Efficient Framework for Semantic Search on Web

A Thesis submitted

for the award of degree of

Doctor of Philosophy

By:

Vikas Jindal

(950903011)

Under the guidance of:

Dr. Seema Bawa, Professor,

Dr. Shalini Batra, Associate Professor,

Computer Science and Engineering Department,

Thapar University, Patiala-147004, INDIA

Computer Science and Engineering Department

Thapar University, Patiala – 147004, INDIA

July, 2016

Certificate

I hereby certify that the work which is being presented in this thesis entitled "**Efficient Framework for Semantic Search on Web**" in partial fulfillment of the requirement for the award of degree of "**Doctor of Philosophy**" submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of Dr. Seema Bawa and Dr. Shalini Batra, refers other research works which have been duly listed in the reference section. The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.

(Vikas Jindal)
Regn. No. 950903011

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

27/July/2016

(Dr. Seema Bawa)
Professor,
Computer Science and Engineering Department,
Thapar University,
Patiala, 147004 (INDIA)

(Dr. Shalini Batra)
Associate Professor,
Computer Science and Engineering Department,
Thapar University,
Patiala, 147004 (INDIA)

Acknowledgment

I would like to express my sincere gratitude to many people who have provided valuable support and guidance in one way or the other during my PhD work, especially **Prof. Seema Bawa**, my principal supervisor, for having faith in me and providing me an opportunity to work under her esteemed guidance. She has been a great source of inspiration for helping me develop my ideas, working patiently on my writing and helping me see the larger perspective. Her short and crisp advises helped me a lot in developing and realizing my research skills along with day-to-day activities of life. The acquired learning during the process of research work with Prof. Bawa as supervisor will always be of immense help in almost every endeavor of life.

I would like to thank **Dr. Shalini Batra,** my supervisor, for all the discussions, reviews and supportive words along the way. She was incredibly supportive and helped move this research in interesting directions. Her positive attitude helped to boost the morale at times when the spirit was low. Many thanks to my doctoral committee members, **Dr. Maninder Singh**, Associate Professor, Computer Science and Engineering Department, Thapar University, **Dr. Inderveer Channa**, Associate Professor, Computer Science and Engineering Department, Thapar University and **Dr. Rajesh Kumar**, Associate Professor, Computer Science and Engineering Department, Thapar University for their valuable insights and regular support in ensuring the progress of my research work.

I am also indebted to Apeejay Stya University for providing me all the support and healthy environment necessary for pursuing my research endeavors. **Dr. Sarbjit Singh,** Honorary Advisor, School of Engineering and Technology, Apeejay Stya University has always been a great source of support and encouragement throughout my research work.

I would express my appreciation especially to my wife, **Kavita** for being remarkably patient and supportive throughout my research work, for putting up with countless hours I spent on my thesis work and for being there when I needed her.

My brother, **Lalit** has always been a great source of moral support during all the ups and downs faced during this journey of research work.

Finally, I am deeply beholden to my parents, **Sh. Nehru Lal Jindal** and **Smt. Usha Jindal** for their immeasurable love and unconditional support. They provided me plenty of freedom to explore the directions I was interested in. This thesis is dedicated to them.

To my parents,
Sh. Nehru Lal Jindal and Smt. Usha Jindal

An investment in knowledge always pays the best interest.
- Benjamin Franklin

Contents

List of Figures

List of Tables

Abstract

With frequent and faster growth of the Web and dependence on the Web for relevant information retrieval, search engines have become the most popular and powerful tool for accessing desired information online. However, it is observed that the Web pages returned by even a renowned search engine are not so accurately useful. The necessity of finding the most relevant information has given rise to the research in the field of semantic search. Traditional Web search methods where basic relevance criteria rely primarily on the presence of query keywords within the returned pages are required to be replaced with more effective semantic search techniques.Semantic based search would be able to provide users a more intelligent form of finding what they are looking for within the global source of information available online.

In this thesis, various approaches for semantic based search on Web have been studied and analyzed resulting in the identification of two broad perspectives of semantic search as elaborated in the chapter on literature review. Fundamental limitations identified in the existing approaches have been major motivation for proposing efficient semantic based search approach. Later a framework for **QU**ery-context based **I**nformation retrieval using **C**orpus **K**nowledge (QUICK) is proposed which has been elaborated in the chapter on proposed framework. Here the Web pages returned by a baseline system in response to original query are used to generate a corpus of words related to the query category. The word tokens which are laying in the close proximity of the query keywords are supposed to be semantically related to the original query. The relative positioning and frequency of the words with respect to the query word is assigned due importance using probabilistic feature of the proposed approach which in turn ensures to have greater probability in reaching to the context of the query. The approach shows the possibility of generating a set of context features in an efficient manner in order to produce a more accurate model of the query topic. This context oriented semantic search approach has been implemented using an open source library of language processing features, NLTK and integrating it with Python language interpreter. The elaborations have been presented in the chapter on design and implementation of QUICK. Category specific user query is entered to a standard search engine in order to retrieve most relevant documents pertaining to that domain. The top-ranked returned documents are stored and techniques are applied for filtering non-lexical tokens like stop-words, non-alphabetic strings. The words laying in the close proximity of the

