CHAPTER 2

LITERATURE REVIEW

Web users today have access to an ocean of WWW and presume complete access to this vast source of information. A search query submitted on web is assumed to yield desired relevant results. However, present search engines deluge the user with popular results, irrespective of the users’ needs. This chapter presents an insight of prevalent context aware web search systems. The chapter begins with the introduction of varied aspects of context and the detailed discussion of issues related to context aware web search systems. It is followed by the techniques to perform retrieval from the WWW. The chapter proceeds further with an overview of intelligent web search systems, agent architectures and the tools and techniques used to develop the intelligent systems.

2.1 Introduction

The notion of context has been widely studied in multiple research disciplines such as computer science applications, cognitive science, linguistics, philosophy, psychology and organizational sciences.

According to Oxford English Dictionary (Webster, 2008), context has two primary senses “the words around a word, phrase, statement etc often used to explain the meaning.

or

“the general circumstances in which an event takes place”

Over the years context has been perceived differently by the distinct experts. Schilit (1994) introduced the concept of context-aware computing and described it as “Context-aware software adapts according to the location of use, the collection of nearby people, hosts, and accessible devices, as well as to changes to such things over time. A system with these capabilities can examine the computing environment and react to changes in the environment”. Hayes (1995) emphasized that like most other English words, meaning
of ‘context’ can only be properly understood when its context is made clear. Hayes identified four senses of context as physical context, linguistic/topical context, conceptual context and deductive context. Ryan et al (1997) and Brown et al (1997) defined context as the user’s location, environment, identity and time etc. Dey (1998) specified context as the user’s emotional state, focus of attention, location and orientation, date and time, and objects and people in the user’s environment.

Edmonds (1999) pointed that context arises from the study of the pragmatics of learning and applying knowledge. Thus, context only makes complete sense when there is transference of knowledge from point of learning to the point of application. It was emphasized that an identifiable context arises through the modelling of features that allow the recognition of a situation in which an inferential model can be applied. Therefore successful application of context may need combination of both artificial intelligence and machine learning techniques. Abowd et al (1999) viewed context as an interaction between the user and an application and defined it as "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the users and applications themselves”.

Lieberman and Selker (2000) defined context as “everything that affects the computation except the explicit input and output”. Dey (2001) recognized that the ability to enhance the behaviour of any application by informing it of the context of its use remains a challenge in the research field and referred context as the location, identity, and state of people, groups and computational and physical objects. Bazire and Br´ezillon (2005) examined 150 different definitions of context from different fields. Some determining factors in the definition of context were identified as the entity concerned by the context, its focus of attention, activity, situation, environment and eventually, an observer.

All these studies reveal that the concept of context is very broad and its definition varies according to the context of its applications. Therefore further study is restricted to the role of context in web search systems.
2.2 Context Aware Web Search Systems

Tim Berner-Lee (Berners-Lee, 1998) in 1990’s incepted the WWW or web1.0, with the notion to link anything to anything. The concept was initiated to provide uniform access to different sources of information independent of the data sources. This concept led to the development of static HTML websites with read-only content. Although a great invention in that era, it supported very limited interactions to the users e.g. navigating the web only through link directories of Yahoo! and DMOZ, e-mails, guest books, etc. The need to boost user interaction over web led to the evolution of Web2.0 also known as Social Web. Social web facilitated the users with read-write resources such as blogs, forums etc and ability to publish their personal multi-media content over web such as Flickr, YouTube etc. However, over the years increased interactivity of the web users’ created the problem of information overload due to which users are overwhelmed with abundance of information.

An initial attempt (Berghel, 1997) to solve this problem led to the development of web search engines. Lawrence (2000) pointed out that web search engines have become an important part of the society as they provide an easy and quick access to an unprecedented amount of information available on WWW. However, prevalent search engines treat search requests in isolation i.e. for a given query, results retrieved by these search engines are identical, independent of the user, or the context in which the user made the request. Thus “one size fits all” model of prevailing web search systems limit diversity and functionality of their output. This derives the need for next generation search engines that can augment search queries with contextual information. This information may be derived explicitly/implicitly or by implementing additional functionality within restricted contexts. Use of context in web search would help in enhancing the diversity on the web.

The concept of Semantic web/ Web 3.0 (Berner-Lee et al, 2001) focuses on developing techniques that would automatically extract information from web user interaction to provide only context relevant content for such users. Allan et al (2002) observed contextual web search as an important long term challenge and defined contextual retrieval as “Combine search technologies and knowledge about query and user context into a single framework in order to provide the most “appropriate” answer for a user’s information
needs.” Finkelstein et al (2002) expressed context as one of the most abused terms in the field, referring to a diverse range of ideas from domain-specific search engines to personalization. Goker and Myrhaug (2002) developed context taxonomy for web search in which context elements were divided into five main categories such as tasks, social, personal, spatio-temporal context and environment context. The task category refers to the goals, tasks or activities of the user. The social category focuses on the social aspects of the user e.g. information about friends and family or his role. Personal context aggregates mental and physical information about the user such as mood, expertise, disabilities etc. The spatio-temporal category includes attributes like time and location and the environmental context refers to user surroundings like things, people and information accessed by the user.

Ingwersen and Jarvelin (2005) presented an integrated model of context stratification for web search. It was anticipated that next generation of retrieval engines created, designed and developed using various models of context would be delivering performance better than that of out-of-context search engines. It was indicated that retrieval of information depends on time, place, history of interaction and task in hand. Such contextual data can be used effectively to constrain retrieval of information thereby reducing the complexity of the retrieval process. Kraft et al (2006) emphasized that contextual web search may represent the next major step in the evolution of web search engines. Mansourian (2008) identified five categories as the main context elements that can affect the search performance of end users. These categories include web users’ characteristics, types of search tools employed, search topic, search situation and features of the retrieved information resources.

