent that she is no longer interested in information about "Australia". The agent's knowledge base $content(\mathbf{B})$ needs to be revised by invoking a belief revision operation such as $\mathbf{B}^{\star}(\neg australia, 0.9)$. The agent's initial theory base and the theory bases after applying the standard AGM Adjustment method, the Maxi-adjustment method, and the Rapid Maxi-adjustment method are shown as follows:

Before Belief Revision

1. $\mathbf{B}(australia \vee brazil) = 0.8$
2. $\mathbf{B}(australia \vee denmark) = 0.7$
3. $\mathbf{B}(\neg canada \vee \neg brazil) = 0.6$
4. $\mathbf{B}(canada) = 0.6$
5. $\mathbf{B}(denmark) = 0.6$
6. $\mathbf{B}(egypt) = 0.6$

Standard AGM revision

7. $\mathbf{B}(\neg australia) = 0.9$
1. $\mathbf{B}(australia \vee brazil) = 0.8$
2. $\mathbf{B}(australia \vee denmark) = 0.7$
5. $\mathbf{B}(denmark) = 0.6$

For all these transmutation methods, the revision procedure will first contract the belief *australia* from the belief set $K = content(\mathbf{B})$. In other words, any explicit beliefs that logically entail *australia* are contracted from the theory base $exp(\mathbf{B})$. Then, the belief $(\neg australia, 0.9)$ is revised into the theory base. It is obvious that the explicit beliefs $\{australia \vee brazil, \neg canada \vee \neg brazil, canada\}$ entail *australia*. According to the (C-) condition, the standard AGM contraction will contract $\neg canada \vee \neg brazil$ since $australia \not< ((\neg canada \vee \neg brazil) \vee australia)$ is derived. According to Definition 4, $degree(\mathbf{B}, (\neg canada \vee \neg brazil) \vee australia) = 0.6 = degree(\mathbf{B}, australia)$ is true. Similarly, *canada* and *egypt* are contracted. The belief *denmark* is retained

Revision by Maxi-adjustment

7. $\mathbf{B}(\neg australia) = 0.9$
1. $\mathbf{B}(australia \vee brazil) = 0.8$
2. $\mathbf{B}(australia \vee denmark) = 0.7$
5. $\mathbf{B}(denmark) = 0.6$
6. $\mathbf{B}(egypt) = 0.6$

Revision by Rapid Maxi-adjustment

7. $\mathbf{B}(\neg australia) = 0.9$
1. $\mathbf{B}(australia \vee brazil) = 0.8$
2. $\mathbf{B}(australia \vee denmark) = 0.7$
3. $\mathbf{B}(\neg canada \vee \neg brazil) = 0.6$
5. $\mathbf{B}(denmark) = 0.6$
6. $\mathbf{B}(egypt) = 0.6$

because $\mathbf{B}(australia \vee denmark) = 0.7 > degree(\mathbf{B}, australia) = 0.6$. For the belief revision process implemented via the Maxi-adjustment method, one more belief *egypt* is retained because the contraction criterion is $\mathbf{B}^{-}(\alpha, i)(\beta) = i$ if $\{\gamma : \mathbf{B}^{-}(\alpha, i)(\gamma) > j_{m-n}\} \cup \Gamma \vdash \alpha$ is true. The minimal subset $\Gamma = \{\neg canada \vee \neg brazil, canada\}$ is developed for the rank with beliefs having entrenchment degree 0.6 in this example. This minimal subset Γ together with other more entrenched beliefs entail *australia*, and so all the sentences of Γ are assigned the degree 0. Other sentences in the same rank but not contained in Γ are retained.

For the Rapid Maxi-adjustment method, the contraction criterion is $\mathbf{B}^{-}(\alpha, i)(\beta) = i$ if $\{\gamma : \mathbf{B}^{-}(\alpha, i)(\gamma) > \mathbf{B}(\beta)\} \cup \{\delta : \mathbf{B}^{-}(\alpha, i)(\delta) = \mathbf{B}(\beta) \wedge Seq(\delta) \leq Seq(\beta)\} \vdash \alpha$ is true. The above condition is true when the fourth belief *canada* is added to the theorem prover to prove *australia*, and so the degree of *canada* is lowered to zero. After this adjustment operation, each belief in this rank (i.e., degree = 0.6) together with any strictly more entrenched beliefs do not entail *australia*. Therefore, more beliefs are retained at the end of the contraction process. The proposed Rapid Maxi-adjustment method is more efficient than the Maxi-adjustment method since there is no need to carry out the time-consuming process of computing the minimal inconsistent subsets in a rank (e.g., the beliefs with entrenchment degree equal 0.6). As demonstrated in this example, the Rapid Maxi-adjustment method is also more effective in terms of fulfilling the minimal belief change criterion than the Maxi-adjustment method does. However, in this thesis, we will validate the qualities of the Rapid Maxi-adjustment approach by conducting empirical evaluations of the method within large adaptive information filtering experiment. The evaluation work and the results will be reported in Chapter 5.

3.3 Expectation Inference Relations

When an intelligent agent attempts to solve a problem, it may not have complete information about the problem domain. However, it may still be useful if the agent can develop tentative solutions in a timely fashion. When more information about the problem domain is obtained later on, the agent must be prepared to alter its tentative conclusion if the new information contradicts previous information from which the tentative conclusion is drawn. This kind of situation prevails in adaptive information retrieval where little information about the retrieval contexts is known at the beginning. However, with the help of users' relevance feedback, more information about the retrieval contexts may be obtained later. The new information about the retrieval contexts requires information agents to revise their beliefs about the situations, and alter their previous decisions about document selection. Nonmonotonic reasoning provides a formal framework for intelligent agents to make quick decisions when they are faced with incomplete and uncertain information. In classical logics the derivability relation $\vdash$ allows an agent to determine if a formula α follows from a set of premises Γ. The set of conclusions are assumed to grow monotonically (i.e., $Cn(\Gamma) \subseteq Cn(\Gamma \cup \{\alpha\})$ for any new information α). The notion of nonmonotonic inference allows an intelligent agent to draw tentative conclusions, and these conclusions can be retracted when more accurate information is available later [KLM90, LM92, Mak93]. Unlike the classical inference relation $\vdash$, the information deduced via a nonmonotonic inference relation $\mid\!\sim$ grows nonmonotonically. In general, $\alpha \mid\!\sim \beta$ means that the piece of information α nonmonotonically entails another piece of information β. According to Gärdenfors and Makinson [GM94], well-behaved nonmonotonic inference relations can be characterised by the following properties. These properties are presented in a way to facilitate the discussion of applying nonmonotonic inference to IR rather than establishing a one to one mapping to the postulates of the AGM belief functions. A relation $\mid\!\sim$ is an *inference relation* iff it satisfies the four postulates:

$$\frac{\alpha \vdash \beta}{\alpha \mid\!\sim \beta} \quad \text{(Supraclassicality)}$$

$$\frac{\alpha \equiv \beta \qquad \alpha \mid\!\sim \gamma}{\beta \mid\!\sim \gamma} \quad \text{(Left Logical Equivalence)}$$

$$\frac{\alpha \mid\!\sim \beta \qquad \beta \vdash \gamma}{\alpha \mid\!\sim \gamma} \quad \text{(Right Weakening)}$$

$$\frac{\alpha \mid\!\sim \beta \qquad \alpha \mid\!\sim \gamma}{\alpha \mid\!\sim \beta \wedge \gamma} \quad \text{(And)}$$

An inference relation $\mid\!\sim$ is a well-behaved *nonmonotonic inference relation* iff it is an inference relation and it satisfies the following four postulates:

$$\frac{\alpha \mid\!\sim \beta \qquad \alpha \wedge \beta \mid\!\sim \gamma}{\alpha \mid\!\sim \gamma} \quad \text{(Cut)}$$

$$\frac{\alpha \mid\!\sim \beta \qquad \alpha \mid\!\sim \gamma}{\alpha \wedge \beta \mid\!\sim \gamma} \quad \text{(Cautious Monotony)}$$

$$\frac{\alpha \mid\!\sim \gamma \qquad \beta \mid\!\sim \gamma}{\alpha \vee \beta \mid\!\sim \gamma} \quad \text{(Or)}$$

$$\frac{\alpha \mid\!\not\sim \neg\beta \qquad \alpha \mid\!\sim \gamma}{\alpha \wedge \beta \mid\!\sim \gamma} \quad \text{(Rational Monotony)}$$

An inference relation $\mid\!\sim$ is an *expectation inference relation* ($\mid\!\sim_K$) iff it is a nonmonotonic inference relation and it additionally satisfies the postulate of *consistency preservation*:

$$\frac{\alpha \mid\!\sim \bot}{\alpha \vdash \bot} \quad \text{(Consistency Preservation)}$$

It has been shown that the set of postulates characterising the expectation inference relations can be translated to the set of postulates which define the AGM belief revision functions [MG91]. Indeed, belief revision and nonmonotonic inference are viewed as two sides of the same coin [GM94]. The relevance of nonmonotonic reasoning with respect to IR has received considerable interest [BH94, BH96, Hun96, Seb94]. Accordingly, nonmonotonic inference provides adaptive information agents with a sound and robust formalism to make decisions regarding document relevance. Gärdenfors and Makinson [MG91, GM94] have examined the interconnections between belief revision and nonmonotonic inference. In general, the interconnection between belief revision and nonmonotonic inference is described by the following relationship:

$$\beta \in K^*_\alpha \equiv \alpha \mid\!\sim_K \beta$$

where K^*_α is the revision of a belief state K with respect to a formula α, and this process is taken as the nonmonotonic inference from α to β given the set K of formulae as background expectations. More specifically, Gärdenfors and Makinson examined the orderings of formulae in K and how the orderings can be used to define a class of nonmonotonic inference relations. They evaluated a subset of the epistemic entrenchment postulates (EE1) to (EE3), and called the orderings as characterised by (EE1) to (EE3) the *expectation orderings.* It was found that both the expectation orderings and the epistemic entrenchment orderings would generate the same class of nonmonotonic inference relations which satisfy the postulates of the expectation inference relations. The formal definition of expectation inference was first proposed by Gärdenfors and Makinson [GM94]. The following definition is based on their proposal with the emphasis on the beliefs in a belief set K:

Definition 7 $\mid\!\sim_K$ *is a comparative expectation inference relation iff there is an ordering* $\leqslant$ *satisfying (EE1) - (EE3) such that the following condition holds:*

$$(C \mid\!\sim_K) \qquad \alpha \mid\!\sim_K \beta \quad \textit{iff} \quad \beta \in Cn(\{\alpha\} \cup \{\gamma \in K \colon \neg\alpha < \gamma\})$$

Expectation inference provides a sound and powerful inference framework for developing the decision making mechanisms in adaptive information agents. It is argued that *conservatively monotonic* IR models are promising because the operational characteristic of IR processes are essentially conservatively monotonic [BSW00]. As characterised by the postulates of *cautious monotony* and *rational monotony*, it is easy to find that the kind of decision making (i.e., document classification) mechanisms underpinned by expectation inference demonstrates conservatively monotonic property because given the fact $\alpha \mid\!\sim_K \gamma$, the expansion $\alpha \wedge \beta \mid\!\sim_K \gamma$ is not always possible. Extending an agent's beliefs such as $\alpha \wedge \beta$ subjects to certain restrictions. In fact, the nonmonotonic axioms such as "Cut", "And", and "Cautious Monotony" have direct counterparts in the set of properties charactering well-behaved IR models [BH94, HW98, BSW00]. The nonmonotonic axiom of "Rational Monotony" also plays an important role in establishing the fundamental property such as QLM of common sense aboutness which characterises the prominent features of IR models [BSW00]. An obvious advantage of applying the AGM belief revision paradigm to develop adaptive information agents is that the learning components and the classification components can be seamlessly integrated in these agents. The learning functions of the agents are characterised by the AGM belief revision functions K^*_α and the classification functions of the agents are underpinned by expectation inference $\alpha \mathrel{|\!\!\!\sim_K} \beta$. In the context of IR, a belief set K represents an agent's perception about a particular retrieval context, and α is the relevance feedback information provided by a user. In general, a relevance feedback can be seen as a refined query or information which leads to the development of a refined query. Therefore, $\alpha \mathrel{|\!\!\!\sim_K} \beta$ represents the evaluation of a document representation β with respect to the refined query α. Since an adaptive information agent functions like a user profile which holds multiple long term recurring queries for a user, the inference process in the classification component of the agent can also be conceptualised as $K \mathrel{|\!\!\!\sim_K} \beta$ where the agent uses all the information about a user's queries and the query context to deduce if a document is relevant or not with respect to these queries.

3.4 The AGM Paradigm in the Context of IR

This section briefly describes some fundamental concepts in IR [BH94, BSW00] so that the assumptions of the AGM belief revision paradigm can be evaluated in the context of IR. Particularly, epistemic entrenchment which underpins the AGM belief functions will be examined with reference to the fundamental IR concepts.

Information Carriers: Information carriers (IC) represent the content of information. Examples of ICs are documents, parts of documents (e.g., a section) and document descriptors, such as keywords. The lowercase letters such as i, j, etc. are used to represent information carriers. The elementary information carriers that cannot be further decomposed are called atomic information carriers. From an application point of view, keywords or terms are elementary enough to be considered as atomic information carriers.

Information Containment: As some information carriers convey more information about a situation(s) than others, it was suggested that information can be partially ordered with respect to information containment, denoted by $\rightarrow_T$ [Bar89b]. $i \rightarrow_T j$ iff information carrier i contains all the information carried by information carrier j.

Information Composition: Information carriers can be composed to form more complex information carriers. For example, information carriers such as *river* and *pollution* can be composed because *river* $\oplus$ *pollution* means the pollution of rivers. More formally, $i \oplus j$ is the smallest information carrier (with respect to the ordering $\rightarrow_T$) that precisely contains the information carried by information carriers i and j. There is a difference between $\oplus$ described here and $\wedge$ used in Boolean retrieval models. The Boolean operator $\wedge$ assumes terms independence. However, it is assumed that $\oplus$ satisfies idempotency, but commutativity and associativity can not be taken for granted because they are dependent on the semantic meanings of associated information carriers. In general, an information language $\mathbb{L}_T$ which is built from a set of terms can be defined [Bru96]: Let IC be a set of information carriers, then, (1)

$IC \subset \mathbb{L}_T$; (2) if $i, j \in \mathbb{L}_T$ then $(i \oplus j) \in \mathbb{L}_T$.

Aboutness: Information retrieval is driven by a process which decides whether a document is about a query. Abstracting from documents and queries renders the IR process as one which decides whether one information carrier is about another. Recently, "Aboutness" has been examined as a by product of research within logic-based information retrieval [Rij89]. Early attempts viewed aboutness as being a model-theoretic relation, that is a document was considered as a sort of model in which the query was interpreted [BH94]. More recent investigations have shown that aboutness is similar in many ways to nonmonotonic consequence [BL98, Bru96, BSW00, WSBC01, AG96]. For example, an information carrier i is deemed to be about information carrier j, denoted $i \models_a j$ if the information borne by j holds in i. In other words, information carrier j is a summary or an abstraction of information carrier i.

Information Preclusion: Not all information carriers can be meaningfully composed because the information that they carry is contradictory. In general, information carriers i and j are said to preclude each other, denoted $i \perp j$ [BH94]. It is natural to assume that any fact precludes its negation (i.e. $i \perp \neg i$). This is the concept of logical consistency in classical logic. Within IR, information carriers naturally preclude each other with respect to the information need of a user [Bru96]. For instance, *apple* $\perp$ *orange* if the user just wants to retrieve information about *apples* rather than *oranges*.

Adaptive information agents are intelligent agents which hold beliefs about retrieval contexts and predict the relevance of documents with respect to these beliefs. Since retrieval contexts will change over time, the agents' beliefs about these retrieval contexts must also be revised promptly and appropriately. The AGM belief functions provide a robust formalism to model the learning components of adaptive information agents. After presenting a document to a user, an information agent will receive the user's relevance feedback α about the document. This feedback information is used to refine the agent's beliefs about the retrieval context, and the process is modelled by the AGM belief revision function K^*_α. After obtaining the latest information about a retrieval context, the information agent decides if certain documents should be retrieved for its users. This process is underpinned by the expectation inference relation $K \mid\!\sim_K d$, where d is the logical representation of a document. The application of expectation inference to adaptive IR is slightly different from its usage in a theoretical context. For instance, the emphasis is not on $\alpha \mid\!\sim_K \beta$, where α and β can be viewed as an individual query and the representation of a document respectively. Since adaptive IR is concerned about retrieving documents with respect to a set of long-term recurring queries, it makes sense to evaluate $K \mid\!\sim_K d$, where K represents a retrieval context which comprises all the related queries. The following discussion is based on the work presented in [LtHB99].

An example is used to illustrate the belief revision process at the conceptual level. If a user is interested in documents about *Japanese*, *Buddhism*, and *Sushi*, her initial information needs can be represented by a belief set: $K = \{japanese, buddhism, sushi\}$.

If the user's information preference shifts from *Japanese* to *English* later on, the information agent will employ the belief revision function to revise the belief of *English* to its belief set i.e. $K^*_{english}$. It is assumed that *English* $\perp$ *Japanese* is true in this context. This information preclusion relation can be detected by observing a positive feedback from a document containing information carrier *English*, and a negative feedback from a document containing information carrier *Japanese*. Because of *English* $\perp$ *Japanese*, a contraction operation $K^-_{japanese}$ must first be invoked to remove the belief *Japanese* from the belief set. In general, to implement preferential preclusion ($\perp$), both the belief revision function and the belief contraction function are involved. The IR process can be expressed in terms of belief revision operations such as $i \perp j \equiv (K^-_j)^*_i$, where $i \in IC_+$ and $j \in IC_-$. The AGM rationale of minimal and consistent changes is quite applicable in adaptive IR. With reference to this example, after incorporating the new belief *English* into the agent's knowledge base, the beliefs of *Sushi* and *Buddhism* should remain because the user is still interested in this information. Moreover, if the belief *english* is in the belief set, the belief $\neg$*english* should not be there. It does not make sense to retrieve documents about *English* and not to retrieve documents about *English* at the same time. The AGM belief revision logic is able to maintain the desirable properties exhibited in adaptive IR processes. Since epistemic entrenchment is used to construct the AGM belief functions, its validity in the context of IR should be examined before applying this formalism to develop adaptive information agents. The five postulates of epistemic entrenchment are examined with respect to the fundamental IR concepts. In general, beliefs are taken as information carriers, and epistemic entrenchment orderings can be interpreted as preference orderings over information carriers in the context of IR. Then, a set of information carriers K is used to partially capture a retrieval context.

(EE1): $\forall a, b, c \in K: \ a \leqslant b \leqslant c$ implies $a \leqslant c$ (transitivity)

In IR, it is believed that a user's information need imposes a preferential ordering on the underlying set of documents [BL98]. Furthermore, it is assumed that the preference relation is irreflexive and transitive. For example, if an information seeker prefers document c over b, and document b over a, then it means that she prefers document c over a. As documents are in fact information carriers, it implies that a transitive preference relation exists among information carriers. Therefore, EE1 is valid in the context of IR.

(EE2): $\forall a, b \in K: \ a \vdash b$ implies $a \leqslant b$ (dominance)

To examine this property in the context of IR, the classical ($\vdash$) derivability relation must first be interpreted in terms of IR concepts. The aboutness relation ($\models_a$) in IR seems a counterpart of the derivability relation in classical logic. The left hand side of the aboutness relation represents a specific information carrier and the right hand side of the aboutness relation is an abstraction about the information carrier. For example, given an aboutness relation such as *salmon* $\models_a$ *fish*, is $salmon \leqslant fish$? If we interpret $\leqslant$ in terms of defeasibility, will "salmon" be a more defeasible information carrier than "fish"? In a query refinement situation, assuming that a user is interested in different kinds of fish, she may try out different information carriers like "salmon", "tuna", "bream", etc. until she finally receives some relevant information from the IR system. In other words, her beliefs about the information carriers change from "salmon" to "tuna", and from "tuna" to "bream". Nevertheless, the information need is still represented by the information carrier "fish". Therefore, the information carrier of "fish" is less defeasible than either "salmon", "tuna", or "bream". So, in a query refinement situation, the aboutness relation demonstrates the characteristic as described by EE2.

Brooks has conducted a phenomenological study about human perception in text-based objects [Bro95]. This study may provide further support of the validity of (EE2) among information carriers. In this study, hierarchical thesauri capturing the semantic relationships such as "generalisation" and "specialisation" among texts were used to evaluate human perception about text relevance given the generalisation or specialisation transformations of texts. These semantic relationships are essentially the *information containment* relations $\rightarrow_T$ discussed in the context of IR. For example, $SoftwareAgent \rightarrow_T DistributedAI \rightarrow_T ComputerSciences \rightarrow_T Sciences.$ In Brooks' study, it was found that the perceived relevance of text would be broken approximately after two semantic steps. For instance, if a user perceives that the information carrier $SoftwareAgent$ is relevant to her needs, both $DistributedAI$

and $ComputerSciences$ may also be considered as relevant (e.g., 2 steps). However, $Sciences$ in general will not be considered as relevant with respect to her needs. In addition, it was found that perceived relevance would be broken immediately if the transformation between information carriers was conducted from the opposite direction [Bro95]. For instance, from $DistributedAI$ to $SoftwareAgent$, a user may find that $SoftwareAgent$ is not really about her preference of $DistributedAI$ which implies other topics such as $DistributedConstraintSatisfaction$ more relevant to her specific interest. Therefore, if the derivability relation appearing in (EE2) is interpreted as the information containment relation, such a postulate captures an information seeker's preference over information carriers. In other words, if $Salmon \rightarrow_T Fish$ is true, $Salmon \leqslant Fish$ can be established because she prefers $Fish$ as much as $Salmon$ given the fact that she likes $Salmon$.

(EE3): $\forall a, b \in K: \; a \leqslant a \wedge b$ or $b \leqslant a \wedge b$ (conjunctiveness)

This property can be linked to the concept of *specificity* in IR. From an IR point of view, more specific terms should generally produce higher precision results. Therefore, if given a choice of information carrier a or information carrier $a \oplus b$ for describing a user's information need, $a \oplus b$ should be the preferred representation of the user's information needs. So, $a \leqslant a \oplus b$ or $b \leqslant a \oplus b$ matches the characteristic of precision oriented IR. Nevertheless, in the context of IR, we must be careful about the semantic clash between information carriers. For example, if $a \perp b$, $a \oplus b$ certainly will not be more useful than a alone. So, it is necessary to add such a condition to the original EE3 if we want to develop the postulates of epistemic entrenchment pertaining to information carriers.

However, the combined result of (EE1) to (EE3) shows that $a = a \wedge b$ or $b = a \wedge b$ is true. In general, the assumption that both information carrier a and the information carrier $a \oplus b$ are about a user's information need is difficult to establish since the user's information need is contingent. For example, if the user actually prefers more general information, either a or b will be a better representation of her need. Hence, in a recall oriented situation[1], it may not be appropriate to state that $a \leqslant a \oplus b$ or $b \leqslant a \oplus b$. The property can only be generalised in that any information carrier a is as useful or entrenched as itself plus another arbitrary information carrier i.e. $a = a \oplus b$ as long as their meanings do not clash.

(EE4): If $K \neq K_{\perp}$, $a \notin K$ iff $\forall b \in K : a \leqslant b$ (minimality)

If a set of information carriers $K = \{$Japanese, Buddhism, Sushi $\}$ is used to represent a user's information needs, the information carrier *Japanese* should only be removed from this set if the user is no longer interested in information about Japanese. In other words, *Japanese* is the least preferred information carrier in the set. So, in general, an information carrier should only be contracted from a belief set if it is the least entrenched information carrier when compared with all other information

[1] This is an IR situation in which the user is interested in retrieving as many relevant information carriers as possible with respect to the given information need.

carriers in the set. This conforms to the property of EE4.

(EE5): $\forall b \in K : b \leqslant a$ implies $\vdash a$ (maximality)

This is another property which can be used to capture the special case in IR. Validity ($\vdash a$) can be interpreted as information carrier a being true in all retrieval situations. The concept of validity can be used by an information agent to handle special information requirements from a user. For instance, if a user wants to specify a query that should not be discarded by the agent under any circumstances, she can assign the maximal entrenchment degree to the corresponding information carriers with respect to an epistemic entrenchment ordering. As these information carriers will be treated as valid formulae by the belief revision formalism, they will be retained in the belief set until the user makes an explicit request to delete them.

In summary, the five postulates of epistemic entrenchment can be translated to the following counterparts which characterise the preference ordering among information carriers in the context of IR:

(IC-EE1): $\forall i, j, k \in \mathbb{L}_T : \ i \leqslant j \leqslant k$ implies $i \leqslant k$

(IC-EE2): $\forall i, j \in \mathbb{L}_T : \ i \models_a j$ implies $i \leqslant j$

(IC-EE3): $\forall i, j \in \mathbb{L}_T :$ if $i \not\leqslant j$, $i \leqslant i \oplus j$ or $j \leqslant i \oplus j$

(IC-EE4): If $K \neq K_{\perp}$, $i \notin K$ iff $\forall j \in \mathbb{L}_T : i \leqslant j$

(IC-EE5): $\forall j \in \mathbb{L}_T : j \leqslant i$ implies $\vdash i$

The close resemblance of the postulates (e.g., IC-EE1 to IC-EE5) characterising the preference orderings of information carriers in IR and that characterising the epistemic entrenchment orderings of beliefs in rational agents provides the theoretical basis to apply the AGM belief revision framework to model changing retrieval contexts and represent these changes as transitions among epistemic states in adaptive information agents. In fact, the work reported in this section is the first attempt of evaluating the validity of the AGM belief functions in the context of IR by analysing the postulates of epistemic entrenchment orderings which underpin the belief functions.

Chapter 4

An Agent-Based Information Filtering System

This chapter illustrates how the AGM belief revision logic is applied to develop adaptive information agents. In particular, the learning and the classification functions of the agents are examined at the computational level. An overview of an agent-based information filtering system (AIFS) is first provided. Issues regarding how to represent documents and users' information needs are then discussed. The computational algorithm which implements the AGM belief functions is discussed. Finally, the learning and the classification (prediction) mechanisms of the adaptive information agents are explained and highlighted with some examples.

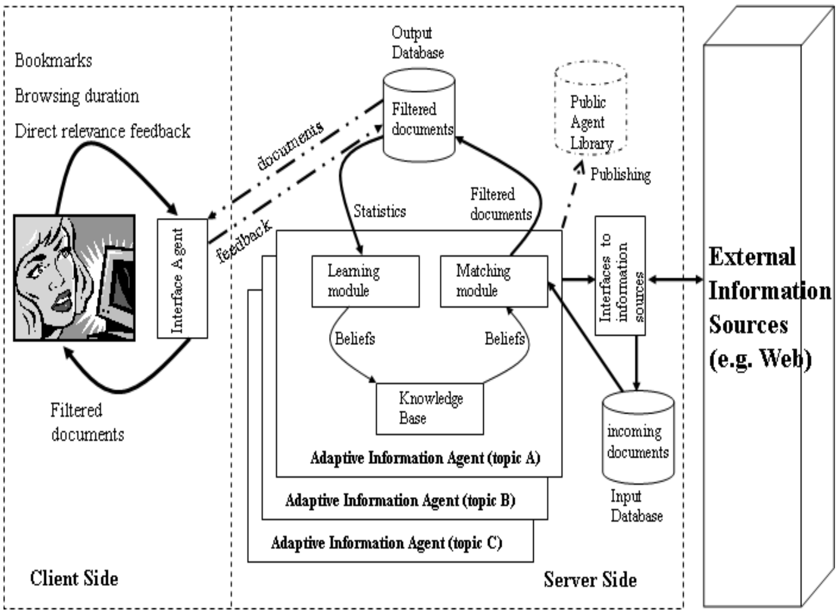

Figure 4.1: The System Architecture of AIFS

4.1 System Architecture

Figure 4.1 depicts the system architecture of an agent-based adaptive information filtering system (AIFS). An interface agent is situated on the client side to communicate with a user. For each information topic of interest, the user instructs the interface agent to instantiate an adaptive information agent on the server side. Therefore, there could be a number of adaptive information agents serving a single user at the same time. However, from the user's point of view, it is a single encapsulated adaptive information agent. The *matching module* of an adaptive information agent compares the logical representation d of each incoming information object Doc (i.e., a document) with the representation K (i.e., the agent's knowledge base) of a retrieval context Ctx. A retrieval context refers to a user's information needs and their background knowledge about a retrieval domain. If there is a sufficiently close match between d and K, the document together with the agent's unique ID will be transferred to the output database. Therefore, every filtered document is associated with the agent who recommends the document. Periodically, an interface agent extracts the filtered documents from the output database and presents these documents to the user based on the matching agent IDs. Presentations of the system's filtering results in the form of summaries are also supported. In this mode, the interface agent only deliver lists of document headings (and URLs for Web pages) to the user. After viewing a particular document, the user may choose to save the document or invoke the feedback mechanism to rate the document. The interface agents also observe the duration that a document is viewed on the display window to infer the relevance of a document. If the review time of a document exceeds a pre-defined threshold, the interface agents will infer that the user considers the document as relevant.

Manual or inferred feedback information is then transferred to the server side and stored along with the corresponding document representation in the output database. At each learning cycle (e.g. after n filtered documents are viewed by a user), the *learning module* of an information agent is activated to analyse the relevance judgement information stored in the output database. The resulting statistical data is used to induce beliefs about a user's interests pertaining to a particular topic. The beliefs are then revised into the corresponding agent's knowledge base through the belief revision operations. In particular, these beliefs which represent the agent's perception of a retrieval context are revised in a *minimal* and *consistent* fashion. Moreover, an off-line process is invoked regularly to mine the term association rules [JBB00] and the information preclusion relations [Bru96] from the output database. These rules are also revised into the agents' knowledge bases through belief revision operations. In AIFS, collaborative filtering is also supported. An information agent trained by a user can be deployed to the public agent library. New users of the system can search for information agents specializing in particular information topics from the agent library. Therefore, archived information agents can recommend documents to new users based on the preferences of similar users. The kernel module (i.e., the learning and the matching modules of adaptive information agents) of AIFS has been implemented. The system was evaluated based on the TREC-AP collection and the Reuters-21578 collection. Details of these experiments are provided in the next chapter.

4.2 Document Representation

Conceptually, there are two levels of document representation in AIFS. At the *physical level*, a document is characterised by a set of terms. Such a term-based representation is commonly found in IR systems [Sal89, SM83]. In the current prototype system, a term is a keyword extracted from a document. At the *symbolic level*, a document is represented by a set of atoms of a classical logic language $\mathcal{L}$. After an information object is retrieved from an external source, the AIFS system will parse the object to extract the text elements. For instance, video, audio, and executable codes are removed at this stage. In addition, non-informative elements such as HTML tags are ignored. The result is a plain text file without any mark-up tags, images, nor embedded executable objects. The stop word removal procedure is followed to remove insignificant common words based on a pre-defined stop word list. The stop word list used in AIFS is developed based on the dictionary found in the SMART system [Sal90] (`ftp://ftp.cs.cornell.edu/pub/smart/`). All text is then folded to lower cases. Non-alphabetic characters are removed from a word because our theorem prover cannot deal with special characters. A stemming procedure is then applied to compute the root form of each word by applying Porter's stemming algorithm [Por80]. For instance, the terms *computer*, *computing*, *computation* are all transformed to *comput* after a stemming process. The TFIDF weighting scheme (also called the "atc" weight in SMART) is applied to compute the TFIDF value of each term [SB88]. In particular, Eq.(2.1) illustrated in Section 2.1 of Chapter 2 is used to compute the term weights. According to previous research, using a small subset of terms to represent a document has led to improved retrieval performance [BS95, Bal97, PB97]. Therefore, only the top n tokens ranked by the TFIDF weights are used as the initial representation of a document. The parameter n is derived by applying a percentage π to the average length of documents cached in AIFS.

At the symbolic level of document representation, each term t present at the physical level is mapped to the ground term of the *positive keyword* predicate (i.e., $pkw(t)$) if the chosen representation language is predicate logic. The intended interpretation of these atoms such as $pkw(t)$ is that they are satisfied in a document Doc (i.e., $Doc \models pkw(t)$) if Doc is taken as a model [CC92, Lal98]. For example, if $Doc = \{text, agent, web, \ldots\}$ is the document representation at the physical level, the corresponding symbolic representation will be $d = \{pkw(text), pkw(agent), pkw(web), \ldots\}$. In Losada and Barreiro's logical IR model [LB99], positive literals of $\mathcal{L}$ represent tokens which are *about* a document, whereas negated literals represent tokens which are not about a document. Nevertheless, in practice, there are usually a large number of tokens which are not about the content of a particular document. Therefore, if negated literals are used to represent documents, it may lead to serious representational and computational problems. Moreover, given the fact that only imperfect characterisations of documents can be achieved [Lal98], it is very difficult to distinguish if a token is not about a document or it is a missing descriptor of the document. The proposed document representation scheme acknowledges the problem of *partiality* in document representation [Lal98]. The uncertainties arising from matching the imperfect characterisations of documents with the partial representations of retrieval contexts are managed through the belief revision operations and the related expectation inference mechanisms. For a more efficient implementation, a term present at the physical level is translated to a propositional letter of a classical propositional language. In fact, it is obvious that if a term t is mapped to the ground term of a predicate $pkw(t)$, it is equivalent to a proposition because the interpretation of the formula is either true or false dependent on whether the term t is contained in the document or not. Therefore, the computationally more expensive first order representation may not bring extra benefit for document representation when compared with the propositional representation. The implemented prototype system supports both the first order and the propositional document representations. However, all the experiments reported in this thesis are based on the propositional document representation. In particular, the classical propositional Horn language $\mathcal{L}_{Horn}$ is used to represent documents and retrieval contexts. With reference to the previous document example, the propositional representation of the document at the symbolic level is simply $d = \{text, agent, web, \ldots\}$. In other words, it is a direct translation from a term to a propositional letter with the interpretation that the proposition is satisfied with respect to the associated document. For a practical implementation of an adaptive IR system, each document representation is augmented with some extra information such as the title of a document, the name of an author, the URL (for a Web document), etc. to facilitate subsequent retrieval of the document.

4.3 Induction of Epistemic Entrenchment Orderings

The AGM belief revision functions and the corresponding expectation inference relations are constructed based on the epistemic entrenchment orderings of beliefs [GM88, GM94]. Therefore, the first step towards building the learning and the classification mechanisms of adaptive information agents is to develop an automated means of inducing the epistemic entrenchment orderings. From the classification point of view, the purpose of entrenchment induction is to identify highly entrenched *beliefs* or *disbeliefs* about a user's information needs so that an information agent can draw sensible conclusions about the relevance of documents with respect to these beliefs. In this sense, the process of entrenchment induction is similar to the process of *feature selection*, which identifies the most prominent subset of features for learning and classification, in the context of machine learning [YP97]. Intuitively, if a term often appears in the set of documents D^+ judged as relevant by a user, it is a good indicator of the user's positive interest [KYMW97]. Accordingly, these positive terms or keywords become the agent's beliefs about the user's information needs. In addition, if a term frequently appears in the set of non-relevant documents D^- judged by the user, it becomes a *disbelief* in the agent's knowledge base.