query keywords are extracted to be used as context vector. The strength of association of the context vector features to the category is calculated and presented in the form of a list. A set of features having best strength of association to the category are selected and treated as the context features of the category to be used for the semantic expansion of the query pertaining to that category. The experiments for the comparison of result set precision of the proposed QUICK based semantic search and the standard keyword based search have been performed and elaborated in the chapter on testing and validation. The proposed semantic based search approach has witnessed a significant improvement over the standard keyword based approach. Finally, the findings of the entire thesis have been concluded along with the potential scope for future directions in the said domain.

Chapter 1

Introduction

On-Line Information Retrieval (IR) has been a field of extensive research for more than half a century. The term was first coined by an American computer scientist Calvin Mooers long back in 1950 [1]. With the advent of World Wide Web in early 1990s, the Web today has become a wide spread movement that has a great influence not only on commerce but also on the social life of the people across the world. The huge repository of data in the form of static and dynamic Web pages has started giving a reflection of society in terms of its behavioral patterns and its impact on global economy. The proliferation of the Web has become an inspiring factor behind the development in Information Retrieval (IR) facilitating the extraction of interesting information from the huge repository.

In the Web's earliest days, people were required to either remember the location of the pages or bookmark the pages for subsequent reference to be able to access the information of interest. In order to manage the rapid growth in the number of Web pages, the human-edited Web directories like Yahoo! directory emerged for organizing the Web pages into a hierarchy of topics. With continuous growth of the Web to its current state, today search engines have become main gateways to the huge amount of Web content for information access about new topics and products. The beauty lies in its functionality of automatically discovering new and modified pages, adding them to databases and indexing them by their keywords and related features. Today, search systems such as Google have profoundly changed the way we access information.

1.1 Background

On-line search engines facilitate to locate relevant and desired information from huge source of information available on Web. Most of the information on Web is presented as natural language text with occasional pictures and graphics. This is convenient for human users to read but difficult for computers to understand. Meaning cannot be inferred from the occurrence of a word e.g. does the word "apple" at a particular site refer to a fruit or a computer? Users share a significant burden in terms of constructing search queries intelligently whereas a general search user is simply unable to do it. Hence it is observed that the most of the renowned search engines return result sets with not so useful pages to the user.

The ultimate objective of a typical IR system involves the representation, storage, organization and access of information items [2]. A typical IR task has input, output and processes:

Input: It consists of an information need in the form of a user query which is generally a text string and a corpus of textual natural language documents.

Output: It consists of a ranked set of documents that are relevant to the user query.

Processes: There are three main processes in an IR system [3]: i) indexing ii) query processing iii) searching and ranking.

Indexing: It is usually observed that not all the parts of an information item are equally important to represent its meaning. Hence it is considered useful to preprocess the information item for selecting the index words from an information item. Indices are the structures used to speed up the search process when the item collection is large and unstructured. The inverted file, one of the most common index structures for text retrieval consists of two elements: Vocabulary and Term occurrences. The vocabulary is the set of all words in the text. For each word in the vocabulary, a list of all the word positions is stored. The set of all those lists is called occurrences.

Query processing: The natural language user query is parsed and compiled into an internal form. Query terms are generally pre-processed by the same algorithms used to select the index objects. Additional query processing like query refinement and reformulation requires the use of external resources such as thesauri or taxonomies.

Searching and ranking: User query is matched against information items with an intention to get a set of relevant information items in response. Text retrieval is based on the assumption that the matching between information items (the documents) and user information need (the query string) can be based on a set of index terms. As apparent, this involves a considerable loss of semantic information when text is replaced by a set of words without consideration of semantically related terms to the query terms. The set of information items returned by the matching step generally constitutes an approximate answer to the information need. The aim of the ranking step is to assess the relevance of the items with respect to user need, thus returning them by decreasing order of approximate relevance.

A typical IR system architecture seeks a query string from the user which is parsed and pre-processed with the intention of finding suitable matches with the indexed terms of the document corpus. The set of matching items lead to the presentation of ranked list of documents in decreasing order of relevance with respect to information need. The pictorial representation of the same has been shown in Figure 1.1.