Tamine-Lechani et al (2010) identified the main context specific dimensions (as depicted in Figure 2.1) that have been widely explored in contextual web search literature. Device context refers to a physical tool that provides the users with direct access to the information such as computer, ubiquitous wireless computer, mobile phone, PDA, etc. Spatio-temporal context incorporates time and location of the devices in addition. Task / Problem context focuses on the basic goal or intention behind the search activity such as fact-finding vs. exploration task (Navarro-Prieto et al, 1999), transactional, informational or navigational (Jansen, 2007) task. Document context originates from the Ingwersen (1994) principle of poly-representation of documents, expressing the cognitive view of information.
retrieval. User’s context is modelled as either the personal context or the social context. Personal context relates either to the physical context such as demographic context e.g age, gender, occupation, religion etc or cognitive/mental context that refers to the desired information context associated with the user’s search query. On the other hand, social context aims at leveraging the search according to the implied preferences of the user’s social community such as friends, neighbours, colleagues etc rather than only the individual. Such a context is of utility in social filtering systems to provide users with personalized recommendations based on their social community preferences. User context is one of the most widely explored facets in web search systems. The following sections explore contextual web search systems with reference to user’s personal and social context.

2.3 Web Search Architecture based on User’s Personal Context

Brin and Page (1998) presented basic architecture of a web search engine as shown in Figure 2.2. A search engine crawls web pages from the WWW, parses them, indexes and stores them in the local repositories. On submission of a user query indexed web pages are used for efficient retrieval. Thus the overall performance of a web search system is determined by the effective working of each component in the search engine.
Tamine-Lechani (2010) presented the basic framework of context aware web search system as shown in Figure 2.3. This framework decomposes the functionality of web search engine in two phases - context modelling and contextual retrieval.
In the first phase, context modelling is performed to appropriately define the context of user’s information needs, commonly known as search context. Subsequently in the next phase contextual retrieval is adapted considering users information selection process. The prevalent techniques used to perform context modelling and contextual retrieval are discussed in the next section.

### 2.4 Context Modelling

The primary techniques (Liu, 2011) used to model the user’s query context in web search engines are

- **Relevance feedback/Query Expansion**
- **Word Sense Disambiguation (WSD)**

#### 2.4.1 Relevance Feedback/Query Expansion

It was the preliminary mechanism adopted to improve the retrieval performance. According to Efthimiadis (1996), query expansion or formulation is a two stage process. Primarily, user constructs the query as an initial attempt to retrieve relevant information from the WWW. In the next stage query is reformulated either through interactive query expansion (IQE) or automatic query expansion (AQE). In IQE approach, search engines rely on user’s input as a feedback to separate the relevant documents from irrelevant ones e.g Rocchio feedback model (Rocchio, 1971). However, users often make poor term selections due to lack of knowledge (Belkin, 2000; Nanas et al, 2003). On the other hand, in AQE (automatic query expansion) approach, information for the query reformulation is implicitly derived by the system through global analysis or local analysis. The methodology involves construction of a matrix or thesaurus of relations between the terms in a global or local sense, to expand the original query with the best related terms. Term clustering, latent semantics indexing (Deerwester et al, 1990) and phrase finder (Jing and Croft, 1994) are all global analysis techniques. Though, these techniques are relatively robust, they consume a considerable
amount of computing resources. In addition, these techniques only focus on the document side and ignore the query side. Thus term mismatch problem is often left unsolved.

On the other hand, local analysis approach such as local clustering (Attar and Fraenkel, 1977) and local context analysis (Xu and Croft, 2000) use only the top ranked documents retrieved by the original query. This approach proves ineffective if the top-n documents lack appropriate terms to be added to the query. Parent et al (2001) developed a model which allowed the users to interact with a concept classification hierarchy to indicate the precise query context. Subsequently web agent uses portions of the hierarchy to expand the initial search query, thereby adding ‘user intent’ to the query. Sieg et al (2004) defined context by the portions of concept hierarchy (such as the Yahoo Directory) that match the user query. Each node of the concept hierarchy maintains a vector representation of the documents contained in the node and all its subcategories. Formerly accessed documents are clustered (an offline process) and the cluster centres form the user’s profile. When a query is issued, all clusters from the user profile having a similarity with the query above a pre-defined threshold level are selected. The selected clusters are then used to further refine the selection. Other approaches for context modelling include hierarchical classifications (Adomavicius and Tuzhilin, 2001), keyword-based vectors or vector classes (Gowan, 2003) and contextual graphs (Vallet et al, 2007).

2.4.2 Word Sense Disambiguation (WSD)

This approach to model the user’s search context endeavours to overcome the inherent ambiguity and incompleteness of the queries submitted by the users. Bates(1986) emphasized that "the probability of two persons using the same term in describing the same thing is less than 20%", and Furnas et al (1987) indicated that ‘the probability of two subjects picking the same term for a given entity ranged from 7% to 18%’. Thus, if the words in a document and the query are correctly disambiguated by matching their senses, performance of the systems may be improved by returning more relevant documents to the user. According to Small et al (1988) ambiguity may be syntactic or semantic. Syntactic ambiguity indicates that a particular word such as “table” can occur as noun as well as verb. On the other hand, semantic ambiguity refers to homonymy or polysemy based on the differences in meaning e.g. the bark of a dog versus the bark of a tree is an example of homonymy and opening a
door versus opening a book is an example of polysemy. Query ambiguity due to homonymy and polysemy is the fundamental problem underlying keyword based search engines.

Krovetz and Croft (1992) studied the relationship of sense matching between query and document. It was discovered that when queries are well formulated and precisely describe the information need, lexical ambiguity is less of an issue. Voorhees (1993) pointed out that performance of retrieval systems degrades due to the difficulty in disambiguating word senses in the short query statements as they provide very little context information to perform disambiguation. Thus, methods for query disambiguation must cope with short query statements in order to improve retrieval performance. Sanderson (1994) performed an insightful investigation into the effect of erroneous disambiguation on information retrieval performance. The introduction of “artificial” ambiguity into a document collection through the novel use of pseudo-words resulted in the reduction of retrieval accuracy to 75%. As a result, it was concluded that disambiguation accuracy needs to be at 90% or greater to observe an improvement in retrieval performance.