The search for an effective entrenchment induction method stems from the area of information theory [Man87, Los99]. In fact, Gärdenfors also pointed out that it would be possible to develop a quantitative evaluations of the "degree of change" based on information theoretic measures (e.g., based on the concept of entropy) [Gär88]. The amount of information I carried by an event e can be measured in terms of bits: $I(e) = -\log_2 Pr(e)$ where $I(e)$ is the information content of an event e and $Pr(e)$ is the probability that the event e occurs. The expected amount of information generated from a system S which consists of multiple events e_i is measured by the *entropy* $H(S)$ and is defined by: $H(S) = -\sum_i Pr(e_i)\log_2 Pr(e_i)$ where $Pr(e_i)$ is the probability of the occurrence of an event e_i in a system S. In addition, the notion of *mutual information* (MI) between two events x and y is used to measure the interdependency between these events and is defined by: $\log_2 \frac{Pr(x \wedge y)}{Pr(x)Pr(y)}$. In the context of IR, MI is often used to measure the association between terms or the dependency between a term t and a class $c \in \{\text{relevant}, \text{non-relevant}\}$. In particular, mutual information for text categorisation tasks is defined as [YP97]:

$$MI(t,c) = \log_2 \frac{Pr(t \wedge c)}{Pr(t)Pr(c)} \tag{4.1}$$

where $MI(t,c)$ is the mutual information between a term t and a class c, and $Pr(t \wedge c)$ is the joint probability that a term appears in a document with a class label c (e.g., relevant or non-relevant). This formulation is suitable for IR tasks because there could be a large number of terms not appearing in a document (i.e., $Pr(\neg t)$), and their absence does not contribute much to the process of classification. Accordingly, the focus is on the mutual information $MI(t, relevant)$ between the presence of a term t and the relevant class rather than $MI(\neg t, c)$. It is interesting to find that the MI measure coincides with the interpretation of entrenched beliefs in the proposed adaptive agent framework. For instance, if a term has strong association with the set of relevant documents (i.e., $MI(t, relevant)$), it becomes a strong belief for representing a user's information need. Based on the notion of entropy, *cross entropy*, also called conditional mutual information, is defined by: $CMI(x_i, y) = \sum_j Pr(y_j|x_i) \log_2 \frac{Pr(y_j|x_i)}{Pr(y_j)}$. Then, *expected cross entropy* $EH(x,y)$ is defined by: $\sum_i Pr(x_i) \sum_j Pr(y_j|x_i) \log_2 \frac{Pr(y_j|x_i)}{Pr(y_j)}$. In fact, expected cross entropy is also referred to as *information gain* in the machine learning research community [Qui86]. So, the following equivalence relation is established: $EH(x,y) = InformationGain(x,y) = H(x) - H(x|y)$ where $H(x|y)$ is the conditional entropy of x given y; the variable y normally refers to a specific feature. Expected cross entropy has been applied to binary text classification problem and the formulation in such a context is [KS97]:

$$EH(t,C) = Pr(t) \sum_{c \in C} Pr(c|t) \log_2 \frac{Pr(c|t)}{Pr(c)} \tag{4.2}$$

where $EH(t,C)$ is the expected cross entropy for a term t with respect to two classes $C = \{\text{relevant}, \text{non-relevant}\}$, and $Pr(c|t)$ is the conditional probability that a document d is associated with a particular class label $c \in C$ given that the term t appears in d. The main difference between the formulation in Eq.(4.2) and the general notion of expected cross entropy $EH(x,y)$ is that Eq.(4.2) (expected cross entropy for text classification) is only normalised by the probability of term appearance instead of averaging the cross entropy CMI by term presence and term absence. Again the intuition of such a formulation is that many terms are not contained in a document. Considering term absence may only increase computational complexity without improving classification accuracy. Because of the success of expected cross entropy for text classification Eq.(4.2), this measure is considered as one of the candidates for entrenchment induction.

Based on the statistical method of Kullback divergence, which is often used to measure the distance between two probability distributions, a measure called *Keyword*

Classifier KC was developed for adaptive text filtering [KYMW97]. The keyword classifier was used to distinguish among *positive keywords* which represent a user's positive information interests, *negative keywords* which indicate what the user dislikes, and *neutral keywords* which are not good indicators of what the user likes or dislikes. Formally, the measure of KC is defined by:

$$KC(t) = \tanh\left(\frac{df(t)}{\alpha}\right) \times \left[Pr(c_1|t)\log_2\frac{Pr(c_1|t)}{Pr(c_1)} - Pr(c_2|t)\log_2\frac{Pr(c_2|t)}{Pr(c_2)}\right] \quad (4.3)$$

where $df(t)$ is the document frequency of a term t and it is simply the number of documents containing t in the collection. The term α is a user defined parameter to control the learning rate. The class value c_1 represents the relevant class and the class value c_2 represents the non-relevant class. The conditional probability $Pr(c_1|t)$ is the estimated probability that a document is relevant given that the term t appears in the document. Observe that the two terms inside the square brackets in Eq.(4.3) are exactly the same elements to be summed in Eq.(4.2) (expected cross entropy for text classification). The only difference is that a substraction instead of an addition is applied to these terms in Eq.(4.3). This similarity may not be purely driven by coincidence, but rather the adoption of slightly different views to model the same reality.

It is believed that the *probability ranking principle* [Rob77] is one of the most influential principles within information retrieval theory [LC01]. This principle suggests that the ranking of documents should be computed based on the odds of the estimated conditional probabilities $Pr(d|c_1)$ and $Pr(d|c_2)$ where c_1 and c_2 represent the relevant class and the non-relevant class respectively. Along the same line of wisdom, the *Odds Ratio* OR was proposed to predict the class values given the presence of a term t in a document [vRHP81]. In particular, Odds Ratio is used to rank documents with respect to a given query based on the appearance of some terms in the documents. Such a ranking is derived by: $R(d, c_1) = \log_2 \frac{Pr(c_1|d)}{Pr(c_2|d)} = \log_2 \frac{Pr(c_1) \prod_j Pr(t_j|c_1)}{Pr(c_2) \prod_j Pr(t_j|c_2)} = \sum_j OR(t_j)m + k$, where $OR(t_j)$ is the odds ratio for a term t_j contained in a document d and m is a Boolean variable indicating if a term appears in the document $(m = 1)$ or not $(m = 0)$. The term k defines a constant to establish the baseline of the document scores. The odds ratio $OR(t)$ for a term t is formally defined by:

$$OR(t) = \log_2 \frac{odds(t|c_1)}{odds(t|c_2)} \tag{4.4}$$

$$odds(x) = \begin{cases} \frac{Pr(x)}{1-Pr(x)} & \text{if } Pr(x) \neq 0 \wedge Pr(x) \neq 1 \\ \frac{\frac{1}{n^2}}{1-\frac{1}{n^2}} & \text{if } Pr(x) = 0 \\ \frac{1-\frac{1}{n^2}}{\frac{1}{n^2}} & \text{if } Pr(x) = 1 \end{cases} \tag{4.5}$$

where $odds(x)$ is the odds for an event x, and n is the total number of training examples (i.e., documents with relevance judgement). In the context of adaptive information agents, n denotes the number of documents viewed by a user.

The candidate methods which are considered for entrenchment induction so far include Expected Cross Entropy for text Eq.(4.2), Mutual Information Eq.(4.1), Keyword Classifier Eq.(4.3), and Odds Ratio Eq.(4.4). As epistemic entrenchment degrees are defined in the unit interval $[0, 1]$, the following formula is used to normalise the raw term score $S(t)$ computed according to the aforementioned measures to the unit interval:

$$SS(t) = \frac{S(t) - S(t)_{min}}{S(t)_{max} - S(t)_{min}} \tag{4.6}$$

where $SS(t)$ is the *scaled* term score and $S(t)$ is the raw term score as derived from one of the candidate methods for entrenchment induction. $S(t)_{max}$ and $S(t)_{min}$ represent the maximal term score and the minimal term score respectively. These values are estimated based on a trial run over the entire document collection. Apart from these candidate methods, the TFIDF measure as defined in Eq.(2.1) is also considered for the task of entrenchment induction. As the TFIDF vector associated with each document is subject to cosine normalisation, it is not necessary to apply Eq.(4.6) to scale the term weights. The standard Rocchio method is used to revise the TFIDF weights of terms. Based on an initial query (e.g., a topic description), a set of positive documents and a set of negative documents, the top n terms ranked by normalised TFIDF weights in the prototype vector are converted to a set of beliefs in an agent's knowledge base. The Rocchio learning method is defined by [Roc71]:

$$\overrightarrow{Q}_{i+1} = \alpha \overrightarrow{Q}_i + \beta \frac{1}{|R|} \sum_{d \in R} \overrightarrow{d} - \gamma \frac{1}{|N-R|} \sum_{d \notin R} \overrightarrow{d} \tag{4.7}$$

where standard parameters (e.g., $\alpha = 1, \beta = 0.75, \gamma = 0.25$) were applied to the Rocchio formula in our experiments. $\overrightarrow{Q}_{i+1}$ is the prototype vector (i.e., a user profile) at time point $i+1$ and $\overrightarrow{Q}_i$ is the prototype vector containing the initial term weights. The term $|R|$ represents the cardinality of the set R of relevant documents judged by a user and the set N represents the total number of documents parsed in a learning cycle. So, $N - R$ is the set of non-relevant documents. If an agent updates its prototype vector $\overrightarrow{Q}_i$ after processing each training document, there is no need to compute the average weights. Therefore, the factors $\frac{1}{|R|}$ and $\frac{1}{|N-R|}$ are not applied. In the experiment related to the Rocchio-based entrenchment induction method, the learning cycle was set to 500 (i.e., $N = 500$). Such a value is derived according to several trial runs for balancing between computational time and retrieval effectiveness. New terms found in a positive training document are used to expand the prototype vector $\overrightarrow{Q}_{i+1}$.

4.4 Representing Users' Information Needs

A retrieval context is mainly characterised by a user's information needs. The user's information needs or preferences are formally represented by the epistemic entrenchment orderings of beliefs in adaptive information agents. At the implementation level, epistemic entrenchment orderings are represented by *finite partial entrenchment ranking* ($\mathbf{B}$) that ranked the sentences of a theory in $\mathcal{L}_{Horn}$ with the minimum possible degree of entrenchment ($\leqslant_{\mathbf{B}}$). Section 4.3 described the intuition and some candidate methods for entrenchment induction. This section describes the standard entrenchment induction method used in the current prototype system and gives a complete example of how to represent a user's information needs in an agent's knowledge base. Figure 4.2 visualises a sample of 10 training documents and the distribution of these documents in D^+ and D^- respectively. Each document represented by a rectangle box contains a set of terms such as $\{business, insurance, system, \ldots\}$. This small training set stored in AIFS's output database will be used for the entrenchment induction example discussed in this section.

Although it was found that the keyword classifier KC performed well for some text filtering tasks [KYMW97], our current experiments show that a modified version of the keyword classifier called MKC is the most effective one among the candidates for entrenchment induction since the MKC method can take into account asymmetric class value distribution typically found in information filtering applications. The details about the empirical evaluations of all the candidate methods are reported in Chapter 5. The MKC method defined below is the default method used in the current prototype system of AIFS:

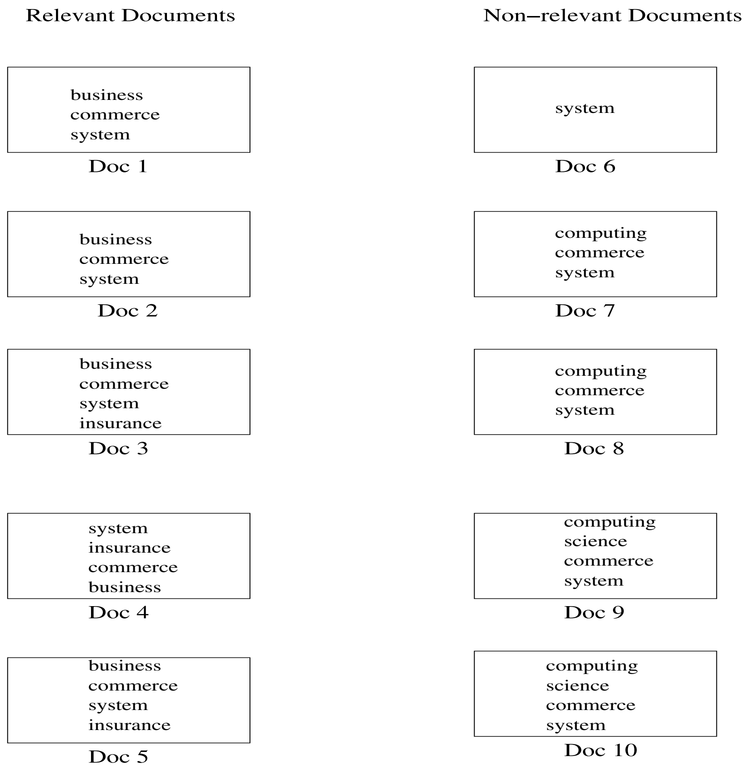

Figure 4.2: Relevant doc. D^+ and Non-relevant doc. D^-

$$MKC(t) = \epsilon \times \tanh\left[\frac{df(t)}{\alpha} \times Pr(c_1|t)\log_2\frac{Pr(c_1|t)}{Pr(c_1)} - \frac{df(t)}{\beta}Pr(c_2|t)\log_2\frac{Pr(c_2|t)}{Pr(c_2)}\right] \tag{4.8}$$

α and β are the learning thresholds for positive terms and negative terms respectively. The negative learning threshold is defined by: $\beta = \frac{Pr(c_2)}{Pr(c_1)} \times \alpha$. In other words, the negative learning threshold is proportional to the estimated probability that an arbitrary document is non-relevant and inversely proportional to the estimated probability that an arbitrary document is relevant. The term $\frac{df(t)}{\alpha}$ or $\frac{df(t)}{\beta}$ is used to select the very positive or negative keywords for belief generation. The hyperbolic tangent function tanh projects the MKC values into the interval $[-1, 1]$. The adjustment factor ϵ ensures that all the entrenchment degrees induced are less than the maximal degree (e.g., 1) because beliefs induced in this way are contractable (defeasible) from the agents' point of view. $Pr(c_1|t) = \frac{df(t_{rel})}{df(t)}$ is the estimated conditional probability that a document is relevant (i.e., class c_1) given that it contains the term t. It is expressed as the fraction of the number of relevant documents which contain the term t over the total number of documents which contain t. Similarly, $Pr(c_2|t) = \frac{df(t_{nrel})}{df(t)}$ is the estimated conditional probability that a document is non-relevant (i.e., class c_2) if it contains the term t. In addition, $Pr(c_1) = \frac{|D^+|}{|D^+|+|D^-|}$ is the estimated probability that a document recommended by an agent is relevant, and $Pr(c_1) = \frac{|D^-|}{|D^+|+|D^-|}$ is the estimated probability that a document is non-relevant. Strictly speaking, a term score returned by $MKC(t)$ should be interpreted as the preference value of the term t driven by a user's specific information needs. According to the definition of finite partial entrenchment ranking defined in Chapter 3, entrenchment degrees are in the unit interval [0,1]. So, it is necessary to convert the raw *preference* values induced by the MKC method to the corresponding epistemic entrenchment degrees. The entrenchment degree $\mathbf{B}(\alpha_t)$ of a belief α_t pertaining to a term t is derived by applying Eq.(4.9) to the corresponding preference value returned by $MKC(t)$. Moreover, to improve computational efficiency, the preference values of terms are compared with a preference threshold λ such that only significant beliefs are induced and revised into the agents' knowledge bases. This procedure is essential for a practical implementation of the belief revision formalism since each belief revision operation is computationally

Terms	$df(t_{rel})$	$df(t_{nrel})$	$MKC(t)$	Formula: α_t	$\mathbf{B}(\alpha_t)$
business	5	0	0.724	$business$	0.605
computing	0	4	-0.631	$\neg computing$	0.473
insurance	3	0	0.510	$insurance$	0.300
science	0	2	-0.361	$\neg science$	0.087
commerce	5	4	0.266	-	-
system	5	5	0	-	-

Table 4.1: Induction of Preference Values by MKC

expensive, and hence the number of revisions should be minimised. According to our empirical study, a large number of trivial belief revision operations are saved if the system focuses on a subset of highly entrenched beliefs. By using an extra filter to remove noisy features, the agents' classification accuracy may be improved because only the reliable information is used to infer document relevance. The minimum entrenchment degree $\mathbf{B}(\alpha_t)$ of an explicit belief α_t representing a user's preference for a term t is derived by:

$$\mathbf{B}(\alpha) = \begin{cases} \frac{(|MKC(t)|-\lambda)}{1-\lambda} & \text{if } |MKC(t)| > \lambda \\ 0 & \text{otherwise} \end{cases} \tag{4.9}$$

A positive $MKC(t)$ implies that the associated term t is a positive keyword. The corresponding belief is represented by a positive literal of $\mathcal{L}_{Horn}$. If the representation language is a classical first order language, the token t will be mapped to the ground term of the pkw predicate (i.e., $pkw(t)$). Since our belief revision engine is language independent, the AIFS system can process beliefs represented by a propositional language or a first order language. However, the experiments reported in Chapter 5 are based on the classical propositional Horn language $\mathcal{L}_{Horn}$. A negative preference value indicates that t is a negative keyword, and the corresponding *disbelief* is represented by a negated proposition such as $\neg t$, or $\neg pkw(t)$ in the case of a first order representation. If the absolute preference value $|MKC(t)|$ is below a threshold value λ, the associated token is considered neutral. Neutral tokens are not represented in the agents' knowledge bases. For the examples described in this chapter, $\epsilon = 0.95$, $\lambda = 0.3$, and the learning threshold $\alpha = 5$ are assumed. Table 4.1 summarises the results of applying Eq.(4.8) and Eq.(4.9) to the training documents depicted in Figure 4.2. The cardinality of the positive training set equals that of the negative training set (i.e., $|D^+| = |D^-| = 5$). The first column in Table 4.1 shows the terms t extracted from the training documents. The second and the third columns show the frequencies of these terms in D^+ and D^- respectively. By applying Eq.(4.8) to the training examples shown in Figure 4.2, the preference value of each term t is computed and listed in the fourth column. The fifth column lists the beliefs induced from the training examples. The last column shows the entrenchment degrees $\mathbf{B}(\alpha_t)$ of the corresponding beliefs α_t. The entrenchment degrees of the beliefs *commerce* and *system* are zero because the preference values of these terms are below the preference threshold λ. The "-" in Table 4.1 indicates that the beliefs are not induced. If the information disclosed in Table 4.1 is used qualitatively, the induced finite partial entrenchment ranking looks like:

$$business > \neg computing > insurance > \neg science$$

Figure 4.3 depicts a learning interface of the current prototype system. The upper panel shows the parameters passed to an adaptive information agent. These parameters include the belief revision algorithm used, the learning thresholds (e.g., α and β), the frequency of learning (i.e., learning cycle), the preference threshold λ, the entrenchment adjustment factor ϵ, the revision sensitivity threshold, and the file names linked to the training document set. These files are used to store the documents as well as the relevance judgement information for the TREC-AP experiments. The revision sensitivity threshold is another mechanism used to minimise the computational cost of belief revision. Only those beliefs with an accumulated change of entrenchment degree greater than the sensitivity threshold since the previous learning cycle will be revised into an agent's knowledge base. The lower left panel in Figure 4.3 listed the ten training documents and the corresponding user's judgement. The lower right panel shows the content of the agent's theory base after learning the user's preferences.

4.5 The Rapid Anytime Maxi-Adjustment Algorithm

Inducing the epistemic entrenchment orderings based on users' preferences over documents is only the first step of a learning process in adaptive information agents. The agents actually learn the users' preferences by revising the corresponding beliefs into the agents' knowledge bases via the AGM belief revision operations in the

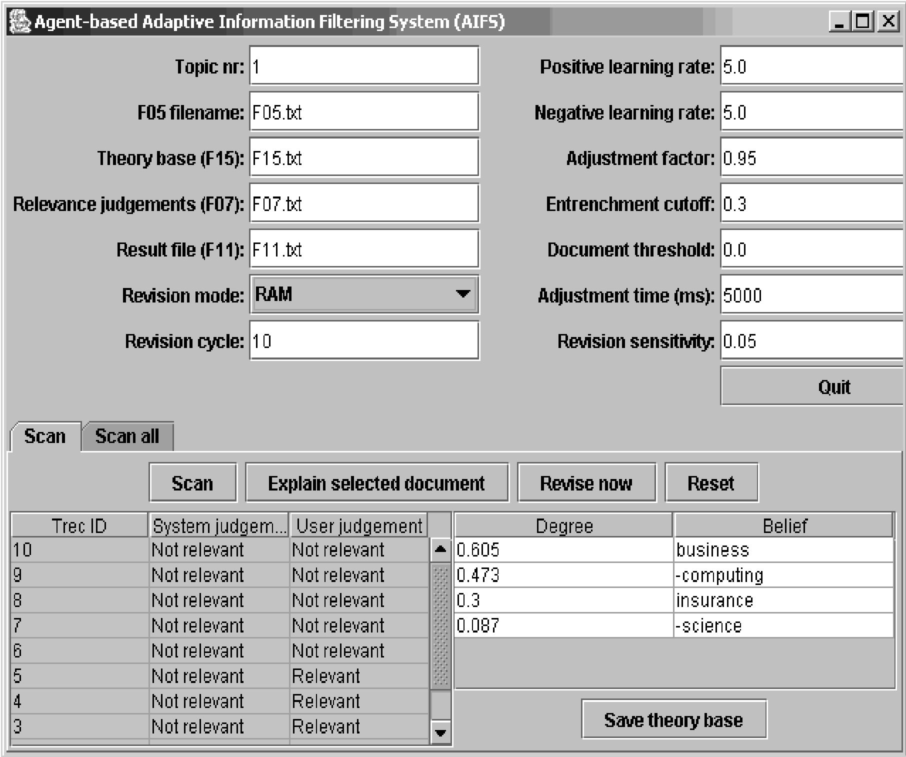

Figure 4.3: Inducing a user's information preferences

light of information derived from users' relevance feedback. At the computational level, the belief revision processes are not only taken as adding or removing beliefs from an agent's knowledge base but also the *transmutations* of the underlying finite partial entrenchment rankings **B**. Chapter 3 illustrated two transmutation methods, namely Maxi-adjustment and RAM. The Rapid Anytime Maxi-adjustment (RAM) method proposed in this thesis is an improvement over the original maxi-adjustment method developed by Williams [Wil96b]. The RAM method is faster than the maxi-adjustment method as demonstrated by our empirical testings reported in Chapter 5. Moreover, the RAM method still adheres to the AGM belief revision principles. For instance, the properties (PER1) - (PER3) of finite partial entrenchment rankings **B** and hence the postulates (EE1) - (EE5) of epistemic entrenchment are maintained for any entrenchment ranking **B** transmuted by the RAM method.

This section illustrates the details of the Rapid Anytime Maxi-adjustment algorithm implemented in the prototype agent system AIFS. Invoking the AGM belief functions and hence the RAM transmutation algorithm involves a classical theorem prover. Early attempts were made to construct our belief revision engine based on the SATEN belief revision system [WS00] which is equipped with a first order theorem prover called Vader. Unfortunately, some fundamental programming problems of the Vader theorem prover prevented us from doing so. Eventually, a brand new belief revision engine was developed on top of the SICStus Prolog system (`http://www.sics.se/ps/sicstus.html`), a commercially available Prolog system. The Prolog inference engine is the work horse to conduct classical theorem proving. Our Java-based agent system utilises the Jasper Java interface supported by SICStus Prolog to communicate with the SICStus inference engine.

The main function $RapidMaxi()$ of the RAM algorithm takes a finite partial entrenchment ranking $OldB$, a belief α, the new entrenchment degree $Ndegree$ of α, and a time limit in milli-seconds as inputs and returns a revised finite partial entrenchment ranking $NewB$ as output. The high level definition of the RAM method presented in Chapter 3 assumes that the belief α is a contingent sentence. The computational algorithm of the RAM method illustrated in this chapter can deal with the exceptional cases (e.g., tautologies). The $RapidMaxi()$ main function first computes the existing entrenchment degree *Odegree* of the belief α by invoking the *Degree* function. The *Degree* function is developed according to Definition 4 defined in Chapter 3. If the entrenchment degree of the corresponding disbelief $\neg\alpha$ equals the maximal degree (i.e., 1 in our current implementation), it means that $\vdash \neg\alpha$ is true. According to the AGM contraction function as defined by the (C-) condition, the corresponding belief set K (i.e., $content(\mathbf{B})$) will not be revised. Under such circumstance, the $RapidMaxi()$ function is terminated by returning the old theory base $OldB$. If this is not the case, the new entrenchment degree $Ndegree$ of α is compared with its existing degree $Odegree$ to determine if a revision function $Revision()$ or a contraction function $Contraction()$ should be called next. In either case, the algorithm exits by returning a new theory base $NewB$.

FUNCTION RapidMaxi(OldB, α, Ndegree, TimeLimit)

    Odegree := $Degree(OldB, \alpha)$

    REMARKS: MaxDegree = 1 in our implementation

    IF $Degree(OldB, \neg\alpha) =$ MaxDegree

        RETURN OldB

    ENDIF

    IF Ndegree $\geq$ Odegree

```
        NewB :=  Revision(OldB, α, Odegree, Ndegree, TimeLimit)
    ELSE
        NewB :=  Contraction(OldB, α, Odegree, Ndegree, TimeLimit)
    ENDIF
    RETURN NewB
END FUNCTION
```

The AGM belief revision function is implemented by the function $Revision()$ which raises the degree of a belief α to $Ndegree$. The $Revision()$ function first checks if the new entrenchment degree equals the minimal degree (i.e. 0 in our implementation). If it is true, a trivial revision is done by returning the existing theory base $OldB$. If the entrenchment degree of $\neg\alpha$ is greater than the minimal degree (i.e., $\neg\alpha \in K$), the contraction function $Contraction()$ must first be invoked to remove $\neg\alpha$ to ensure that the new belief set $K = content(\mathbf{B}_{NewB})$ remains consistent. If $\neg\alpha \notin K$ holds, a revision operation (i.e., raising the degree of α to $Ndegree$ based on $OldB$) is performed immediately. One of the main tasks of the $Revision()$ function is to identify and extract the problematic segment of beliefs $ProblemB$ from the existing theory base $OldB$. Therefore, the algorithm will transmute the entrenchment degrees of beliefs in the problematic segment with the help of a theorem prover. In this sense, $HighB$ and $LowB$ represent the segments of the existing theory base $OldB$ which are not affected by the belief revision operation. Therefore, beliefs from these segments are simply copied to the new theory base $NewB$. The $Cut()$ function extracts a segment of beliefs from a given theory base according to a starting point and an ending point expressed by the ranks of the beliefs residing at these points. The $Rank()$ function converts a given entrenchment degree to the corresponding rank with respect to a theory base. A portion of the new theory base $newB$ is safely constructed by $NewB := HighB + (\alpha, Ndegree)$ since the entrenchment degrees of the beliefs in $HighB$ are not affected by the belief revision operation. As such an operation does not invoke the theorem prover, it can be finished quickly. The FOR ... NEXT loop enumerates each element of $ProblemB$ and carries out the main revision function. For each looping, the $ElaspsedTime()$ function returns the elapsed time since the $Revision()$ function is executed, and this elapsed time is compared with the maximum duration $TimeLimit$ allowed for a belief revision operation. If the elapsed time exceeds the time limit, the main loop is terminated and the $Revision()$ function will return the unaffected theory base segments plus any revised beliefs from the problematic theory base segment. Therefore, the returned approximation of the new theory base $NewB$ is guaranteed to maintain the properties of finite partial entrenchment rankings. Essentially, for each sentence $\beta = ProblemB[x].belief$ from the problematic belief segment $ProblemB$, it is necessary to check if any beliefs ranked strictly higher than β can classically entail ($\vdash$) β. If this is the case, the property (PER1) of finite partial entrenchment rankings (i.e., the dominance property (EE2)) is violated, and so the minimal change to restore (PER1) is to raise the degree of β to $Ndegree$ or $degree(\mathbf{B}_{NewB}, \alpha \rightarrow \beta)$ depending on which one is closer to the existing degree of β.

The $Beliefs()$ function extracts a set of sentences from a given theory base segment. The $Proved()$ function returns true if a sentence (e.g., $ProblemB[x].belief$) is a logical consequence of the set of sentences (i.e., axioms) currently held in the theorem prover. The axioms are added to the theorem prover via the $AddAxioms()$ function.

```
FUNCTION  Revision(OldB, α, Odegree, Ndegree, TimeLimit)
    REMARKS: MinDegree = 0 in our implementation
    IF Ndegree = MinDegree
        RETURN OldB
    ENDIF
    NegDegree :=  Degree(OldB, ¬α)
    IF NegDegree > MinDegree
        OldB :=  Contraction(OldB, ¬α, NegDegree, MinDegree, TimeLimit)
    ENDIF
    REMARKS: Theory base without  ¬α
    IF Ndegree  > Odegree
        HighB := Cut(OldB, Rank(MaxDegree), Rank(Ndegree))
        ProblemB := Cut(OldB, Rank(Ndegree)+1, Rank(Odegree))
```

```
LowB := Cut(OldB, Rank(Odegree)+1, Rank(MinDegree))

NewB := HighB  +  (α, Ndegree)

FOR x := 1 TO NoElements(ProblemB)

     IF ElapsedTime() > TimeLimit AND TimeLimit > 0

        EXIT

     ENDIF

     IF ProblemB[x].belief =  α

        SKIP NEXT

     ENDIF

     prover := NEW TheoremProver()

     AddAxioms(prover, Beliefs(

                  Cut(NewB, Rank(MaxDegree), Rank(Ndegree))))

     IF Proved(prover, ProblemB[x].belief)

         ProblemB[x].degree := Ndegree

         NewB := NewB + ProblemB[x]

     ELSE

         REMARKS: beliefs ranked higher than ProblemB[x].belief
```

```
            FOR y := 1 TO x - 1

                IF  NOT Exist(NewB, ProblemB[y].belief)

                    AddAxioms(prover, ProblemB[y].belief)

                ENDIF

                IF  Proved(prover, ProblemB[x].belief)

                    ProblemB[x].degree := ProblemB[y].degree

                    EXIT

                ENDIF

            NEXT

            NewB := NewB + ProblemB[x]

        ENDIF

    NEXT

    REMARKS: Existing normalised lower end theory base

    REMARKS: Could be empty

    NewB := NewB + LowB

ELSE
```

REMARKS: IF Ndegree = Odegree AND $\alpha \notin exp(\mathbf{B}_{OldB})$

```
        IF NOT  Exist(OldB, α)
            NewB := NewB +  (α, Ndegree)
        ENDIF
    ENDIF
    RETURN NewB
END FUNCTION
```

The AGM belief contraction function is implemented by $Contraction()$ which lowers the existing degree $Odegree$ of a belief α to $Ndegree$. The $Contraction()$ function first checks if the new entrenchment degree $Ndegree$ equals the existing degree $Odegree$. If this is true, a trivial contraction is done by returning the existing theory base $OldB$. For a non-trivial contraction operation, the problematic segment of beliefs $ProblemB$ and the segments $HighB$ and $LowB$ which are supposed to be intact are identified based on the existing theory base $OldB$. A portion of the new theory base $NewB$ is constructed by copying the beliefs from segment $HighB$. According to the AGM principle, if a belief α is contracted from a belief set, the other beliefs which entail α should also be contracted. Therefore, all the sentences of $HighB$ are also added to the theorem prover to test if any beliefs ranked strictly higher than α logically entail α. The main FOR ...NEXT loop contracts sentences from the problematic theory base $ProblemB$ by lowering the entrenchment degrees of the affected beliefs to $Ndegree$. For each loop, the $ElaspsedTime()$ function returns the elapsed time since the contraction function is executed, and this elapsed time is compared with the maximum duration $TimeLimit$ allowed for a belief contraction operation. If the elapsed time exceeds the time limit, the main loop is terminated and the $Contraction()$ function will return the unaffected theory base segments plus any revised beliefs from the problematic theory base segment so far. Therefore, the returned new theory base $NewB$ still satisfies the postulates of finite partial entrenchment rankings. Essentially, for each sentence $\beta = ProblemB[x].belief$ of the problematic theory base segment $ProblemB$, it is necessary to check if any beliefs ranked strictly higher than β can classically entail ($\vdash$) β. If this is the case, the property (PER1) of finite partial entrenchment ranking (i.e., the dominance property (EE2)) is violated, and so the minimal change for restoring the property (PER1) is to lower the degree of β to $Ndegree$. If the belief α should still appear in the new theory base $NewB$, the affected belief β will be copied to the new theory base after its entrenchment degree is revised to $Ndegree$; otherwise the beliefs β as well as α are simply excluded from the copying operation. If a belief β from $ProblemB$ together with other sentences from $HighB$ entails α, the axiom β must be removed from the theorem prover before testing the remaining beliefs in $ProblemB$ otherwise every remaining belief can prove α. The $RemoveAxioms()$ function is used to remove axioms from the theorem prover. At the end of the $Contraction()$ function, the intact segment $LowB$ from the old theory base will also be copied to the new theory base $NewB$ if the new entrenchment degree $Ndegree$ is greater than the minimal degree.

```
FUNCTION    Contraction(OldB, α, Odegree, Ndegree, TimeLimit)

    REMARKS: MinDegree = 0 in our implementation

    IF Ndegree = Odegree

        RETURN OldB

    ENDIF

    HighB := Cut(OldB, Rank(MaxDegree), Rank(Odegree) - 1)

    ProblemB := Cut(OldB, Rank(Odegree), Rank(Ndegree) - 1)

    LowB := Cut(OldB, Rank(Ndegree), Rank(MinDegree))

    NewB := HighB

    prover := NEW TheoremProver()

    AddAxioms(prover, Beliefs(HighB))

    FOR x := 1 TO NoElements(ProblemB)

             IF ElapsedTime() > TimeLimit AND TimeLimit > 0

                EXIT

             ENDIF

             IF ProblemB[x].belief =  α

                SKIP NEXT
```

```
            ENDIF

            AddAxioms(prover, ProblemB[x].belief)

            IF Proved(prover, α)

               IF Ndegree > MinDegree

                  ProblemB[x].degree := Ndegree

                  NewB := NewB + ProblemB[x]

               ENDIF

               RemoveAxioms(prover, ProblemB[x].belief)

            ELSE

                REMARKS: {γ : B⁻(α, i)(γ) ⩾ B(β)} ∪ {β} ⊬ α

                NewB := NewB + ProblemB[x]

            ENDIF

      NEXT

      IF Ndegree > MinDegree

              NewB := NewB + (α, Ndegree)

              NewB := NewB + LowB

      ENDIF
```

```
    RETURN NewB

END FUNCTION
```

4.6 Mining Contextual Information

A retrieval context refers to a user's information needs as well as the background information about these needs. For example, the background knowledge such as "Oracle is a database product" can be used by an information seeker to retrieve information objects about database products. Formally, the "is-a" relationship is expressed by the association rule $oracle \rightarrow database$. Other semantic relationships can also be represented by logical implications. For instance, a synonym relationship can be represented by a biconditional $\leftrightarrow$. Indeed, information agents can make use of this kind of beliefs to enhance their learning and classification functions. This process is similar to query expansion by means of manually or automatically constructed thesauri [Gre98]. However, the background knowledge should be context sensitive because each information seeker may have different interpretations about term associations. For instance, $java \rightarrow indonesia$ is true for a tourist visiting Indonesia, but $java \rightarrow programming$ is true for a computer programmer specialising in Java programming. Therefore, association rule mining techniques have been explored to dynamically extract term associations pertaining to retrieval domains [FH96, LSC$^+$98, JBB00]. Based on the TREC-4 routing tasks and the AP-90 document set, it was found that context sensitive association rules were more effective than manually constructed static thesaurus such as WordNet [JBB00].