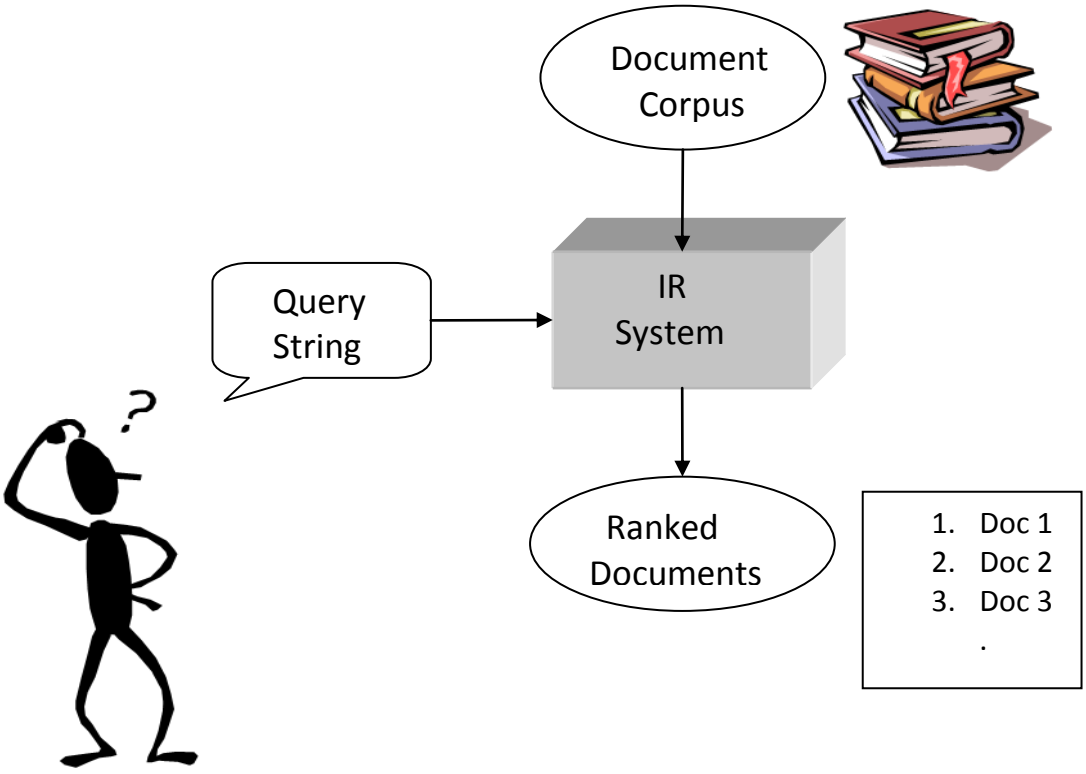

Figure 1.1: A typical IR system architecture

Similarly in case of a typical Web search system, the information need of the user is presented in the form of query string which is processed and matched against the index space of the document corpus and the relevant information is presented to the user in the form of ranked list of documents. Here the document space is generated through a spider program which crawls on the hyperlinked Web for fetching the topically relevant web pages. The pictorial representation of a typical Web search system is given in Figure 1.2.