Jing and Tzoukermann (1999) computed words similarity through semantic distances between words using the local contexts of words within a single document and the lexical co-occurrence information in the set of documents. The methodology improved 11-point average precision by 8.6%. However, generation of context vectors and computing their distance along with the computation of term-term similarities through mutual information is computationally expensive. Gonzalo et al (1999) used manual annotations of IR-Semcor collection to establish that indexing using WordNet synsets brings significant improvements to text retrieval for large queries. This strategy proved better than Voorhees (1993) approach as it utilized both the synonym terms and the word senses. Leake and Scherle (2001) developed PRISM; a system to monitor and extract the description of the user's task context in order to predict the information likely of their interest. Subsequently search queries were dispatched to special-purpose search engines tailored towards the individual user's needs. Thus, underlying search engines determined the retrieval performance of the system.

Billhardt et al (2002) developed an indexing and retrieval method based on the vector space model with term dependencies to obtain semantically richer representations of documents. Results depicted that incorporation of term dependencies in the retrieval process...
performed statistically significantly better than the classical vector space model with IDF weights. However, the approach works well only for queries with many relevant documents whereas such queries represent usually more general or vague information needs. Liu et al (2004) utilized WordNet to determine the synonyms, hyponyms, words from its definition and its compound words for making possible additions to the query. Results established that WordNet based approach yields improvements between 23% and 31% over the best-known results on the TREC 9, 10 and 12 collections for short queries. Wen et al (2004) presented probabilistic model for contextual information retrieval to overcome the problems of incompatible context, noisy context and incomplete query. The query log was used as the key to build effective contextual retrieval model. However, analysing query logs to identify the user’s context indicates that users will not get relevant results in the first chance.

Kraft et al (2005) developed Y!Q an architecture of large-scale contextual search system with components like content analysis, query planning and rewriting framework and contextual ranking. For contextual web search, model allows users or publishers to specify context that can be used for automatically augmenting search queries. Though the approach seems good, system was evaluated for very small number of queries. Kraft et al (2006) investigated query rewriting (QR), rank-biasing (RB) and iterative filtering meta-search (IFM) approaches for contextual search. All three approaches worked well and IFM outperformed others both in terms of recall and relevance at an added engineering cost.

Vallet et al (2007) presented an ontology based approach to dynamically discard the out-of-context preferences of the users. However developing ontologies for the varied domains is a tedious task. Bozzon et al (2007) performed lexical analysis for query reformulation and used the variation in query clarity as well as the Part-Of-Speech pattern transitions to study user’s search actions. Efficacy of the approach was established over a log of 2.4 million queries. However developing generalized models of Part-Of-Speech patterns for millions of users is a cumbersome task. Wei et al (2009) presented context sensitive synonym discovery approach for web search queries based on co-click analysis. In addition, concept based synonyms through query segmentation were also devised. This approach significantly improved the accuracy from 40% to more than 80% as compared to thesaurus based synonym replacement method. The approach fails when some concepts are more popular than the others and the same clicks are performed with different intents.
All the above context modelling approaches are either too interactive and require more effort from the user’s to identify and indicate their search query context or ignore the user feedback and automatically identify the query context that may not be the instant context of the user’s query. For present day web users contextually relevant retrieval is most important.

### 2.5 Contextual Retrieval

It aims at delivering the right information to the user within the right context. Contextual retrieval is achieved by integrating the user’s context model in any of the retrieval steps- document re-ranking, document retrieval or by maintaining the contextual web repositories indexed by the search engines. The current size of world-wide web\(^1\) is approximately 50 billion pages and continues to grow. The unprecedented growth of Internet has resulted in considerable focus on Web crawling / spidering techniques. Crawlers are defined as “software programs that traverse the World Wide Web information space by following hypertext links and retrieve web documents by standard HTTP protocol” (Cheong, 1996). Crawlers are used to populate the local repositories with web pages to be indexed later using various ranking algorithms (Cho and Garcia-Molina, 2000).

Prevalent search engines use generic web crawlers that automatically traverse the web by downloading documents and following links from page to page. Thus, these search engines cover only the surface web or “publicly indexable Web” (Lawrence and Giles, 1999) whereas large portion of the WWW is dynamically generated. Such content requires the users to fill out forms and submit the queries. This covert side of the WWW is popularly known as the Hidden/Deep/Invisible Web. Bergman, (2001) pointed that the invisible web contains 400 to 550 times more information as compared to the surface web. Hidden web content is often stored in specialized databases (Lin and Chen, 2002). For example, the IMDB movie-review database contains a plethora of useful information regarding movies. However, standard crawlers have limited capabilities to access this information (Sizov et al., 2003). Thus, retrieval of information from the web should be from both the dimensions; surface and hidden. Thesis employs both the surface and deep web crawlers to populate the local repositories.

---

1. [http://www.worldwebsize.com](http://www.worldwebsize.com)
2.5.1 Deep Web Crawler

Studies have revealed that size of deep web is very large as compared to the surface web. Bergman (2001) discovered that size of only 60 popular deep web sites together is 40 times larger than the known surface web. He et al (2007) exposed that deep web constitutes 307,000 sites, 450,000 databases, and 1,258,000 interfaces. It covers broad range of high quality articles, research papers etc which have immense utility for diverse categories of users like academicians, librarians and scholars etc. The volume of the hidden web is estimated to increase more rapidly with the advancements in web technology trends. As more web applications are developed using server-side scripts like java server pages(JSP), active server pages(ASP/ASP.net) and hypertext pre-processor (PHP) etc and databases like MySql, Oracle etc, amount of hidden web content will grow more. However, traditional search engines often deprive the users from the deep/dynamic web content because it requires locating the entry points to deep web source i.e. database driven websites with search interfaces, to automate the form filling task and generate event to submit the form. Thus the research in deep web is mainly focused on overcoming the challenges posed by these systems.