The association rule mining technique employed by the AIFS prototype system is based on the Apriori algorithm because it has been successfully applied to text mining applications [FH96, JBB00]. The Apriori algorithm was originally used to conduct data mining over transactional databases [AS94]. Formally, a database is conceptualised by a set of transactions $\mathcal{D}$, where each transaction $t \in \mathcal{D}$ consists of a set of items X also called *itemset* in data mining. A finite set of items $\mathcal{I} = \{i_1, i_2, \ldots, i_n\}$ is often used to represent the physical objects such as consumer products, Web pages, financial instruments, etc. present in data mining applications. Therefore, each transaction t can be seen as a subset of $\mathcal{I}$ (i.e., $t \subseteq \mathcal{I}$). In general, the number of items contained in an itemset is called the *size* of the itemset. For instance, if an itemset X consists of k items, it is called a k-itemset. It is assumed that items within an itemset are kept in lexicographic order. An association rule is an implication of the form $X \rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \emptyset$. In addition, two quantitative measures, *rule support* and *rule confidence*, are used to represent the significance of the association rules. The association rule $X \rightarrow Y$ holds in a transactional database $\mathcal{D}$ with support s if $s\%$ of transactions in $\mathcal{D}$ contain $X \cup Y$. In other words, the rule support s represents the joint probability $Pr(X \wedge Y)$ that a transaction $t \in \mathcal{D}$ contains the items from both X and Y. Moreover, the association rule $X \rightarrow Y$ has rule confidence c with respect to $\mathcal{D}$ if $c\%$ of transactions in $\mathcal{D}$ that contain X also contain Y. In other words, rule confidence c represents the conditional probability $Pr(Y|X)$ that a transaction $t \in \mathcal{D}$ will contain the set of items Y if the set of items X is present in t. This conditional probability can also be expressed in terms of: $c = \frac{support(X \wedge Y)}{support(X)}$ according to Bayes' theorem. The followings are the general meanings of rule confidence and support often referred to in the data mining literature:

$$support(X \to Y) = \frac{\text{Number of transactions containing } X \wedge Y}{\text{Total number of transactions}} \tag{4.10}$$

$$confidence(X \to Y) = \frac{\text{Number of transactions containing } X \wedge Y}{\text{Number of transactions containing } X} \tag{4.11}$$

Given a set of transactions $\mathcal{D}$, the data mining problem is to find all the association rules with support and confidence greater than the user specified minimum support $minsup$ and the minimum confidence $minconf$ respectively. The Apriori algorithm decomposes the association rule mining problem into two sub-problems. Firstly, the sets of items satisfying the minimum support are identified. Itemsets with minimum support are called large itemsets. For example, L_k represents the set of large itemsets with each itemset of size k. The second step is to use the large itemsets to construct the association rules. For every large itemset $l \in L$, find all the non empty subsets of l. Then, for every such subset x, generate a rule of the form $x \to (l - x)$ if the confidence of the rule is greater than $minconf$ (i.e., $\frac{support(l)}{support(x)} > minconf$). The following algorithm is used to find large itemsets:

$L_1 = \{\text{large itemsets with size } k = 1\}$

`FOR` $(k = 2, L_{k-1} \neq \emptyset, k++)$

$C_k = \text{Apriori-gen}(L_{k-1})$ // Generate candidate itemsets

FORALL transactions $t \in \mathcal{D}$ DO

$C_k^t = \text{Subset}(C_k, t)$ // subsets contained in t

`FORALL` $s \in C_k^t$ DO

$s.count++$ // increment the count of a candidate itemset

`END`

`END`

$L_k = \{s \in C_k | s.count \geq minsup\}$

`NEXT`

`Sets of large itemsets` $= \bigcup_k L_k$

With the Apriori algorithm, the first database scan is used to find large itemsets with size 1 (i.e.,L_1). For any subsequent pass k, the large itemsets L_{k-1} found in the $(k-1)$ pass are used to generate the candidate itemsets C_k using the Apriori-gen function. The merit of the Apriori algorithm is that a smaller candidate itemset C_k is generated for database scanning by first joining the L_{k-1} large itemsets confirmed in the previous pass and then deleting those candidate itemsets which contain subsets not in L_{k-1}. The basic intuition is that any subsets of a large itemset must also be large. Based on such an intuition, the Apriori algorithm aims at minimising the computational time wasted on generating and counting the hopeless itemsets. The join step of the Apriori-gen function can be characterised by a SQL statement as below:

INSERT INTO C_k

SELECT $p.item_1, p.item_2, \ldots, p.item_{k-1}, q.item_{k-1}$

FROM L_{k-1} p, L_{k-1} q

WHERE $p.item_1 = q.item_1, \ldots, p.item_{k-2} = q.item_{k-2}, p.item_{k-1} < q.item_{k-1}$;

Before a database scanning begins to count the large itemsets of size k, the following prune step in the Apriori-gen function is conducted to delete the hopeless candidate itemsets. The Subset(C_k, t) function can easily be implemented using an efficient *hash* tree data structure. The candidate itemsets of C_k are stored in a hash tree and the items from a transaction t are used to hash such a tree.

FORALL itemsets $c \in C_k$ DO

FORALL $k-1$ subset s of c DO

IF $(s \notin L_{k-1})$

delete c from C_k

BREAK

ENDIF

END

END

In the context of text mining for IR, a transaction is taken as a document, and a database is seen as the collection of documents. An item refers to a token or term present in a document. For the discussion in this thesis, a token is a single keyword contained in a document. Accordingly, an itemset is simply a set of terms. The Apriori algorithm is applied to discover the associations among terms in a document collection. Since the prototype agent system AIFS employs the Horn logic $\mathcal{L}_{Horn}$ as its representation language, each term association rule must satisfy the property of Horn clauses. For instance, the consequent (i.e., the right hand side) of an association rule contains a single item only. Therefore, the Apriori algorithm is applied as usual to find large itemsets. However, the procedure of rule generation is implemented in a slightly different way. For each large itemset $l \in L$, every non-empty item $x \in l$ is used to develop the consequent of a rule. If the rules generated by the template $Subsets(l-x) \rightarrow x$ satisfy $minconf$, they will be included in the rule set representing the background knowledge about a retrieval context. The $Subsets()$ function is used to generate all the non-empty subsets from $(l - x)$. In practice, a parameter k is used to constrain the subsets with sizes $\leq k$ so as to speed up the rule generation processes. To convert the term association rules to beliefs in an agent's knowledge base, the entrenchment degree $\mathbf{B}(\alpha)$ is derived by multiplying the rule support s and rule confidence c by an adjustment factor ξ. In the current prototype system, the adjustment factor ξ is tuned based on empirical evidence to optimise the retrieval effectiveness of the system.

Apart from term association rules, an information agent can also make use of other semantic relationships such as *information preclusion* [BH94, Bru96] to characterise a retrieval context so as to enhance retrieval effectiveness. An information preclusion relation such as $\alpha \perp \beta$ indicates that an information carrier α (represented by a sentence α of a $\mathcal{L}_{Horn}$) precludes another information carrier β. For example, $text \perp multimedia$ may hold if an information seeker only wants to retrieve documents about "text" but not about "multimedia". It should be noted that the information preclusion relations are driven by users' specific information needs, and so these relations are context sensitive. Accordingly, using automated methods to induce these relationships is desirable. Statistical information generated from AIFS's output database can be used to induce context sensitive information preclusion relations. For instance, the statistical data as depicted in Table 4.1 provides a valuable source for *mining* the information preclusion rules. Formally, an information preclusion relation between two terms $\alpha \perp \beta$ can be represented by a rule $\alpha \rightarrow \neg\beta$. In the current prototype system, only strict preclusion rules are induced. For any term t from the term table such as the one depicted in Table 4.1, if $df(t_{rel}) > 0$ and $df(t_{nrel}) = 0$, t is added to a set L. Similarly, for any term t satisfying $df(t_{nrel}) > 0$ and $df(t_{rel}) = 0$, it is added to a set R. Then, for each term $t_i \in L$, generate a rule $t_i \rightarrow \neg t_j$ for each $t_j \in R$. The entrenchment degree of such a rule is derived by: $Pr(t_i) \times Pr(t_j) \times \xi$, where a term probability $Pr(t) = \frac{df(t)}{N}$. $df(t)$ is the document frequency of a term t (i.e., the number of documents containing t) and N is the total number of documents reviewed by a user. These training documents are cached in AIFS's output database. The adjustment factor ξ is used to tune the entrenchment degrees of rules. For instance, based on the data presented in Table 4.1, the information preclusion rule $(business \rightarrow \neg science, 0.95)$ is induced if $\xi = 9.5$ is assumed. The following background knowledge is used for the learning and classification examples discussed in Section 4.8. In the context of IR, the first rule represents a synonym relationship. For instance, the term "business" is considered as equivalent to the term "commerce" from the perspective of a particular information seeker. The second and the third beliefs describe the classification knowledge perceived by the information seeker. For instance, "Insurance" is a kind of "Business", and "Computing" belongs to the "Sciences" discipline. The last association illustrate an *information preclusion* relation.

$$(business \leftrightarrow commerce,\ 0.95)$$

$$(insurance \rightarrow business,\ 0.95)$$

$$(computing \rightarrow science,\ 0.95)$$

$$(science \rightarrow \neg business,\ 0.95)$$

4.7 Predicting Document Relevance

The primary function of any IR system is to determine if a document Doc is relevant with respect to a given retrieval context Ctx. In fact, only the representation d of Doc and the representation K of Ctx are being evaluated in IR systems. Therefore, the *semantic correspondence* between Ctx and Doc can only be approximated by evaluating d with respect to K. Matching between retrieval contexts and documents is a binary classification problem (e.g., a document d is assigned the class label relevant or non-relevant). The advantage of the proposed belief-based adaptive information agent framework is that richer representations of the retrieval contexts Ctx can be developed, and the representations K (i.e., belief sets) can be refined by means of the AGM belief revision function and users' relevance feedback. In addition, both the learning and the classification behaviour of adaptive information agents can be predicted based on the axioms characterising the AGM logic. The belief-based agent framework also facilitates the development of the explanation functions of adaptive information agents since the agents' decisions can be justified based on formal deduction. Expectation inference provides a sound and powerful framework to reason about the relevance of documents with respect to a retrieval context. The notion of expectation inference $\alpha \mid\!\sim_K \beta$ states that a rational agent expects β to be true if it believes α is true and its existing beliefs (expectation) K together with α logically entail β. Moreover, expectation inference is closely related to belief revision in the sense that an agent believes β if β is in the agent's belief set K after the belief revision operation K^*_α. In other words, $\beta \in K^*_\alpha$ implies $\alpha \mathrel{|\!\!\sim_K} \beta$ with the set of beliefs K as background information.

In adaptive information retrieval, information agents revise their beliefs K about the retrieval contexts Ctx (e.g., users' current information needs, users' IR goals, users' background, semantic information of retrieval domains, etc.) according to users' relevance feedback. A given user's relevance feedback can be viewed as a refined query q. An information agent infers if an incoming document is relevant with respect to the refined query and other background information (expectation) K. As can be seen, the learning functions and the classification functions in adaptive information agents closely resemble the processes of belief revision and the processes of expectation reasoning. The learning functions and the classification functions of the agents are characterised by K^*_q and $q \mathrel{|\!\!\sim_K} d$ respectively. In particular, if $d \in K^*_q$ is true, the agents will conclude that $q \mathrel{|\!\!\sim_K} d$. In practice, what an agent would like to infer is whether an incoming document d is relevant with respect to K, the representation of a retrieval context which includes a set of long-term queries. Therefore, the document classification functions of adaptive information agents are underpinned by expectation inference of the form $K \mathrel{|\!\!\sim_K} d$ rather than $q \mathrel{|\!\!\sim_K} d$ which emphasises inference pertaining to each individual query. The idea of establishing document relevance by evaluating a document with respect to a retrieval context was also examined by Nie [NBL95]. However, the inference $K \mathrel{|\!\!\sim} d$ is characterised by *counterfactual* in their logic-based IR model. In addition, it has also been pointed out that estimating the probability associated with a *conditional* such as $q \rightarrow d$ within the logical imaging framework leads to a more sensible conclusion about the relevance of a document if ambiguous terms present in a retrieval context [Cre98]. The proposed classification framework $q \mid\!\!\sim_{K} d$ in information agents are akin to the aforementioned approaches. At the computational level, the *degree* function defined in Chapter 3 provides a gradated assessment of documents with respect to retrieval contexts. For instance, the notion $degree(\mathbf{B}, \alpha)$ is used to assess an element α of the logical representation d of a document Doc with respect to an agent's theory base $\mathbf{B}$ (i.e., a finite representation of a retrieval context). Given the fact that both d and K are only partial representations of the underlying Doc and Ctx, the classification processes are inherently uncertain. Document ranking is often used by IR models to deal with the uncertainties arising in matching Doc with Ctx [MBVL99, MM98, PB99, SM83]. In particular, several similarity measures have been used for document ranking [LB99, SM83, TKS00]. In general, these measures try to approximate the semantic correspondence between a document Doc and the pertaining retrieval context Ctx. To combine the advantage of quantitative ranking and symbolic reasoning, an entrenchment-based similarity measure Eq.(4.12) is used to evaluate documents in AIFS. Indeed in the quest of *common sense aboutness*, the authors have indicated that by employing certain weighting schemas in conjunction with the non-monotonic models, it may be able to simulate a form of *conservative monotonicity* which is believed to be the desirable behaviour for IR models [BSW00]. The entrenchment-based similarity measure represents an initial attempt along this direction.

$$Sim_{\leqslant}(Ctx, Doc) = \frac{\sum_{\alpha \in d}[degree(\mathbf{B}, \alpha) - degree(\mathbf{B}, \neg\alpha)]}{|S|} \tag{4.12}$$

The basic idea is that a document Doc is characterized by a set of positive literal $d = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$. If an agent's knowledge base K, which represents a retrieval context Ctx, nonmonotonically entails an atom $\alpha_i \in d$ via $\mid\!\!\sim_K$, a positive contribution to the overall similarity score is made because of the possible *semantic correspondence* between Ctx and Doc. The gradated assessment of the likelihood of $K \mid\!\!\sim_K \alpha_i$ is derived from $degree(\mathbf{B}, \alpha_i)$ where $K = content(\mathbf{B})$. On the other hand, if K entails the negation of an atom α_i in d, it demonstrates the possible *semantic distance* between Ctx and Doc. Therefore, the entrenchment degree of $K \mid\!\!\sim_K \neg\alpha_i$ is subtracted from the similarity score. This approach is similar to the paradigm of assumption-based reasoning where a plan or design is evaluated against a set of constraints stored in an agent's knowledge base [Kra97]. In the context of document classification, the negation of a document representation is the assumption which should be tested against the agent's knowledge base. The set $S = \{\alpha \in d : degree(\mathbf{B}, \alpha) > 0 \vee degree(\mathbf{B}, \neg\alpha) > 0\}$ contains the literals which are nonmonotonically entailed by the agent's knowledge base K.

In logic-based IR, it has been proposed that a document can be characterised by a conjunct of atoms [CC92]. However, it seems that an alternative is to represent a document by a disjunct of atoms if we accept the fact that document characterisation is imperfect and partial. For instance, given a document partially indexed by tokens

$\{html, web\}$, it is more appropriate to assume that the document is about *html* or *web* because there is uncertainty if the document is really about HTML scripting or Web surfing in general. Therefore, the proposed entrenchment-based similarity measure evaluates each atom $\alpha_i \in d$ individually. In fact, terms are also seen as disjoint possible worlds in logical imaging for IR [Cre98]. An intuitive example follows: assuming that an information seeker's preferences about "Web" and "Music" are partially represented by $K = \{(web \rightarrow internet,\ 0.6), (Web,\ 0.6), (music,\ 0.4)\}$ in an agent's knowledge base, and a document indexed by the tokens $d = \{mp3, internet\}$ is being evaluated by the agent. Should this document be recommended by the agent? The user may be interested in this document because "Web" is about the "Internet" and "MP3" is a popular form of archiving musical items on the Internet. If the document is represented by $d = mp3 \wedge internet$, $degree(\mathbf{B}, d) = 0$ is derived. In other words, the agent is totally uncertain about the relevance of the document with respect to the retrieval context K. On the other hand, based on the entrenchment-based similarity measure Eq.(4.12), a positive similarity score is derived because of $degree(\mathbf{B}, internet) > 0$. Therefore, the proposed logical characterisation of documents and the entrenchment-based similarity measure $Sim_{\leqslant}(Ctx, Doc)$ seem more effective in dealing with the issue of partiality in IR. A high positive score derived from Eq.(4.12) implies that an agent is certain that there is semantic correspondence between a document Doc and a retrieval context Ctx. If the similarity score is zero, the agent is totally uncertain about the relevance of Doc with respect to Ctx. However, a high negative similarity score implies that an agent is quite certain that the document Doc is irrelevant with respect to Ctx. Based on the similarity scores derived from Eq.(4.12), a ranking of documents (i.e. $(\preceq, Doc)$) can be formed to describe the relative relevance of the documents with respect to a retrieval context. If a document delivery threshold is employed, the agents can also make binary decisions about the relevance of incoming documents.

The entrenchment-based similarity measure has the advantage that the computed similarity scores (i.e., the conclusions) can be explained and justified based on a retrieval context (e.g., the relationships among terms). Four explanation templates are implemented in our adaptive information agent system AIFS to justify an agent's classification decisions. The notation [variable] means that the variable inside the square brackets will be instantiated during execution time.

1. **Item** $[\alpha]$ **is requested.**

   If $\alpha \in d$ is an explicit belief captured in $exp(\mathbf{B})$, the agent's theory base.

2. **Item** $[\beta]$ **is requested, and item** $[\alpha]$ **is associated with it.**

   For $\alpha \in d$ and $\beta \in exp(\mathbf{B})$, $degree(\mathbf{B}, \beta \rightarrow \alpha) > 0$ is deduced from $content(\mathbf{B})$.

3. **Item** $[\beta]$ **is requested, which precludes item** $[\alpha]$.

   For $\alpha \in d$ and $\beta \in exp(\mathbf{B})$, $degree(\mathbf{B}, \beta \rightarrow \neg\alpha) > 0$ is deduced from $content(\mathbf{B})$.

4. **Neither support nor rejection.**

   For $\alpha \in d$, $degree(\mathbf{B}, \alpha) = 0$ and $degree(\mathbf{B}, \neg\alpha) = 0$ are deduced from $content(\mathbf{B})$.

Example 1: Explaining an Agent's Decision

Assuming that a retrieval context Ctx is characterised by an agent's theory base as follows:

$$\begin{aligned} exp(\mathbf{B}) = \{ & (internet \rightarrow softbot,\ 0.850), \\ & (softbot \rightarrow spider,\ 0.850), \\ & (spider \rightarrow crawler,\ 0.850), \\ & (crawler \rightarrow \neg music,\ 0.023), \\ & (internet,\ 0.300)\} \end{aligned}$$

The first three beliefs represent term associations. The fourth belief is the *information preclusion* relation driven by a user's specific information needs. The last belief represents the user's current interest in "Internet". If a document $Doc = \{internet, spider, music, mp3\}$ is presented to the agent, the agent's prediction and explanation will look like the one depicted in Figure 4.4. In this example, a user who is interested in information about the "Internet" may also be interested in "MP3 music" which is among the *cool* items available on the Internet. Even though the preference of "Internet" may preclude the general interest of "Music" as described by the preclusion rule and the other term association rules in the agent's knowledge base, such a preclusion is not strong enough to totally rule out the user's possible interest in the document Doc which is about "Internet Spider for MP3 Music". As Doc is partially relevant to the retrieval context Ctx, the agent should recommend this document to its user.

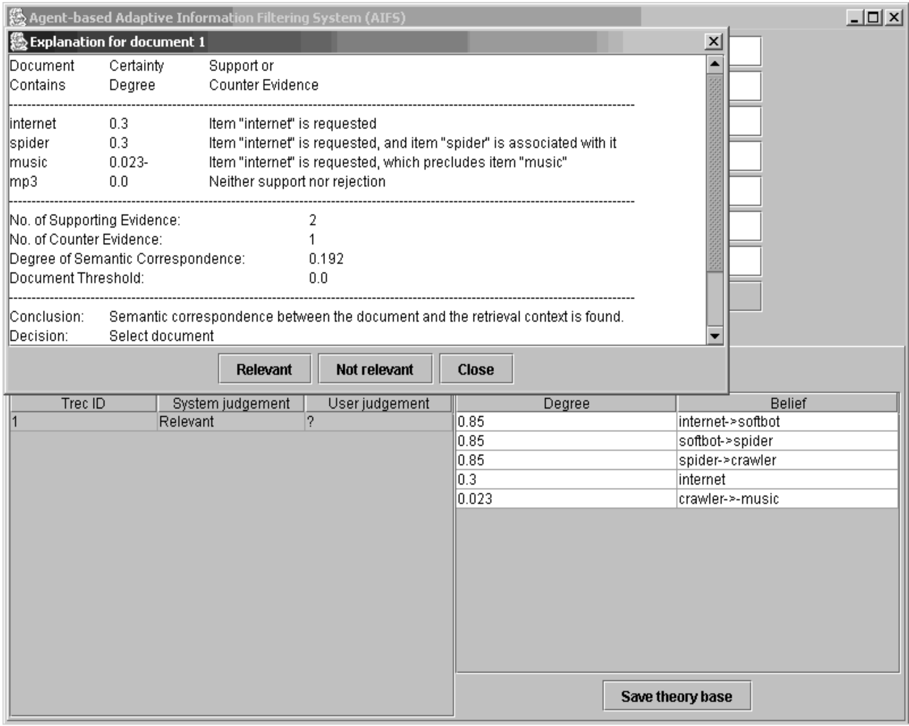

Figure 4.4: Justifying an agent's decision based on the explanation templates

4.8 Examples of Learning and Classification

The following examples illustrate how an adaptive information agent learns a retrieval context based on a user's relevance feedback and how the agent classifies documents with respect to the changing retrieval context. It is assumed that at time ($t0$), the agent does not know the user's preference except the background knowledge about the particular information retrieval domain. Therefore, the retrieval context at time ($t0$) is represented by the term association rules and the information preclusion relation only. At time ($t1$), the user is interested in documents about "Insurance". Through the user's relevance feedback, the agent learn the new belief *insurance* and other related beliefs pertaining to the retrieval context. The user's involvement is minimal and the agent can autonomously learn the user's possible interests related to "Insurance". At time ($t2$), the user's interest shifts from "Insurance" to "Computing". The information agent revises its beliefs about the current retrieval context by incorporating *computing* and other related beliefs into its knowledge base while contracting the contradictory information by executing the Rapid Anytime Maxi-adjustment algorithm. At each stage, only the implicit beliefs relevant for our discussion are shown.

Example 2: Learning at Time t0

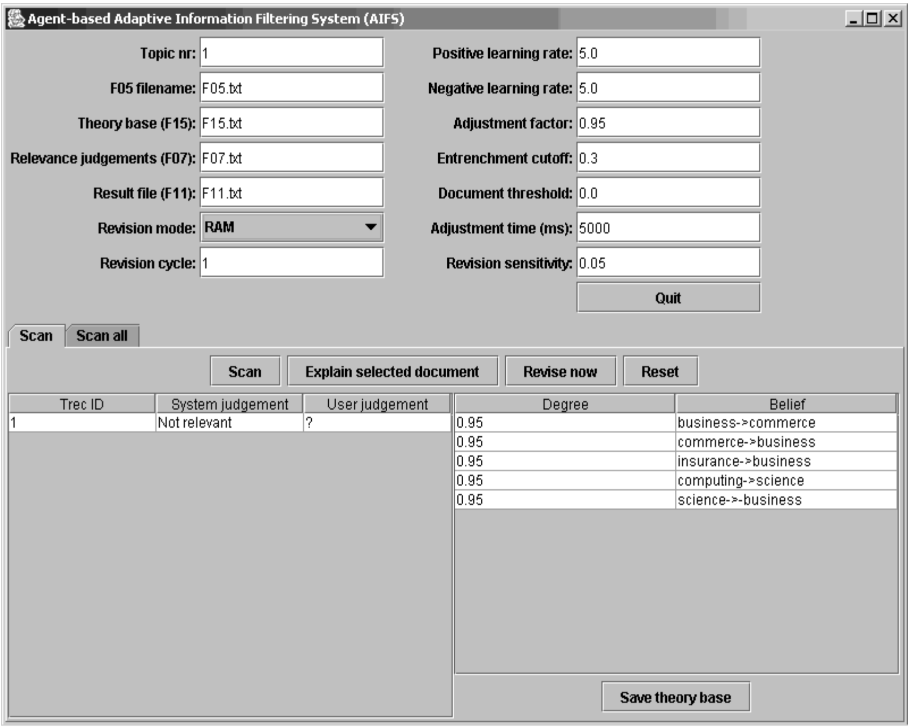

Figure 4.5: Learning at Time $t0$

It is assumed that only the following four rules induced based on the text mining methods described in Section 4.6 are captured in $exp(\mathbf{B}_{t0})$. The implicit beliefs which are derived from $exp(\mathbf{B}_{t0})$ are not listed along K_{t0} since they are not relevant for our discussion for the time being. It should be noted that $K_{t0} = content(\mathbf{B}_{t0})$. The theory base of the adaptive information agent is depicted in Figure 4.5. The lower right panel displays the agent's current theory base.

$$K_{t0} = \{(business \leftrightarrow commerce,\ 0.950),$$
$$(insurance \rightarrow business,\ 0.950),$$
$$(computing \rightarrow science,\ 0.950),$$
$$(science \rightarrow \neg business,\ 0.950), \ldots\}$$

Example 3: Learning at Time t1

The user informs the agent about their information needs by providing relevance feedback. For example, if two documents characterised by the term "Insurance" are judged as relevant by the user, the corresponding belief $(insurance, 0.087)$ is induced according to the procedure described in Section 4.4. A low entrenchment value (e.g., 0.087) is deliberately chosen in this example to indicate that an entrenchment value is not the same as a probability value. In particular, the belief $(insurance, 0.087)$ should not be interpreted that the chance of the item "Insurance" being requested by the user is low. For document ranking, the relative ranking rather than the absolute entrenchment degrees of beliefs is important. In fact, the entrenchment degrees of beliefs can be adjusted by means of the λ parameter of Eq.(4.9) discussed in Section 4.4. The next step in the learning process is to revise this belief into the agent's knowledge base via $\mathbf{B}^{\star}(insurance, 0.087)$. The RAM algorithm that implements $\mathbf{B}^{\star}(\alpha, i)$ is illustrated in Section 4.5. Since *insurance* is a new belief, its original degree of acceptance $j_m = degree(\mathbf{B}_{t0}, insurance)$ equals zero. As the new entrenchment degree $i = 0.087$ is greater than j_m, the revision function of the RAM algorithm is invoked:

$$\mathbf{B}^{\star}(insurance, 0.087) =$$

$$(\mathbf{B}^{-}(\neg insurance, 0))^{+}(insurance, 0.087)$$

Formula:α	$\mathbf{B}(\alpha)$ Before $(t1)$	$\mathbf{B}(\alpha)$ After $(t1)$
$business \leftrightarrow commerce$	0.950	0.950
$insurance \rightarrow business$	0.950	0.950
$computing \rightarrow science$	0.950	0.950
$science \rightarrow \neg business$	0.950	0.950
$insurance$	0.000	**0.087**
$business$	0.000	**0.087**
$commerce$	0.000	**0.087**
$\neg science$	0.000	**0.087**
$\neg computing$	0.000	**0.087**

Table 4.2: The retrieval context K_{t1} at time $(t1)$

By executing the RAM algorithm, the before and after images of the information agent's knowledge base K_{t1} (i.e. $content(\mathbf{B}_{t1})$) are tabulated in Table 4.2. The upper section of the table represents the agent's explicit beliefs (i.e. $exp(\mathbf{B}_{t1})$), and the lower section delimited by a horizontal line shows some of the agent's implicit beliefs. As demonstrated in this example, the user only needs to provide direct relevance feedback for the token "Insurance" and the agent can autonomously learn the user's other possible interests such as "Business" and "Commerce". The degree of acceptance $degree(\mathbf{B}_{t1}, business) = 0.087$ of the implicit belief *business* is computed according to Definition 4 defined in Section 3.1 of Chapter 3. Incorporating the belief *business* into the agent's knowledge base corresponds to our intuition of how the retrieval context should be revised at this point of time. Since "Insurance" is a kind of "Business", if the agent believes that the user may be interested in information objects about "Insurance", there is a good reason for the agent to believe that the user may also be interested in information objects about "Business". Similarly, the belief *commerce* is also automatically revised into the agent's knowledge base and the degree of acceptance of the belief *commerce* is 0.087.

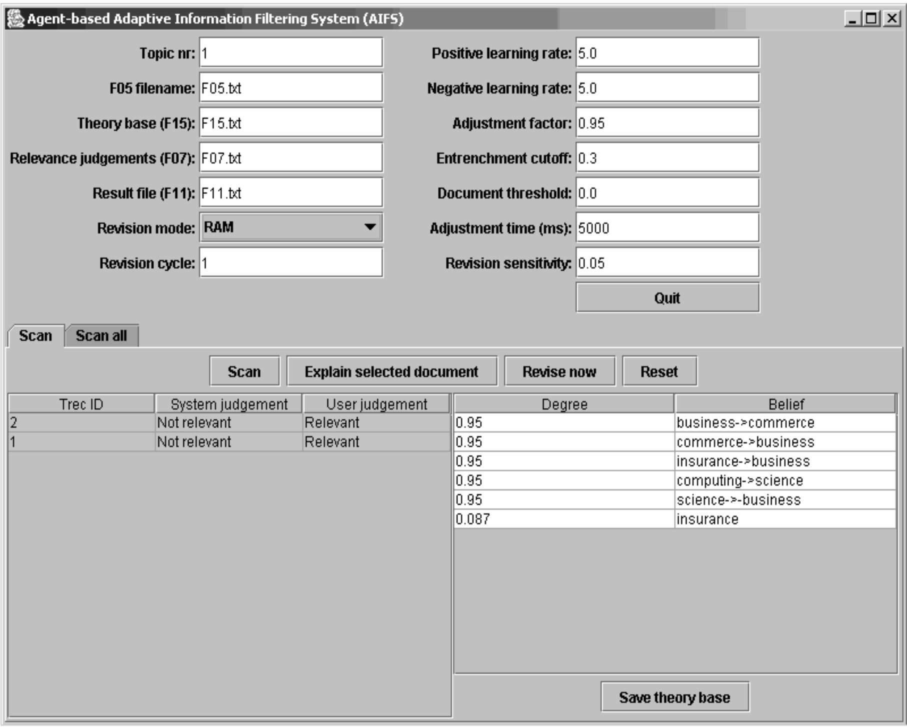

Figure 4.6: Learning at Time $t1$

The advantage of applying the AGM belief revision logic to construct the learning mechanisms of adaptive information agents is that the amount of direct relevance feedback required from human users can be minimised because the agents can infer the users' changing information needs based on formal deduction. In other words, the *learning autonomy* of the adaptive information agents is enhanced. Moreover, it is possible to explain such a learning behaviour by showing the relationships between terms. For example, the reason why the agent deduces that the user may be interested in documents about "Commerce" is that the token "Commerce" and the token "Business" are correlated in the current retrieval context. Therefore, documents about "Commerce" may contain information about "Business" as well. The agent's theory base at the end of time ($t1$) is depicted in Figure 4.6. The lower left panel shows that two documents are judged relevant by the user. The lower right panel displays the agent's theory base after learning the new belief based on the user's relevance feedback at time ($t1$). In summary, at the end of time ($t1$), the agent's knowledge base contains:

$$
\begin{aligned}
K_{t1} = \{&(business \leftrightarrow commerce, 0.950),\\
&(insurance \rightarrow business, 9.950),\\
&(computing \rightarrow science, 0.950),\\
&(science \rightarrow \neg business, 0.950),\\
&(insurance, 0.087),\\
&(business, 0.087),\\
&(commerce, 0.087),\\
&(\neg science, 0.087),\\
&(\neg computing, 0.087), \ldots\}
\end{aligned}
$$

Example 4: Learning at Time t2

If the user's interest shifts from "Insurance" to "Computing", the agent's perception K about the current information context Ctx can be refined based on the user's relevance feedback pertaining to the token "Computing". Other related changes can automatically be inferred by the agent. Assuming that four documents characterised by the token "Computing" are judged as relevant by the user, the belief $(computing, 0.473)$ is induced according to the entrenchment induction process described in Section 4.4. The belief revision operation $\mathbf{B}^{\star}(computing, 0.473)$ is then invoked to revise the agent's beliefs about the current retrieval context.

As the implicit belief $\neg computing$ exists in K_{t1}, the new belief *computing* cannot be revised into the agent's knowledge base unless its negation $\neg computing$ is contracted first. In addition, the implicit beliefs $(\neg business, 0.473)$ and $(\neg insurance, 0.473)$ are also deduced by the agent if the belief $(computing, 0.473)$ is accepted. These deduced implicit beliefs together with the agent's existing beliefs also lead to logical inconsistency ($\perp$) in the agent's knowledge base. Since it does not make sense if a user is interested in "Computing" and not interested in "Computing" at the same time, the existing belief $\neg computing$ that represents the user's previous interest should be retracted.

The computational advantage of the transmutation-based AGM belief revision is that a theory revision can be conducted based on a finite theory base $\mathbf{B}$. By executing the RAM algorithm to raise the entrenchment degree of *computing* from

0 to 0.473, the problematic theory base segment $ProblemB$ must first be identified. In this example, $ProblemB$ is the one bound by the $[higher, lower]$ entrenchment degrees such that $[higher < 0.473, lower > 0]$. It is easy to see that the explicit belief $insurance$ is the least entrenched belief causing inconsistency in the entire knowledge base K. Therefore, it should be contracted first. According to the RAM algorithm, the following procedure will be executed:

$$\mathbf{B}^{\star}(computing, 0.473) =$$

$$(\mathbf{B}^{-}(\neg computing, 0))^{+}(computing, 0.473)$$

The implicit belief ($\neg computing$, 0.087) was introduced to the agent's knowledge base at time ($t1$). The contraction part of the RAM algorithm $\mathbf{B}^{-}(\neg computing, 0)$ lowers the entrenchment degree of the belief $insurance \in ProblemB$ to zero because:

If $\forall_{\beta \in ProblemB}\{\gamma \in \mathbf{B} : \ \mathbf{B}^{-}(\neg computing, i)(\gamma) \geq \mathbf{B}(\beta)\} \ \cup \{\beta\} \ \vdash \ \neg computing$

Then $\mathbf{B}^{-}(\neg computing, i)(\beta) = i$

In this example, the result is obvious because the only explicit belief in $ProblemB$ is $(insurance, 0.087)$. As the belief $insurance$ together with other beliefs with higher entrenchment degrees (i.e., the set of explicit beliefs $(insurance, 0.087)$, ($insurance \rightarrow business$, 0.950), ($science \rightarrow \neg business$, 0.950), and ($computing \rightarrow science$, 0.950)) logically entail ($\vdash$) the belief $\neg computing$, it should be assigned the same entrenchment degree as $\neg computing$ according to the RAM algorithm. In this case, the new degree of $\neg computing$ equals zero. In other words, the belief $insurance$ is assigned zero entrenchment degree and is contracted from the theory base $\mathbf{B}_{t2}$. After contracting the belief *insurance* from the theory base, the agent's knowledge base K_{t2} is consistent and the belief *computing* can be added to K_{t2} to represent the new retrieval context at time ($t2$). As can be seen, the RAM algorithm adheres to the AGM principle of minimal and consistent belief revision. After revising the belief *computing* into the agent knowledge base, the causes of inconsistency in K_{t2} are $(insurance, 0.087)$, $(insurance \rightarrow business, 0.950)$, $(science \rightarrow \neg business, 0.950)$, and $(computing \rightarrow science, 0.950)$. The minimal change to the agent's knowledge base such that the knowledge base remains consistent is to give up the least significant belief $(insurance, 0.087)$ rather than one of the significant beliefs with entrenchment degree 0.950. The before and after images of the filtering agent's knowledge base K_{t2} are tabulated in Table 4.3.