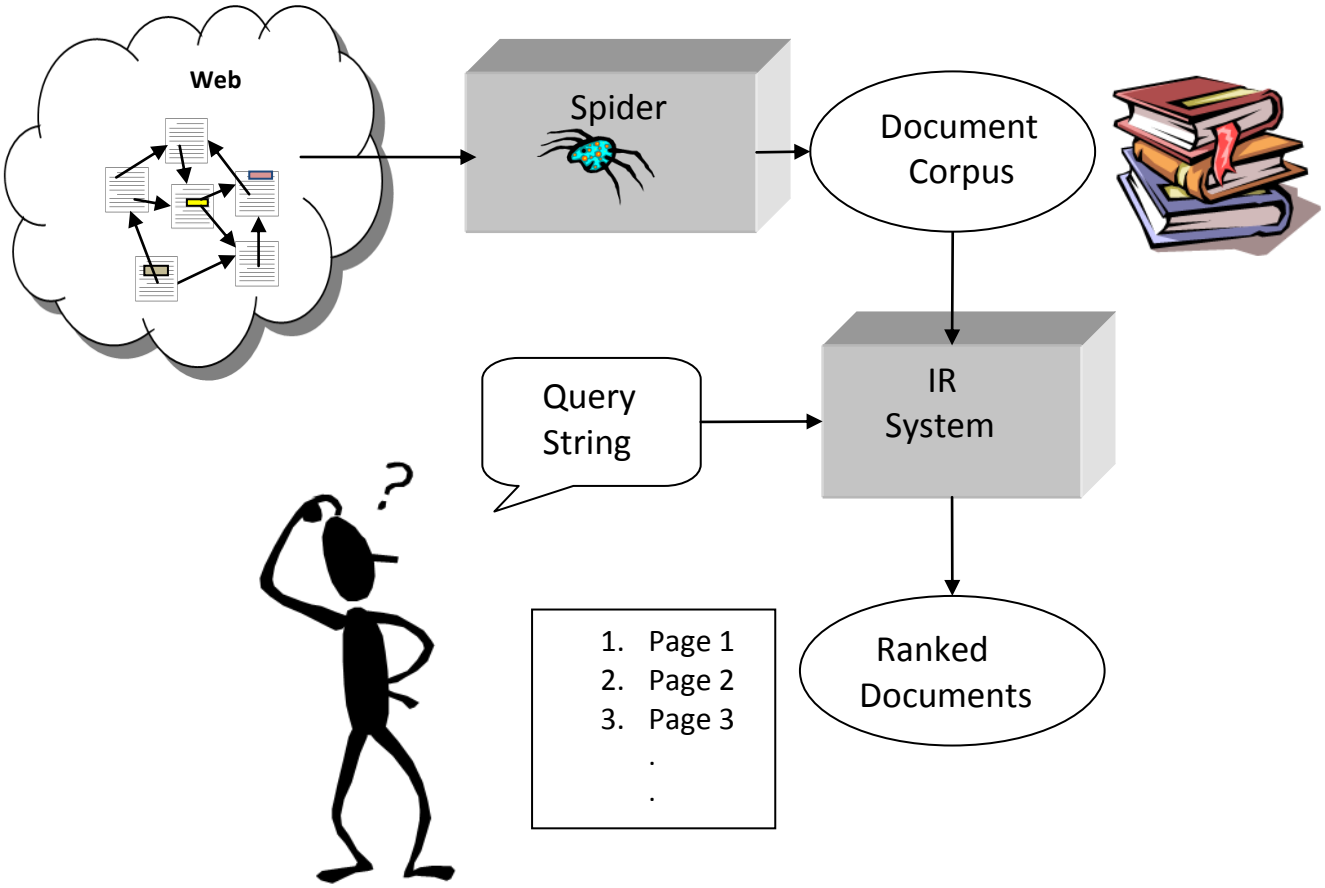

Figure 1.2: Web Search System

1.2 The syntactic search

The term syntactic search stems from the syntax, which is the rule set or grammar of a language used to structure and convey information where there is no concern with the meaning of the sentence with respect to a certain context. On the other hand, semantics deals with the actual meaning of that information. The present day Web is still based on HTML, which describes how

information is to be displayed and laid out on a Web page for humans. A general HTML document does not contain any special tagging to express its meaning; which is difficult to parse by a computer program.

A huge amount of text information on Web is written using natural languages. Being unstructured, most of the text information is hard to access compared to other well-structured information sources in the form of relational databases. This is due to the fact that reading and understanding text requires ability to disambiguate text fragments at different levels such as syntax and semantics. With huge information space in unstructured manner, human users rely on keyword based search engines to locate information and answer their queries. [The Web] is still based on HTML, which describes how information is to be displayed and laid out on a Web page for humans.

1.2.1 The syntactic search models

A number of IR models have been developed by IR research fraternity with syntactic perspective across the globe such as Boolean Model, Vector Space Model and Probability Model [2, 5]. These models have been further extended and refined by research fraternity with an intention to mitigate some of the drawbacks in the original models. E.g. Extended Boolean Model, Fuzzy Set Model, Generalized Vector Spaced Model, Latent Semantic Indexing Model [2]. In addition, a selected study of machine learning based IR models including Neural Network Model, Symbolic Learning and Genetic Algorithm Model has been provided by [6]. However, it is observed that although extended and refined models have a sound logical foundation, they did not achieve better performance compared to the classical ones. Hence only classical models have been deployed in practice for Web search applications such as online search engines.

The meaning of information retrieval can be very broad. However as an academic and research field of study, it had got different variants of definitions.

Definition 1 [5]: Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).

Definition 2 [2]: An Information Retrieval model is a quadruple [D, Q, F, sim] where:

- D is a set of (logical representation of) documents.

- Q is a set of (logical representation of) queries.
- F is a framework for modeling documents, queries, and their relationships.
- sim(q_i , d_j) is a ranking function which associates a real number with a query $q_i \in Q$ and a document representation $d_j \in D$. Such ranking defines an ordering among the documents with respect to the query q.

In information retrieval, keywords or concepts are referred to as index terms used as imprecise semantic representation of a document. A data structure called inverted index or inverted file [5] is used for the information retrieval purposes. To build a model, we think first of how to represent documents and user information needs. Given these representations, the framework in which they can be modeled is then conceived. This framework should also provide the intuition for constructing a ranking function. For instance, for the classic Boolean model, the framework comprises of sets of documents and the standard operations on sets. For the classic vector-space model, the framework comprises of a t-dimensional vector space and linear algebra operations on vectors. For the classic probabilistic model, the framework is made of sets, standard probability operations, and the Bayes' theorem.

Here three of the most popular classical text IR models are discussed in order to give a broad idea of the prevailing methods for information retrieval. These are: **Boolean**, **Vector** and **Probabilistic**.

Boolean model

The Boolean Model is one of the simplest retrieval model based on set theory and Boolean algebra. Document is represented as a collection of index terms present in the document and query is boolean expression on terms. Here:

- D: sets of index terms occurring in each document. Terms are treated as logic propositions, denoting whether the term is either present: 1 or absent: 0 in the document. Documents can thus be seen as the conjunction of their terms.
- Q: queries represented as a Boolean expression comprising index terms and logic operators (AND $\wedge$, OR $\vee$, NOT$\neg$) which can be normalized to a disjunction of conjunctive vectors.
- F is a Boolean algebra over sets of terms and sets of documents.