Raghavan and Garcia-Molina (2001) introduced ‘HiWeb’ architecture, a task-specific approach to crawl the hidden web pages. However the architecture had limited capability to recognize and respond to dependencies between multiple search interfaces and for processing the forms with partial values. Cope et al (2003) developed a decision tree based technique to automatically discover the web pages with search interfaces. The algorithm incorporated long rules and large number of features in training samples which may lead to over-fitting. Lage et al (2004) presented a technique for collecting hidden web pages for data extraction based on web wrappers. However this technique was limited to access only the web sites having common navigation patterns. A technique for understanding web query interfaces through best effort parsing with hidden syntax was proposed by Zhang et al (2004a). Fontes and Silva (2004) developed SmartCrawl strategy to automatically generate the queries to crawl the deep web pages. Smartcrawl had limited capabilities to analyse the web pages, thus it also indexed the web pages with no results or errors. Ntoulas et al (2005) developed a strategy for automatic query generation to excavate the high quality gold mine of deep web data.
The approaches to automatically fill the forms were presented by Caverlee et al. (2004) and Xiang et al. (2008). Alvarez et al. (2007) presented an approach to automatically fill and submit only the domain specific forms of deep web databases. Barbosa and Freire (2007) presented focused crawlers to automatically learn patterns of promising links with hidden web pages, to reduce the manual interference. However, the study ignored filtering of web pages with ‘custom’ search or ‘site’ search forms. Retrieved forms were highly heterogeneous and it could be time consuming to train the link classifier. Xu et al. (2007) achieved good precision and recall using ontology-based deep web classification which included a category ontology model and a deep web vector space model (VSM). However, there are no standard ontologies available for varied categories of deep web applications.

Madhavan et al. (2008) presented techniques for indexing deep websites with multiple search interfaces. A random forest algorithm based on weighted feature scheme to find search interfaces was proposed by Deng et al. (2008). The algorithm seemed highly sensitive towards the changes in training data set. Ye et al. (2008) improved this algorithm by adding one more classifier. Though the accuracy of their system is good, certain assumptions made by the authors with respect to “get” and “post” methods are not valid. Zhang et al. (2010) developed a form-based crawler to gather the pages with at least one search form. Nevertheless this approach involved great deal of human intervention.

2.5.2 Surface Web Crawler

Due to the dynamic nature of Web and limited bandwidth, storage, and computational resources, search engines cannot index every web page and even the covered portion of the web cannot be monitored continuously for changes (Menczer, 2000). Therefore more sophisticated crawling approaches such as focused web crawling have become trendier. Focused crawling is designed to traverse the promising links that lead to target pages and avoid off the topic pages. Chakrabarti et al. (1999) presented a “focused crawler” that targeted on the retrieval of web pages relevant to a particular topic. The methodology employed a naïve Bayesian classifier and ODP web taxonomy to classify and extract the web pages with desired topics of interest. Diligent et al. (2000) pointed that the major problem in
focused crawling is to perform appropriate credit assignment to different documents along a crawl path. This ensured that short-term gains are not pursued at the expense of less-obvious crawl paths that ultimately yield larger sets of valuable pages. To solve this problem, a focused crawling algorithm to build the context model of topically relevant pages was developed. The approach performed better than standard focused crawling approach.

Aggarwal et al (2001) developed an intelligent crawling technique based on self-learning mechanism, to dynamically adapt the specific structures of the relevant predicates. The ability of learning crawler to reuse the information learned from a given crawl helped in improving subsequent crawls. Chakrabarti et al (2002) modified the focused crawling approach by assigning better priorities to the unvisited URLs in the crawl frontier. Using topics from Dmoz, the approach cut down the fraction of false positives by 30-90%. Bergmark et al (2002) incorporated ‘tunneling’ technique in best-first focused crawler approach. The results established that combination of focussed crawling with tunnelling could be an effective tool for building digital libraries. Angkawattanawit and Rungsawang (2002) developed an approach that utilized seed URLs, topic keywords and URL relevance predictors build from previous crawl logs. The approach improved harvest rate. Sizov et al (2003) presented BINGO!, a focused crawling approach that identified the positively identified documents of a topic from the crawled dataset and utilized it to re-train the classifier periodically. An ontology-based approach to compute the page relevance for focussed crawling was developed by Ehrig and Maedche (2003). Methodology improved the harvest rate as compared to the keyword based focused crawler. Chau and Chen (2003) investigated algorithms to improve the performance of vertical search engine spiders. The study addressed three approaches- breadth-first graph-traversal algorithms with no heuristics, a best-first traversal algorithm that used a hyperlink-analysis heuristic and a spreading-activation algorithm based on modelling the web as a neural network. Results confirmed that Hopfield Net and BFS spiders both scored significantly better than the PageRank spider.

Yunming et al (2004) developed iSurfer that used an incremental method to learn the page classification model and a link prediction model. Experimental results on various topics demonstrated that incremental learning method improves harvest rate with a few initial
samples. Other focussed crawlers include online-deal forums (Yih et al, 2004), web blogs (Glance et al, 2005), web forums (Limanto et al, 2005), recipe crawler (Li et al, 2006) and Board Forum Crawler (Guo et al, 2006) etc. Yuan et al (2007) presented domain specific resource discovery approach by integrating the various existing approaches and evaluated it with respect to breadth first and best first search approaches over the music, education sports and arts domains. Shen (2008) developed an ontology-based adaptive topical crawling technique to improve the correlativity of the topical web crawler. The approach employed ontology categorizer to evaluate the topic correlation of web pages and extracted more topical features to enrich characteristic set. The experiments portrayed that algorithm improved the topic crawling performance.