Formula:α	$\mathbf{B}(\alpha)$ Before ($t2$)	$\mathbf{B}(\alpha)$ After ($t2$)
$business \leftrightarrow commerce$	0.950	0.950
$insurance \rightarrow business$	0.950	0.950
$computing \rightarrow science$	0.950	0.950
$science \rightarrow \neg business$	0.950	0.950
$computing$	0.000	**0.473**
$insurance$	0.087	**0.000**
$science$	0.000	**0.473**
$\neg business$	0.000	**0.473**
$\neg insurance$	0.000	**0.473**
$\neg commerce$	0.000	**0.473**
$business$	0.087	**0.000**
$commerce$	0.087	**0.000**

Table 4.3: The retrieval context K_{t2} at time ($t2$)

The degree of acceptance of the implicit beliefs such as *science*, $\neg business$, $\neg insurance$, and $\neg commerce$ are computed according to Definition 4. After con-

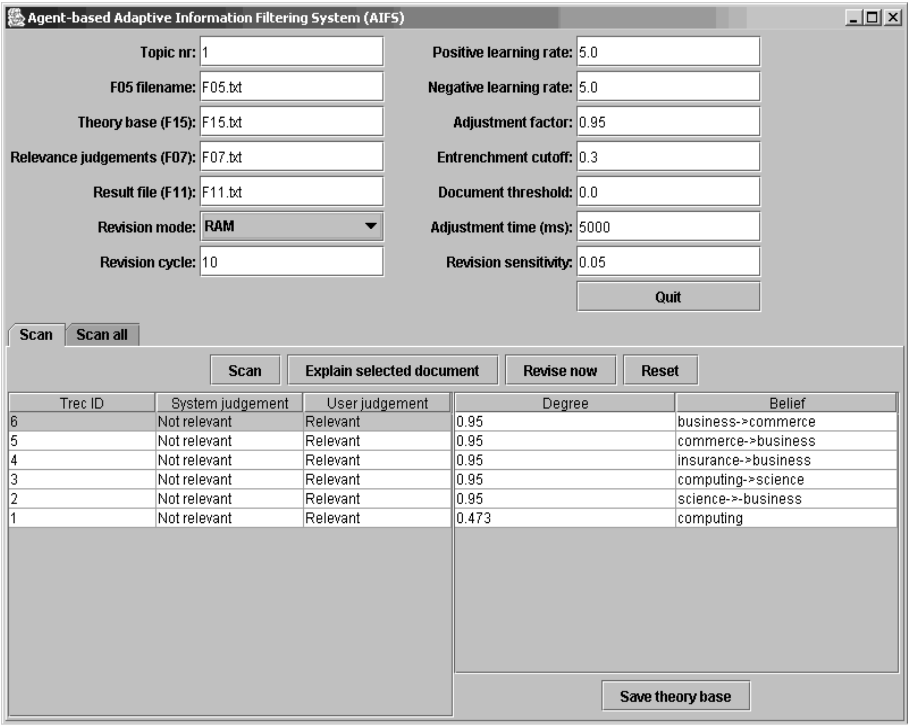

Figure 4.7: Learning at Time $t2$

tracting the belief *insurance* from $exp(\mathbf{B}_{t2})$, $business \notin content(\mathbf{B}_{t2})$ is established and so is $commerce \notin content(\mathbf{B}_{t2})$. Accordingly, the degrees of acceptance of these beliefs are zero. By taking the user's relevance feedback for a single item "Computing", the agent can automatically deduce that the user may no longer require information objects about "Insurance", "Business", and "Commerce". This illustrates how the AGM logic based learning mechanism can improve the adaptive information agents' learning autonomy. It should be noted that if the belief *insurance* is firmer than the belief $science \rightarrow \neg business$, the principle of minimal belief change makes the agent contract the information preclusion relationship $science \rightarrow \neg business$. Since "Business" is less likely to preclude "science", both the user's information needs of

"Insurance" and "Computing" can co-exist in the agent's knowledge base. On the other hand, if the entrenchment degree of *computing* is lower than that of *insurance* at time ($t2$), the agent should still revise the belief *computing* into its knowledge base and contract the belief *insurance* because the user's current interest is more likely about "Computing" rather than "Insurance". The agent's theory base at the end of time ($t2$) is depicted in Figure 4.7. The lower left panel shows that additional four documents are judged relevant by the user. The lower right panel displays the agent's theory base after learning the new belief based on the user's recent relevance feedback. In summary, at the end of time ($t2$), the agent's knowledge base K_{t2} contains:

$$\begin{aligned}
K_{t2} = \{&(business \leftrightarrow commerce, 0.950),\\
&(insurance \rightarrow business, 0.950),\\
&(computing \rightarrow science, 0.950),\\
&(science \rightarrow \neg business, 0.950),\\
&(computing, 0.473),\\
&(science, 0.473),\\
&(\neg business, 0.473),\\
&(\neg insurance, 0.473),\\
&(\neg commerce, 0.473), \ldots\}
\end{aligned}$$

Example 5: Matching at Time t1

The document classification mechanisms in adaptive information agents are based on the entrenchment-based similarity measure Eq.(4.12) which is underpinned by *expectation inference*. The similarity measure Eq.(4.12) can be used for document ranking which corresponds to multi-class classification with each document being assigned a rank (i.e., a class label), or for "Yes/No" recommendation which corresponds to binary classification with each document being assigned either the class label of *Relevant* or *Non-relevant*. With the binary classification mode, a document delivery threshold θ is used to divide the documents into two classes. Any documents with similarity scores greater than the delivery threshold are assigned to the relevant class and the documents will be dispatched to the users. With reference to the learning examples discussed before, if the following four documents are presented to the agent at time $(t1)$ and $(t2)$, the classification results will be:

$$d_1 = \{insurance, business, commerce\}$$

$$d_2 = \{insurance, business, computing\}$$

$$d_3 = \{computing, business, science\}$$

$$d_4 = \{computing, agent, science\}$$

$$Sim_{\leqslant}(Ctx_{t1}, Doc_1) = \frac{(0.087+0.087+0.087)-0.000}{3} = 0.087$$

$$Sim_{\leqslant}(Ctx_{t1}, Doc_2) = \frac{(0.087+0.087)-0.087}{3} = 0.029$$

$$Sim_{\leqslant}(Ctx_{t1}, Doc_3) = \frac{0.087-(0.087+0.087)}{3} = -0.029$$

$$Sim_{\leqslant}(Ctx_{t1}, Doc_4) = \frac{0-(0.087+0.087)}{2} = -0.087$$

$$\therefore Doc_4 \preceq Doc_3 \preceq Doc_2 \preceq Doc_1$$

The notation $\preceq$ represents the preferential ordering (i.e., relevance) of documents with respect to a retrieval context Ctx. For example, $Doc_x \preceq Doc_y$ means that Doc_y is at least as preferred or relevant as Doc_x with respect to a retrieval context. More precisely, the semantic correspondence between a document Doc_i and the Ctx_{t1} is approximated by the net entrenchment degree of the logical representation d_i of the document Doc_i. The above ranking corresponds to our intuition of document relevance. At time $(t1)$, the user is interested in documents about "Insurance", "Business", and "Commerce". However, the user is not interested in documents about "Computing" nor "Science" in general. Therefore, the retrieval context Ctx_{t1} is about the "Business" world but not about "Science". There may be semantic correspondence between Doc_1 and Ctx_{t1}, and the document Doc_1 should be ranked the highest in the list. Doc_2 is partially corresponding to Ctx_{t1}, and so it should be ranked higher than Doc_3 and Doc_4. Doc_3 seems about "Business Computing" rather than "Business", and so Doc_3 is ranked lower than Doc_2. It is obvious that Doc_4 is not about the "Business" world at all, and so it is ranked the lowest with respect to Ctx_{t1}. For binary classification, a document delivery threshold $\theta = 0$ is assumed. Accordingly, only the documents Doc_1 and Doc_2 which are really about "Insurance" and "Business" will be recommended by the agent. Justification of such a document ranking is based on the underlying entrenchment-based entailment (i.e., expectation inference). For instance, $insurance$ is an explicit belief captured in the agent's knowledge base, and so a document characterised by the token "Insurance" contributes a positive value to the overall similarity score. Furthermore, since "Insurance" is a kind of "Business" defined by the classification rule *insurance* $\rightarrow$ *business*, documents characterised by a token "Business" may also be relevant with respect to the user's information needs. As can be seen, expectation inference opens the door to a more explanatory information retrieval process. Figure 4.8 shows an example of how the prototype agent system AIFS computes and explains the entrenchment-based similarity score. In this example, a binary classification decision for document Doc_2 is made based on the document delivery threshold θ.

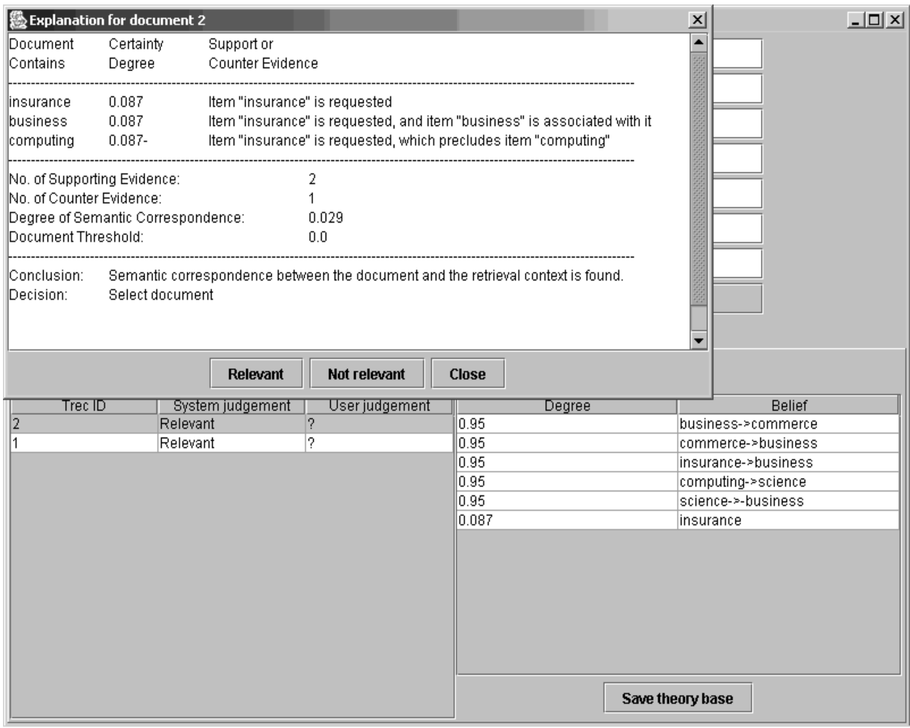

Figure 4.8: Classification and Explanation at Time $t1$

Example 6: Matching at Time t2

$$Sim_{\leqslant}(Ctx_{t2}, Doc_1) = \frac{0.000-(0.473+0.473+0.473)}{3} = -0.473$$

$$Sim_{\leqslant}(Ctx_{t2}, Doc_2) = \frac{0.473-(0.473+0.473)}{3} = -0.158$$

$Sim_{\leqslant}(Ctx_{t2}, Doc_3) = \frac{(0.473+0.473)-0.473}{3} = 0.158$

$Sim_{\leqslant}(Ctx_{t2}, Doc_4) = \frac{(0.473+0.473)-0.000}{2} = 0.473$

$$\therefore D_1 \preceq D_2 \preceq D_3 \preceq D_4$$

The user's information need has shifted from "Insurance" to "Computing" at time ($t2$), and so the retrieval context Ctx_{t2} at time ($t2$) is about "Computing" and "Science" in general. The four documents can be ranked again based on the agent's classification mechanism. The ranking corresponds to our intuition about document relevance with respect to Ctx_{t2}. The document Doc_4 is more likely to be semantically corresponding to Ctx_{t2} because the document is about "Computing Science". The measure $Sim_{\leqslant}(Ctx_{t2}, Doc_4)$ is able to capture this reality by returning the highest positive similarity score. Therefore, the document Doc_4 is ranked the highest in the list. On the other hand, Doc_1 is totally incompatible with the retrieval context Ctx_{t2}. The semantic distance between Doc_1 and Ctx_{t2} is approximated by the sum of the entrenchment degrees $degree(\mathbf{B}_{t2}, \neg insurance)$, $degree(\mathbf{B}_{t2}, \neg business)$ and $degree(\mathbf{B}_{t2}, \neg commerce)$. As $Sim_{\leqslant}(Ctx_{t2}, Doc_1)$ returns the smallest similarity score, the document Doc_1 is ranked the lowest in the list. Documents Doc_3 is partially relevant with respect to the retrieval context Ctx_{t2}, and so it is ranked higher than Doc_2. Figure 4.9 shows another example of how the prototype agent system AIFS computes and explains the entrenchment-based similarity score at time ($t2$). In this example, a binary classification decision and the corresponding explanation for document Doc_2 is made.

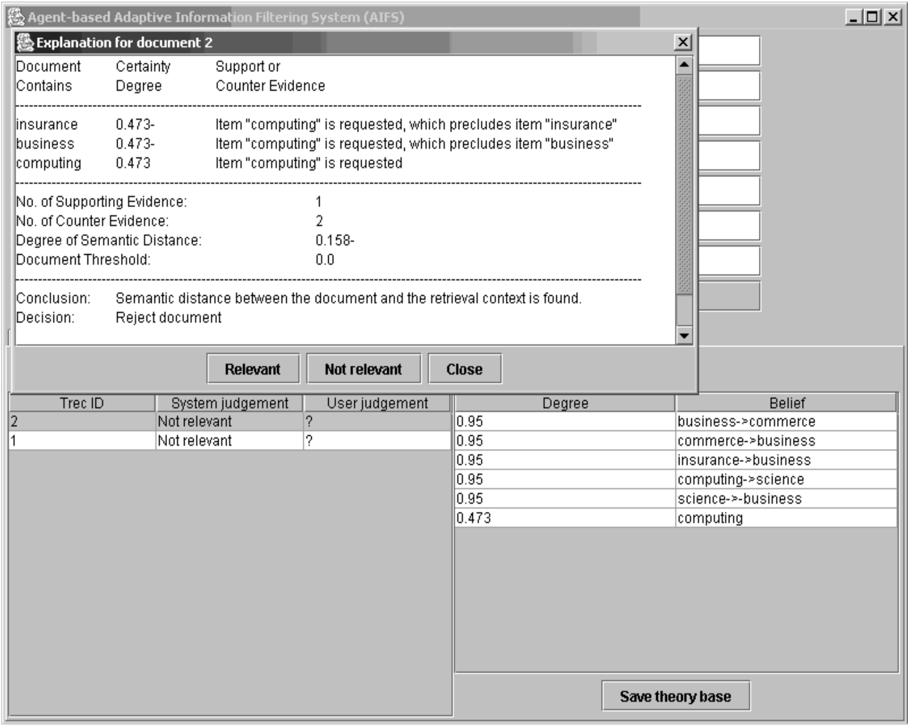

Figure 4.9: Classification and Explanation at Time $t2$

Chapter 3 discussed the theory of belief revision in a broad sense via the AGM belief revision framework. In addition, how the AGM belief revision functions and the corresponding expectation inference relations is applied to adaptive information retrieval is illustrated at the conceptual level. Chapter 4 further describes the computational details of the belief revision based adaptive information model and illustrates how the computational algorithms can be applied to develop an agent-based adaptive filtering system AIFS. In particular, the epistemic entrenchment induction method and the RAM belief revision algorithm are examined. An entrenchment-based similarity measure which combines the power of expectation inference and quantitative ranking is applied to build the agent system's classification mechanism. An example is used to explain how AIFS's learning and classification mechanisms work. The task now is to evaluate AIFS in a much larger practical setting.

Chapter 5

Experiments and Results

To evaluate IR models, one needs to consider at least three aspects: performance measures, document collections, and evaluation procedures. This chapter describes how the proposed belief-based adaptive information agent model is evaluated, and reports the results of our initial experiments. Basically, the kernel module of the agent-based adaptive information filtering system (AIFS) was tested against two benchmark collections: the TREC-AP collection and the Reuters-21578 collection which are widely used in IR and machine learning research.

5.1 The Performance Measures for IR

Previous studies in information retrieval have used a variety of measures to evaluate the effectiveness of IR systems. However, each measure has its merits and limitations. One well-known measure often employed in machine learning research is referred to as *classification accuracy*. For binary classification problem such as text filtering, only two class values (e.g., relevant vs. non-relevant) are considered. Classification accuracy for binary classification problems can then be defined with respect to a contingency table as depicted in Table 5.1. In the contingency table, the letters a, b, c, d represent the number of documents classified to the respective categories. For example, a represents the number of documents classified as relevant by the agent and these documents are really relevant with respect to a user's specific information needs. With respect to the contingency table, classification accuracy can be formally defined by:

$$Accuracy = \frac{a+d}{a+b+c+d} \tag{5.1}$$

	Relevant document	Non-relevant document
Agent predicted relevant document	a (true positive)	b (false positive)
Agent predicted non-relevant document	c (false negative)	d (true negative)

Table 5.1: Contingency Table for Binary Classification Problem

Generalisation of classification accuracy for any number of class values (e.g., the categorisation problem in IR) is done by dividing the sum of diagonal elements (i.e., the number of correctly classified objects) by the sum of all table elements: $Accuracy = \frac{\sum_i x_{ii}}{\sum_i \sum_j x_{ij}}$, where x_{ij} is the element in the i-th row and the j-th column of the contingency table. For instance, the category b in Table 5.1 is referred to as

x_{12} in the generalised accuracy measure. For text filtering applications, class value distribution is often asymmetric (e.g., many documents falling into the non-relevant class). Under such circumstance, high classification accuracy can be trivially achieved by a classifier by simply classifying all the objects into the majority class (i.e., the non-relevant class).

Precision and *Recall* are two widely used measures in IR research [SM83]. One of the class values corresponds to the target class (e.g., relevant) for which precision and recall should be maximised. An ideal IR system would have both the precision and the recall values equal 1. Nevertheless, perfect precision can be trivially obtained by an IR system that does not classify documents into the target class, while perfect recall can be trivially achieved by a system that classifies all documents into the target class. Therefore, using just one of these measures alone is not sufficient to validate the effectiveness of the underlying IR models. *Precision* is defined as the proportion of retrieved relevant documents in the set of all retrieved documents. For a binary classification problem such as text filtering, precision can be estimated with reference to the contingency table:

$$Precision = \frac{a}{a+b} \tag{5.2}$$

In the exceptional case that there is no document retrieved by an IR system (i.e., no object falling into the target class), there will be no classification error in the result set. Accordingly, the maximum precision of 1 is achieved trivially. On the other hand, *recall* is defined as the proportion of retrieved relevant documents out of the set of all relevant documents. Accordingly, recall can be estimated by the following formula:

$$Recall = \frac{a}{a+c} \tag{5.3}$$

In case that there is no relevant document in a collection, an IR system can trivially achieve the maximum recall because the number of retrieved relevant documents always equals the number of relevant documents. A better approach than employing the precision measure or the recall measure alone to evaluate the effectiveness of IR systems is to use the F-measure. The F-measure is a weighted combination of precision and recall values [vR79]. The relative significance of precision and recall is expressed by the β parameter. The F-measure is formally defined by:

$$F_\beta = \frac{(1+\beta^2) Precision \times Recall}{\beta^2 Precision + Recall} \tag{5.4}$$

If the parameter β equals 0, the particular F_0 measure is equivalent to the Precision measure. On the other hand, if the parameter β takes the value of ∞, the corresponding F_∞ measure is the same as Recall. As can be seen, the value of β can be chosen between 0 and ∞. To emphasise the importance of precision, the value of β should be less than 1. On the other hand, to emphasise the importance of recall, the value of β should be greater than 1. As stated in Chapter 1, one of the requirements of effective IR systems is their capabilities of maximising both precision and recall in

IR processes. Accordingly, the parameter $\beta = 1$ is chosen to evaluate the effectiveness of our adaptive information agents. With reference to the contingency table 5.1, the F-measure can also be expressed as: $F_\beta = \frac{(1+\beta^2)a}{(1+\beta^2)a+b+\beta^2 c}$. In the case of singularity such as $a + b + c = 0$, it can be interpreted that an IR system does not retrieve any document given that there is no relevant document in the collection. Under such a circumstance, the value of F_β equals 1. Since the value of the β parameter is set to 1 in our experiments, the following equivalent measure is used to assess the performance of the adaptive information agents:

$$F_1 = \frac{2a}{2a + b + c} \tag{5.5}$$

However, one drawback of the F-measure is that its value is not easily comprehended by ordinary information seekers [SSS98]. In practice, it may not be easy for a user to judge the relative importance of recall and precision. For instance, it may be difficult for an ordinary information seeker to assert that recall is twice as important as precision in their particular IR context. Nevertheless, the F-measure is considered the most effective measure for evaluating the performance of IR systems among the measures of classification accuracy, raw precision, and raw recall. Recently, the TREC text filtering evaluations have been using the utility measure to assess the performance of IR systems [Hul98]. A utility-based measure assigns a value or cost to each document based on whether it is retrieved or not, and whether it is relevant or not. With reference to Table 5.1, one can imagine that each category is associated with a utility parameter. For instance, the category a is associated with a reward A, the category b is associated with a cost B and so on. For a binary classification problem, a general utility function is defined by:

$$Utility = A \times a + B \times b + C \times c + D \times d \tag{5.6}$$

The utility parameters A, B, C, D determine the relative value of the corresponding categories a, b, c, d. The parameters A and D are generally considered as rewards, and the parameters B and C are considered as penalties. The larger the utility score, the better an IR system is performing. However, the utility measure is not perfect either. Utility scores vary widely from topic to topic depending on the actual number of relevant documents in the respective topics. In addition, converting the utility scores to a standard measure for comparison across topics requires complex scaling and normalisation procedures. Finally, the linear utility function treats all relevant documents with the same preference value even though a user may find that a particular document is more important than the others in a collection.

Nevertheless, the utility measure is effective for evaluating IR models which deal with problems characterised by asymmetric class values distribution (e.g., many non-relevant documents vs. a few relevant documents). For instance, the precision of an IR system retrieving one non-relevant document is the same as another IR system retrieving ten thousand non-relevant documents if there is no relevant document in the chosen topic. An evaluation metric based on utility functions can alleviate the above problem by penalising the latter case more than the former. Therefore, a utility function based measure is a good choice for evaluating IR systems that deal with text filtering tasks. To facilitate the comparison between our experimental results with that of the other IR or machine learning experiments, all the aforementioned measures will be used to evaluate the performance of our adaptive information agents.

5.2 The Collections

There are three main components in every document collection namely, documents, topics, and relevance judgements. The experiments presented in this thesis are based on both the TREC-AP collection and the Reuters-21578 collection.

5.2.1 The TREC-AP Collection

The TREC-AP collection consists of 3 years of Associated Press newswire covering the period from 1988 to 1990. This collection is distributed in the TREC TIPSTER disks 1-3. The annual TREC conference is co-sponsored by NIST and the Information Technology Office of the Defence Advanced Research Project Agency (DARPA) as part of the TIPSTER text retrieval research program (`http://trec.nist.gov/`). There are $84,678$ documents (254 mega bytes) for the year 1989 in the TREC-AP collection, $79,919$ documents (237 mega bytes) for the year 1988, and $78,321$ documents (237 mega bytes) for the year 1990. Our experiments only utilise the AP-89 (disk 1) documents because it is the one with the most comprehensive relevance judgements. The average document length of the AP-89 subset is 137 words. The TREC-AP documents are ordered roughly by date. The AP newswire covers a broad variety of domains such as economics, trade, technology, etc. These documents are tagged using SGML to allow easy parsing. Each document in this collection has a distinct title field marked by the SGML tag `<HEAD>`, and a distinct body field marked by the SGML tag `<TEXT>`. Both the title field and the main body text were used in our experiments. The following is an example of an AP news story formatted by the SGML tags:

```
<DOC>
<DOCNO> AP890101-0001 </DOCNO>
<FILEID>AP-NR-01-01-89 2358EST</FILEID>
<FIRST>r a PM-APArts:60sMovies     01-01 1073</FIRST>
<SECOND>PM-AP Arts: 60s Movies,1100</SECOND>
<HEAD>You Don't Need a Weatherman To Know '60s Films Are Here</HEAD>
<HEAD>Eds: Also in Monday AMs report.</HEAD>
<BYLINE>By HILLEL ITALIE</BYLINE>
<BYLINE>Associated Press Writer</BYLINE>
<DATELINE>NEW YORK (AP) </DATELINE>
<TEXT>
   ``Film is a very powerful art medium,'' he said. ``I believe it
very accurately reflects not only the prevailing but the coming
trends. It's because film writers, like other writers, are
perceptive people. They get the message of what's going on.''
</TEXT>
</DOC>
```

An AP document does not contain any fields to indicate if the corresponding story is about a particular topic or category. So, the topics related to the AP newswire actually refer to the topic descriptions created as part of the TREC evaluation procedure. A topic can be seen as a representation of a user's specific information need (i.e., a query). The format of the TREC topics has evolved over time. Generally, the topic descriptions become shorter in recent TREC experiments. The original ad hoc topics 1-50 used in TREC-1 contain the longest descriptions. Each topic description contains multiple fields and lists of concepts related to the topic. The experiments reported in this thesis used the TREC topics 1-50 to represent a wide variety of initial information needs of a hypothetical user. An example of a TREC topic can be found in Appendix B. In our experiments, each TREC topic description was treated as a document, and they were parsed along with the AP-89 document collection based on the standard text pre-processing and TFIDF weighting procedure similar to the one employed in the SMART system [Sal90]. For example, the stop word list as defined in SMART was used to remove insignificant common words from the collection and then Porter's stemming algorithm [Por80] was applied to compute the root form of each word. Non-alphabetic characters are removed from a word because our theorem prover cannot deal with special characters. Finally, the TFIDF weighting scheme (also called the "atc" weight in SMART) was applied to compute the TFIDF weight of each term [SB88]. After text pre-processing, there are $131,906$ unique terms found in the AP-89 collection. For each indexed TREC topic description, the top 10 terms with the highest TFIDF weights were then selected to represent an initial query (i.e., a user's initial information need). The top n tokens ranked by the TFIDF weights were used to represent an AP newswire story. For all the results presented in this thesis, $n = 50$ was employed. Previous studies revealed that using a small number of terms to represent a document produced better retrieval results [BS95, Bal97, PB97]. A pilot run of our experiments showed that the setting of $n = 50$ produced better result in terms of F_1 scores when compared with that of other parameter settings such as $n = 30$, $n = 100$, etc. The actual size of the AP-89-50 subset with each document represented by no more than 50 terms is $122,298$KB.

Relevance judgements are among the most important elements of any corpora. To assess the effectiveness of IR systems, a list of relevant documents pertaining to each topic is compiled in advance. This list of documents is called the relevance judgement which defines the possibly changing information needs pertaining to a hypothetical user. By comparing an IR system's predictions with the user's relevance judgements, the IR system's effectiveness can then be estimated. For the TREC-AP corpus, not all the documents were assessed manually. A *pooling* method was used to construct a list of documents which were predicted as relevant by the majority of the participating IR systems in TREC [VH99]. The particular sampling method used in TREC was to take the top 100 documents retrieved in each submitted run for a given topic and then these documents were merged into a pool for assessment by human experts. It is assumed that a document not in the list of relevance judgement is non-relevant. Our experiments related to the AP-89 collection employed the relevance judgement file provided by TREC [Con02]. A sample format of the relevance judgement file is as follows:

```
1 0 AP880212-0161 0
1 0 AP880216-0139 1
```

```
1 0 AP880216-0169 0
1 0 AP880217-0026 0
1 0 AP880217-0030 0
```

The first field (column) represents the TREC topic number; the second field is not used in our experiments. The remaining fields (columns) contain the TREC document ID and the relevance judgement respectively. If the relevance field contains "1", it means that the associated document is relevant with respect to the given topic; otherwise it is non-relevant. In addition, if a particular topic and document ID combination is not found in the relevance judgement file, the document is assumed non-relevant for that particular topic.

5.2.2 The Reuters-21578 Collection

The Reuters corpus of newswire stories is widely used for IR and machine learning research. The documents of the Reuters-21578 collection appeared on the Reuters newswire in 1987. These documents were first assembled by Reuters Ltd. and the Carnegie Group Inc. and then refined by David Lewis. Since then, the Reuters-21578 test collection has been made publicly available from the following Web site:

`http://www.research.att.com/~lewis`

Similar to the TREC-AP collection, the documents in the Reuters-21578 collection were marked up with the SGML tags and each document was assigned a document ID corresponding to the chronological order of appearance of the newswire stories.

A sample document of the Reuters-21578 corpus is depicted in Appendix C. Among the five categories “Exchange”, “Orgs”, “People”, “Places”, and “Topics”, a human assessor decided which categories a document belonged to. Our experiments only dealt with the “Topics” category. The Reuters-21578 topics are about economic subjects such as “coconut”, “gold”, “inventories”, “money-supply”, etc.. If a document has been assigned to one or more topics, there will be corresponding topic names delimited by the tags `<D>` and `</D>` inserted in the `<TOPICS>` field of the document. For example, a document belonging to the topic “cocoa” will have the entry `<TOPICS><D>cocoa</D></TOPICS>` inserted in the topic field. As can be seen, the representation of relevance judgement information in the Reuters-21578 collection is different from that employed in the TREC-AP collection. In our experiments, a computer program was developed to parse every document and create a relevance judgement file with the same format as the one employed in the TREC-AP collection.

There are 135 topics in the Reuters-21578 collection and each one of them is sequentially assigned a topic number in our experiments. There could be multiple relevance judgement records generated for a single document if there are more than one topic names in the `<TOPICS>` field. As in the TREC-AP collection, if a relevance judgement record is not found given a topic number and a document ID, the corresponding document is assumed non-relevant. Only the first 20 topics were used in our experiments. Among these topics, there are topics with many relevant documents (e.g., topic 1 - acq) and also topics with no relevant document at all (e.g., topic 4 -

austral). Therefore, this subset of topics seems sufficient to represent different kinds of class value distributions corresponding to various retrieval situations. Text appearing in the `<TITLE>` field or the `<BODY>` field was used to represent a document in our experiments. The same text pre-processing procedure as applied to the TREC-AP collection was used to parse the Reuters-21578 collection. Table 5.2 lists the first 20 topics of the Reuters-21578 collection and the corresponding number of relevant documents pertaining to each topic:

Topic No.	Description	No. of Relevant Documents
1	acq	2366
2	alum	57
3	austdlr	4
4	austral	0
5	barley	51
6	bfr	0
7	bop	105
8	can	3
9	carcass	68
10	castor-meal	0
11	castor-oil	2
12	castorseed	1
13	citruspulp	1
14	cocoa	73
15	coconut	6
16	coconut-oil	7
17	coffee	139
18	copper	65
19	copra-cake	3
20	corn	237

Table 5.2: The Reuters-21578 Topics & Relevant Documents

There are two standard subsets of the Reuters-21578 collection for batch learning tasks. One of them is called "Modified Lewis Split" and the another is the "Modified Apte Split". The main difference between these two subsets is that the latter contains documents belonging to at least one topic. With batch mode learning tasks, a document collection is often divided into a training set and a test set. For our adaptive learning and filtering task, such a split is not required because an information agent learns as soon as a document is presented. In other words, there is no training period to develop an initial user model. Our experiments used the "Modified Lewis Split" but without the actual splitting. The original "Modified Lewis Split" contains 19,813 documents. Nevertheless, there were only 19,702 documents used in our experiments because there were some documents with empty text body after our stop word removal procedure. There are 31,568 unique terms found in the "Modified Lewis Split" subset of the Reuters-21578 collection. The average number of words per document is 45. The reason why the "Modified Lewis Split" instead of the "Modified Apte Split" was used is that more realistic IR scenarios are preferred. For instance, there could be no document satisfying a user's information needs in real IR situations, and these cases are represented in the "Modified Lewis Split" document subset.

5.3 Evaluation Procedures

The evaluation procedure for our adaptive information agents is based on the adaptive information filtering task of the seventh Text REtrieval Conference (TREC-7) [Hul98]. The primary objective of the TREC forum is to encourage research in text retrieval based on large text collections, and to facilitate the exchange of ideas among industry, academia, and the government. The emphasis on individual experiments evaluated within a common setting has proven to be a major strength of TREC. The main reason for employing the adaptive filtering bench-marking procedure of TREC is that their method provides a realistic assessment of adaptive IR systems. By using the TREC-7 bench-marking procedure to evaluate our agent system, it becomes possible to compare the performance of the proposed belief-based information agent model with that of other well-known IR models. The TREC-7 adaptive filtering task assumes that an IR system will make a binary decision of document relevance as soon as a document arrives. Such an assumption is more akin to the scenarios of on-line interactive IR processes.

For the TREC-7 adaptive information filtering task, each IR system is only provided with a set of topic descriptions (i.e., a user's initial interests) based on the TREC topics 1-50 originally used in the ad hoc retrieval task of TREC-1. Training documents are **not** available to develop an initial user profile. Documents arrive sequentially and an IR system needs to make an immediate decision if the current document is relevant or not (i.e., binary classification) with respect to the current user profile. Therefore, the adaptive filtering task is considered much more difficult than the batch filtering or routing tasks where an IR system is supplied with a set of training examples to learn the information needs of a hypothetical user before the system starts to make recommendations. The terms "routing", "filtering", and "ad hoc retrieval" are somewhat confusing in TREC. In all the TREC experiments, *ranked* text retrieval with respect to a query is called “ad hoc retrieval”. Ranked text filtering is referred to as “routing”, whereas binary text filtering in which a “yes” or “no” decision must be made as each document arrives is referred to as “text filtering”. For the TREC-7 adaptive filtering task, each participating system starts with a query derived from a topic description. An IR system processes documents one at a time according to their chronological order. If the system decides to retrieve a document, it obtains the relevance judgement associated with the document. Then, the IR system uses the relevance judgement to refine its user profile. So, only retrieved documents are used as learning examples. To simulate the interactive relevance feedback environment, the relevance judgement information associated with each document should not be read by the system before a prediction about the current document is produced. In TREC-7, there was no mechanism to enforce this policy and it was entirely up to the participating systems to follow this procedure based on self discipline. Apart from the relevance judgement information, participating systems were allowed to use the TREC document collection other than the AP corpus to develop collection frequency statistics such as IDF or auxiliary data structures such as automatically generated thesauri.

As each participating system returns an unordered set of documents instead of a ranking, the evaluation measure is quite different from the measures used for ad hoc or routing tasks in TREC. Classical set-based evaluation measures such as raw precision and recall do not behave gracefully for topics with asymmetric class value distributions. The adaptive filtering task of TREC-7 used utility functions to measure the quality of the retrieved documents. In particular, such a quality metric is computed as a function of the benefit of retrieving a relevant document and the cost of retrieving an irrelevant document. In TREC-7, two utility functions namely F1 and F3 were used. In general, the F1 measure favours precision-oriented IR systems and the F3 measure favours recall-oriented IR systems:

$$F1 = 3 \times a - 2 \times b \tag{5.7}$$

$$F3 = 4 \times a - b \tag{5.8}$$

With reference to the contingency table 5.1, a and b are the number of relevant and non-relevant documents retrieved respectively. However, our experimental procedure differed from the TREC-7 evaluation method in that the agent system was allowed to use rejected documents as training examples to refine its knowledge base because our belief revision model can learn *beliefs* as well as *disbeliefs*. Moreover, only a subset of the TREC-AP collection (AP-89) was used in our experiments. The TREC-7 results indicated that the adaptive filtering task was a very challenging problem even for the sophisticated IR systems [Hul98]. Therefore, adopting the TREC-7 adaptive filtering bench-marking procedure to assess our belief-based agent system AIFS has the added advantage of examining the scalability of the system for large and complex text filtering tasks.

5.4 Experiment on Entrenchment Induction

All the experiments reported in this thesis were conducted on Intel Pentium III 800MHz PCs with 256MB main memory running under Windows2000. Though the agent system was also tested on Sun Microsystems' Enterprise Server under SunOS 5.7, for the reason of consistency, only the performance data collected from the Pentium III based PC platforms are reported. Since the induction of epistemic entrenchment orderings is a crucial step for applying belief revision and non-monotonic reasoning to adaptive information agents, the first experiment aimed at identifying an effective and efficient method to induce epistemic entrenchment orderings representing information seekers' information preferences. All the test runs were performed based on the Reuters-21578 collection in this experiment. The candidate induction methods which were subject to empirical testing included expected cross entropy for text (EH) Eq.(4.2), mutual information (MI) Eq.(4.1), the original version of the keyword classifier (KC) Eq.(4.3), modified keyword classifier (MKC) Eq.(4.8), odds ratio (OR) Eq.(4.4) and Eq.(4.5), and normalised TFIDF Eq.(2.1) with Rocchio updating Eq.(4.7). If a candidate entrenchment induction method produces term scores outside the unit interval [0,1], the terms scores will be scaled to the unit interval by a linear function Eq.(4.6). The scaling process was performed by first parsing the entire document collection to identify the maximal raw term score $S(t)_{max}$ and the minimal term score $S(t)_{min}$. Then, a second pass was followed and the prescribed induction method was invoked to induce the beliefs and their associated entrenchment degrees. All the candidate methods used the same adjustment factor ϵ to adjust the entrenchment degrees so that induced beliefs would not be assigned the maximal degree 1 which is attached to tautologies only. The same belief revision method (e.g., Rapid Anytime Maxi-adjustment RAM) and document scoring procedure were applied to each induction method. For each candidate method, the effectiveness measures such as the F_1 measure, F1 utility and F3 utility were collected for 20 runs corresponding to the first 20 topics of the Reuters-21578 collection. These results were macro-averaged to facilitate comparison. The final result for the six entrenchment induction methods is depicted in Table 5.3.

Induction Method	F_1 measure	F1 Utility	F3 Utility
Odds Ratio	0.365	112.5	160.1
Mutual Information	0.117	10.8	63.2
Expected Cross Entropy	0.046	-113.3	-56.9
TFIDF+Rocchio	0.301	90.2	144.6
Keyword Classifier	0.386	124.1	188.3
Modified Keyword Classifier	0.486	160.4	285.1

Table 5.3: Comparison of Entrenchment Induction Methods

Among the evaluated entrenchment induction methods, the method that was adopted from expected cross entropy for binary text classification [KS97] performed worst. As can be seen in Eq.(4.2), the raw term score is mainly derived from the sum of the two conditional probabilities $Pr(Rel|t)$ and $Pr(Nrel|t)$ and normalised by the probability of term appearance $Pr(t)$. Even if a term t appears frequently in both the set of relevant documents D^+ and the set of non-relevant documents D^- (e.g., $Pr(Rel|t)$ and $Pr(Nrel|t)$ are even), the raw term score $S(t)$ based on the expected cross entropy for text could be high. This entrenchment induction method does not correspond to our intuition about epistemic entrenchment orderings. A term t often found in D^+ should be a highly entrenched belief about a user's information need (i.e., positive keyword). On the other hand, if the term t often appears in D^-, it is a highly entrenched disbelief (i.e., negative keyword). If t appears frequently in both D^+ and D^-, it is not a good indicator (i.e., neither belief nor disbelief) of what the user likes or dislikes (i.e., neutral keyword). It is also observed that if a term t often appears in D^- only, a medium term score may be generated according to Eq.(4.2). Unfortunately, there is no way to distinguish if it is a belief or disbelief. Such a distinction is important in our belief-based classification framework since beliefs in an agent's knowledge base are used to infer relevant documents and disbeliefs are used to reject non-relevant documents. Without such a distinction, it is possible that disbeliefs could be mistakenly used to infer a user's positive information need. Consequently, poor retrieval performance was observed.