- sim is defined based on the consideration that a document is said to be relevant to a query if its index terms satisfy the query expression.

Example

Assume we have the query q = retrieval $\wedge$ (text $\vee$ ¬multimedia).

This query comprises of three different terms: *retrieval, text and multimedia*, and it can be written in a disjunctive normal form as q_{dnf} = [(1,1,1)$\vee$(1,1,0)$\vee$(1,0,0)], where each of the components is a binary-weighted vector associated with the tuple (retrieval, text, multimedia) [2]. These binary weighted vectors are called the conjunctive components of q_{dnf}.

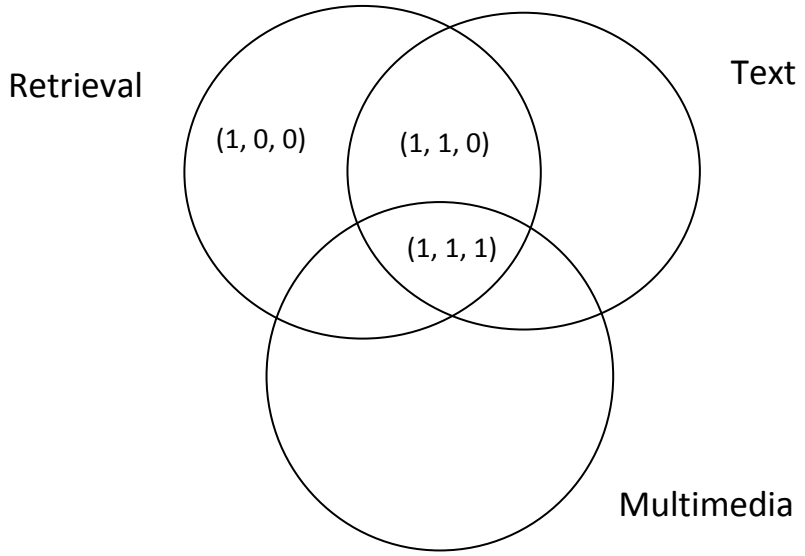

Figure 1.3: The three conjunctive components for the query q = retrieval$\wedge$(text$\vee$ ¬multimedia).

Figure 1.3 shows the set of documents containing the word *retrieval*, the set of documents containing the word *text*, and the set of documents containing the word *multimedia*. Given the query *q*, the subsets of documents that satisfy the query are: i) those containing all the three terms: (1, 1, 1) ii) those containing the word *retrieval*, but neither *text* nor *multimedia*: (1, 0, 0) and iii) those containing the word *retrieval* and *text*, but not *multimedia*: (1, 1, 0).

In view of its simplicity, the Boolean model had been popular for more than three decades until the emergence of online search engines. Unfortunately the Boolean model suffers from two major drawbacks. First, binary criterion (i.e. a document is predicted to be either relevant or non relevant) lacks in providing a proper basis for ranking the retrieved results, which is likely to lead low precision levels when the retrieval space is too big. Second, it is not always easy for most users to translate an information need into a Boolean expression with logic operators, which significantly decreases the usability of the approach. Third, in the absence of exact match between query terms and document terms in view of frequently changing senses of the existing words and addition of new words, approach seems to defeat its purpose to a significant level.

Vector-space model

In an attempt to overcome the limitations of Boolean model in terms of binary weight assignments to documents, the vector-space model (VSM) proposes a framework in which partial matching is possible. The non-binary weights assigned to document terms and query terms are ultimately used to compute the degree of similarity between each document stored in the system and the user query. The approach leads to ranked list of resulting documents in decreasing order of similarity. This is considerably more precise in terms of better match with the user information need in comparison to Boolean model approach.

Following the notation:

- D: documents are represented by a vector of words or index terms occurring in the document. Association of each term t_i in the document d_j is measured through a positive, non-binary associated weight $w_{i,j}$.
- Q: queries are represented as a vector of words or index terms occurring in the query. Each term t_i in the query q has a positive, non-binary associated weight $w_{i,q}$ representing association of term with query.
- F: is an algebraic model over vectors in a t-dimensional space.
- sim: estimates the degree of similarity of a document dj to a query q as the correlation between the vectors dj and q. This correlation can be quantified, for instance, by the cosine of the angle between the two vectors:

$$sim(\vec{q}, \overrightarrow{d_j}) = \cos(\vec{q}, \overrightarrow{d_j}) = \frac{\vec{q}.\overrightarrow{d_j}}{|\vec{q}| \times |\overrightarrow{d_j}|} = \frac{\sum_{i=1}^{t} w_{i,q} \times w_{i,j}}{\sqrt{\sum_{i=1}^{t} w_{i,q}^2} \times \sqrt{\sum_{i=1}^{t} w_{i,j}^2}} \quad (1.1)$$