Pal et al (2009) developed focused crawling based on content and link structure analysis. The methodology allowed the crawler to go through several irrelevant pages to get to the next relevant one when the current page is irrelevant. The approach performed better than the BFS crawler over E-Business, Nanotechnology, Politics, and Sports topics. Batsakis et al (2009) presented a novel learning crawler based on Hidden Markov Model (HMM). Results revealed that all crawlers achieved their maximum performance when a combination of web page content and (link) anchor text was used for assigning download priorities to web pages. It was observed that the new HMM crawler improved the performance of the original HMM crawler and also outperformed classic focused crawlers in searching for specialized topics.

Most of the focused crawlers discussed above employed local search algorithms such as Breadth-first Search and Best-first Search to decide the order in which the targets URLs are visited, which can result in low-quality domain-specific collections (Qin et al, 2004 and Chau et al, 2005). The limitation of local search algorithms is that they traverse the web space by visiting the neighbours of previously visited web nodes. Thus conventional focussed crawlers miss out relevant web pages if there is no chain of links that connects some starting page to that relevant page. Furthermore, the focused crawlers give up crawling in a direction before reaching the final target pages if there existed some irrelevant web pages between the start page and the target page. As a result, local search algorithms based focused crawlers are able to find relevant pages only within a limited sub-graph of the web that surrounds the
starting URL set and miss any relevant web pages outside this sub-graph. To deal with this challenge, global search algorithms such as nature inspired algorithms are explored for building domain specific collections.

2.5.3. Nature Inspired Algorithms based Web Crawlers

Nature inspired algorithms are a branch of artificial intelligence techniques that mimic the metaphor of biological evolution. These algorithms are categorized into evolutionary algorithms that are inspired by the notion of ‘survival of the fittest’ and the swarm intelligence techniques that are inspired by the social behaviour of swarms. Bra et al (1994) developed the first crawling algorithm inspired by nature - the Fish-Search algorithm. The algorithm begins with a set of seed web urls relevant to the query and collects more pages by following the web pages that best match the original keyword query. To identify the potential set of relevant web documents binary priority values were assigned. However, using binary approach, all relevant pages were assigned the same priority values. Thus using Fish Search algorithm it becomes difficult to determine the appropriate cut off levels apriori, possibly resulting in load unfriendly search behaviours. To overcome these limitations, Hersovici et al (1998) developed Shark-Search algorithm for web crawling. The approach used vector space model for assigning non binary priority values to candidate pages. The priority values were computed by taking into account, page content, anchor text, text surrounding the links and the priority value of parent pages. Shark search algorithms performed better than the Fish search algorithm in downloading more relevant documents.

Chen et al (1998) developed an agent-based Itsy Bitsy spider using genetic algorithm technique. The spider was given user defined seed web pages as input to crawl the related web pages based on the links and keyword indexing. Based on user evaluation, the genetic algorithm spider obtained higher recall value than that of the local search algorithm based best-first search spider. However, values were not statistically different. Rennie and McCallum (1999) used reinforcement learning for focussed crawling. The approach measured the probability of a hyperlink being linked to a relevant page based on the texts in the hyperlink neighbourhood. Subsequently the crawler followed the link with the highest probability value. InfoSpider also known as ARACHNID (Adaptive Retrieval Agents Choosing Heuristic Neighbourhoods’ for Information Discovery) is another focused crawler
developed using genetic algorithm (Menczer and Belew, 2000). InfoSpider consists of a set of computation units with single genotypes that differ considerably from the Itsy Bitsy spider. The neural network and the other components of the genotype made it possible for a single agent to autonomously determine and adapt its behaviour according to both the local context of the search and the personal preferences of the user.

Srinivasan et al (2002) introduced the idea of web crawling agents and used InfoSpider to crawl the web pages focused in the field of biomedical information. The approach performed better than conventional focussed crawlers. Menczer et al (2004) analyzed the computational complexity of prevalent crawling techniques such as Breadth-First, Best-First, PageRank, Shark-Search, and InfoSpiders. It was observed that evolutionary crawler such as InfoSpider was extremely efficient and scalable than conventional techniques and was no more expensive than the blind search algorithms such as Breadth-First. InfoSpider achieved the best performance/cost ratio whereas greedy algorithms such as Best-First search required large amount of time to sort through candidate links and were less scalable. Shokouhi et al (2005) developed an intelligent crawler namely Gcrawler that used genetic algorithm for improving its crawling performance. Gcrawler estimated the best path for crawling and used genetic algorithm to expand its initial keywords set during the crawling. This was the first crawler that acted intelligently without any relevance feedback or training.

Dziwiński and Rutkowska (2008) developed an approach for hypertext graph crawling using ant colony algorithm. Using an ant as an agent in a hypertext graph significantly reduced the downloading of irrelevant documents to download a given number of relevant documents. The approach performed better than Shark-Search crawling algorithm and best-first search strategy utilizing a queue for the central control of the crawling process. Zhang et al (2009) presented a multiagent system framework based on the idea of ants to perform focused crawling. However the performance of the system was not evaluated. Yohanes et al (2011) observed that Genetic algorithm based focussed crawler traverses the web search space more comprehensively than traditional focused crawler over education, computer and sports domains.
The above discussed focused crawlers though reduce the overheads of traversing irrelevant web pages to some extent, they still have a common limitation; they need to perform broad search before actually getting the desired web urls. Thus, to eliminate the need to crawl WWW, meta-search techniques have been developed as discussed in the next section.