It should be noted that information gain was regarded as one of the most effective feature selection methods [YP97]. Indeed, information gain is equivalent to expected cross entropy. However, the main difference between our experiment and the one reported by Yang and Pedersen [YP97] is that the output of their feature selection mechanism was consumed by a K-Nearest Neighbours (KNN) classifier or a linear regression model. However, our induced entrenchment degrees are reasoned about by the belief revision engine. Indeed, there is a mis-match between how the entrenchment degrees are induced based on the expected cross entropy method and the way how these entrenchment degrees are interpreted by the belief revision engine. It is believed that the correspondence between a feature selection method and a particular classification model is crucial for improved IR performance [YP97]. The second difference is that Eq.(4.2) represents a specialisation of the general expected cross entropy measure. In fact, only term presence (e.g., $Pr(Rel|t)$ and $Pr(Nrel|t)$) is taken into account in Eq.(4.2). Nevertheless, both term presence and term absence (e.g., $Pr(Rel|\neg t)$) are included in the general expected cross entropy formulation. So, Eq.(4.2) is not exactly the same as the notion of information gain often referred to in the machine learning community [Qui86].

The Mutual Information measure Eq.(4.1) was tested for entrenchment induction as well. Particularly, $MI(t, Rel) = \log_2 \frac{Pr(t \wedge Rel)}{Pr(t)Pr(Rel)}$ was used to induce the entrenchment degree. According to our experiment, the MI method is not effective as reflected by the low F_1 value, F1 utility score, and F3 utility score. The reason is that the MI measure favours rare terms which does not correspond to our intuition about epistemic entrenchment orderings. A term seldom appearing in a document collection does not necessarily imply that it is a highly entrenched belief representing the most preferred information item. In fact, Eq.(4.1) can be expressed by:

$$MI(t, Rel) = \log_2 Pr(t|Rel) - \log_2 Pr(t) \tag{5.9}$$

From Eq.(5.9), it is easy to observe that given the same $Pr(t|Rel)$, a rare term (i.e., low $Pr(t)$) will have a higher $MI(t, Rel)$ score. Such a highly weighted term would be converted to a highly entrenched belief even though the underlying term may not be a strong indicator of a user's positive information need. The epistemic entrenchment induction methods which are based on Odds Ratio (OR) and normalised TFIDF with Rocchio term re-weighting produce better IR results when compared with that of the MI and EH methods. The OR method seems slightly better than the TFIDF method. Odds Ratio was proposed for document ranking [vRHP81], but it is not the best method for epistemic entrenchment induction. The reason is that the Odds Ratio as defined in Eq.(4.4) generates a high term score for a term t if $Pr(t|Rel)$ is much higher than $Pr(t|Nrel)$. Indeed, the Odds Ratio based induction method partially corresponds to our intuition about epistemic entrenchment orderings. Therefore, the resulting IR performance is moderate. However, the Odds Ratio method did not perform as well as the Keyword Classifier method because it assigned very positive terms with high term score and very negative terms with low term score. This principle is suitable for document ranking because negative information items should be presented after all the positive information items. However, in the context of epistemic entrenchment induction, very negative terms can also be used as *disbeliefs* by the information agent to reject non-relevant items. The original formulation of Odds Ratio Eq.(4.4) is not effective in inducing disbeliefs which are quite useful for rejecting non-relevant documents.

Surprisingly, the entrenchment induction method based on normalised TFIDF re-weighted by the Rocchio method produced comparable performance to that of the Odds Ratio based induction method. TFIDF is an effective method to identify the most discriminatory terms for document representation. Nevertheless, a representative term in a document does not necessarily imply that it is a good representation (i.e., a belief) of a user's information need. Therefore, the original intention of testing this method was to provide a baseline result to compare with other information theoretic approaches. The reason why this method performed better than the other information theoretic methods such as mutual information may be that the Rocchio method is effective with respect to re-weighting the terms based on the set of positive training examples D^+ and the set of negative training examples D^- [SB90]. These re-weighted terms (and hence the induced beliefs) can more or less represent a user's current information needs. In our experiment related to the TFIDF method, only normalised positive TFIDF weights (i.e., in the interval $[0, 1]$) were used. Therefore, disbeliefs, which can be used to reject non-relevant documents, were not induced.

Besides, there is still a fundamental mis-match between the TFIDF weights and the epistemic entrenchment degrees. For example, a term with a very high TFIDF weight in a positive document and a low TFIDF weight in a negative document (e.g., because of its low term frequency) may still generate a moderate term score by Rocchio updating because negative documents are only penalised by a small factor (e.g., 0.25), whereas positive documents are rewarded by a high rating factor (e.g., 0.75). Accordingly, a moderately entrenched belief could be induced even though the underlying term is more likely to be considered as a neutral keyword because of its even appearance in D^+ and D^- respectively. This may explain why entrenchment induction based on normalised TFIDF is not as effective as the keyword classifier (KC) method. According to our experiment, the normalised TFIDF method for entrenchment induction is quite inefficient. The change of one term weight may affect all the other term weights because of the cosine normalisation procedure. Even though such a normalisation process may not be computationally expensive, performing belief revision (i.e., raising or lowering the entrenchment degree) for every affected term (i.e., belief) is very time consuming. According to our empirical study, on average it took 2.4 seconds more to process a document if the normalised TFIDF method rather than the MKC method was used for entrenchment induction. This result indicates that the TFIDF method for entrenchment induction is not appealing for large IR applications.

Both the KC method Eq.(4.3) and the modified MKC method Eq.(4.8) outperformed other entrenchment induction methods. Even if the term $Pr(c|t) \log_2 \frac{Pr(c|t)}{Pr(c)}$, where $c \in \{\text{Relevant}, \text{Non-relevant}\}$, appearing in both Eq.(4.3) and Eq.(4.8) is exactly the same as the one appearing in the expected cross entropy formula Eq.(4.2), the resulting term scores are quite different as derived from the respective methods.

The main difference is that the conditional probability $Pr(Nrel|t)$ will lower the overall term score in the keyword classifier formulas. Moreover, without the presence of a term t in the relevant set D^+ (i.e., $Pr(Rel|t) = 0$), a negative term score is derived from $Pr(Nrel|t)$. Such a negative term score exactly reflects the entrenchment degree of the corresponding *disbelief*. In other words, if a term t only appears in the non-relevant set D^-, it may become a strong disbelief of a user's information need. This disbelief can then be used by the agent to reject non-relevant documents. Accordingly, the combined recall and precision score F_1 is better than that as obtained via other methods. In addition, the modified keyword classifier MKC method is appreciably better than the KC method because it can take into account the asymmetric distribution of class values (e.g., many non-relevant documents vs. relevant documents). If the majority class (e.g., non-relevant) is not the target class (e.g., relevant), the term score as derived by Eq.(4.3) is dominated by the negative terms. As a consequence, the agent's knowledge base will only be filled with many highly entrenched disbeliefs. While the disbeliefs can help improve precision in IR, it does not help retrieve relevant documents at all.

The modified keyword classifier Eq.(4.8) takes into account the possible asymmetric class value distribution by weighting positive evidence and negative evidence with different factors (i.e., the α and β parameters). In information filtering situations, the positive learning threshold α is set much higher than the negative learning threshold β. Therefore, a disbelief will only be induced if the corresponding term is found from a large number of non-relevant documents. This assumption better captures the reality in information filtering situations where there is a relatively higher chance that an arbitrary term is found in non-relevant documents. To be considered as a negative keyword (disbelief), the term must appear quite frequently in the non-relevant set D^-. Our experimental results confirmed the above observation. Upon closer examination of the agent system's knowledge base, it was found that the knowledge base was dominated by disbeliefs when the KC method was used to induce entrenchment orderings. However, when the MKC method was used, the agent's knowledge base contained more evenly distributed beliefs and disbeliefs. This is the reason why the agent system's retrieval performance is better when the MKC method is used. As a result, the MKC method is adopted as the standard method to induce the epistemic entrenchment orderings within AIFS.

5.5 Experiment on Adaptive Learning

One important property of adaptive information agents is their abilities to continuously learn users' changing information needs so as to improve the agents' retrieval performance over time. If our adaptive agents are effective in learning users' information needs, improvement of the agents' retrieval performance should be observed over time. To evaluate the agent system's learning effectiveness, the F_1 scores were plotted against the number of training examples encountered by the system. The F_1 measure is used instead of the F1 utility or the F3 utility because the utility values are related to the number of relevant documents in a document collection. For example, if there are 10 relevant documents out of the first 1000 documents in a collection and an agent performs quite well so that all the 10 relevant documents are retrieved, the utility score is positive. However, if there is no relevant document in the following 1000 documents and the agent is still performing well by rejecting all the non-relevant documents, the utility score will drop to zero. Therefore, the utility functions are not good indicators for the agents' adaptive learning performance. On the other hand, the F_1 measure allows us to monitor if an agent's performance is really changing independent of the number of relevant documents contained in a collection. To evaluate an agent's learning and classification performance over time, the document collection is evenly divided into several subsets to observe the agent's performance in different periods. The hypothesis is that if the proposed belief revision framework for adaptive information agents is effective, the F_1 scores should improve over time. In other words, an up turning curve should be observed. To test this hypothesis, both the Reuters-21578 and the AP-89 collections were used. A set of topics, some with many relevant documents and some with few or no relevant documents, were used to test the agents' learning performance under various retrieval situations.

For TREC topic 10 (112 relevant documents) and topic 17 (106 relevant documents), the hypothesis is confirmed in that the agent's classification performance in the last period (documents 70,000 - 80,000) is improved when compared with the first period (documents 1 - 10,000). These results are plotted in Figure 5.1 and Figure 5.2

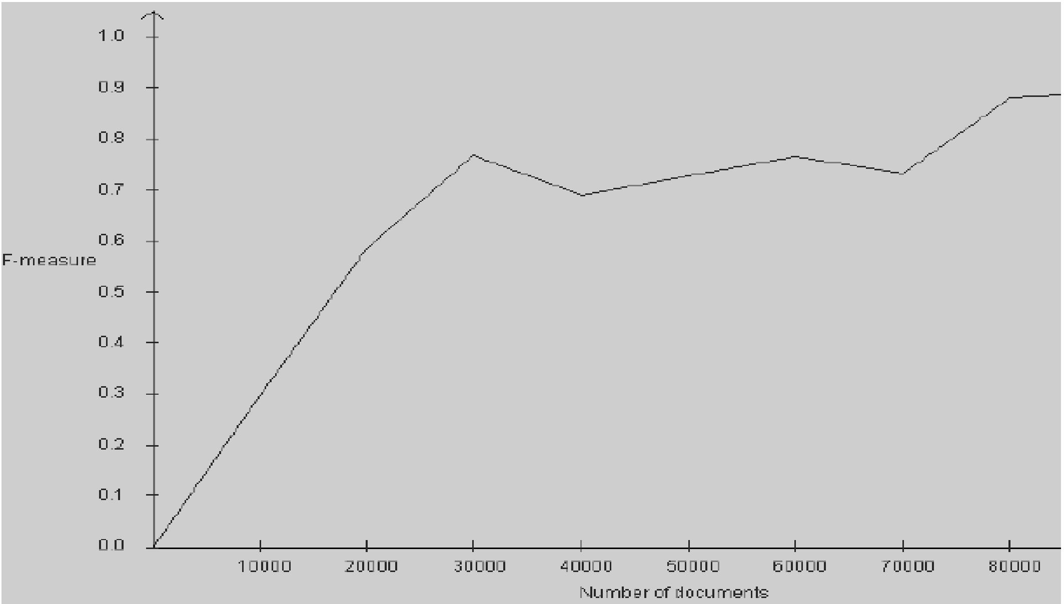

Figure 5.1: Adaptive Learning (TREC Topic 10)

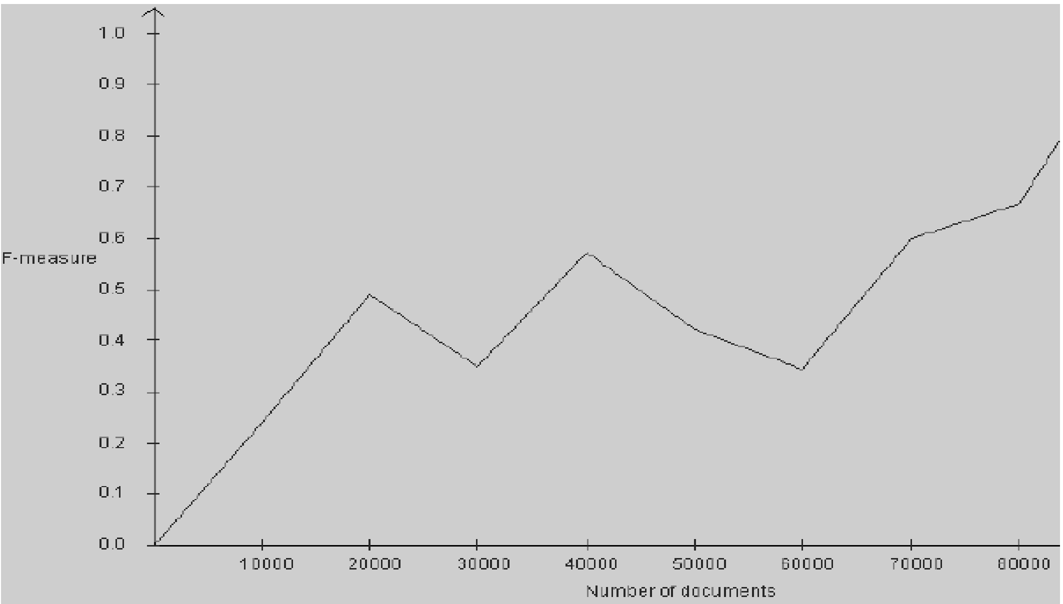

Figure 5.2: Adaptive Learning (TREC Topic 17)

respectively. The plotting after the 80,000 document point should be ignored because the number of documents in this interval is not the same as that of the previous in-

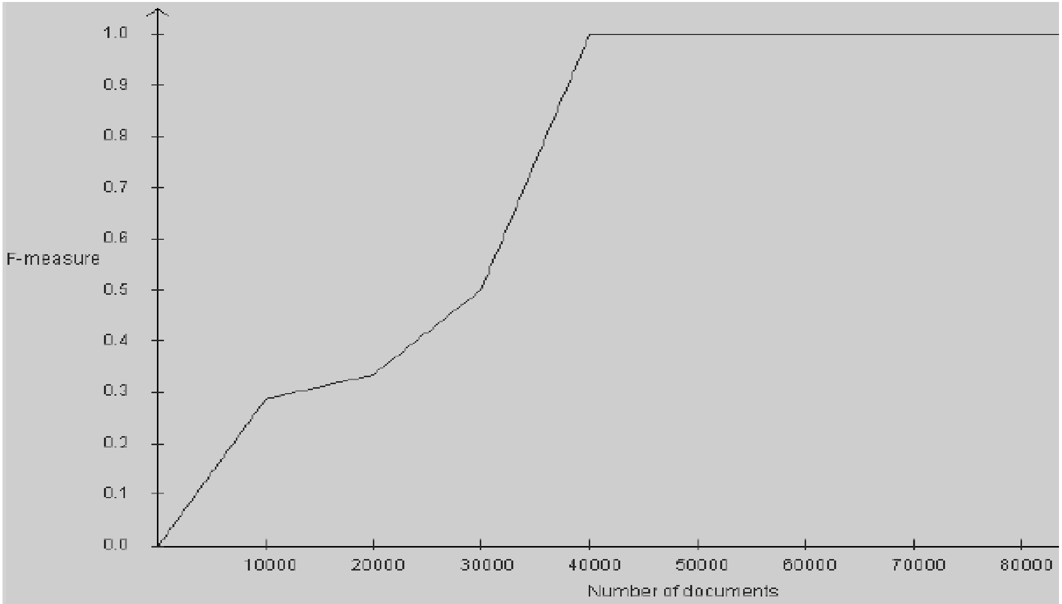

Figure 5.3: Adaptive Learning (TREC Topic 37)

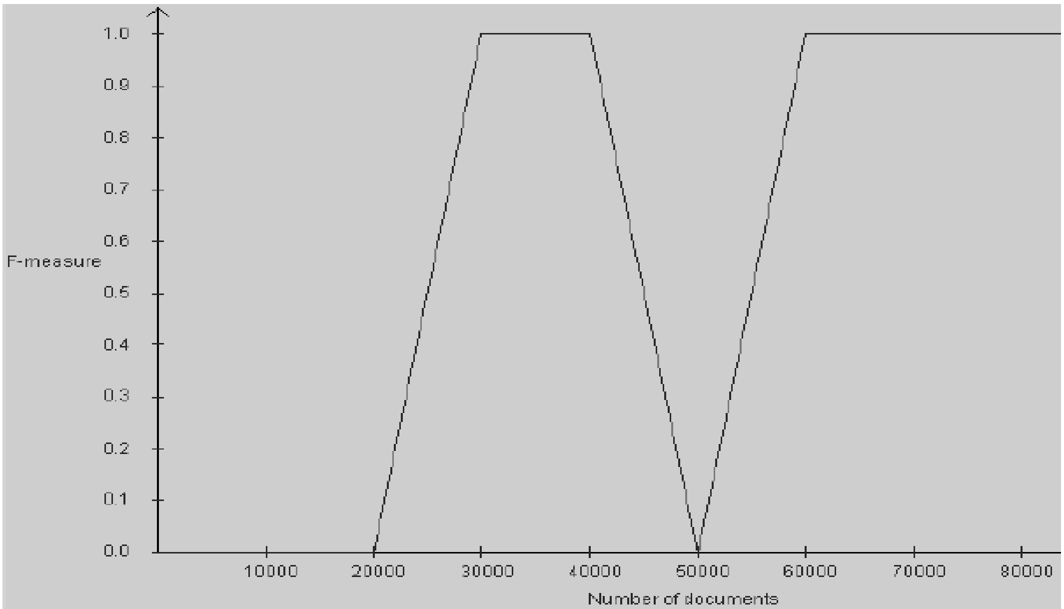

Figure 5.4: Adaptive Learning (TREC Topic 21)

tervals. There is fluctuation in the middle periods. This fluctuation of performance is expected since a user's information needs will change over time. Therefore, the fluctuated figures reflect the periods when an agent is learning new information needs. In general, the lowest points in these fluctuated periods are still higher than the point plotted after the initial period (documents 1 - 10,000). This indicates that the agent has acquired the user's basic information interests after the initial learning period. For TREC topic 37 (7 relevant documents), the agent can also learn the user's preferences quickly even with a small number of positive training examples as demonstrated in Figure 5.3. This is reflected by the positive F_1 score after the first period (documents 1 - 10,000). The agent's performance keeps improving since then. The maximal filtering performance is obtained during the fourth period (documents 30,000 - 40,000), and this performance is maintained since then. The maximal filtering performance achieved after the fourth period is due to the stable interests of the user exhibited in these later learning periods.

However, for TREC topic 21 (2 relevant documents) as depicted in Figure 5.4, the agent's learning process is not so obvious. Particularly, there is a serious drop of performance in a relatively late period (documents 40,000 - 50,000) when improved performance is expected. Upon closer examination, it was found that there was one relevant document in the corresponding document subset. Unfortunately, the agent system rejected all the documents. Consequently, the F_1 score dropped to zero. Such a zero score does not necessarily indicate that the agent performed very poorly. In general, if there is only a small number of relevant documents with respect to a given information need, it will be quite difficult for an agent to learn an accurate user model.

Therefore, the agent's classification performance may fluctuate considerably due to missing some relevant documents. As a whole, with few positive training examples, it is quite difficult for the agent to learn a user's positive information needs. It seems that the proposed belief revision learning model is effective if there is a reasonable number of positive training examples to gradually train the agent. The proposed learning model may suffer from the same weakness pertaining to the state of the art machine learning algorithms such as the Boosting method [SSS98]. However, the advantage of the belief revision model is that it enables an agent to learn incrementally rather than learning in a batch mode manner.

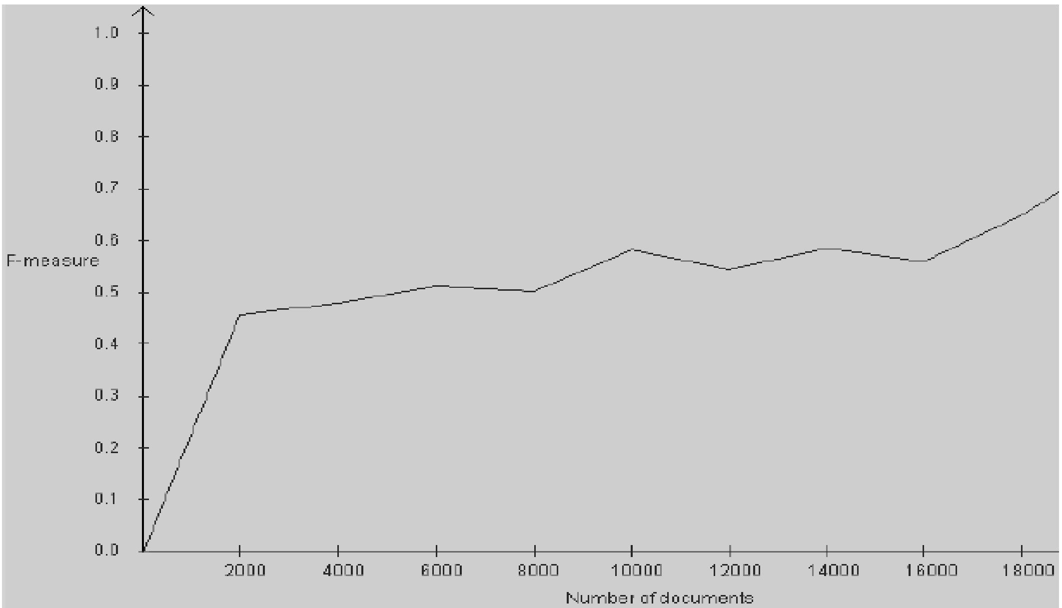

Figure 5.5: Adaptive Learning (Reuters-21578 Topic 01)

The adaptive learning performance of the information agents was also examined with reference to the runs based on the Reuters-21578 collection. Figure 5.5 and Figure 5.6 depict two such runs. For Reuters topic 1 (2366 relevant documents) and

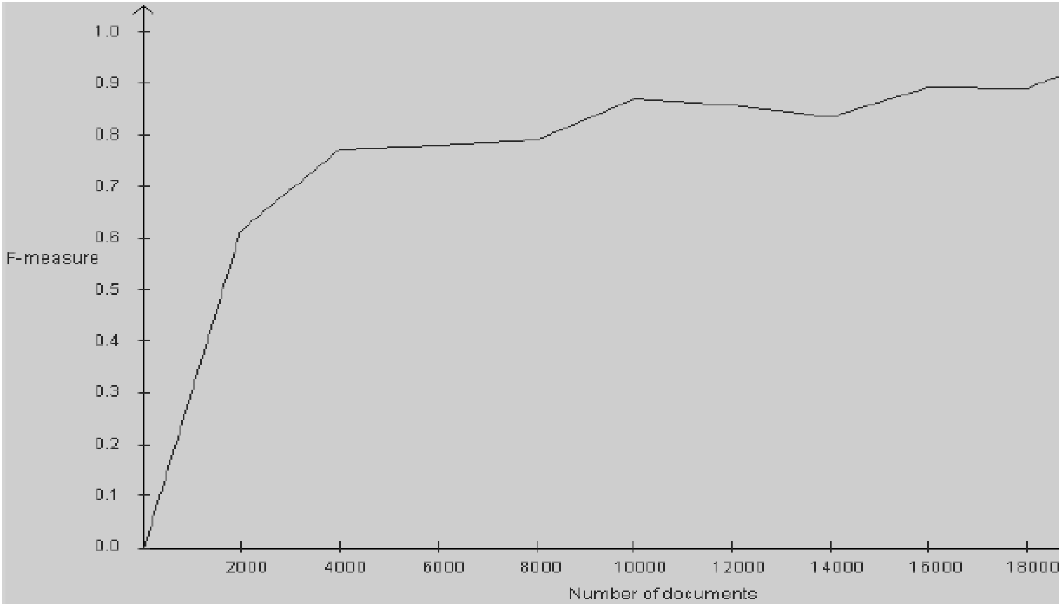

Figure 5.6: Adaptive Learning (Reuters-21578 Topic 20)

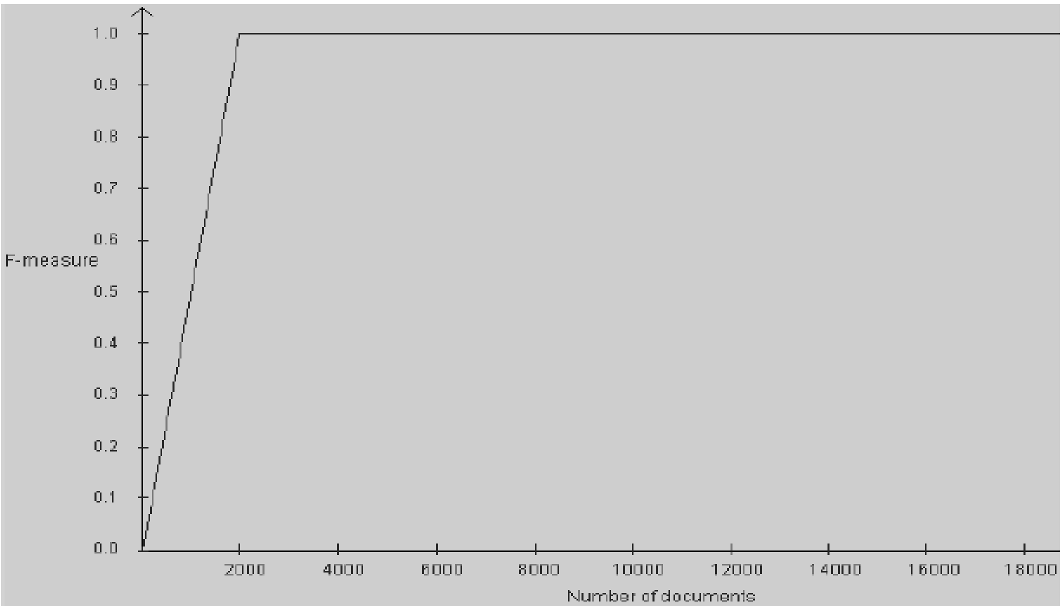

Figure 5.7: Adaptive Learning (Reuters-21578 Topic 10)

topic 20 (237 relevant documents), adaptive learning was realised by the improved classification performance recorded in the last period (documents 16,000 - 18,000)

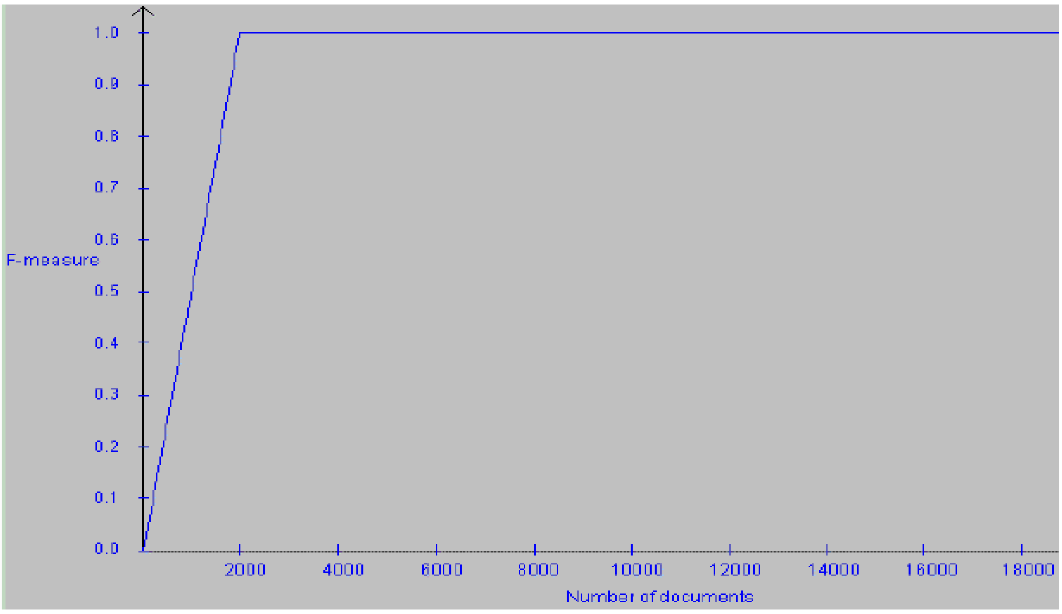

Figure 5.8: Adaptive Learning (Reuters-21578 Topic 12)

when compared with that obtained in the initial period (documents 1 - 2,000). These results confirm that the agent system can learn a retrieval context over time and employ the acquired knowledge to make more accurate classification decisions. Again the plotting after the 18,000 document point should be ignored because there are insufficient documents to compute an agent's average performance figure after that period. There is not as large a fluctuation in the middle periods as demonstrated by the test runs based on the AP-89 collection (e.g., Figure 5.2). This observation can be explained in that the information needs of the hypothetical user as represented in many Reuters topics are more or less stable. In fact, the retrieval effectiveness as demonstrated by many previous studies based on the Reuters-21578 collection is generally better than that obtained based on the TREC AP collection [SSS98, YP97]. The learning and classification tasks based on the Reuters-21578 collection are considered easier than that of the TREC-AP collection. In our experiment, the agent system AIFS was able to learn rapidly (e.g., a sharp up-turning curve) during the initial learning period. Further investigation found that there were quite a number of relevant documents among the first 2,000 documents for both topics. Because of the effective belief revision mechanism, the agent system was quite responsive to these positive training examples and was able to learn high quality user profiles early in its learning cycle. The agents' learning and classification performance for topics with few or no relevant documents in the Reuters-21578 collection is different from that obtained based on the AP-89 collection. For Reuters topic 10 (0 relevant documents) and topic 12 (1 relevant document), the agent system was able to learn as effectively and quickly as the other topics where a large number of positive training examples existed. As shown in Figure 5.7 and Figure 5.8, after the initial learning periods, the agent system was able to maintain its maximal filtering performance in the subsequent periods. This difference can be explained in that the information needs as presented in most of the Reuters topics are relatively stable. Once the agent system learns the beliefs about the users' information needs, the system can make use of its knowledge to predict the relevance of forth-coming documents. In the case that there is no relevant document for a topic, the agent system can make use of its disbeliefs effectively to reject all non-relevant documents.

5.6 Experiment on Transmutation Methods

The objective of running this experiment is to evaluate the various belief revision methods. Strictly speaking, it is the corresponding adjustment methods being tested in an empirical setting. According to previous theoretical analysis, the Maxi-adjustment method should be superior to the standard AGM adjustment method because the beliefs of a user's information needs will be retained unless there is really a reason (e.g., a logical implication) to support the contraction [Wil96a, LtHB01a, LtHB01b]. However, the computational cost of the Maxi-adjustment method is a major concern when it is applied to process large and complex applications. It is believed that the computationally more efficient *Anytime Maxi-adjustment* method can produce close approximations of the results as generated by the Maxi-adjustment method [Wil97]. The *Anytime* approach is theoretically sound, but to our knowledge its effectiveness and efficiency in large real-life applications are yet to be validated. So, one of the goals of this experiment is to examine the properties of the Anytime mechanism in an empirical setting. In addition, as the Rapid Anytime Maxi-adjustment method (RAM) is proposed in this thesis, this experiment also aims at evaluating the effectiveness of the RAM method and comparing its performance with that achieved by the Anytime Maxi-adjustment method.

Two text filtering tasks were used to examine the various adjustment methods. The first text filtering task was conducted based on the TREC topic 22 with the largest number of relevant documents (524) in the AP-89 collection. It is also one of

Adjustment Methods	Time	F_1 measure	F1 Utility	F3 Utility
Standard AGM	12H44M45S	0.129	-121	-23
Maxi-adjust	271H7M56S	0.236	-93	176
Anytime Maxi-adjust	48H50M43S	0.241	-88	192
Rapid Anytime Maxi-adjust	32H28M18S	0.248	-86	198

Table 5.4: Comparison of Adjustment Methods (TREC Topic 22)

the most time-consuming runs among the 50 TREC topics. Moreover, TREC topic 37 with a few relevant documents (7) was also used to test the various adjustment methods. The motive of using these two TREC topics to test the various adjustment methods is to compare their performance under quite different retrieval situations (e.g., one with many beliefs to learn and one with only a few beliefs to learn). In order to test the scalability of the belief revision framework, the AP-89 collection was used instead of the Reuters-21578 collection. The results are depicted in Table 5.4 and Table 5.5 respectively. The time limit applied to both the anytime Maxi-adjustment method and the RAM method was $5000ms$ in this experiment. For a retrieval domain with many positive beliefs, it takes substantially longer time for the Maxi-adjustment method to learn and revise the beliefs into an agent's knowledge base. In fact, it is 22 times longer than using the standard AGM adjustment method.

This problem can be explained based on the current learning and revision method. In all our experiments, the standard document pre-processing approach is to take the top 50 terms with the highest TFIDF weights to represent a document. Based on our current entrenchment induction method (MKC), a relevant document may have up to 50 beliefs induced and revised into a theory base. If all the terms are new to the agent system, these beliefs will have exactly the same entrenchment degree. In other words, it is possible to have 50 or even more beliefs with the same rank pertaining to an epistemic entrenchment ordering. The maxi-adjustment algorithm computes the minimal inconsistent subsets of beliefs if there is more than one belief in a particular rank during a belief revision operation. The minimal subset computation is exponential with time complexity $O(2^n)$ in the worst case, where n is the number of beliefs in a particular rank. As demonstrated in this empirical testing, the Maxi-Adjustment method does not scale up well for demanding applications such as text filtering for a large document collection. The agent system took 271 hours to filter the AP-89 collection if the Maxi-adjustment method was used. Moreover, it is surprising to find that the learning effectiveness of the Maxi-adjustment method is not better than that of the Anytime Maxi-adjustment method nor the RAM method. The reason may be that although the Maxi-adjustment method can theoretically retain more beliefs in an agent's knowledge base than the other methods do, these beliefs may not be significant (e.g., beliefs with low entrenchment degrees). Some of these insignificant beliefs may eventually cause the agent system to make inaccurate classification decisions. Accordingly, both the F1 and the F3 utility scores achieved by the corresponding agent were low because of the penalty applied to the wrong classification.

The standard AGM adjustment method produces the fastest belief revision operation. This method was invoked in the agent system by setting zero (i.e., no time limitation) for the time limit parameter in the Anytime standard AGM adjustment procedure. The complete algorithm of the standard AGM adjustment method is documented in Appendix D. However, the learning effectiveness of the agent system based on the standard AGM adjustment method is not as good as that of the other adjustment methods. After a closer examination of the agent system's theory base, it was found that some useful beliefs were not captured in the agent's theory base when the standard AGM adjustment method was applied. The reason is that the standard AGM belief revision operation will contract any contradictory beliefs as well as beliefs with entrenchment degrees lower than or equal to these contradictory beliefs from an agent's knowledge base. This finding confirms previous theoretical analysis in that the standard AGM belief revision operator may not be suitable for adaptive text filtering applications [LtHB01b]. In our initial experiment, both the Anytime Maxi-Adjustment method and the Rapid Anytime Maxi-Adjustment method (RAM) produced promising results. The performance figures of these two methods are comparable, but the RAM method is slightly better. Since the RAM method does not involve the computation of minimal inconsistent subsets when belief contraction takes place, it is faster than the Anytime Maxi-Adjustment method as validated by the respective text filtering tasks. In addition, the F_1 measure, F1 utility and F3 utility achieved by the RAM method are also slightly better than those obtained based on the Anytime Maxi-Adjustment method. It was found that some disbeliefs in the agent's theory base after applying the RAM method did not exist in the agent system's theory base if the Anytime Maxi-Adjustment method was applied. These beliefs might be lost during a belief revision operation when the time limit of the Anytime Maxi-adjustment method was exceeded. As the RAM method is faster than the Anytime Maxi-adjustment method (e.g., no minimal subsets of beliefs are computed), there is less chance that some significant beliefs are lost because the time limit of a belief revision operation is reached. As a result, more accurate document classification is achieved based on a larger number of reliable beliefs about the current retrieval context.

Adjustment Methods	Time	F_1 measure	F1 Utility	F3 Utility
Standard AGM	0H48M13S	0	0	0
Maxi-adjust	4H49M02S	0.767	12	16
Anytime Maxi-adjust	2H45M16S	0.769	13	19
Rapid Anytime Maxi-adjust	2H26M33S	0.769	13	19

Table 5.5: Comparison of Adjustment Methods (TREC Topic 37)

When there were only a few positive examples to be learnt from a topic, the various adjustment methods, except standard AGM adjustment, produced comparable results as depicted in Table 5.5. The standard AGM adjustment method produced the poorest result measured in terms of the F_1 measure, F1 utility, and F3 utility. After examining the agent's theory base, it was observed that some positive beliefs learnt by the other adjustment methods were not present if the standard AGM adjustment method was applied. This is perhaps caused by the fact that some beliefs with equal or lower entrenchment degree may be contracted from the agent's knowledge base along with the belief causing the inconsistency. Without an adequate representation of the retrieval context, the agent could not identify the relevant documents. Accordingly the F_1 measure, F1 utility and F3 utility were all zeros when the standard adjustment method was invoked.