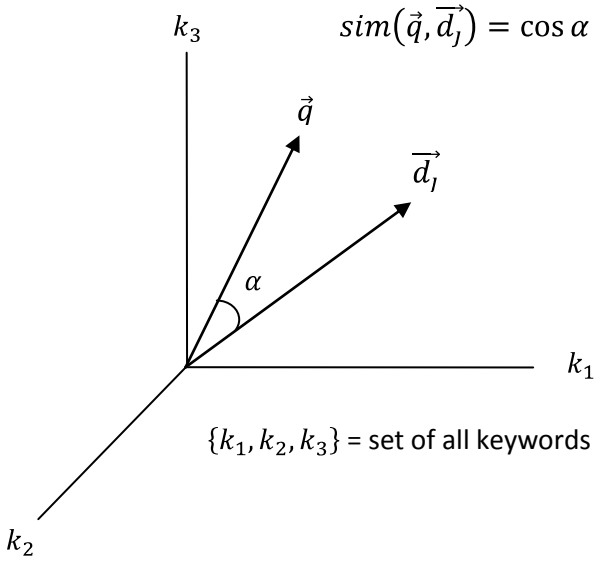

Figure 1.4: The 3-D representation of $sim(\vec{q}, \overrightarrow{d_j})$

$sim(\vec{q}, \overrightarrow{d_j})$ gets a non-binary weight (since $w_{i,j} > 0$ and $w_{i,q} > 0$) ultimately computing the degree of similarity between document d_j and query q. This leads to ranked list of documents in decreasing order of similarity retrieving documents even if it matches the query partially. A minimum threshold value can always be identified for retrieving a minimum number of documents based on the degree of similarity. The 3-D representation of the similarity of a document dj to a query q is shown in Figure 1.4.

The question arises how to calculate the weights of index terms with respect to document and with respect to query. Term Frequency, Inverse Document Frequent (TF-IDF) is one of the most popular models for measuring the weights of index terms. The weight of term t_i in document d_j is calculated as:

$$w_{i,j} = tf_{i,j} \times idf_i = \frac{freq_{i,j}}{max_l freq_{l,j}} \times log\frac{N}{n_i} \quad (1.2)$$

Where:

• N = total number of documents in the search space

• n_i = number of documents where the term t_i appears

• $freq_{i,j}$ = frequency of the term t_i in the document d_j

• $max_l freq_{l,j}$ = maximum frequency of any term t_l in the document d_j

The term frequency factor $tf_{i,j}$ indicates the association of term t_i with document d_j. Inverse document frequency factor idf_i indicates the spread of term t_i in the whole document search space. This helps to overcome the insignificant effect of frequently used words (like may, ought, that, this) which may occur otherwise. Since for such cases, $N \approx n_i$, which makes the factor $log \frac{N}{n_i}$ as zero, finally making $w_{i,j}$ as zero.

Although approximate matching of a document to query helps to make the set of the relevant documents more precise with respect to user's information need in comparison to binary matching, both the models assume the index terms to be mutually independent neglecting term dependencies.

Probabilistic model

In an attempt to capture the IR problem in a probabilistic framework, the probabilistic retrieval model ranks a document in decreasing order of probability that it belongs to relevant set of documents (i.e. relevant to the information need). Mathematically, it is termed as $P(R|q, d_j)$, given a query q and a collection of documents D, d_j is a document in D. R⊆ D contains exactly the relevant documents to q (the ideal answer set). Following the notation:

• D: set of documents where each document is considered as a vector of words or index terms occurring in a document. Each pair (t_i, d_j), has a binary associated weight 1 or 0, denoting the presence or absence of the term t_i in the document d_j.

• Q: query is represented by a vector of words or index terms that occur in the query. Each pair (t_i, q) has a binary weight 1 or 0, denoting the presence or absence of the term in the query.

• F: a probabilistic model that ranks documents in decreasing order of probability of relevance to the query.

• sim: measures the degree of similarity of a document d_j to a query q in terms of probability of d_j lying in R, set of relevant documents for q. This is measured in the probabilistic model as the means of relevance, as given by:

$$sim(d_j, q) = \frac{P(R|d_j)}{P(\neg R|d_j)} \quad (1.3)$$

where $\neg R$ denotes the set of non relevant documents, $P(R|d_j)$ is the probability of d_j being relevant to the query q, and $P(\neg R|d_j)$ is the probability of d_j being non relevant to q.