2.5.4 Web Meta-Search

In this digital era search engine websites like Google, Bing and Yahoo! etc are guiding the whole web community. Since inception to this date, these search engines have indexed billions of web pages from the internet through their sophisticated generic crawlers, thus formulating biggest web repositories. Therefore large number of meta-search approaches have been developed that take advantages of conventional search engines. Meta-search engines do not maintain any local repositories of their own. Instead, they take a user’s request and dynamically pass it to varied heterogeneous databases and compile their results to be presented to the user. The primary advantages of a meta-search engine over a single search engine are increased coverage and a consistent interface e.g. ‘MetaCrawler’ (Selberg and Etzioni,1995) softbot is one of the earliest meta-search engine to fetch web urls from Galaxy2, InfoSeek3, Lycos4, Open Text5, WebCrawler6 and Yahoo7 web search engines. In order to increase the precision of the meta-search results, Gauch et al (1996), Howe and Dreilinger (1997) and Benitez et al (1998) developed flexible meta-search engines such as ProFusion, SavvySearch and MetaSEEK respectively. These meta-search engines employ distinct techniques to identify the search engine for submitting the user query e.g ProFusion considers the performance, the predicted subject of the query, and the user’s explicit search engine preferences. MetaSEEK utilizes past results and the user’s keywords for source selection, while SavvySearch allows users to specify a “category” to determine the sources to be searched. Lawrence and Giles (1998) presented architecture of a meta-search tool ‘Inquirus’ that focused on making certain user preferences explicit. These preferences defined a search strategy for source selection, query modification, and result scoring features to understand the user’s needs.

2. http://galaxy.einet.net/galaxy.html
All the meta-search tools eliminate the need to perform multiple searches over varied search engines; yet they undergo the limitations of underlying search engines. Meta-search tools have limited capabilities to handle contextual information needs of the users and index the deep web. It is predicted (Spivack, 2007) that future web -WWW 4.0 will be the amalgamation of artificial intelligent techniques and agents i.e. intelligent web. Therefore current research is focused to develop intelligent contextual web search agents that attempt to learn the context sensitive information needs of the users with minimum user interaction and thus provide context relevant results from both the surface and the deep web.

### 2.6 Intelligent Web Search Systems

Since inception to this date, growth of computing has been marked by the important and continuous trends like ubiquity, interconnection, intelligence, delegation and human-orientation (Wooldridge, 2003). These trends present major challenges for software developers. With the trends like ubiquity and interconnection, modern computing platforms and information environments are no longer stand-alone systems. They have become tightly connected both with each other and their users’ realizing the idea of global computing e.g Internet, World Wide Web etc. In the present WWW, application programs are not limited to simple input–compute–output kind of operational structure. They show ‘reactive’ behaviour that is they maintain a long term, ongoing interaction with their environment; they do not simply compute some function of an input and then terminate e.g. online banking systems, web servers, computer operating systems, process control systems etc. From the software development point of view, “reactive” systems are much harder to engineer efficiently as compared to simple functional systems. These systems are known as agents. Agents are defined as the computational entities such as a software programs or robots that act autonomously on behalf of their users, across open and distributed environments like World Wide Web, to solve the growing number of complex problems.

Wooldridge and Jennings (1995) defined agents as “An agent is a computer system that is situated in some environment and that is capable of autonomous action in this environment in order to meet its design objectives”. Weiss (1999) describe information agents as “Agents that have access to multiple, potentially heterogeneous and geographically
distributed information sources. Information agents have to cope with the increasing complexity of modern information environments and retrieve, analyze, manipulate and integrate information from different sources.” Intelligent agents are increasingly being used for modelling rational behaviours in a wide range of distributed applications. Wooldridge and Jennings (1995) indicated that an intelligent agent is one that is capable of flexible autonomous action in order to meet its design objectives where flexibility means three things such as reactivity, pro-activeness and social ability. Reactivity implies the ability of intelligent agents to perceive their environment, and respond in a timely fashion to changes that occur, pro-activeness means the capability of intelligent agents to exhibit goal-directed behaviour by taking the initiative and social-ability refers to the ability of intelligent agents to interact with other agents (and possibly humans) in order to satisfy their design objectives.

In practice, ‘single agent systems’ are rare. Majority of the problem solving environments contain multiple agents, formulating a multi-agent system. A multi-agent system (MAS) is a loosely coupled network of software agents that interact with each other to solve problems that are beyond the individual capacities or knowledge of each problem solver. In a multiagent system, agents interact by exchanging messages through some computer network infrastructure. For successful interactions, agents imbibe the ability to cooperate, coordinate and negotiate with each other, in much the same way as humans cooperate, coordinate, and negotiate with other people in their everyday lives.

2.6.1 MultiAgent Systems for Web Search

Multi-agent systems have been used in wide variety of applications ranging from relatively small systems for personal assistance to open, complex, mission critical systems for industrial applications (Jennings and Wooldridge, 1998). Internet is perceived as the perfect domain for multiagent systems due to its intrinsically distributed nature and the amount of information available. The distributed information retrieval task deals with the collection of information from multiple and usually heterogeneous information sources that exist in a distributed environment such as World Wide Web.
Clark and Lazarou (1997) designed a multiagent system to cope with the problem of distributed information retrieval. The system employed task specific agents such as user agent, extractor agent and the information source agent, information supply facilitator agent, information request facilitator agent, the matchmaker agent and the descriptor agent. The extractor agent detects web pages with relevant information, extract and represents this information in an attribute-based format. The user agent transforms user query into a KQML-style message structure to allow easy inspection and transformation by other agents. The transformed query is then sent to the corresponding information request facilitator. The information request facilitator agent receives the queries from user agents and attempt to find potential information support facilitators to answer the query. Thus information request facilitator agent performs query reformulation, consult and subscribe to the matchmaker agent and metadata caching. The matchmaker agent on receiving the attribute value coordinates with the descriptor agents to place topic in the topic hierarchy.