The Maxi-adjustment method, Anytime Maxi-adjustment method, and Rapid Anytime Maxi-adjustment method achieved comparable learning effectiveness. Again it took longer to filter the AP-89 collection if the Maxi-adjustment method was applied. The additional time was consumed while the Maxi-adjustment method processed some disbeliefs with the same rank. In fact, there were only a few disbeliefs with the same entrenchment degree (i.e., in the same rank) in the agents' theory bases for this filtering task. Consequently, the Maxi-adjustment method did not consume substantially more time to learn the hypothetical user's changing information needs when compared with the RAM method. The small time difference between the Maxi-adjustment method and the Anytime Maxi-adjustment method also indicated that the anytime feature was only invoked occasionally to terminate a belief revision operation when the Anytime Maxi-adjustment procedure was executed. However, after examining the agents' theory bases, it was found that some disbeliefs captured in the agent's theory base when the Maxi-adjustment method was applied did not appear in the agent's theory base if the anytime Maxi-adjustment method or the RAM method was invoked. This probably explains the small difference of the F_1 measure, F1 utility, and F3 utility when these methods were invoked. For this filtering task, the Anytime Maxi-adjustment method produced the same set of positive beliefs as the RAM method did. This is the reason why their learning effectiveness is the same. However, the Anytime Maxi-adjustment method still consumed a bit more time to learn the retrieval context because it enumerated the minimal inconsistent subsets in several entrenchment ranks.

As a whole, this preliminary experiment provides new empirical evidence to support the concept of *anytime* belief revision [Wil97]. The Anytime Maxi-adjustment method achieves comparable learning effectiveness to that of the Rapid Anytime Maxi-adjustment method if there is a small number of beliefs to be learnt by the agents. However, if there are many beliefs to be learnt from a retrieval topic, the RAM method is more promising than the Anytime Maxi-adjustment method in terms of both learning effectiveness and computational efficiency. Therefore, the RAM method is applied to our belief-based information agent system AIFS. The remaining experiments reported in this thesis are all based on the RAM adjustment algorithm and the MKC entrenchment induction method.

5.7 Evaluation of AIFS based on the Reuters-21578 Collection

The overall filtering performance of the AIFS prototype system based on the Reuters-21578 collection is depicted in Figure 5.9. The system was evaluated against 20

Reuters topics. The first column in Figure 5.9 shows the topic number, and the second column lists the number of relevant documents pertaining to each topic. These figures represent the actual number of relevant documents judged by human assessors. The remaining columns show the classification accuracy, recall, precision, F_1 measure, F1 utility, F3 utility, and filtering time in seconds. The last row in Figure 5.9 shows the average result across topics. The proposed agent system achieves an average F_1 of 0.486, an average F1 utility score of 160.4, and an average F3 utility score of 285.1. The average time of filtering a topic for the "Modified Lewis Split" subset (19,702 documents) of the Reuters-21578 collection is 1,791.1 seconds (around 30 minutes). Therefore, on average our belief-based agent system AIFS spends about 0.091 second to learn and to classify if a document is relevant with respect to a user's changing information needs. These efficiency figures produce concrete evidence that the proposed logical framework is feasible for the development of adaptive information agents which deal with large on-line information retrieval tasks.

From among the 20 topics, the most time consuming one is topic 1 which involves a significant number of belief revision operations to learn both beliefs (i.e., positive keywords) and disbeliefs (i.e., negative keywords). The processing time related to topic 1 represents our worst case of filtering for the Reuters-21578 collection. The agent spent about 1.04 seconds to process (e.g., classification and learning) a document in this worst case scenario. However, the effectiveness result of topic 1 is quite good. The classification accuracy, the F_1 score, the F1 utility, and the F3 utility are 0.907, 0.533, 2098, 3674 respectively.

Topic	Rel.Doc.	Accuracy	Recall	Precision	F-measure	F1	F3	Seconds
1	2366	0.907	0.444	0.666	0.533	2098	3674	20417
2	57	0.997	0.263	0.555	0.357	21	48	622
3	4	0.999	0.000	1.000	0.000	0	0	315
4	0	0.999	1.000	0.000	0.000	-8	-4	254
5	51	0.999	1.000	0.739	0.850	117	186	328
6	0	1.000	1.000	1.000	1.000	0	0	313
7	105	0.995	0.048	0.385	0.085	-1	12	552
8	3	0.999	0.000	0.000	0.000	-32	-16	313
9	68	0.997	0.235	0.533	0.327	20	50	872
10	0	1.000	1.000	1.000	1.000	0	0	259
11	2	0.999	0.000	1.000	0.000	0	0	260
12	1	1.000	1.000	1.000	1.000	3	4	1995
13	1	1.000	1.000	1.000	1.000	3	4	2270
14	73	0.999	0.945	0.793	0.863	171	258	598
15	6	0.999	1.000	0.231	0.375	-22	4	335
16	7	0.999	0.000	1.000	0.000	0	0	320
17	139	0.996	0.986	0.655	0.787	267	476	3656
18	65	0.998	0.985	0.604	0.749	108	214	661
19	3	0.999	0.000	0.000	0.000	-4	-2	302
20	237	0.994	0.945	0.685	0.794	466	793	1180
Avg.		0.994	0.593	0.642	0.486	160.4	285.1	1791.1

Figure 5.9: Overall Results of AIFS by Reuters-21578 Topics

In order to gain more insight into the performance of the belief-based agent system, a base line agent system (VSpace) was developed and applied to the same filtering task. All the experimental conditions remained the same except that the VSpace agent system was developed based on the Vector Space model [SM83] and using the Rocchio learning method [Roc71] to revise the term weights captured in a

Topic	Accuracy	Recall	Precision	F-measure	F1	F3	Seconds
1	0.896	0.264	0.667	0.378	1248	1872	1032
2	0.998	0.246	0.875	0.384	38	52	329
3	0.999	0.000	1.000	0.000	0	0	266
4	0.999	1.000	0.000	0.000	-8	-4	250
5	0.998	0.843	0.623	0.717	77	120	311
6	1.000	1.000	1.000	1.000	0	0	250
7	0.995	0.476	0.500	0.488	50	100	368
8	0.999	0.000	0.000	0.000	-12	-6	266
9	0.997	0.074	0.625	0.136	9	14	372
10	1.000	1.000	1.000	1.000	0	0	252
11	0.999	0.000	1.000	0.000	0	0	256
12	1.000	1.000	1.000	1.000	3	4	264
13	0.999	1.000	0.500	0.667	1	3	259
14	0.999	0.808	0.922	0.861	167	226	348
15	0.999	1.000	0.667	0.800	12	18	261
16	0.999	0.000	1.000	0.000	0	0	269
17	0.998	0.755	0.963	0.847	307	412	429
18	0.998	0.554	0.783	0.649	88	124	347
19	0.999	0.000	1.000	0.000	0	0	264
20	0.993	0.511	0.883	0.647	331	452	447
Avg.	0.993	0.527	0.750	0.479	115.6	169.4	342.0

Figure 5.10: Overall Results of VSpace by Reuters-21578 Topics

prototype vector representing the hypothetical user's information needs. The Vector Space model together with the Rocchio learning method is a well-known quantitative approach for developing IR and IF systems. This approach has been successfully applied to process large and complex IR tasks [Sal90]. In the base line system, the document pre-processing procedure is exactly the same as the belief-based agent system AIFS. The performance figures of the base line agent system are depicted in

Topic	Accuracy	Recall	Precision	F-measure	F1	F3	Seconds
1	0.011	0.180	-0.001	0.155	850	1802	19385
2	-0.001	0.017	-0.320	-0.027	-17	-4	293
3	0.000	0.000	0.000	0.000	0	0	49
4	0.000	0.000	0.000	0.000	0	0	4
5	0.001	0.157	0.116	0.133	40	66	17
6	0.000	0.000	0.000	0.000	0	0	63
7	0.000	-0.428	-0.115	-0.403	-51	-88	184
8	0.000	0.000	0.000	0.000	-20	-10	47
9	0.000	0.161	-0.092	0.191	11	36	500
10	0.000	0.000	0.000	0.000	0	0	7
11	0.000	0.000	0.000	0.000	0	0	4
12	0.000	0.000	0.000	0.000	0	0	1731
13	0.001	0.000	0.500	0.333	2	1	2011
14	0.000	0.137	-0.129	0.002	4	32	250
15	0.000	0.000	-0.436	-0.425	-34	-14	74
16	0.000	0.000	0.000	0.000	0	0	51
17	-0.002	0.231	-0.308	-0.060	-40	64	3227
18	0.000	0.431	-0.179	0.100	20	90	314
19	0.000	0.000	-1.000	0.000	-4	-2	38
20	0.001	0.434	-0.198	0.147	135	341	733
Avg.	0.001	0.066	-0.108	0.007	44.8	115.7	1449.1

Figure 5.11: Comparison (AIFS vs. VSpace) by Reuters-21578 Topics

Figure 5.10, and the comparison between the AIFS agent system and the base line system is shown in Figure 5.11. In Figure 5.11, a positive value represents how much the belief-based agent system out-performs the base line system, and a negative figure indicates that the belief-based agent system is inferior to the base line system. All the positive figures are highlighted in Figure 5.11.

The average F_1, F1 utility, and F3 utility achieved by the base line system are

0.479, 115.6, and 169.4 respectively. On average it only took 0.017 second to process a document. So, the Vector Space based information agent model is more efficient than our belief-based adaptive information agent model. In fact, the base line system is 5 times faster than the belief-based agent system in filtering the Reuters-21578 documents. However, in terms of learning and classification effectiveness, the belief-based agent system AIFS outperforms the base line agent in these filtering tasks. It is shown that the average F_1, F1 utility and F3 utility achieved by AIFS are all superior to the equivalent values produced by the base line agent system. The differences are 0.007, 44.8, and 115.7 respectively as depicted in the last row of Figure 5.11. The last column (total execution time in seconds) indicates how many seconds more are consumed by the belief-based agent system to process the documents. Since each filtering topic has distinct characteristics (e.g., number of relevant documents), the average F_1 score, F1 utility, and F3 utility across different topics may not be an elegant way to show the overall performance of the system. A better approach is to carry out a topic-by-topic comparison among systems. In this experiment, the AIFS system out-performed the VSpace system in 7 topics if their performance was measured in terms of the F_1 scores, whereas the VSpace system out-performed the AIFS system in only 4 topics. If the IR effectiveness is measured in terms of the F1 utility or the F3 utility, the number of topics that AIFS performed better than VSpace was also more than the number of topics that VSpace performed better than AIFS. As a whole, the initial experimental result shows that the belief-based agent model out-performs the vector space based agent model in many of the filtering tasks based on the Reuters-21578 collection. The cost of achieving this improved retrieval performance is spending about 0.073 seconds more to process each document. This small computational cost seems acceptable even for demanding interactive information retrieval activities.

5.8 Evaluation of AIFS based on the TREC-AP Collection

The overall filtering performance of the AIFS system against the AP-89 collection is depicted in Figure 5.12 and Figure 5.13 respectively. Figure 5.12 illustrates the result pertaining to the first 25 TREC topics, and Figure 5.13 shows the result of the remaining 25 topics as well as the average scores. In each table, the first column shows the topic number, and the second column lists the number of relevant documents pertaining to a topic. These figures represent the actual number of relevant documents judged by the TREC assessors. The remaining columns show the classification accuracy, recall, precision, F_1 score, F1 utility, F3 utility, and filtering time in seconds. The last row in Figure 5.13 shows the average result across topics. AIFS achieved an average F_1 of 0.175, an average F1 utility score of -7.7, and an average F3 utility score of 27.1. These performance figures are not as good as that obtained based on the Reuters-21578 collection. However, given that the TREC-AP based adaptive IR represents much more difficult learning and classification tasks than that based on the Reuters-21578 collection, it is more appropriate to compare the results

Topic	Rel.Doc.	Accuracy	Recall	Precision	F-measure	F1	F3	Seconds
1	135	0.998	0.030	0.364	0.055	-2	9	11,578
2	130	0.997	0.146	0.128	0.136	-26	2	109,032
3	95	0.998	0.137	0.236	0.173	-45	10	59,943
4	28	0.999	0.036	0.250	0.063	-3	1	12,605
5	25	0.999	0.200	0.833	0.323	13	19	9,027
6	129	0.998	0.047	0.109	0.065	-80	-25	86,018
7	86	0.999	0.000	0.000	0.000	-10	-5	12,778
8	24	0.999	0.000	0.000	0.000	-16	-8	12,979
9	48	0.999	0.688	0.340	0.455	-17	68	11,990
10	112	0.998	0.732	0.589	0.653	132	271	113,195
11	174	0.997	0.420	0.386	0.402	-13	176	92,115
12	268	0.997	0.041	0.345	0.073	-9	23	44,375
13	64	0.999	0.547	0.729	0.625	79	127	10,736
14	48	0.999	0.000	0.000	0.000	-2	-1	8,903
15	41	0.999	0.024	0.083	0.038	-19	-7	23,104
16	61	0.999	0.049	0.200	0.079	-15	0	59,402
17	106	0.998	0.566	0.364	0.442	-30	135	192,390
18	36	0.999	0.000	0.000	0.000	-22	-11	21,041
19	144	0.998	0.167	0.375	0.231	-8	56	26,839
20	63	0.998	0.206	0.250	0.226	-39	13	136,857
21	2	0.999	0.000	0.000	0.000	-10	-5	18,110
22	524	0.993	0.201	0.325	0.248	-86	198	116,898
23	113	0.998	0.707	0.482	0.573	68	234	67,976
24	113	0.998	0.257	0.244	0.250	-93	26	72,939
25	23	0.999	0.000	0.000	0.000	-12	-6	63,297

Figure 5.12: Overall results of AIFS by TREC Topics (1-25)

produced by different systems based on the TREC-AP collection. The average time for filtering a topic of the AP-89 dataset ($84,678$ documents) is $34,772.8$ seconds (around 9 hours and 39 minutes). Therefore, on average our belief-based agent system spends about 0.41 second to classify if a document is relevant and not, and at the same time uses the relevance feedback information attached to a document to learn the hypothetical user's changing information needs. The average filtering time per

Topic	Rel.Doc.	Accuracy	Recall	Precision	F-measure	F1	F3	Seconds
26	23	0.999	0.087	0.222	0.125	-8	1	40,006
27	5	0.999	0.400	0.222	0.286	-8	1	11,830
28	25	0.999	0.200	0.135	0.161	-49	-12	47,362
29	3	0.999	0.333	1.000	0.500	3	4	10,073
30	0	0.999	1.000	0.000	0.000	-2	-1	9,302
31	0	1.000	1.000	1.000	1.000	0	0	9,278
32	4	0.999	0.000	0.000	0.000	-2	-1	8,079
33	10	0.999	0.000	1.000	0.000	0	0	8,281
34	1	0.999	0.000	0.000	0.000	-4	-2	14,645
35	1	0.999	0.000	0.000	0.000	-2	-1	8,889
36	2	0.999	0.000	0.000	0.000	-8	-4	8,659
37	7	0.999	0.714	0.833	0.769	13	19	8,793
38	201	0.998	0.059	0.522	0.107	14	37	19,552
39	0	0.999	1.000	0.000	0.000	-4	-2	8,576
40	118	0.998	0.051	0.250	0.085	-18	6	12,224
41	30	0.999	0.000	1.000	0.000	0	0	8,509
42	60	0.999	0.017	0.500	0.032	1	3	10,820
43	49	0.999	0.020	0.048	0.029	-37	-16	33,760
44	67	0.999	0.058	0.364	0.102	-2	9	10,945
45	1	0.999	0.000	1.000	0.000	0	0	9,323
46	19	0.999	0.000	0.000	0.000	-4	-2	8,880
47	40	0.999	0.000	1.000	0.000	0	0	10,542
48	16	0.999	0.125	1.000	0.222	6	8	8,988
49	26	0.999	0.154	0.308	0.205	-6	7	8,312
50	1	0.999	0.000	0.000	0.000	-2	-1	8,886
Avg.		0.999	0.208	0.341	0.175	-7.7	27.1	34,772.8

Figure 5.13: Overall results of AIFS by TREC Topics (26-50)

TREC-AP document is 4.5 times longer than the average time spent on learning and classifying a Reuters-21578 document by the same agent system. The reason is that the AP collection is much larger and there are significantly more beliefs to be learnt by the agent system. Nevertheless, this efficiency figure indicates that the proposed logical framework is feasible for the development of adaptive information agents to process complex IR tasks since on average the agent system only needs less than half a second to learn and classify a document. The worst case in terms of computational efficiency in this experiment is TREC topic 17. It took about 2.2 seconds to process a document. Such a response time is still acceptable to on-line interactive information retrieval tasks because it may take up to a few seconds for a human user to respond to a message generated by a computer system.

Topic	Accuracy	Recall	Precision	F-measure	F1	F3	Seconds
1	0.998	0.311	0.467	0.373	30	120	8959
2	0.998	0.000	1.000	0.000	0	0	8939
3	0.999	0.253	0.480	0.331	20	70	8304
4	0.999	0.357	0.185	0.244	-58	-4	7401
5	0.999	0.240	0.462	0.316	4	17	7572
6	0.999	0.481	0.596	0.532	102	206	8369
7	0.999	0.244	0.313	0.275	-29	38	8463
8	0.999	0.000	0.000	0.000	-2	-1	7476
9	0.999	0.604	0.333	0.429	-29	58	7680
10	0.999	0.473	0.757	0.582	125	195	8376
11	0.997	0.753	0.354	0.482	-85	285	8583
12	0.997	0.041	0.524	0.076	13	34	11907
13	0.999	0.906	0.817	0.859	148	219	8004
14	0.999	0.146	0.700	0.241	15	25	7782
15	0.999	0.609	0.325	0.424	-29	48	7561
16	0.999	0.295	0.375	0.330	-6	42	7959
17	0.999	0.443	0.671	0.534	95	165	8416
18	0.990	0.861	0.035	0.068	-1603	-724	7423
19	0.998	0.159	0.767	0.264	55	85	9144
20	0.999	0.539	0.872	0.667	92	131	7972
21	0.999	1.000	0.200	0.333	-10	0	7081
22	0.994	0.042	0.667	0.079	44	77	13177
23	0.999	0.381	1.000	0.551	129	172	8518
24	0.999	0.027	0.600	0.051	5	10	8610
25	0.999	0.522	0.160	0.245	-90	-15	7499

Figure 5.14: Overall Results of VSpace by TREC Topics (1-25)

The performance figures of our base-line agent system (VSpace) are depicted in

Topic	Accuracy	Recall	Precision	F-measure	F1	F3	Seconds
26	0.999	0.043	0.200	0.071	-5	0	7513
27	0.999	0.200	1.000	0.333	3	4	7262
28	0.999	0.000	1.000	0.000	0	0	7530
29	0.999	0.000	1.000	0.000	0	0	7090
30	0.999	1.000	0.000	0.000	-18	-9	7042
31	0.999	1.000	0.000	0.000	-6	-3	7290
32	0.999	0.000	1.000	0.000	0	0	7389
33	0.999	0.100	1.000	0.182	3	4	7314
34	0.999	0.000	1.000	0.000	0	0	7112
35	1.000	1.000	1.000	1.000	3	4	7092
36	0.999	0.000	0.000	0.000	-2	-1	7263
37	0.999	0.429	0.429	0.429	1	8	7332
38	0.998	0.045	0.450	0.081	5	25	9018
39	0.999	1.000	0.000	0.000	-2	-1	7272
40	0.998	0.644	0.325	0.432	-88	146	8643
41	0.999	0.033	1.000	0.065	3	4	7575
42	0.999	0.033	0.071	0.045	-46	-18	7925
43	0.999	0.100	0.389	0.209	-1	17	7802
44	0.999	0.299	0.513	0.377	22	61	8052
45	0.999	0.000	1.000	0.000	0	0	7066
46	0.999	0.579	0.523	0.550	13	34	7563
47	0.999	0.000	0.000	0.000	-12	-6	7558
48	0.999	0.000	0.000	0.000	-2	-1	7402
49	0.999	0.577	0.319	0.411	-19	28	7556
50	0.999	0.000	1.000	0.000	0	0	7051
Avg.	0.999	0.335	0.518	0.249	-24.2	31.0	7977.7

Figure 5.15: Overall Results of VSpace by TREC Topics (26-50)

Figure 5.14 and Figure 5.15 respectively. Figure 5.14 shows the base line result for TREC topics 1-25, and Figure 5.15 shows the base line result for TREC topics 26-50. The average figures are shown in the last row of Figure 5.15. In terms of computational efficiency, the vector space based agent model is a sure winner. It took $7,977.7$ seconds (2 hours and 12 minutes) to process a TREC topic on average, and 0.09 second to process a document of the TREC-AP collection. The base line system is at least 4.5

Topic	Accuracy	Recall	Precision	F-measure	F1	F3	Seconds
1	0.000	-0.281	-0.103	-0.318	-32	-111	2,619
2	-0.001	0.146	-0.872	0.136	-26	2	100,093
3	-0.001	-0.116	-0.244	-0.158	-65	-60	51,639
4	0.000	-0.321	0.065	-0.181	55	5	5,204
5	0.000	-0.040	0.371	0.007	9	2	1,455
6	-0.001	-0.435	-0.487	-0.467	-182	-231	77,649
7	0.000	-0.244	-0.313	-0.275	19	-43	4,315
8	0.000	0.000	0.000	0.000	-14	-7	5,503
9	0.000	0.084	0.007	0.026	12	10	4,310
10	-0.001	0.259	-0.168	0.071	7	76	104,819
11	0.000	-0.333	0.032	-0.080	72	-109	83,532
12	0.000	0.000	-0.179	-0.003	-22	-11	32,468
13	0.000	-0.359	-0.088	-0.234	-69	-92	2,732
14	0.000	-0.146	-0.700	-0.241	-17	-26	1,121
15	0.000	-0.585	-0.242	-0.386	10	-55	15,543
16	0.000	-0.246	-0.175	-0.251	-9	-42	51,443
17	-0.001	0.123	-0.307	-0.092	-125	-30	183,974
18	0.009	-0.861	-0.035	-0.068	1581	1550	13,618
19	0.000	0.008	-0.392	-0.033	-63	-29	17,695
20	-0.001	-0.333	-0.622	-0.441	-131	-118	128,885
21	0.000	-1.000	-0.200	-0.333	0	-5	11,029
22	-0.001	0.159	-0.342	0.169	-130	121	103,721
23	-0.001	0.326	-0.518	0.022	-61	62	59,458
24	-0.001	0.230	-0.356	0.199	-98	16	64,329
25	0.000	-0.522	-0.160	-0.245	78	9	55,798

Figure 5.16: Comparison (AIFS vs. VSpace) by TREC Topics (1-25)

times faster than the belief-based agent system. A direct comparison between the result of AIFS and that of VSpace is depicted in Figure 5.16 and Figure 5.17. In general, a positive performance figure such as the F1 score indicates that the AIFS system out-performs the base-line system, whereas a negative figure implies that AIFS is not performing as well as the base-line system. However, positive figures in the last columns (execution time in seconds) of Figure 5.16 and Figure 5.17 mean that the

Topic	Accuracy	Recall	Precision	F-measure	F1	F3	Seconds
26	0.000	0.044	0.022	0.054	-3	1	32,493
27	0.000	0.200	-0.778	-0.047	-11	-3	4,568
28	0.000	0.200	-0.865	0.161	-49	-12	39,832
29	0.000	0.333	0.000	0.500	3	4	2,983
30	0.000	0.000	0.000	0.000	16	8	2,260
31	0.001	0.000	1.000	1.000	6	3	1,988
32	0.000	0.000	-1.000	0.000	-2	-1	690
33	0.000	-0.100	0.000	-0.182	-3	-4	967
34	0.000	0.000	-1.000	0.000	-4	-2	7,533
35	-0.001	-1.000	-1.000	-1.000	-5	-5	1,797
36	0.000	0.000	0.000	0.000	-6	-3	1,396
37	0.000	0.285	0.404	0.340	12	11	1,461
38	0.000	0.014	0.072	0.026	9	12	10,534
39	0.000	0.000	0.000	0.000	-2	-1	1,304
40	0.000	-0.593	-0.075	-0.347	70	-140	3,581
41	0.000	-0.033	0.000	-0.065	-3	-4	934
42	0.000	-0.016	0.429	-0.013	47	21	2,895
43	0.000	-0.080	-0.341	-0.180	-36	-33	25,958
44	0.000	-0.241	-0.149	-0.275	-24	-52	2,893
45	0.000	0.000	0.000	0.000	0	0	2,257
46	0.000	-0.579	-0.523	-0.550	-17	-36	1,317
47	0.000	0.000	1.000	0.000	12	6	2,984
48	0.000	0.125	1.000	0.222	8	9	1,586
49	0.000	-0.423	-0.011	-0.206	13	-21	756
50	0.000	0.000	-1.000	0.000	-2	-1	1,835
Avg.	0.000	-0.127	-0.177	-0.075	16.6	-3.9	26,795.1

Figure 5.17: Comparison (AIFS vs. VSpace) by TREC Topics (26-50)

AIFS system takes longer time to conduct information filtering. The highlighted cells in Figure 5.16 and Figure 5.17 indicate that AIFS performed better than VSpace. In terms of the average F_1 measure, the base-line agent system is slightly better than AIFS (e.g., -0.075). However, the average F1 utility of AIFS is better than that of the base-line system (e.g., 16.6). The base-line system achieved a slightly better average F3 utility (e.g., -3.9). For a topic by topic comparison based on the F_1 measure, there are 23 topics where the belief-based agent system performs better than or as well as the base-line agent system. Based on the F1 utility scores, there are 21 topics where AIFS performs better than or at least as well as the base-line system. AIFS performs better than or as well as the base-line system over 20 TREC topics if it is assessed based on the F3 utility scores. As a whole, we are unable to draw a conclusion that the belief-based information agent model out-performs the vector space based information agent model. The AIFS model produced comparable result as that achieved by the VSpace model. The base line agent system is definitely more efficient than the belief-based agent system in this experiment. The slightly higher average F1 utility score of AIFS than that of VSpace demonstrates the potential of a belief-based IR model. Since this is the first implementation of a belief revision based information agent model, there is still room for improvement in terms of computational efficiency and IR effectiveness in the future. Given the fact that it is quite difficult to develop logic-based IR models [CRSR95], the AIFS system is the first fully operational logic-based IR system which can process IR tasks based on large document collections.

Apart from examining the base line system, more insights about the performance of AIFS can be obtained by comparing its classification effectiveness with that of the participating systems in TREC-7 [Hul98]. Since the SIGMA information agent system (Chapter 2, Section 2.4) is the only adaptive information agent system participated in the adaptive filtering task of TREC-7, it makes sense to compare SIGMA's performance with that of AIFS. Figure 5.18 and Figure 5.19 depict the F1

Topic	F1	F3	SIGMA F1	SIGMA F3	Diff. F1	Diff. F3
1	-2	9	-535	-73	533	82
2	-26	2	-645	-82	619	84
3	-45	10	-484	23	439	-13
4	-3	1	-477	-269	474	270
5	13	19	-827	-317	840	336
6	-80	-25	-1708	-611	1628	586
7	-10	-5	-1096	-369	1086	364
8	-16	-8	-529	-260	513	252
9	-17	68	-407	-231	390	299
10	132	271	-897	97	1029	174
11	-13	176	-1187	3	1174	173
12	-9	23	-826	256	817	-233
13	79	127	-123	55	202	72
14	-2	-1	-1667	-722	1665	721
15	-19	-7	-1475	-440	1456	433
16	-15	0	-581	-310	566	310
17	-30	135	-151	194	121	-59
18	-22	-11	-652	-476	630	465
19	-8	56	-199	29	191	27
20	-39	13	-146	193	107	-180
21	-10	-5	-93	-29	83	24
22	-86	198	-844	1055	758	-857
23	68	234	266	478	-198	-244
24	-93	26	-904	-288	811	314
25	-12	-6	-163	-51	151	45

Figure 5.18: Comparison (AIFS vs. SIGMA) by TREC Topics (1-25)

and F3 utility scores as obtained by AIFS and SIGMA. Figure 5.18 shows the comparison over TREC topics 1-25, and Figure 5.19 shows the comparison over TREC topics 26-50. The last row in Figure 5.19 depicts the average figures from AIFS and SIGMA, and their differences. Since only the F1 and F3 utility scores are available from TREC-7 (`http://trec.nist.gov/pubs/trec7/t7_proceedings.html`), comparison between AIFS and SIGMA is done based on these two measures only. The

Topic	F1	F3	SIGMA F1	SIGMA F3	Diff. F1	Diff. F3
26	-8	1	-12	24	4	-23
27	-8	1	-55	-15	47	16
28	-49	-12	-222	-102	173	90
29	3	4	-138	-63	141	67
30	-2	-1	-101	-48	99	47
31	0	0	-116	-58	116	58
32	-2	-1	-79	-37	77	36
33	0	0	-384	-154	384	154
34	-4	-2	-19	-7	15	5
35	-2	-1	-130	-65	128	64
36	-8	-4	-133	-55	125	51
37	13	19	-114	-52	127	71
38	14	37	-1864	-779	1878	816
39	-4	-2	-10	5	6	-7
40	-18	6	-1599	-366	1581	372
41	0	0	-527	-208	527	208
42	1	3	-294	-164	295	167
43	-37	-16	-552	-275	515	259
44	-2	9	-746	-159	744	168
45	0	0	-754	-318	754	318
46	-4	-2	-479	-232	475	230
47	0	0	-868	-441	868	441
48	6	8	-161	-73	167	81
49	-6	7	-208	-12	202	19
50	-2	-1	-14	-7	12	6
Avg.	-7.7	27.1	-518.6	-116.1	510.9	143.2

Figure 5.19: Comparison (AIFS vs. SIGMA) by TREC Topics (26-50)

first column in Figure 5.18 and Figure 5.19 depicts the TREC topic numbers. The second and the third columns depict the F1 and F3 scores of AIFS, and the forth and the fifth columns show the F1 and F3 scores of SIGMA. A positive figure in the last two columns means that AIFS outperforms SIGMA in a particular TREC topic. All the positive figures in these two columns are highlighted. By comparing the F1 scores (the sixth column), AIFS outperforms SIGMA in all topics except TREC topic 23. By comparing the F3 scores (the last column), AIFS outperforms SIGMA in 42 topics. Apparently, our belief-based adaptive information agent system AIFS achieved much better IR performance than that of the SIGMA information agent system. However, we cannot conclude that AIFS is definitely more effective than SIGMA because AIFS was applied to filter the AP-89 subset only. Moreover, both accepted and rejected documents were used by AIFS to learn a user's changing information needs.

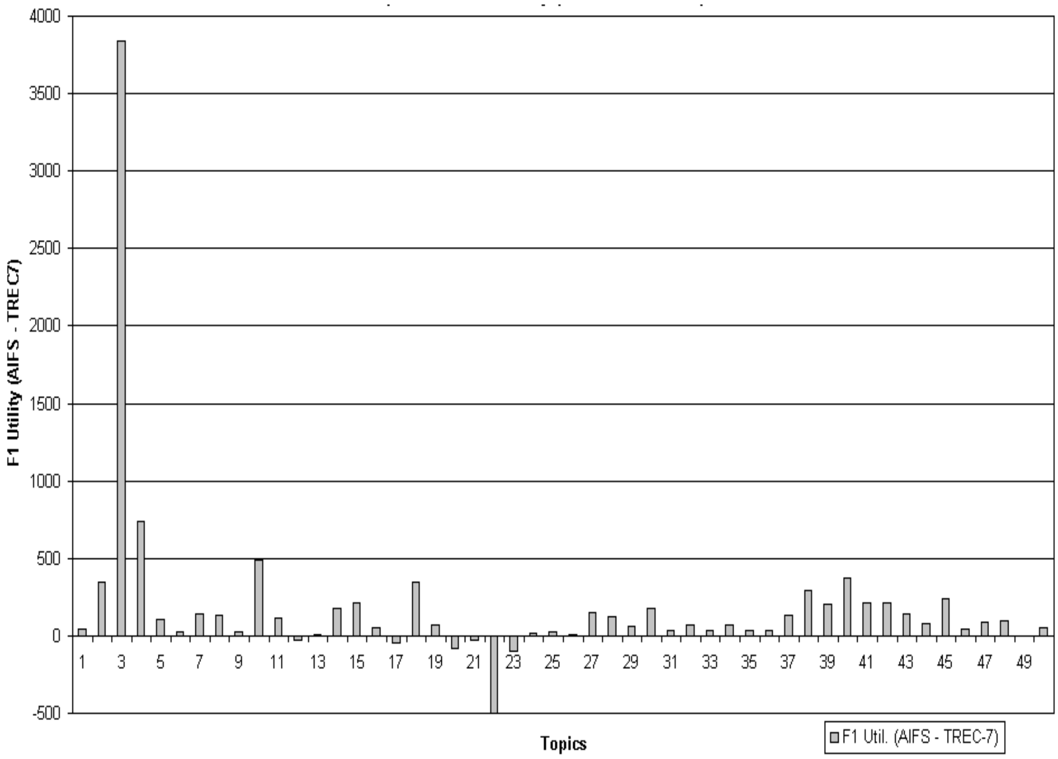

Figure 5.20: Comparison of F1 utility (AIFS vs. Average TREC-7)

Figure 5.20 plots AIFS's F1 utilities against the average F1 utilities of all the participating systems in the adaptive filtering task of TREC-7 over the 50 TREC topics. Each bar in Figure 5.20 represents the difference between the F1 score obtained from AIFS and the average F1 score obtained from the TREC-7 participants for the

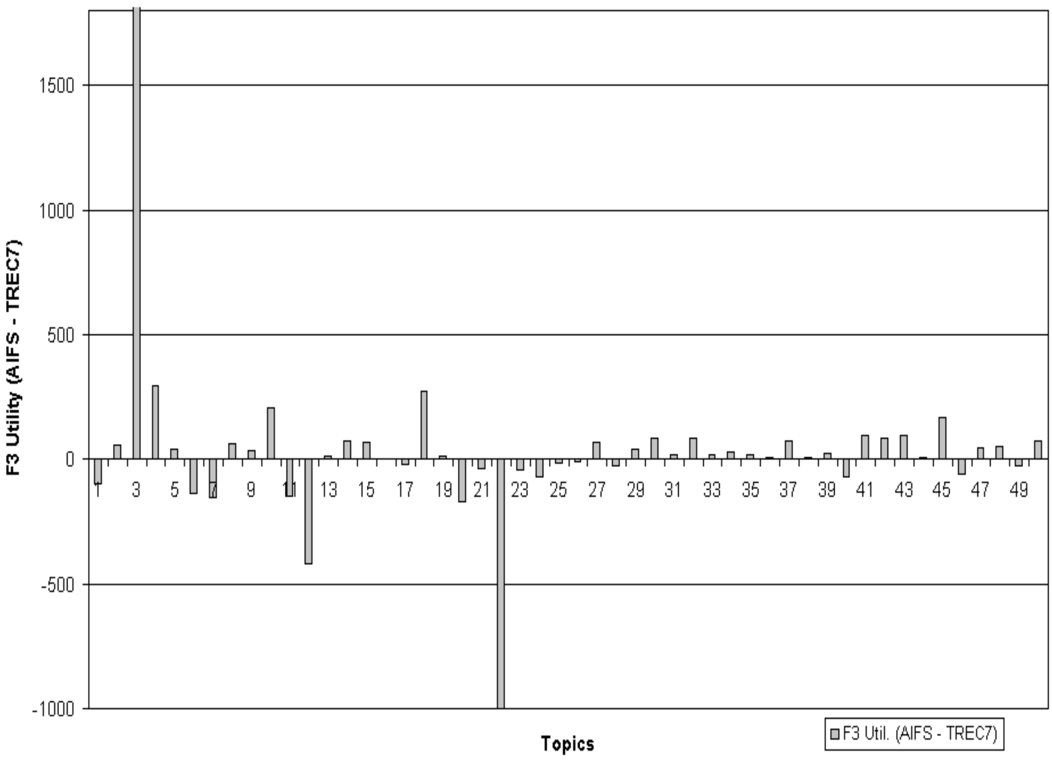

Figure 5.21: Comparison of F3 utility (AIFS vs. Average TREC-7)

corresponding topic. The average F1 scores from the TREC-7 participants in the adaptive information filtering task are treated as base line figures for our comparison. If there is a bar above the x-axis, it means that the performance of AIFS is better than the average performance of the TREC-7 participants for that particular topic. A bar below the x-axis indicates that the performance of AIFS is not as good as the average performance of the TREC-7 participants for the particular topic. As can be seen, AIFS's performance is better than the average performance of the TREC-7 participants in the adaptive information filtering task in many occasions (in 44 TREC topics). In addition, figure 5.21 plots AIFS's F3 utilities against the average F3 utilities among the participating systems of TREC-7 over the 50 TREC topics. A bar above the x-axis means that AIFS's performance is better than the average performance of the TREC-7 participants. AIFS's F3 utilities are above the average of the TREC-7 participants for the adaptive filtering task in 32 TREC topics. Although our experiment is slightly different from the adaptive filtering task of TREC-7 (e.g., only the AP-89 dataset is processed by AIFS), these topic-by-topic comparisons provide the basis for an initial assessment of AIFS's general performance. In general, the belief revision based information agent model is promising as demonstrated by the results produced by our very first prototype system. The precise figures of the topic-by-topic comparison between AIFS and the participating systems of the TREC-7 adaptive information filtering task are given in Appendix F.