The further reformulation and computation requires a little elaboration [2]. Using Bayes' rule, we may write:

$$sim(d_j, q) = \frac{P(d_j|R) \times P(R)}{P(d_j|\neg R) \times P(\neg R)} \quad (1.4)$$

Assuming that $P(R)$ and $P(\neg R)$ are same for all the documents in the collection, and considering term independence assumption, $P(d_j|R) = \prod_{i=1}^{t} P(t_i|R)$ we have:

$$sim(d_j, q) \sim \frac{P(d_j|R)}{P(d_j|\neg R)} \sim \frac{\prod_{i=1}^{t} P(t_i|R)}{\prod_{i=1}^{t} P(t_i|\neg R)} \quad (1.5)$$

Assuming a function g(t, d) where g(t, d) = 1 if term t appears in the document d, and 0 otherwise, the previous formula can be reformulated as:

$$sim(d_j, q) \sim \frac{\left(\prod_{g(t_i,d_j)=1} P(t_i|R)\right) \times \left(\prod_{g(t_i,d_j)=0} P(\neg t_i|R)\right)}{\left(\prod_{g(t_i,d_j)=1} P(t_i|\neg R)\right) \times \left(\prod_{g(t_i,d_j)=0} P(\neg t_i|\neg R)\right)} \quad (1.6)$$

The term $P(t_i|R)$ stands for the probability that the index term t_i is present in a document randomly selected from the set R. $P(\neg t_i|R)$ stands for the probability that the index term t_i is not present in a document randomly selected from the set R. The probabilities associated with the set ¬R have meanings which are analogous to the ones just described. Taking logarithms, recalling that $P(t_i|R) + P(\neg t_i|R) = 1$, and ignoring factors which are constant for all documents in the context of the same query, we can finally write:

$$sim(d_j, q) \sim \sum_{i}^{t} w_{i,q} \times w_{i,j} \times \left(log \frac{P(t_i|R)}{1 - P(t_i|R)} + log \frac{1 - P(t_i|\neg R)}{P(t_i|\neg R)} \right) \quad (1.7)$$

where $w_{i,q}$={0,1} indicates the absence/presence of the term t_i in the query q and $w_{i,j}$ ={0,1} indicates the absence/presence of the term t_i in the document d_j

Since R is unknown a priori, simplifying assumptions can be made such as:

$P(t_i|R)$ = 0.5 and constant for all index terms t_i.

$P(t_i|\neg R) = \frac{n_i}{N}$ where n_i is the number of documents that contain t_i and N is the total number of documents.

Once an initial subset of documents v is retrieved and ranked by the probabilistic model, the probabilities can be refined to:

$P(t_i|R) = \frac{|v_i|}{|v|}$, v_i is the set of retrieved documents containing t_i.

$P(t_i|\neg R) = \frac{n_i - |v_i|}{N - |v|}$, considering that the non-retrieved documents are not relevant.

Following this process recursively we get:

$$P(t_i|R) = \frac{|v_i| + \frac{n_i}{N}}{|v| + 1} \qquad (1.8)$$

The idea of adding the factor $\frac{n_i}{N}$ in the numerator of equation (1.8) above is to cover the case of non-retrieval of relevant documents which will make v_i = 0. Also the factor 1 has been added in the denominator for the similar reason.

$$P(t_i|\neg R) = \frac{n_i - |v_i| + \frac{n_i}{N}}{N - |v| + 1} \qquad (1.9)$$

Initially the set of documents is required to be estimated into a set of relevant documents and a set of non-relevant documents. Also the term frequencies are not considered while assigning weights to the documents. Despite these limitations, variations of the probabilistic model have lead to the development of one of the most successful ranking models, BM25 [12, 13]. The first system to incorporate this function was the Okapi information retrieval system, implemented at London's City University in the 1980s and 1990s. This ranking methodology takes into account the present/absence of relevant information and incorporates a document-specific component, which measures term frequencies and documents lengths.

Other Extended IR Models

Out of the many refined and extended models of information retrieval, Latent Semantic Indexing (LSI) [5, 14] has gained a special attention recently. Based on a concept termed as Singular Value Decomposition, it can be considered as extension of the vector space model. The main idea behind is to map each document vector and query vector lowering the dimensional space in terms of concepts. This is based on the intuition that the terms that are having similar co-occurring words are brought together when it is arranged to lower the dimensional space of terms/documents [5]. One of the main benefits of this approach is its ability to take care the classical problems of synonymy and polysemy. The main limitation in its wide applicability is its computation cost in case Singular Value Decomposition is significant. Moreover, LSI also has difficulties in expressing queries with boolean conditions. Machine learning techniques like Artificial Neural Network (ANN), Symbolic Learning, and Genetic Algorithm (GA) [2] have also been implanted in the information retrieval task. Another approach is the text classification concerned with tasks of classifying documents into a set of pre-defined topics or categories utilizing, in particular, supervised learning with training data also known as inductive learning [15]. However, it has been tested and validated that these models do not have much capability to overtake the classical models [6].