Moukas et al (1998) presented Amalthaea - a multiagent ecosystem to assists users cope with information overload on the World Wide Web. Amalthaea represents an artificial ecosystem of evolving agents - the information filtering (IF) agent and information discovery (ID) agent that cooperate and compete. The information filtering agents track interests of the users for personalization. Information discovery agents perform information resource handling, adaptation to information sources, and find and fetch the actual information with respect to user’s interests. Choi et al (1999) presented a multiagent learning approach for information retrieval on the web in which each agent collaboratively learned its environment from user's relevance feedback using a neural network mechanism. Rhodes and Maes (2000) developed JITIR (just-in-time information retrieval) agents to proactively retrieve and present information based on a person’s context. JITIR agents provided information from an unordered set of documents and did not require any hand-coding of documents or special domain-dependent techniques.

McDermott and O'Rierdan (2002) developed a multi-agent information trading system, a suitable test-bed for studying agent cooperation, coordination and negotiation. Shaban et al (2002) presented cooperative multiagent system to overcome the pitfalls of current web-based information retrieval tools. It consists of a distributed group of intelligent
agents that perceive users’ desires, set goals, exploit all efforts to achieve these goals, filter and customize results and present them in appropriate formats. Pant and Menczer (2002) developed a web intelligence tool called MySpiders, a threaded multiagent system designed for information discovery from the Internet. It was established that augmenting search engines with adaptive intelligent search agents leads to significant competitive advantages.

Chen and Yang (2003) presented multi-agent hybrid learning approach to enhance personalized information filtering. System employed four agents - user agent that acts as an interface agent between the user and the system for all interactions. Information retrieval agent to retrieve web-documents from search engine using queries created from user profile. Clustering agent to classify the retrieved web-documents into several clusters and learning agent to update user profile based on user’s behaviour information. Mirikitani and Kushchhu (2003) presented E.Coli search - a prototyped multi-agent information retrieval system for the World Wide Web. E.Coli system simulated intelligent behaviour at the local level by evaluating the relevance of the document with respect to the users query and at the global level through rapid reproduction leading to invasion of search agent in information rich areas of the search space.

Enembreck et al (2004), developed MAIS (Multi-agent system for Internet Search), an agent based system for searching the web. MAIS architecture involved Personal Agent (PA), Library Agent (LA), Filter Agent (FA) and Search Agents (SA). PA acts as an interface between the users and performs management of user’s favourite sites in cooperation with LA and management of search services in cooperation with FA and SA. MAIS improved accuracy as compared to standard search engines. Herrera-Viedma et al (2004) presented a Fuzzy Linguistic multi-agent model for information gathering from the web based on collaborative filtering techniques. Multi-agent model involved four cooperative agents- interface agent, collaborative filtering agent, task agent and information agents. Interface agent acts as a mediator between the user and the collaborative filtering agent. This agent communicates the user query, information need category and user identity to the collaborative filtering agent. Subsequently collaborative filtering agent communicates the user query to the task agent, fetches more relevant documents from the task agent and generates recommendations based on the information need category expressed by the user.
Zhang et al (2004b) developed multi-agent system for peer-to-peer (P2P) network-based information retrieval (IR). The approach presented a mediator-free multi-agent framework incorporating the initial formation and reorganization of agent-view structures as well as the context-sensitive searching algorithms. The approach demonstrated significant increase in information retrieval system performance. Zhang and Lesser (2006) further extended this infrastructure to handle multiple and concurrent search sessions.

Bañares-Alcántaraa et al (2005) presented MASH (Multi-Agent Search Engine) that employed domain ontology to search the relevant web pages. MASH involved one user agent (UA) to interact with the user, n internet agents (IA) to access, retrieve, rate and filter the information from the web, one weight agent (WA) to support the search and supply alternative queries when requested and one coordinator agent (CA) to coordinate the overall process, particularly during splitting and amalgamation of the tasks performed by the IA. MASH was implemented using JADE multiagent environment. Woerndl and Groh (2005) designed “Cairsweb” (Context-Aware Information Retrieval in the Semantic Web) to provide personalized services based on user context. The “Cairsweb” architecture included personalization agent, context agent, identity agent and search agent. The personalization agent behaved like “Semantic Web browser” to manage user’s information needs. Context agent managed the user’s dynamic and transient information that includes current physical location, last search request, current devices, possibly virtual location etc. Identity agent managed static information about the user e.g. the user profile that does not change very often. The Search agent proactively matched the user’s information needs with the information provided by the information source.

Lhotska and Prieto(2007) designed a multi-agent system for supporting medical diagnostics and monitoring in the area of cardiology (the ADAM system). The system employed two kinds of agents- Builder agent and Searcher agent. The various activities performed by the builder agent included loading of graphical user interface to receive input from the user, to display the result generated in the process, build ontology and perform creation, communication and analysis and destruction of the searcher agent. Searcher agents searched the articles from PubMed as per the class assigned by the builder agent and sent them back to the builder agent. Addis et al (2008) presented ‘WIKI.MAS’ a generic architecture for multiagent information retrieval. System encompassed multiple software
agents for information retrieval like information agent, filter agents, task agents, interface agents and middle agent etc to perform the various tasks. Ivanova and Momtchev (2010) developed a multiagent system for knowledge rich personalized information retrieval. System involves query linguistic analysis agent (QLAA) to analyze and perform initial query refinements of the user query using WorldNet and send the refinements to the coordination agent (CA). CA sends request to Hits Agent (HA) for delivery hits of existing variants and requests Ontology Based Enrichment Agent (OBEA), Thesaurus-Based Enrichment Agent (TBEA) and Corpus-Based Enrichment Agent (CBEA) to get additional query refinement proposals. Subsequently, coordination agent returns estimated proposals to the searcher, to ascertain the clear and unambiguous expression of information needs.

Addis et al (2012) developed ‘XMAS’ a generic multiagent architecture to retrieve, filter and reorganize information according to user interests. During this project it was observed that multiagent technology is quite effective to design and realize concrete information retrieval applications. Birukou et al (2012) developed ‘Implicit’ a multi-agent recommendation system to support web search for groups or communities of people. The model used implicit knowledge observed from the search behaviour of groups of users to improve web search for a community of people with similar but specific interests. In Implicit, agents observe behaviour of their users to learn about the “culture” of the community with specific interests. ‘Implicit’ facilitates sharing of knowledge about relevant links within the community by means of recommendations. The agents also recommend contacts, i.e., who in the community would be the right person to contact for a specific topic. Implicit approach improved the quality of the web search in terms of precision and recall.