Claritech is among the best adaptive filtering systems in TREC-7 [Hul98]. The comparison between the F1 and F3 utilities of AIFS and that of Claritech shows that AIFS is not as good as Claritech. However, AIFS's performance is close to that of Claritech based on a topic-by-topic comparison. There are 24 topics where AIFS's F1 scores are higher than that of Claritech, and there are 20 topics where AIFS's F3 scores are higher than that of Claritech. Nevertheless, there are 26 topics where Claritech's F1 scores are better than that of AIFS, and there are 30 topics where Claritech's F3 scores are better than that of AIFS. The precise figures of the topic-by-topic comparison between AIFS and Claritech are tabulated in Appendix G.

Chapter 6

Conclusions and Future Directions

6.1 Conclusions

The AGM belief revision framework provides a rigorous theoretical foundation to build the next generation of adaptive information agents. The logical language provides the expressive power to represent complex retrieval contexts. The AGM belief functions formally characterise the agents' learning activities, and ensure that the abstraction of retrieval contexts is revised in a *minimal* and *consistent* fashion. Therefore, information matching in these agents adheres to the *logical uncertainty principle*. In addition, expectation inference provides a sound and robust framework to develop the agents' classification mechanisms which enhance *proactive* IR. The close connection between belief revision and expectation inference allows a seamless integration of the learning and the classification functions in adaptive information agents. The belief-based information agent system is more effective than the vector space based information agent system for the adaptive filtering tasks conducted based on the Reuters-21578 collection. Moreover, the belief-based information agent system is also efficient in dealing with large IR applications. It takes less than half second to filter a document for the AP-89 collection which contains over eighty thousand documents. The belief revision based IR model is among a few implemented logic-based IR models, and is the first logic-based IR model with both IR effectiveness and computational efficiency successfully evaluated based on large IR bench-marking collections such as the Reuters-21578 collection and the AP-89 collection. This is the first research providing concrete evidence that logic-based IR model is not only effective but also efficient for large and realistic IR applications. In addition, the work reported in this thesis also demonstrates the first large scale implementation and validation of the AGM belief revision framework in an empirical setting.

6.1.1 Entrenchment Induction

Mutual Information MI and a variant of the Expected Cross Entropy EH are not effective because of the mis-match between how an entrenchment degree is induced and how the entrenchment degree is interpreted by the belief-based classification model. Both the Keyword Classifier KC and Modified Keyword Classifier MKC which are based on the statistical method of Kullback Divergence are effective. MKC is the most effective entrenchment induction method measured in terms of the F_1 score, F1 utility and F3 utility for the IR tasks performed based on the Reuters-21578 collection.

6.1.2 Adaptive Learning

The belief revision based adaptive information agents can learn changing retrieval contexts. Their learning performance is reflected by the improvement of the F_1 scores over time in several IR tasks based on the TREC-AP collection or the Reuters-21578 collection. However, if there are only a few relevant documents (i.e., positive training examples) in an IR topic, the agents' learning and classification effectiveness may fluctuate because of insufficient information to learn an accurate representation of a retrieval context.

6.1.3 Transmutation Methods

Several transmutation methods which implement the AGM belief functions have been evaluated based on two adaptive filtering tasks of the TREC-AP collection. The standard transmutation method which exactly implements the AGM belief revision functions is not as effective as the Maxi-adjustment method or the Rapid Maxi-adjustment method. The Maxi-adjustment transmutation method is not as effective as its *anytime* counterpart and is the least efficient transmutation method. The Anytime Maxi-adjustment method and the Anytime Rapid Maxi-adjustment (RAM) method produced comparable performance if there are not many positive beliefs about a retrieval context to be learnt from. Nevertheless, the RAM method is more efficient than the Anytime Maxi-adjustment method when there are many beliefs to be revised into an agent's knowledge base. Through our experiments, it is confirmed that the *anytime* approximations of the AGM belief revision operations are both effective and efficient for large real-life applications.

6.1.4 General Discussion

With reference to the set of desirable features of intelligent IR systems discussed in Section 1.3 of Chapter 1, the belief revision based adaptive information agent model is promising in many aspects. The belief revision based information agents can *autonomously classify documents* from a large stream of incoming documents with minimum human intervention. These agents are *adaptive* since they can constantly revise their knowledge bases in accordance with the changing retrieval contexts and predict the relevance of documents with respect to the revised retrieval contexts. The belief-based agent system is *scalable* for processing large and complex IR applications. The first prototype of belief revision based information agent system only requires less than half second to process a document for the AP-89 collection. The belief revision based information agents are also *proactive* because they can make use of the relationships among information items to infer the possibly interesting documents which are not explicitly requested by the users. In addition, these agents are *explanatory* as they can justify their classification decisions based on the relationships among information items. Finally, for the requirement of balanced precision and recall IR behaviour, the initial experiments show that the average precision of the agent system is slightly higher than its average recall for both the TREC-AP and the Reuters-21578 runs.

However, this may not be such a bad feature given the fact that information seekers are often overwhelmed by too much rather than too little information.

6.2 Future Directions

The current research work in belief revision based adaptive information agents represents an initial study towards applying theoretical AI models to practical IR applications. During the course of this research, it was found that there were other issues and research questions related to the current study. However, because of the limited time, these issues are left to be tackled by future research.

6.2.1 Discovering Contextual Knowledge

During the course of developing the belief revision based IR model, it was found that automated means of learning IR contexts is critical for the success of the information agents. Essentially, the entrenchment induction procedure concerns about inducing users' information preferences. The more challenging induction task is the discovery of the corresponding IR contexts where the users' information needs arise. This contextual information is essential for the agents to infer the users' implicit information needs. For instance, given the contextual knowledge that science students who are interested in items described by the term "Java" are probably studying computer sciences (i.e., $java \wedge science \rightarrow computer$), and students with major in computer sciences normally learn programming (i.e., $computer \rightarrow programming$), an information agent will recommend documents about "Computer Programming" to a user who is a science student issuing a query of "Java". It is possible for the agents to discover such contextual knowledge from a student enrolment database. However, mining such contextual knowledge is not a trivial task. Indeed, there are still many outstanding research questions to be tackled in the field of text mining. The text mining framework proposed in this thesis is based on the well-known association rule mining techniques. The notions of rule support and rule confidence in association rule mining are probabilistic measures, whereas epistemic entrenchment orderings do not satisfy the basic probability axioms in general. This gives rise to a fundamental research challenge of handling the mis-match between these two paradigms. In addition, a novel method which is based on a rigid measure of term property is proposed to induce the information preclusion relationships. For instance, only the absolute positive terms and negative terms are selected to construct the information preclusion rules. From the initial experiments, it was found that not many such rigid rules exist in a collection. On the other hand, relaxing the rigid selection criterion to allow more terms to go into the rule generation processes jeopardises the recall of the information agents because some of the relevant items are mistakenly identified as non-relevant. A more effective text mining method for the discovery of information preclusion relationships is required.

6.2.2 Belief Revision for Abductive Classification

A formal analysis of the relationship between the AGM belief revision and minimal abduction has been conducted [PNF95]. In the context of adaptive information agents, entrenchment-based abduction can be applied to enhance the agents' information matching (classification) functions. The proposed belief revision based learning model revises an agent's knowledge base only if explicit user's relevance feedback is received. Then, subsequent classification is conducted by matching the revised knowledge about the current retrieval context with incoming document characterisations. However, it is possible for the agents to *abduct* document relevance given little or no relevance feedback from the users. Such an information matching capability is akin to the one found in the recently proposed probabilistic relevance model [LC01]. If a document representation is not logically entailed by an agent's knowledge base, a *shadow* belief revision operation can be invoked to *minimally* revise the knowledge base such that the revised beliefs entail the document characterisation. If the minimal changes satisfy certain criteria in terms of epistemic entrenchment, the corresponding document is deemed relevant. Unlike the revision processes triggered by users' relevance feedback, the abducted sentences are not physically added to the agents' knowledge bases. In general, such a classification method can improve the recall of the information agent system since some partially relevant documents may be considered relevant after the abduction process. An entrenchment-based abductive framework can be seen as a direct implementation of the *logical uncertainty principle* for IR. Some technical issues need to be resolved before such an abductive reasoning framework can be applied to develop operational information agent systems. The fundamental issue is the development of the entrenchment-based document selection criterion.

6.2.3 Further Optimisation of the Information Agent Model

Symbolic IR models are computationally expensive and so are the adaptive information agents built on top of such models. Optimisation techniques (e.g., feature selection based on TFIDF, removing less entrenched beliefs, reducing the frequency of belief revision, using a subset of the propositional language, etc.) were applied to the belief revision based agent model so that an efficient prototype system could be built. As indicated from our preliminary experiments, the belief revision based agent model is less efficient than its vector space based counterpart. It is going to be a long battle to develop a symbolic information agent model that is as efficient as a purely quantitative model. One possible approach to improve the efficiency of the existing information agent model is to apply Latent Semantic Indexing (LSI) [DDF+90] to reduce the dimensionality of the document space before applying the belief revision logic for learning and classification in information agents. However, since LSI is also computationally expensive, it remains a problem for optimising the computational efficiency of on-line adaptive information agents. An alternative is to explore a phrase-based rather than word-based document representation scheme. The research hypothesis is whether a phrase-based document representation can reduce the number of tokens required for representing documents, and hence the number of belief revision operations taken to learn a retrieval context. Given a smaller knowledge base in an agent, the time spent on inferring document relevance (i.e., classification) may also be reduced. Finally, with the advances in theorem proving techniques, it is possible to optimise the *degree*() function which is the work horse of the belief revision algorithm. Consequently, both the learning and the classification processes in adaptive information agents become more efficient.

6.2.4 Possibilistic Information Agents

It has been proved that the numerical counterparts of the *epistemic entrenchment orderings* are the necessity measures [DP91]. For instance, $\alpha \leqslant \beta$ is equivalent to $N(\alpha) \leq N(\beta)$ for any $\alpha, \beta \in \mathcal{L}$; $\leqslant$ represents the epistemic entrenchment ordering between α and β. The ordering induced by the necessity measure such as $N(\alpha)$ is represented by $\leq$. For any possibilistic formula such as (α, m), the greatest lower bound certainty m derived from the necessity measure equals the degree of acceptance of the corresponding formula $degree(\mathbf{B}, \alpha)$. This correspondence not only establishes the close connection between the AGM belief revision and *possibilistic logic*, but also provides an alternative for modelling the learning and classification functions of adaptive information agents. In a possibilistic knowledge base, inconsistencies among beliefs are allowed. An inconsistency tolerant possibilistic deduction framework is used to draw conclusions based on the most reliable (certain) subset of information in a knowledge base. A recent psychological study has shown that the nonmonotonic postulates characterising possibilistic logic are compatible with the characteristics of human reasoning [NBR00]. It is also reported that possibilistic rather than probabilistic reasoning is closer to the kind of *approximate reasoning* exercised by human experts [RN98]. Therefore, it is intuitively attractive to apply possibilistic logic to model adaptive IR situations where inconsistent retrieval contexts may arise. If the information needs pertaining to different topics are captured by a single knowledge base, the chance of developing an inconsistent knowledge base increases. A possibilistic framework for IR has been explored [LtHB01c, LtHB01b]. From a theoretical stand point, the learning processes in information agents may be sped up because the computations spent on maintaining consistent knowledge bases are saved. However, the extra computational costs involved in finding maximal consistent belief sets for deducing document relevance (i.e., classification) may outweigh the agents' efficiency gains obtained during learning. This is a severe problem if the information agents are deployed for on-line interactive IR where the agents' on-line performance is mainly influenced by the classification processes. Empirical studies are needed to examine the advantages and drawbacks of the possibilistic information agent model.

6.2.5 Knowledge Fusion and Collaborative Filtering

When a large number of information agents are deployed for IR activities, there will be a need to share the domain knowledge among the agents or sharing the agents among an information seeking community. The former activities are related to the research topic of knowledge fusion, whereas the latter may be empowered by collaborative filtering. For knowledge fusion, the main issue is how to combine several epistemic entrenchment orderings into a coherent one while retaining as much information from individual entrenchment orderings as possible. The theories about knowledge fusion among multiple knowledge bases need to be explored. A related issue is how to share the knowledge acquired by the adaptive information agents among the users within an information seeking community. With these capabilities, the advantages of both content-based filtering (the focus of this thesis) and collaborative filtering are unified under a single information agent architecture. To implement collaborative filtering, an exploration of the techniques for generalising contextual knowledge acquired by a group of information agents or specialising the contextual knowledge acquired by a generic information agent is needed. Furthermore, a more sophisticated agent library structure should be sought to facilitate the re-use of the contextual knowledge acquired by the adaptive information agents.

6.2.6 Web-based Adaptive Information Agents

The Internet and the Web present very challenging IR problems. As reported from previous studies, queries passed to the Internet search engines are often short and incomplete. This indicates that information seekers have difficulties in expressing their implicit information needs by artificial query languages. Even for a domain specific query, a search engine may return thousands of hits. The adaptive information agents proposed in this thesis are quite applicable to solve the problem of information overload on the Web. The client-server agent architecture depicted in Chapter 4 represents a feasible solution to the Web search problem. With the help of the server side adaptive information agents, information seekers residing on the client sides are pushed with relevant Web information (e.g., Web pages or net news). This kind of service is particularly useful for satisfying users' long-term recurring information needs. The development of a proxy server housing the adaptive information agents, the wrappers components interfaced with external Internet search engines or other information agents, and the intelligent user interface agents which can constantly monitor users' on-line actions will certainly complement the current prototype agent system and make the system fully operational on the Web. To improve the *external validity* of the current research work, huge Web collections (e.g., from the TREC archive) can be used to examine the scalability power of the Web-based adaptive information agents. Moreover, usability studies involving real information seekers can be performed to evaluate the Web-based agent system as a whole.

6.2.7 Adaptive Information Agents for E-commerce

The belief revision based adaptive information agents can be applied to other related applications to improve the external validity of the underlying agent model. For instance, the first two essential stages in agent-mediated electronic commerce are needs identification and product brokering [GMM98]. During these stages, profiles of consumers' requirements for products are created by intelligent agents. Based on a consumer's feedback about her current product preferences, a profiling agent can constantly revise the content of the corresponding consumer profile and reason about the consumer's actual requirements with respect to her latest product preferences. This scenario is quite similar to the adaptive IR processes. In fact, profiling consumers' needs and recommending products can be seen as a special case of the general adaptive IR processes. In the context of electronic commerce, information objects are about consumer products. An initial investigation into the framework of applying the belief revision agent model to adaptive consumer profiling and product recommendation has been performed [LtHB00]. In addition, a novel belief revision based negotiation model has also been proposed [Lau02a]. However, more work is required to develop and evaluate the belief revision based adaptive profiling or negotiation system for electronic commerce.

Appendix A

Publications Related to Adaptive Information Agents

Journal Publications

1. R.Y.K. Lau. The State of the Art in Adaptive Information Agents. *International Journal on Artificial Intelligence Tools*, 11(1):19–61, March 2002.

2. R.Y.K. Lau, A.H.M. ter Hofstede, and P.D. Bruza. Belief Revision for Adaptive Information Filtering Agents. *International Journal of Cooperative Information Systems*, 10(1-2):57–79, March–June 2001.

3. R.Y.K. Lau., A.H.M. ter Hofstede, and P.D. Bruza. Maxi-Adjustment and Possibilistic Deduction for Adaptive Information Agents. *Journal of Applied Non-Classical Logics*, 11(1-2):169–201, 2001.

Refereed Conferences

4. R.Y.K. Lau, A.H.M. ter Hofstede, and P.D. Bruza. A Study of Belief Revision in the Context of Adaptive Information Filtering. In Lucas C. Hui and Dik L. Lee, editors, *Proceedings of the Fifth International Computer Science Conference (ICSC'99)*, volume 1749 of *Lecture Notes in Computer Science*, pages 1–10, Hong Kong, China, 1999. Springer.

5. R.Y.K. Lau, A.H.M. ter Hofstede, P.D. Bruza, and K.F. Wong. Belief Revision and Possibilistic Logic for Adaptive Information Filtering Agents. In Babak Hamidzadeh, editor, *Proceedings of the Twelfth IEEE International Conference on Tools with Artificial Intelligence*, pages 19–26, British Columbia, Canada, November 13–15 2000. IEEE Press.

6. R.Y.K. Lau, A.H.M. ter Hofstede, and P.D. Bruza. Nonmonotonic Reasoning for Adaptive Information Filtering. In Michael Oudshoorn, editor, *Proceedings of the Twenty-Fourth Australasian Computer Science Conference (ACSC2001)*, pages 109–116, Gold Coast, Australia, January 29–February 2 2001. IEEE Press.

7. R.Y.K. Lau. Belief Revision and Data Mining for Adaptive Information Agents. In H. R. Arabnia, editor, *Proceedings of the 2001 International Conference on Artificial Intelligence (IC-AI'2001)*, pages 504–509, Las Vegas, Nevada, June 25–28 2001. CSREA Press.

Refereed Workshops

8. R.Y.K. Lau, A.H.M. ter Hofstede, and P.D. Bruza. Applying Maxi-adjustment to Adaptive Information Filtering Agents. In Chitta Baral and Miroslaw Truszczyński, editors, *Proceedings of the 8th International Workshop on Non-Monotonic Reasoning NMR'2000*, Breckenridge, Colorado, April 2000. Available from `http://xxx.lanl.gov/html/cs.AI/0003073`.

9. R.Y.K. Lau, A.H.M. ter Hofstede, and P.D. Bruza. A Logic-Based Approach for Adaptive Information Filtering Agents. In Fumio Mizoguchi, Leon Sterling, and Seng Wai Loke, editors, *Proceedings of the 1st Pacific Rim International Workshop on Intelligent Information Agents (PRIIA 2000)*, volume 2112 of *Lecture Notes in Artificial Intelligence*, pages 269–279, Melbourne, Australia, August 28 2000. Springer.

Appendix B

An example of a TREC topic

```
<top>
<head> Tipster Topic Description
<num> Number: 001
<dom> Domain: International Economics
<title> Topic: Antitrust Cases Pending
<desc> Description:
Document discusses a pending antitrust case.
<narr> Narrative:
To be relevant, a document will discuss a pending antitrust case and
will identify the alleged violation as well as the government entity
investigating the case.
<con> Concept(s):
1.  antitrust suit, antitrust objections, antitrust investigation,
antitrust dispute
<fac> Factor(s):
<def> Definition(s):
Antitrust - Laws to protect trade and commerce from unlawful
restraints and monopolies or unfair business practices.
</top>
```

Appendix C

An Example of a Reuters-21578 document


```
<REUTERS TOPICS="YES" LEWISSPLIT="TRAIN" CGISPLIT="TRAINING-SET"
 OLDID="5544" NEWID="1">
<DATE>26-FEB-1987 15:01:01.79</DATE>
<TOPICS><D>cocoa</D></TOPICS>
<PLACES><D>el-salvador</D><D>usa</D><D>uruguay</D></PLACES>
<PEOPLE></PEOPLE>
<ORGS></ORGS>
<EXCHANGES></EXCHANGES>
<COMPANIES></COMPANIES>
<TEXT>&#2;
<TITLE>BAHIA COCOA REVIEW</TITLE>
<DATELINE>    SALVADOR, Feb 26 - </DATELINE>
<BODY>Showers continued throughout the week in
the Bahia cocoa zone, alleviating the drought since early
January and improving prospects for the coming temporao,
although normal humidity levels have not been restored,
Comissaria Smith said in its weekly review.
</BODY>
</TEXT>
</REUTERS>
```

Appendix D

The Standard AGM Adjustment Algorithm

```
FUNCTION AnytimeAGM(OldB, α, Ndegree, TimeLimit)
    Odegree := Degree(OldB, α)
    REMARKS: MaxDegree = 1 in our implementation
    IF Degree(OldB, ¬α) = MaxDegree
        RETURN OldB
    ENDIF
    IF Ndegree ≥ Odegree
        NewB := Revision(OldB, α, Odegree, Ndegree, TimeLimit)
    ELSE
        NewB := AGMContraction(OldB, α, Odegree, Ndegree, TimeLimit)
```

```
ENDIF

RETURN NewB

END FUNCTION

FUNCTION AGMContraction(OldB, α, Odegree, Ndegree, TimeLimit)

REMARKS: MinDegree = 0 in our implementation

IF Ndegree = Odegree

   RETURN OldB

ENDIF

HighB := Cut(OldB, Rank(MaxDegree), Rank(Odegree) - 1)

ProblemB := Cut(OldB, Rank(Odegree), Rank(Ndegree) - 1)

LowB := Cut(OldB, Rank(Ndegree), Rank(MinDegree))

NewB := HighB

FOR x := 1 TO NoElements(ProblemB)

         IF ElapsedTime() > TimeLimit AND TimeLimit > 0

            EXIT

         ENDIF

         IF ProblemB[x].belief =  α

            SKIP NEXT

         ENDIF

         IF  Degree(OldB, ProblemB[x] ∨ α) > Odegree
```

```
            NewB := NewB + ProblemB[x]

          ELSE

             IF Ndegree > MinDegree

                ProblemB[x].degree := Ndegree

                NewB := NewB + ProblemB[x]

             ENDIF

          ENDIF
```

NEXT

```
IF Ndegree > MinDegree
```

`        NewB := NewB +` $(\alpha, \text{Ndegree})$

```
        NewB := NewB + LowB

ENDIF
```

RETURN NewB

END FUNCTION

Appendix E

The Anytime Maxi-Adjustment Algorithm

The Anytime Maxi-adjustment algorithm illustrated in this section is developed based on Williams' idea presented in [Wil97]. However, it is not a re-production of Williams' algorithm. In particular, this algorithm is based on the concept of identifying and extracting the $ProblemB$ segment of finite partial entrenchment ranking to produce the closest approximation of a belief revision operation rather than based on the $MoveUp()$ and $MoveDown()$ functions discussed in [Wil97]. The algorithm illustrated in this section is optimised to avoid the enunmeration of the minimal subsets in a particular entrenchment rank if it is not really necessary.

FUNCTION AnytimeMaxi(OldB, α, Ndegree, TimeLimit)

Odegree := $Degree(OldB, \alpha)$

REMARKS: MaxDegree = 1 in our implementation

```
IF Degree(OldB, ¬α) = MaxDegree
    RETURN OldB
ENDIF
IF Ndegree ≥ Odegree
    NewB := Revision(OldB, α, Odegree, Ndegree, TimeLimit)
ELSE
    NewB := MaxiContraction(OldB, α, Odegree, Ndegree, TimeLimit)
ENDIF
RETURN NewB
END FUNCTION


FUNCTION MaxiContraction(OldB, α, Odegree, Ndegree, TimeLimit)
IF Ndegree = Odegree
    RETURN OldB
ENDIF
HighB := Cut(OldB, Rank(MaxDegree), Rank(Odegree) - 1)
ProblemB := Cut(OldB, Rank(Odegree), Rank(Ndegree) - 1) - {α, Odegree}
LowB := Cut(OldB, Rank(Ndegree), Rank(MinDegree))
NewB := HighB
prover := NEW TheoremProver()
```

AddAxioms(prover, Beliefs(HighB))

REMARKS: The main loop begins here

REMARKS: Enumeration by entrenchment rank rather than individual belief

FOR rank := 1 TO NoRanks(ProblemB)

```
IF ElapsedTime() > TimeLimit AND TimeLimit > 0
   EXIT
ENDIF
```

REMARKS: Extracting all beliefs from a rank

SingleRankB = Cut(ProblemB, rank, rank)

REMARKS: May need to compute minimal subsets entailing α

```
 IF NoElements(SingleRankB) > 1
   AddAxioms(prover, Beliefs(SingleRankB))
   IF NOT Proved(prover, α)
      NewB := NewB + SingleRankB
   ELSE
      RemoveAxioms(prover, Beliefs(SingleRankB))
      problemsets := SortedPowerSet(SingleRankB)
```

REMARKS: problemsets e.g., [a, b, c, ab, bc, ac, abc] with entrenchment degrees

```
      minimalsubsets := {}
```

REMARKS: This loop may be very computational expensive $O(2^n)$

```
FOR x := 1 TO NoElements(problemsets)
   IF (NOT SuperSet(problemsets[x], minimalsubsets))
      AddAxioms(prover, Beliefs(problemsets[x]))
      IF Proved(prover, α)
         minimalsubsets := minimalsubsets + problemsets[x]
      ENDIF
      RemoveAxioms(prover, Beliefs(problemsets[x]))
   ENDIF
NEXT
changedset := Union(minimalsubsets)
unchangedset := SingleRankB - changedset
IF NOT unchangedset = {}
       NewB := NewB + unchangedset
       AddAxioms(prover, Beliefs(unchangedset))
ENDIF
IF Ndegree > MinDegree
   FOR y := 1 TO NoElements(changedset)
       changedset[y].degree := Ndegree
   NEXT
   NewB := NewB + changedset
```

```
      ENDIF
   ENDIF
```

REMARKS: There is only one belief in this rank

REMARKS: Use the same procedure as RAM

```
 ELSE
   onebelief := SingleRankB
   AddAxioms(prover, onebelief.belief)
   IF Proved(prover, α)
      IF Ndegree > MinDegree
         onebelief.degree := Ndegree
         NewB := NewB + onebelief
      ENDIF
      RemoveAxioms(prover, onebelief.belief)
   ELSE
      NewB := NewB + onebelief
   ENDIF
 ENDIF
```

NEXT

```
IF Ndegree > MinDegree
        NewB := NewB + (α, Ndegree)
```

```
        NewB := NewB + LowB

    ENDIF
```

RETURN NewB

END FUNCTION

Appendix F

AIFS vs. TREC-7 Adaptive Filtering Systems

Figure F.1 and Figure F.2 depict the F1 and F3 utility scores as obtained by AIFS and the adaptive information filtering systems participated in TREC-7. Figure F.1 shows the comparison over TREC topics 1-25, and Figure F.2 shows the comparison over TREC topics 26-50. The last row in Figure F.2 depicts the average figures from AIFS and the filtering systems in TREC-7, and their differences. Since only the F1 and F3 utility scores are available from TREC-7 proceeding Web site: (`http://trec.nist.gov/pubs/trec7/t7_proceedings.html`), comparison between AIFS and the adaptive filtering systems in TREC-7 is done based on these two measures only. The first column in Figure F.1 and Figure F.2 depicts the TREC topic numbers. The second and the third columns depict the F1 and F3 scores of AIFS, and the forth and the fifth columns show the average F1 and F3 scores achieved by the adaptive filtering systems in TREC-7. A positive figure in the last two columns means that AIFS's result is better than the average performance of the adaptive filtering systems participated in TREC-7 for a particular topic. All the positive figures in these two columns are highlighted. By comparing the F1 scores (the sixth column), AIFS's performance is better than the average performance of the TREC-7 adaptive filtering systems in **44** topics. By comparing the F3 scores (the last column), AIFS's performance is better than the average performance of the TREC-7 adaptive filtering systems in **32** topics. It should be noted that a topic-by-topic comparison is necessary since each topic represents an IR task with quite different characteristic. Unless an accurate normalisation procedure that takes into account the intrinsic characteristic of each topic can be developed, computing the mean and standard deviation based on the figures across the various topics does not lead to a more accurate evaluation among different IR models. Apparently, the performance of our belief-based adaptive information agent system AIFS is better than the average performance of the TREC-7 adaptive filtering systems in more than half of the TREC topics.

Topic	F1	F3	T7 Avg F1	T7 Avg F3	Diff F1	Diff F3
1	-2	9	-41.8	106.2	39.8	-97.2
2	-26	2	-374.0	-54.1	348.0	56.1
3	-45	10	-3880.6	-1831.7	3835.6	1841.7
4	-3	1	-740.9	-295.6	737.9	296.6
5	13	19	-96.5	-20.0	109.5	39.0
6	-80	-25	-108.6	112.7	28.6	-137.7
7	-10	-5	-150.6	148.4	140.6	-153.4
8	-16	-8	-148.0	-72.4	132.0	64.4
9	-17	68	-45.2	33.5	28.2	34.5
10	132	271	-354.0	65.6	486.0	205.4
11	-13	176	-125.0	326.1	112.0	-150.1
12	-9	23	18.5	439.5	-27.5	-416.5
13	79	127	71.0	115.9	8.0	11.1
14	-2	-1	-177.2	-76.4	175.2	75.4
15	-19	-7	-227.8	-75.8	208.8	68.8
16	-15	0	-67.2	0.6	52.2	-0.6
17	-30	135	17.8	153.8	-47.8	-18.8
18	-22	-11	-372.5	-283.2	350.5	272.2
19	-8	56	-80.5	40.9	72.5	15.1
20	-39	13	41.3	184.7	-80.3	-171.7
21	-10	-5	15.9	30.8	-25.9	-35.8
22	-86	198	411.5	1196.2	-497.5	-998.2
23	68	234	170.6	278.6	-102.6	-44.6
24	-93	26	-113.7	96.2	20.7	-70.2
25	-12	-6	-38.4	9.5	26.4	-15.5

Figure F.1: Comparison AIFS vs. Filtering Systems in TREC-7 for Topics (1-25)

Topic	F1	F3	T7 Avg F1	T7 Avg F3	Diff F1	Diff F3
26	-8	1	-11.9	8.5	3.9	-7.5
27	-8	1	-156.2	-68.3	148.2	69.3
28	-49	-12	-173.2	13.0	124.2	-25.0
29	3	4	-62.3	-36.7	65.3	40.7
30	-2	-1	-178.9	-86.8	176.9	85.8
31	0	0	-35.6	-17.2	35.6	17.2
32	-2	-1	-71.0	-88.2	69.0	87.2
33	0	0	-36.3	-17.8	36.3	17.8
34	-4	-2	-71.1	-33.9	67.1	31.9
35	-2	-1	-36.3	-19.9	34.3	18.9
36	-8	-4	-39.6	-12.5	31.6	8.5
37	13	19	-121.9	-54.5	134.9	73.5
38	14	37	-281.3	27.8	295.3	9.2
39	-4	-2	-211.8	-23.8	207.8	21.8
40	-18	6	-390.1	78.2	372.1	-72.2
41	0	0	-211.8	-94.2	211.8	94.2
42	1	3	-208.3	-80.5	209.3	83.5
43	-37	-16	-178.5	-110.1	141.5	94.1
44	-2	9	-77.4	1.9	75.4	7.1
45	0	0	-238.5	-170.3	238.5	170.3
46	-4	-2	-44.2	59.2	40.2	-61.2
47	0	0	-88.3	-47.5	88.3	47.5
48	6	8	-89.2	-41.2	95.2	49.2
49	-6	7	-8.3	35.5	2.3	-28.5
50	-2	-1	-51.3	-73.4	49.3	72.4
Avg.	-7.7	27.1	-189.4	-4.5	181.7	31.5

Figure F.2: Comparison AIFS vs. Filtering Systems in TREC-7 for Topics (26-50)

Appendix G

Comparison AIFS vs. Claritech

Figure G.1 and Figure G.2 depict the F1 and F3 utility scores as obtained by AIFS and Claritech which is among the best adaptive filtering system in TREC-7. The first column in Figure G.1 and Figure G.2 depicts the TREC topic numbers. The second and the third columns depict the F1 and F3 scores of AIFS, and the forth and the fifth columns show the average F1 and F3 scores achieved by Claritech. A positive figure in the last two columns means that AIFS's result is better than that of Claritech for that particular topic. All the positive figures in these two columns are highlighted.

Topic	AIFS F1	AIFS F3	Claritech F1	Claritech F3	Diff F1	Diff F3
1	-2	9	1	128	-3.0	-119.0
2	-26	2	-13	140	-13.0	-138.0
3	-45	10	13	139	-58.0	-129.0
4	-3	1	-15	71	12.0	-70.0
5	13	19	27	50	-14.0	-31.0
6	-80	-25	-20	166	-60.0	-191.0
7	-10	-5	110	398	-120.0	-403.0
8	-16	-8	-38	-19	22.0	11.0
9	-17	68	34	152	-51.0	-84.0
10	132	271	233	607	-101.0	-336.0
11	-13	176	96	581	-109.0	-405.0
12	-9	23	250	901	-259.0	-878.0
13	79	127	172	283	-93.0	-156.0
14	-2	-1	-11	5	9.0	-6.0
15	-19	-7	-84	-67	65.0	60.0
16	-15	0	-5	28	-10.0	-28.0
17	-30	135	135	333	-165.0	-198.0
18	-22	-11	9	34	-31.0	-45.0
19	-8	56	-3	20	-5.0	36.0
20	-39	13	93	328	-132.0	-315.0
21	-10	-5	-13	9	3.0	-14.0
22	-86	198	698	2078	-784.0	-1880.0
23	68	234	360	775	-292.0	-541.0
24	-93	26	67	387	-160.0	-361.0
25	-12	-6	-62	21	50.0	-27.0

Figure G.1: Comparison AIFS vs. Claritech in TREC-7 for Topics (1-25)

Topic	AIFS F1	AIFS F3	Claritech F1	Claritech F3	Diff F1	Diff F3
26	-8	1	-9	24	1.0	-23.0
27	-8	1	5	21	-13.0	-20.0
28	-49	-12	-41	-18	-8.0	6.0
29	3	4	-34	-17	37.0	21.0
30	-2	-1	-49	-22	47.0	21.0
31	0	0	-51	-24	51.0	24.0
32	-2	-1	-30	-9	28.0	8.0
33	0	0	-61	-10	61.0	10.0
34	-4	-2	-38	-14	34.0	12.0
35	-2	-1	-77	-31	75.0	30.0
36	-8	-4	-30	-7	22.0	3.0
37	13	19	-90	-46	103.0	65.0
38	14	37	34	108	-20.0	-71.0
39	-4	-2	-37	-11	33.0	9.0
40	-18	6	33	442	-51.0	-436.0
41	0	0	-34	-28	34.0	28.0
42	1	3	26	82	-25.0	-79.0
43	-37	-16	-23	26	-14.0	-42.0
44	-2	9	-9	34	7.0	-25.0
45	0	0	-21	-10	21.0	10.0
46	-4	-2	2	143	-6.0	-145.0
47	0	0	-42	-20	42.0	20.0
48	6	8	-26	-13	32.0	21.0
49	-6	7	-15	-18	9.0	25.0
50	-2	-1	-30	-15	28.0	14.0
Avg.	-7.7	27.1	28.0	162.0	-35.4	-135.2

Figure G.2: Comparison AIFS vs. Claritech in TREC-7 for Topics (26-50)

Bibliography

[AB99] G. Amati and P.D. Bruza. A logical approach to query reformulation motivated from belief change. Proceedings of the Workshop on Logical and Uncertainty Models for Information Systems (LUMIS), 1999. Available from http://www .dcs.gla.ac.uk/lumis99/programme-40. html.

[ACL+OO] Y. Aridor, D. Carmel, R. Lempel, A. Soffer, and Y.S. Maarek. Knowledge agents on the web. In Matthias Klusch and Larry Kerschberg, editors, *Proceedings of the 4th International Workshop on Cooperative Infomnation Agents (CIA'2000),* volume 1860 of *Lecture Notes in Artificial Intelligence,* pages 15–26, Boston, Massachusetts, July 2000. Springer-Verlag.

[AFJM95] R. Armstrong, D. Freitag, T. Joachims, and T. Mitchell. Webwatcher: A learning apprentice for the world wide web. In *AAAI Spring Symposium on Information Gathering,* pages 6–12, 1995.

[AG96] G. Amati and K. Georgatos. Relevance as deduction: a Logical View of Information Retrieval. In F. Crestani and M. Lalmas, editors, *Proceedings of the Second Workshop on Information Retrieval, Uncertainty and Logic WIRUL'96,* pages 21–26. University of Glasgow, Glasgow, Scotland, 1996. Technical Report TR-1996-29.