These studies established the suitability of multiagent systems for web search applications. Multiagent systems provide various advantages like scalability, stability and load balancing. Some tasks like learning profiles or searching the web can be very hard indeed. Agent coordination protocols tend to distribute the work between different agents running in different processes, threads, machines or networks uniformly and naturally. Using multiple agents the system can be updated or expanded without affecting the normal operations. Since agents are independent systems, they can be created and modified
individually while other agents continue to provide services. Whenever an agent crashes, other agents can coordinate the distribution of services and ensure their execution.

However, the prevalent multiagent systems discussed above focus on enhancing the retrieval from only one dimension of web; surface web. The users while searching for information on the internet presume complete access to the web and provision of context relevant results. Hence it becomes the responsibility of a search engine to satisfy the user in this respect.

### 2.6.2 MultiAgent Architectures

Agent architectures are the fundamental mechanisms underlying the development of autonomous components such as agents that support effective behaviour in real-world, dynamic and open environments. According to Wooldridge and Jennings (1995), various multiagent architectures are as follows:

- **Logic-Based Architecture** It employs the "traditional" approach i.e. symbolic AI to build artificially intelligent systems. These systems take symbolic representation of its environment and its desired behaviour and then syntactically manipulate it to illustrate intelligent behaviour. Since human knowledge is symbolic in nature, it becomes easier to encode. However, developing accurate description of the real world problems is a tedious task.

- **Reactive Architecture** It is based on the notion that intelligent decision making is inherently connected to the environment an agent occupies. Intelligent behaviour is represented as a direct mapping of situation to action and is a result of the interaction the agent maintains with its environment. This architecture is suitable for dynamic environments and is simpler in design than logic-based agents. However, reactive agents do not employ models of their environment which makes it difficult to design agents that learn from experience.
Layered (hybrid) Architecture This architecture supports both reactive and deliberative agent behaviour by arranging subsystems as the layers of a hierarchy such as horizontal layering (Ferguson, 1991) and vertical layering (Müller, 1994). In horizontal layering, all layers are directly connected to the sensory input and action output acting like an agent. Although simplicity of design is it major advantage, its complexity is high due to the large number of possible interactions between horizontal layers—\(m^n\) (where \(n\) is the number of layers and \(m\) is the number of actions per layer). Vertical layer architecture eliminates some of these issues as the sensory input and action output are each dealt with by at most one layer each. This mechanism reduces the number of interactions between layers to \(m^2(n-1)\). Nevertheless architecture depends on all layers and is not fault tolerant, so if one layer fails, the entire system fails.

Belief-Desire-Intention Architecture It is one of the most popular agent architectures in which intelligent behaviour is determined by the manipulation of data structures representing beliefs, desires, and intentions of the agent (Rao and Georgeff, 1995). The architecture has its roots in philosophy to understand practical reasoning which refers to the task of deciding the course of actions in order to attain the goals. The various multiagent systems developed using BDI architecture include Procedural Reasoning System (PRS) (Georgeff and Lansky, 1987), JAM (Huber, 1999), dMARS (d’Inverno et al, 1998), JACK (Howden et al, 2001) and JADEX (Pokahr et al, 2005).

2.6.3 Tools for Building MultiAgent Systems

Programming languages, platforms and development tools are important components that affect the dissemination and use of agent technologies across different application domains. The success of multi-agent systems is largely determined by the availability of appropriate technologies to implement the concepts and techniques underlying multi-agent systems.
Yoav (1993) developed Agent-Oriented Programming (AOP) to directly program agents in terms of mentalistic notions such as belief, desire and intention. AOP models an application as a collection of components called agents that are characterized by autonomy, pro-activity and an ability to communicate. Bordini et al (2006) reviewed several agent-oriented languages that are available today.

The various approaches used to develop the agent oriented programming languages include declarative languages (that are partially characterized by their strong formal nature, normally grounded on logic), imperative approaches (e.g. JACK (Evertsz et al, 2004)) and hybrid approaches (e.g. 3APL (Dastani et al, 2004), Jason (Bordini et al, 2005), IMPACT (Subrahmanian et al, 2000) and AF-APL (Agent Factory Agent Programming Language) (Yaov, 1993)). Various programming languages provide integrated development environments (IDEs) to simplify and enhance the productivity by automating tedious coding tasks e.g. Jason is distributed with an IDE that provides a graphical interface for editing system configuration file and AgentSpeak code for the individual agents. The JACK Development Environment (JDE) is a full featured commercial IDE for the JACK BDI agent platform (Winikoff, 2005) developed by Agent Oriented Software Ltd. It is based on the JACK Agent Language (JAL). AgentBuilder is another agent platform directly based on Agent-Oriented Programming (AOP), originally defined by Shoham (Yaov, 1993) and developed by Acronymics Inc. Other platforms to develop multiagent systems include TuCSon (Tuple Centre Spread over the Network) (Omicini and Zambonelli, 1999), JADE (Java Agent DEvelopment Framework) (Belifemine, Caire and Greenwood, 2007) etc.

This work presents a model of Contextual Web Search based on BDI Multiagent architecture in the next chapter. The model primarily attempts to identify the contextual information needs of the users and subsequently performs contextual retrieval from both the surface and deep web. Many off the shelf conventional crawlers could be utilized for contextual retrieval from the WWW. However these crawlers have limitation that they need to perform broad search before getting the desired web urls. Previous studies established the superiority of nature inspired algorithm based focussed crawlers over conventional crawlers. The following chapter compares various nature inspired algorithms for their suitability in development for surface and deep web meta-crawlers.