[AGM85] C.E. Alchourrón, P. Gardenfors, and D. Makinson. On the logic of theory change: partial meet contraction and revision functions. *Journal of Symbolic Logic,* 50:510-530, 1985.

[AS94] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, editors, *VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases,* pages 487–499, Santiago de Chile, Chile, September 12–15 1994. Morgan Kaufmann Publishers.

[Ba197] M. Balabanovic. An adaptive web page recommendation service. In W. Lewis Johnson and Barbara Hayes-Roth, editors, *Proceedings of the First International Conference on Autonomous Agents (Agents'97)*, pages 378–385, Marina del Rey, California, February 5-8, 1997. ACM Press.

[Bal98] M. Balabanovic. Exploring versus exploiting when learning user models for text recommendation. *User Modeling and User-Adapted Interaction*, 8(1-2):71-102, 1998.

[Bar89a] H. Barlow. Unsupervised learning. *Neural Computation*, l(3):295-311, 1989.

[Bar89b] K. Barwise. Logic and information. In K. Jon Barwise, editor, *The Situation in Logic*, pages 37–57. CSLI Publications, 1989.

[BC92] N. Belkin and W. Croft. Information Filtering and Information Retrieval: Two sides of the same coin? *Communications of the ACM*, 35(12):29-38, 1992.

[BGMS98] B. Bessant, E. Grégoire, P. Marquis, and L. Saddotis. Combining non-monotonic reasoning and belief revision: A practical approach. In Fausto Giunchiglia, editor, *Proceedings of the 8th International Conference on Artificial Intelligence: Methodology, Systems, and Applications (AIMSA-98)*, volume 1480 of *LNAI*, pages 115-128. Springer, September 21–23 1998.

[BGNR96] F. Bergadano, D. Gunetti, M. Nicosia, and G. Ruffo. Learning logic programs with negation as failure. In L. Raedt, editor, *Advances in Inductive Logic Programming*. IOS Press, New York, New York, 1996.

[BH94] P.D. Bruza and T.W.C. Huibers. Investigating Aboutness Axioms Using Information Fields. In W.B. Croft and C.J. van Rijsbergen, editors, *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 112–121, Dublin, Ireland, July 1994. ACM Press.

[BH96] P.D. Bruza and T.W.C. Huibers. A Study of Aboutness in Information Retrieval. *Artificial Intelligence Review*, 10:1–27, 1996.

[BH98] K. Bharat and M. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In W. Bruce Croft, Alistair Moffat, C. J.

van Rijsbergen, Ross Wilkinson, and Justin Zobel, editors, *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-98)*, pages 104–111, Melbourne, Australia, August 1998. ACM Press.

[BL98] P.D. Bruza and B. van Linder. Preferential Models of Query by Navigation. In F. Crestani, M. Lalmas, and C.J. van Rijsbergen, editors, *Information Retrieval: Uncertainty and Logics*, volume **4** of *The Kluwer International Series on Information Retrieval*, pages 73–96. Kluwer Academic Publishers, 1998.

[BP98] D. Billsus and M. Pazzani. Learning collaborative information filters. In J. Shavlik, editor, *Proceedings of the 15th International Conference on Machine Learning*, pages 46–54, Madison, Wisconsin, July 1998. Morgan Kaufmann Publishers.

[BP99] D. Billsus and M.J. Pazzani. A personal news agent that talks, learns and explains. In Brian Drabble, editor, *Proceedings of the Third International Conference on Autonomous Agents (Agents'99)*, pages 268–275, Seattle, Washington, May 1–5 1999. ACM Press.

[BPR+99] F. Bergadano, A. Puliafito, S. Riccobene, G. Ruffo, and L. Vita. Java-based and secure learning agents for information retrieval in distributed systems. *Information Sciences*, 113(1-2):55-84, 1999.

[Bro95] T.A. Brooks. People, Words, and Perceptions: A Phenomenological Investigation of Textuality. *Journal of the American Society for Information Science*, 46(2):103-115, 1995.

[Bru96] P.D. Bruza. Intelligent Filtering using Nonmonotonic Inference. In *Proceedings of the 1st Australian Document Computing Symposium*, pages 1–7, Melbourne, Australia, March 1996. Department of Computer Science, RMIT.

[BS95] M. Balabanovic and Y. Shoham. Learning information retrieval agents: Experiments with automated web browsing. In *Proceedings of the AAAI Spring Symposium on Information Gathering from Heterogenous Distributed Resources*, pages 13–18, Palo Alto, California, 1995.

[BSA94] C. Buckley, G. Salton, and J. Allan. The effect of adding relevance information in a relevance feedback environment. In W. Bruce Croft and C. J. van Rijsbergen, editors, *Proceedings of the Seventeenth Annual*

International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 292–300, Dublin, Ireland, July 1994. ACM Press.

[BSWOO] P. Bruza, D. Song, and K. Wong. Aboutness from a commonsense perspective. *Journal of the American Society for Information Science*, 51(12):1090-1105, 2000.

[CBS90] V. Chaitanya, A. Bharati, and R. Sangal. Natural language processing, complexity theory and logic. In *Foundations of Software Technology and Theoretical Computer Science*, page 410. Springer, 1990.

[CC92] Y. Chiaramella and J. P. Chevallet. About retrieval models and logic. *The Computer Journal*, 35(3):233-242, June 1992.

[CHSS98] H. Chen, A. Houston, R. Sewell, and R. Schatz. Internet browsing and searching: User evaluations of category map and concept space techniques. *Journal of the American Society for Information Science*, 49(7):604-618, 1998.

[Coh95] W. Cohen. Text categorization and relational learning. In A. Prieditis and S. Russell, editors, *Proceedings of ICML-95, 12th International Conference on Machine Learning*, pages 124–132, Tahoe City, California, July 9–12, 1995. Morgan Kaufmann Publishers.

[Con02] Text REtrieval Conference. TREC: English Relevance Judgments, 2002. Available from *http ://trec.nist.gov/data/qrels_eng/index. html.*

[Cor00] Compaq Computer Corporation. Alta Vista Net Guide, 2000. Available from *http ://www.altavista.com/.*

[Cre98] Fabio Crestani. Logical Imaging and Probabilistic Information Retrieval. In M. Lalmas F. Crestani and C.J. van Rijsbergen, editors, *Information Retrieval: uncertainty and logics*, pages 247–279. Kluwer Academic Publishers, Norwell, Massachusetts, 1998.

[CRSR95] F. Crestani, I. Ruthven, M. Sanderson, and C.J. van Rijsbergen. The troubles with using a logical model of IR on a large collection of documents. In D.K. Harman, editor, *proceedings of the fourth Text REtrieval Conference (TREC-4)*, Gaithersburg, Maryland, November 1995. NIST.

[CvR95] F. Crestani and C.J. van Rijsbergen. Information retrieval by logical imaging. *Journal of Documentation*, 51(1):3-17, 1995.

[Dal88] M. Dalal. Investigations into a theory of knowledge base revision: Preliminary report. In *Proceedings of the 7th National Conference on Artificial Intelligence (AAAI'88)* pages 475–479, St. Paul, Minnesota, August 1988.

[DDF+90] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391-407, 1990.

[DF93] S. Dixon and N. Foo. Connections between the ATMS and AGM Belief Revision. In Ruzena Bajcsy, editor, *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 534–539, Chambery, France, 1993. Morgan Kaufmann Publishers.

[DH73] R. Duda and P. Hart. *Pattern Classification and Scene Analysis.* John Wiley & Sons, New York, New York, 1973.

[DIU98] J. Delgado, N. Ishii, and T. Ura. Content-based collaborative information filtering: Actively learning to classify and recommend documents. In Matthias Klusch and Gerhard Weiß, editors, *Proceedings of the 2nd International Workshop on Cooperative Information Agents 11: Learning, Mobility and Electronic Commerce for Information Discovery on the Internet*, pages 206–215, Paris, France, July 4–7, 1998. Springer-Verlag. LNAI 1435.

[dK86] J. de Kleer. An assumption-based truth maintenance system. *Artificial Intelligence*, 28:127-162, 1986.

[DP91] D. Dubois and H. Prade. Epistemic entrenchment and possibilistic logic. *Artificial Intelligence*, 50(2):223-239, 1991.

[Dre81] F. Dretske. *Knowledge and the Flow of Information.* The MIT Press, Cambridge, Massachusetts, 1981.

[dV97] Alvaro del Val. Nonmonotonic Reasoning and Belief Revision: Syntactic, Semantic, Foundational, and Coherence Approaches. *Journal of Applied Non-Classical Logics*, 7(2):213-240, 1997.

[DWR97] N. Davies, R. Weeks, and M. Revett. Information agents for the World Wide Web. In Hyacinth S. Nwana and Nader Azarmi, editors, *Software Agents and Soft Computing*, volume 1198 of *Lecture Notes in Computer Science*, pages 81–99. Springer-Verlag, 1997.

[EM01] H. Ekbia and A. Maguitman. Context and relevance: A pragmatic approach. In Varol Akman, Paolo Bouquet, Richmond Thomason, and Roger A. Young, editors, *Modeling and Using Context,* pages 156–169. Springer-Verlag, 2001.

[FH96] R. Feldman and H. Hirsh. Mining associations in text in the presence of background knowledge. In Evangelos Simoudis, Jia Wei Han, and Usama Fayyad, editors, *Proceedings of the 2nd International Conference on Knowledge Discovery in Databases and Data Mining,* pages 343–346, Portland, Oregon, August 1996. AAAI Press.

[FK96] I. Ferguson and G. Karakoulas. Multiagent learning and adaptation in an information filtering market. In Sandip Sen, editor, *Working Notes for the AAAI Symposium on Adaptation, Co-evolution and Learning in Multiagent Systems,* pages 28–32, Stanford, California, 1996.

[Gal92] J. R. Galliers. Autonomous Belief Revision and Communication. In P. Gardenfors, editor, *Belief Revision,* pages 220–246. Cambridge University Press, Cambridge, UK, 1992.

[Gär88] P. Gardenfors. *Knowledge in flux: modeling the dynamics of epistemic states.* The MIT Press, Cambridge, Massachusetts, 1988.

[Gär90] P. Gardenfors. The dynamics of belief systems: Foundations vs. Coherence theories. *Revue Internationale de Philosophie,* 172:24-46, 1990.

[Gär92] P. Gardenfors. Belief revision: An introduction. In P. Gardenfors, editor, *Belief Revision,* pages 1–28. Cambridge University Press, Cambridge, UK, 1992.

[GCS98] S. Green, P. Cunningham, and F. Somers. Agent mediated collaborative web page filtering. In Matthias Klusch and Gerhard Weiß, editors, *Proceedings of the 2nd International Workshop on Cooperative Information Agents 11: Learning, Mobility and Electronic Commerce for Information Discovery on the Internet,* volume 1435 of *Lecture Notes in Artificial Intelligence,* pages 195–205. Springer-Verlag, July 4–7 1998.

[Geo96] K. Georgatos. Ordering-based representations of rational inference. In José Julio Alferes, Luis Moniz Pereira, and Ewa Orlowska, editors, *Proceedings of the European JELIA Workshop (JELIA-96): Logics in Artificial Intelligence,* pages 176–191. Springer, September 30–October 3 1996.

[GM88] P. Gardenfors and D. Makinson. Revisions of knowledge systems using epistemic entrenchment. In Moshe Y. Vardi, editor, *Proceedings of the Second Conference on Theoretical Aspects of Reasoning About Knowledge,* pages 83–95, Pacific Grove, California, March 1988. Morgan Kaufmann.

[GM94] P. Gardenfors and D. Makinson. Nonmonotonic inference based on expectations. *Artificial Intelligence,* 65(2):197-245, 1994.

[GMM98] R. Guttman, A. Moukas, and P. Maes. Agent-mediated electronic commerce: A survey. *Knowledge Engineering Review,* 13(2):147-159, June 1998.

[Gre98] Warren R. Greiff. A theory of term weighting based on exploratory data analysis. In W. Bruce Croft, Alistair Moffat, C. J. van Rijsbergen, Ross Wilkinson, and Justin Zobel, editors, *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-98),* pages 11–19, Melbourne, Australia, August 24–28 1998. ACM Press, New York.

[GSK+99] N. Good, J. Schafer, J. Konstan, A. Borchers, B. Sarwar, J. Herlocker, and J. Riedl. Combining collaborative filtering with personal agents for better recommendations. In *Proceedings of the 6th National Conference on Artificial Intelligence (AAAI-99),* pages 439–446, Menlo Park, California., July 18–22 1999. MIT Press.

[Har77] W.L. Harper. Rational Conceptual Change. In *Proceedings of the 1976 Meeting of the Philosophy of Science Association,* volume 2, pages 462-494. Philosophy of Science Association, 1977.

[Hod93] W. Hodges. Logical Features of Horn Clauses. In D.M. Gabbay, C.J. Hogger, J.A. Robinson, and J. Siekmann, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming,* volume 1, pages 449–503. Clarendon Press, Oxford, UK, 1993.

[Hul98] D. Hull. The TREC-7 Filtering Track: Description and Analysis. In E.M. Voorhees and D.K. Harman, editors, *Proceedings of the seventh Text REtrieval Conference (TREC-7),* pages 33–56, Gaithersburg, Maryland, November 9–11 1998. NIST. Available from *http://trec.nist.gov/pubs/trec7/t7_proceedings.html.*

[Hun95] A. Hunter. Using default logic in information retrieval. In Christine Froidevaux and Jurg Kohlas, editors, *Proceedings of the ECSQARU European Conference on Symbolic and Quantitive Approaches to Reasoning and Uncertainty,* volume 946 of *Lecture Notes in Artificial Intelligence,* pages 235–242. Springer-Verlag, July 1995.

[Hun96] A. Hunter. Intelligent text handling using default logic. In *Proceedings of the IEEE Conference on Tools with Artificial Intelligence,* pages 34–40. IEEE Computer Society Press, 1996.

[Hun97] A. Hunter. Using Default Logic for Lexical Knowledge. In Dov Gabbay, editor, *Proceedings of the first international joint conference on qualitative and quantitative practical reasoning (ECSQARU FAPR'97),* volume 1244 of *Lecture Notes in Computer Science*, pages 322–335. Springer-Verlag, 1997.

[HvL96] T. Huibers and B. van Linder. Formalising Intelligent Information Retrieval Agents. In *Proceedings of the 18th British Computer Society Annual Information Retrieval Colloqium,* pages 125–143, Manchester, UK, 1996. Manchester Metropolitan University.

[HW98] T. Huibers and B. Wondergem. Towards an axiomatic aboutness theory for information retrieval. In M. Lalmas F. Crestani and C.J. van Rijsbergen, editors, *Information Retrieval: uncertainty and logics,* pages 297-318. Kluwer Academic Publishers, Norwell, Massachusetts, 1998.

[JBB00] W. Jie, S. Bressan, and O. Beng. Mining term association rules for automatic global query expansion: methodology and preliminary results. In Qing Li, Z. Meral Ozsoyoglu, Roland Wagner, Yahiko Kambayashi, and Yanchun Zhang, editors, *Proceedings of the First International Conference on Web Information Systems Engineering,* volume 1, pages 366–373, Hong Kong, China, 2000. IEEE Computer Society.

[JFM97] T. Joachims, D. Freitag, and T. Mitchell. Webwatcher: A tour guide for the world wide web. In Martha E. Pollack, editor, *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence,* pages 770–775, Nagoya, Japan, August 23–29, 1997. Morgan Kaufmann Publishers.

[JSW98] N. Jennings, K. Sycara, and M. Wooldridge. A roadmap of agent research and development. *Journal of Autonomous Agents and Multi-Agent Systems*, 1(1):7-38, 1998.

[Kel98] K. Kelly. The learning power of belief revision. In Itzhak Gilboa, editor, *Theoretical Aspects of Reasoning About Knowledge: Proceedings of the Seventh Conference (TARK 1998)*, pages 111–124, San Francisco, California, 1998. Morgan Kaufmann Publisher.

[KF95] G. Karakoulas and I. Ferguson. A computational market for information filtering in multi-dimensional spaces. In *Working Notes of the AAAI Fall Symposium on AI Applications in Knowledge Navigation and Retrieval*, 1995.

[KF98] G. Karakoulas and I. Ferguson. Applying SIGMA to the TREC-7 filtering track. In E.M. Voorhees and D.K. Harman, editors, *Proceedings of the seventh Text REtrieval Conference*, pages 258–263, Gaithersburg, Maryland, November 1998. NIST.

[Kir98] S. Kirsch. The future of Internet Search: Infoseek's experiences searching the Internet. *ACM SIGIR FORUM*, 32(2):3-7, 1998.

[KLM90] S. Kraus, D. Lehmann, and M. Magidor. Nonmonotonic Reasoning, Preferential Models and Cumulative Logics. *Artificial Intelligence*, 44(1):167–207, 1990.

[Kra97] G. Kraetzschmar. *Distributed Reason Maintenance for Multiagent Systems*. PhD thesis, School of Engineering, University of Erlangen-Nurnberg, Erlangen, Germany, 1997.

[Kri71] Saul Kripke. Semantical considerations on modal logic. In Leonard Linsky, editor, *Reference and modality*, pages 63–73. Oxford University Press, Oxford, UK, 1971.

[KS97] D. Koller and M. Sahami. Hierarchically classifying documents using very few words. In Douglas H. Fisher, editor, *Proceedings of ICML-97, 14th International Conference on Machine Learning*, pages 170–178, Nashville, Tennessee, 1997. Morgan Kaufmann Publishers, San Francisco, California.

[KYMW97] T. Kindo, H. Yoshida, T. Morimoto, and T. Watanabe. Adaptive personal information filtering system that organizes personal profiles automatically. In Martha E. Pollack, editor, *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, pages 716-721, Nagoya, Japan, August 23–29, 1997. Morgan Kaufmann publishers.

[Lal98] M. Lalmas. Logical models in information retrieval: Introduction and overview. *Information Processing & Management,* 34(1):19-33, 1998.

[Lan97] J. Lang. Possibilistic Logic: Algorithms and Complexity. In J. Kohlas and S. Moral, editors, *Handbook of Algorithms for Uncertainty and Defeasible Reasoning.* Kluwer Academic Publishers, Norwell, Massachusetts, 1997.

[Lau02a] R.Y.K. Lau. Belief Revision for Adaptive Negotiation Agents. In *Proceedings of the Fourth Workshop on Agent Mediated Electronic Commerce (AMEC IV) (Poster Section),* Bologna, Italy, July 16 2002. Available from **http://www.eecs.harvard.edu/amec4.**

[Lau02b] R.Y.K. Lau. The State of the Art in Adaptive Information Agents. *International Journal on Artificial Intelligence Tools,* 11(1):19-61, March 2002.

[Law00] S. Lawrence. Context in web search. *IEEE Data Engineering Bulletin,* 23(3):25-32, 2000.

[LB98] M. Lalmas and P.D. Bruza. The Use of Logic in Information Retrieval Modeling. *Knowledge Engineering Review,* 13(3):263-295, 1998.

[LB99] D. Losada and A. Barreiro. Using a Belief Revision Operator for Document Ranking in Extended Boolean Models. In Fredric Gey, Marti Hearst, and Richard Tong, editors, *Proceedings of the 22nd Annual ACM Conference of Research and Development in Information Retrieval (SIGIR'99)*, pages 66–72, Berkeley, California, 1999. ACM Press.

[LC01] V. Lavrenko and W. Croft. Relevance-based language models. In W. Bruce Croft, David J. Harper, Donald H. Kraft, and Justin Zobel, editors, *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-01)*, pages 120–127, New Orleans, Louisiana, September 9–13 2001. ACM Press.

[LDV99] H. Lieberman, N. Van Dyke, and A. Vivacqua. Let's Browse: A Collaborative Browsing Agent. *Knowledge-Based Systems,* 12:427-431, 1999.

[Lev77] I. Levi. Direct Inference. *The Journal of Philosophy,* 74:5-29, 1977

[LG98] S. Lawrence and C. Giles. Context and page analysis for improved web search. *IEEE Internet Computing,* 2(4):38-46, 1998.

[Lie95] H. Lieberman. Letizia: An agent that assists web browsing. In Chris S. Mellish, editor, *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence,* pages 924–929, Montreal, Canada, August 20–25, 1995. Morgan Kaufmann publishers.

[LM92] D. Lehmann and M. Magidor. What does a conditional knowledge base entail? *Artificial intelligence,* 55(1):1–60, 1992.

[Los99] R. Losee. *The Science of Information: Measurement and Applications.* Academic Press, New York, New York, 1999.

[LR92] M. Lalmas and C.J. van Rijsbergen. A Logical Model of Information Retrieval based on Situation Theory. In *Proceedings of the 14th British Computer Society Annual Information Retrieval Colloqium,* pages 1–13, Lancaster, United Kingdom, 1992.

[LRJ94] B. Logan, S. Reece, and K. S. Jones. Modelling information retrieval agents with belief revision. In W. Bruce Croft and C. J. van Rijsbergen, editors, *Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* pages 91–100, Dublin, Ireland, 1994.

[LSC+98] S.H. Lin, C.S. Shih, M.C. Chen, J.M. Ho, M.T. Ko, and Y.M. Huang. Extracting classification knowledge of internet documents with mining term associations: A semantic approach. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98),* Information Retrieval Experiments, pages 241–249, Melbourne, Australia, 1998.

[LtHB99] R.Y.K. Lau, A.H.M. ter Hofstede, and P.D. Bruza. A Study of Belief Revision in the Context of Adaptive Information Filtering. In Lucas C. Hui and Dik L. Lee, editors, *Proceedings of the Fifth International Computer Science Conference (ICSC'99),* volume 1749 of *Lecture Notes in Computer Science,* pages 1–10, Hong Kong, China, 1999. Springer.

[LtHB00] R.Y.K. Lau, A.H.M. ter Hofstede, and P.D. Bruza. Adaptive Profiling Agents for Electronic Commerce. In Joan Cooper, editor, *Proceedings of the 4th CollECTeR Conference on Electronic Commerce,* Breckenridge, Colorado, April 2000. Available from `http://www.collecter.org/past_collecter.htrnl`.

[LtHB01a] R.Y.K. Lau, A.H.M. ter Hofstede, and P.D. Bruza. Belief Revision for Adaptive Information Filtering Agents. *International Journal of Cooperative Information Systems,* 10(1-2):57-79, March–June 2001.

[LtHB01b] R.Y.K. Lau, A.H.M. ter Hofstede, and P.D. Bruza. Maxi-Adjustment and Possibilistic Deduction for Adaptive Information Agents. *Journal of Applied Non-Classical Logics,* 11(1-2):169-201, 2001.

[LtHB01c] R.Y.K. Lau, A.H.M. ter Hofstede, and P.D. Bruza. Nonmonotonic Reasoning for Adaptive Information Filtering. In Michael Oudshoorn, editor, *Proceedings of the Twenty-Fourth Australasian Computer Science Conference (ACSC2001),* pages 109–116, Gold Coast, Australia, January 29-February 2 2001. IEEE Press.

[Mae94] P. Maes. Agents that reduce work and information overload. *Communications of the ACM,* 37(7):31-40, July 1994.

[Mak93] D. Makinson. General Patterns in Nonmonotonic Reasoning. In D.M. Gabbay, C.J. Hogger, J.A. Robinson, and D. Nute, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming,* volume 3, pages 35–110. Oxford University Press, Oxford, UK, 1993.

[Man87] M. Mansuripur. *Information Theory.* Prentice-Hall, Englewood Cliffs, New Jersey, 1987.

[MB00] F. Menczer and R. Belew. Adaptive retrieval agents: Internalizing local context and scaling up to the web. *Machine Learning,* 39(2-3):203-242, 2000.

[MBVL99] M. Martin-Bautista, M. Vila, and H. Larsen. A Fuzzy Genetic Algorithm Approach to an Adaptive Information Retrieval Agent. *Journal of the American Society for Information Science,* 50(9):760-771, 1999.

[MG91] D. Makinson and P. Gardenfors. Relations between the logic of theory change and the nonmonotonic logic. In André Fuhrmann and M. Morreau, editors, *The Logic of Theory Change,* pages 185–205. Springer-Verlag, 1991.

[MM98] A. Moukas and P. Maes. Amalthaea: An evolving information filtering and discovery system for the WWW. *Journal of Autonomous Agents and Multi-Agent Systems,* 1(1):59-88, 1998.

[MRF+90] G. A. Miller, Beckwith R., C. Fellbaum, D. Gross, and K. J. Miller. Introduction to wordnet: An on-line lexical database. *Journal of Lexicography*, 3(4):234-244, 1990.

[MSG97] U. Manber, M. Smith, and B. Gopal. Webglimpse: combining browsing and searching. In *Proceedings of the USENIX Technical Conference*, pages 195–206, Anaheim, USA, January 1997. ACM Press.

[MSST93] M. Meghini, F. Sebastiani, U. Straccia, and C. Thanos. A Model of information Retrieval based on Terminological Logic. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 298–307, Pittsburgh, PA, 1993. ACM Press.

[NBL95] J.Y. Nie, M. Brisebois, and F. Lepage. Information retrieval as counterfactual. *The Computer Journal*, 38(8):643-657, 1995.

[NBROO] R. Neves, J.F. Bonnefon, and E. Raufaste. Rationality in human nonmonotonic inference. In Chitta Baral and Miroslaw Truszczynski, editors, *Proceedings of the 8th International Workshop on Non-Monotonic Reasoning NMR '2000*, Breckenridge, Colorado, April 2000. Available from **http://xxx.lanl.gov/html/cs.AI/0003073.**

[NCP91] F. Nunamaker, J. Chen, and T. Purdin. System development. *Journal of Management Information Systems*, 7(3):89-106, 1991.

[Nie89] J.Y. Nie. An Information retrieval Model based on Modal Logic. *Information Processing & Management*, 25(5):477-491, 1989.

[Oar97] D.W. Oard. The state of the art in text filtering. *User Modeling and User-Adapted Interaction*, 7(3):141-178, 1997.

[OKW+97] J. Odubiyi, D. Kocur, S. Weinstein, N. Wakim, S. Srivastava, C. Gokey, and J. Graham. SAIRE: A scalable agent-based information retrieval engine. In W. Lewis Johnson and Barbara Hayes-Roth, editors, *Proceedings of the 1st International Conference on Autonomous Agents*, pages 292–300, Marina del Rey, California, February 5-8, 1997. ACM Press.

[O'L97] D.E. O'Leary. The Internet, Intranets, and the AI Renaissance. *IEEE Computer*, 30(1):71-78, 1997.

[OS95] A. O'Riordan and H. Sorensen. An intelligent agent for high-precision text filtering. In *Proceeding of the International Conference on Information and Knowledge Management (CIKM'95)*, pages 205–211, Baltimore, Maryland, 1995. ACM Press.

[PB97] M.J. Pazzani and D. Billsus. Learning and revising user profiles: The identification of interesting web sites. *Machine Learning*, 27(3):313-331, 1997.

[PB99] H. Proper and P. Bruza. What is Information Discovery About? *Journal of the American Society for Information Science*, 50(9):737-750, 1999.

[PMB96] M.J. Pazzani, J. Muramatsu, and D. Billsus. Syskill and Webert: Identifying interesting web sites. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 54–61, Portland, Oregon, 1996. MIT Press.

[PNF95] M. Pagnucco, A. Nayak, and N. Foo. Abductive reasoning, belief expansion and nonmonotonic consequence. In H. Decker, U. Geske, A. Kakas, C. Sakama, D. Seipel, and T. Urpi, editors, *Proceedings of the ICLP'95 Joint Workshop on Deductive Databases and Logic Programming and Abduction in Deductive Databases and Knowledge-based Systems*, pages 143–158, Shonan Village Center, Japan, June 1995.

[Por80] M. Porter. An algorithm for suffix stripping. *Program*, 14(3):130-137, 1980.

[PW95] Pavlos Peppas and Mary-Anne Williams. Constructive modelings for theory change. *Notre Dame Journal of Formal Logic*, 36(1):120-133, Winter 1995.

[Qui86] J. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81-106, 1986.

[Rij86] C.J. van Rijsbergen. A non-classical logic for information retrieval. *The Computer Journal*, 29(6):481-485, 1986.

[Rij89] C.J. van Rijsbergen. Towards an Information Logic. In N.J. Belkin and C.J. van Rijsbergen, editors, *Proceedings of the 12th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 77–86, Cambridge, Massachusetts, June 1989. ACM Press.

[RN98] E. Raufaste and R. Neves. Empirical evaluation of possibility theory in human radiological diagnosis. In *Proceedings of the 13th European Conference on Artificial Intelligence ECAI98*, pages 124–128, Sussex, UK, 1998. Universities of Brighton.

[RNM+94] P. Resnick, I. Neophytos, S. Mitesh, P. Bergstrom, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 Conference on Computer Supported Cooperative Work (CSCW'94)*, pages 175–186. Addison-Wesley, 1994.

[Rob77] S. Robertson. The probability ranking principle in IR. *Journal of Documentation*, 33:294-304, 1977.

[Roc71] J. Rocchio. Relevance Feedback in Information Retrieval. In G. Salton, editor, *The SMART retrieval system:experiments in automatic document processing*, pages 313–323. Prentice-Hall, Englewood Cliffs, New Jersey, 1971.

[Sal89] G. Salton. *Automatic Text Processing–The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, Reading, Massachusetts, 1989.

[Sal90] G. Salton. Full text information processing using the smart system. *Database Engineering Bulletin*, 13(1):2-9, March 1990.

[Sal91] G. Salton. Developments in automatic text retrieval. *Science*, 253:974-980, August 1991.

[SB88] G. Salton and C. Buckley. Term-weighted approaches to automatic text retrieval. *Information Processing and Management*, 24(5):513-523, 1988.

[SB90] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41(4):288-297, 1990.

[SDG+85] R.W. Schvaneveldt, F.T. Durso, T.E. Goldsmith, T.J. Breen, N.M. Cooke, R.G. Tucker, and J.C. De Maio. Measuring the structure of expertise. *International Journal of Man-Machine Studies*, 23(6):699-728, 1985.

[Seb94] F. Sebastiani. A Probabilistic Terminological Logic for Modelling Information Retrieval. In W.B. Croft and C.J. van Rijsbergen, editors, *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 122–130, Dublin, Ireland, July 1994. ACM Press.

[Seb98] F. Sebastiani. On the role of logic in information retrieval. *Information Processing & Management*, 34(1):1-18, 1998.

[SM83] G. Salton and M.J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, New York, 1983.

[SM95] U. Shardanand and P. Maes. Social information filtering: Algorithms for automating "word of mouth". In *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, volume 1, pages 210–217. ACM, 1995.

[SOO97] H. Sorensen, A. O'Riordan, and C. O'Riordan. Profiling with the INFOrmer text filtering agent. *J. UCS: Journal of Universal Computer Science*, 3(8):988-1010, August 1997.

[Spo87] W. Spohn. Ordinal conditional functions: A dynamic theory of epistemic states. In W. Harper and B. Skyrms, editors, *Causation in Decision, Belief Change and Statistics*, volume 2, pages 105–134. D. Reidel, Dordrecht, Netherlands, 1987.

[SSH99] B. Shapira, P. Shoval, and U. Hanani. Experimentation with an information filtering system that combines cognitive and sociological filtering integrated with user stereotypes. *Decision Support Systems*, 27(1-2):5–24, 1999.

[SSS98] R. Schapire, Y. Singer, and A. Singhal. Boosting and Rocchio applied to text filtering. In W. Bruce Croft, Alistair Moffat, Cornelis J. van Rijsbergen, Ross Wilkinson, and Justin Zobel, editors, *Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval*, pages 215–223, Melbourne, Australia, 1998. ACM Press.

[Sta81] R. Stalnaker. Probability and conditionals. In William L. Harper, R. Stalnaker, and Glenn Pearce, editors, *Ifs: conditionals, belief, decision, chance, time*, pages 107-128. Reidel/Kluwer, 1981.

[TKSOO] D. Tauritz, J. Kok, and I. Sprinkhuizen-Kuyper. Adaptive Information Filtering using Evolutionary Computation. *Information Sciences*, 122(2-4):121-140, 2000.

[TNMH97] K. Takahashi, Y. Nishibe, I. Morihara, and F. Hattori. Collecting shop and service information with software agents. *Applied Artificial Intelligence*, 11(6):489-500, 1997.

[Tsu00] H. Tsukimoto. Extracting rules from trained neural networks. *IEEE Transactions on Neural Networks*, 11(2):377-389, March 2000.

[VH99] E. Voorhees and D. Harman. Overview of the Eighth Text REtrieval Conference (TREC-8). In E.M. Voorhees and D.K. Harman, editors, *Proceedings of the eighth Text REtrieval Conference (TREC-8)*, pages 1–24, Gaithersburg, Maryland, November 17–19 1999. NIST. Available from `http://trec.nist.gov/pubs/trec8/t8_proceedings.html`.

[vR79] C.J. van Rijsbergen. *Information Retrieval.* Butterworths, London, United Kingdom, 1979.

[vRHP81] C. van Rijsbergen, D. Harper, and M. Porter. The selection of good search terms. *Information processing and Management*, 17(2):77-91, 1981.

[WD00] M. Wooldridge and K. Decker. Agents on the net: Infrastructure, Technology, Applications. *IEEE Internet Computing*, 4(2):46-48, March/April 2000.

[WFG01] I. Wang, N. Fiddian, and W. Gray. Using market mechanisms to control agent allocation in global information systems. *Lecture Notes in Computer Science*, 2113:547-556, 2001.

[Wil94] M.-A. Williams. Transmutations of knowledge systems. In Jon Doyle, Erik Sandewall, and Pietro Torasso, editors, *KR'94: Principles of Knowledge Representation and Reasoning*, pages 619–629, Bonn, Germany, May 24–27 1994. Morgan Kaufmann Publishers.

[Wil95] M.-A. Williams. Iterated theory base change: A computational model. In Chris S. Mellish, editor, *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1541–1547, Montréal, Canada, August 20–25, 1995. Morgan Kaufmann Publishers.

[Wil96a] M.-A. Williams. Applications of belief revision. In B. Freitag, H. Decker, M. Kifer, and A. Voronkov, editors, *Transactions and Change in Logic Databases*, volume 1472 of *Lecture Notes in Computer Science*, pages 287–316. Springer, September 23–27, 1996.

[Wi196b] M.-A. Williams. Towards a practical approach to belief revision: Reason-based change. In Luigia Carlucci Aiello, Jon Doyle, and Stuart Shapiro, editors, *KR '96: Principles of Knowledge Representation and Reasoning,* pages 412–420, Cambridge, Massachusetts, 1996. Morgan Kaufmann Publishers.

[Wi197] M.-A. Williams. Anytime belief revision. In Martha E. Pollack, editor, *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence,* pages 74–79, Nagoya, Japan, August 23–29, 1997. Morgan Kaufmann Publishers.

[WJ95] M. Wooldridge and N. Jennings. Intelligent Agents: Theory and Practice. *Knowledge Engineering Review,* 10(2):115–152, 1995.

[WS00] M.-A. Williams and A. Sims. SATEN: An Object-Oriented Web-Based Revision and Extraction Engine. In Chitta Baral and Miroslaw Truszczynski, editors, *Proceedings of the 8th International Workshop on Non-Monotonic Reasoning NMR '2000,* Breckenridge, Colorado, April 2000. Available from http ://xxx.lanl.gov/html/cs.AI/0003073.

[WSBC01] K. Wong, D. Song, P. Bruza, and C. Cheng. Application of aboutness to functional benchmarking in information retrieval. *ACM Transactions on Information Systems,* 19(4):337-370, October 2001.

[XC96] J. Xu and W.B. Croft. Query expansion using local and global document analysis. In Hans-Peter Frei, Donna Harman, Peter Schauble, and Ross Wilkinson, editors, *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* pages 4–11, Zurich, Switzerland, 1996.

[YKLOO] Edmund Yu, Ping Koo, and Elizabeth Liddy. Evolving intelligent text-based agents. In Carles Sierra, Gini Maria, and Jeffrey S. Rosenschein, editors, *Proceedings of the 4th International Conference on Autonomous Agents (AGENTS-OO),* pages 388–395, Atalonia, Spain, June 3–7, 2000. ACM Press.

[YP97] Y. Yang and J. Pedersen. A comparative study on feature selection in text categorization. In Douglas H. Fisher, editor, *Proceedings of ICML-97, 14th International Conference on Machine Learning,* pages 412-420, Nashville, Tennessee, July 8–12, 1997. Morgan Kaufmann Publishers, San Francisco, California.

[ZH00] S. Zelikovitz and H. Hirsh. Improving short text classification using unlabeled background knowledge. In Patrick Langley, editor, *Proceedings of ICML-00, 17th International Conference on Machine Learning*, pages 1183–1190, Stanford, California, June 29–July 2, 2000. Morgan Kaufmann Publishers, San Francisco, California.