Link Structure Analysis

In a scenario of huge availability of information on the Web, it becomes necessary to work on measuring the relevance of documents for Web-based information retrieval systems. In a classical scenario of IR systems, the relevance was treated to be measured based on the idea of similarity between the query terms and document terms whether the underlying model is Boolean, Vector space or Probabilistic. There is another alternative approach for measuring the relevance of the documents with respect to information need called link structure analysis as worked out by [5]. It has been mentioned in [5], "Link analysis is one of the many factors considered by Web search engines in computing a composite score for a given query (i.e. for ranking documents)". Link structure analysis is based on discovering and modeling the hyperlink structure of the web graph. This can help in discovering the similarity between the Web pages based on the hyperlinks connecting the two Web pages and the importance of the Web pages based on the number of hyperlinks pointing to a particular Web page. The idea of citation analysis seems to be a motivating factor behind this approach. The technique is used to measure the impact of original research articles based on the number of other scholarly articles

referring or citing that piece of research in order to reach to new findings [5]. A reference to an article is treated as confidence reposed in the authority of the referred article by the referring article. In a similar fashion, hyperlink to a Web page A from another page B is treated as B reposing confidence in the authority of A. However, it was noted that mere the count of the number of in-bound links is not sufficient to reflect the authority or importance a Web page as practiced in earlier link analysis techniques. As a result, some more robust and influential link analysis techniques cropped up, popularly known as the HITS algorithm [16], and the PageRank [17].

The HITS algorithm is based on the idea that if the creator of page *p* provides a link to *q*, then *p* reposes confidence in the authority of page *q*. In [16] Kleinberg introduced a link analysis algorithm called Hyperlink-Induced Topic Search (HITS) based on the intuition that links represent human judgment. The algorithm associates two types of weights to the hypertext document. It assumes for any query topic, the existence of a set of authority Web pages or sites that are relevant and important with respect to the query topic and the existence of some hub sites containing links to relevant and important sites. An authority page has to be pointed by many hub pages whereas a good hub would be pointing to many authority pages [16].

PageRank [17], the algorithm used by one of the most popular search engine, Google is also based on a similar recursive propagation idea [18] as in the HITS algorithm. The rank of a document depends on the rank of its parents.

$$r(v) = \alpha \sum_{w \in pa[v]} \frac{r(w)}{|ch[w]|} \qquad (1.10)$$

Where r(v) is the PageRank value assigned to page v, pa[v] is the set of v's parent nodes and ch[w] is the set of v's child nodes, α is the normalization constant [18]. Each parent of v contributes a weight directly proportional to its own PageRank value and inversely proportional to the number of its outgoing links.

1.2.2 Motivation for improvement

Though the keyword-based approach to search engine design has provided users with a means to accessing the Web, keyword-based searching is limited. Indexing words from HTML files within

a vast centralized search database and presenting keyword matches to human users seems to be less effective.

As on May 2014, there are approximately 975,262,468 web servers as analyzed by an internet services company, Netcraft. This is approximately 16 million more than those on previous month as analyzed by the same company. In recent times, Web has become a huge, open, at the same time, universal repository of information and knowledge which is continuously expanding at a significant pace. A rapidly growing number of user generated documents poses special difficulties to information retrieval. Current information retrieval systems are primarily based on the premise that the meaning of a document is inferred from the occurrence or absence of terms in it whereas precise vocabulary for query formation is often difficult to predict while searching for relevant information on Web. In order to retrieve more relevant documents minus non-relevant documents, the query has to be refined by the user in an appropriate manner. With continuous increase in the Web content with diversified terminology, it becomes impossible for the user to anticipate the terms which can be used to enter a query for reaching to the set of relevant documents filtering the non-relevant ones [7]. Moreover, query length continuously tends to be small in size as per earlier trend. Hitwise, a significant representative of online consumer behavior, in 2009 reiterates in its study that the average query length was 2.30 words, the same as that reported in [8] a decade ago. Though there are queries with length five or more words but the number of such queries is very less. In such circumstances, vocabulary problem becomes even worse along with the fact that user often does not know exactly what he is looking for and/or he is not clearly able to describe it in words.

Another useful observation complementing the above points is given in [9] that most web queries fall into one of the three categories namely i) Informational ii) Navigational and iii) Transactional. In information queries, a user seeks to satisfy his information needs pertaining to a domain. In such cases, users generally are not aware of specific terms or vocabulary which should be appropriately used for describing his information need. Hence, this type of queries seem to be the best case for finding out the related terms in context of the query which would make the user's search experience more satisfying. Whereas navigational queries and transactional queries are usually communicated with specific terms/words related to a particular URL or Web-mediated transactional activity respectively generally known to the user.

Generally a natural language like English contains no two words with identical meaning still there are many words and phrases that have different meanings in different contexts. Present keyword-based search techniques do not consider the semantic aspect of the words used in queries. Thus, using such keyword-based searching comes with the risk of generating unwanted results, providing users with less relevant information stack that they must sift through to find what they are looking for.

The format of the present web page architecture is human readable. In other words, machines can hardly process the content of the web pages so as to reach to the desired relevant information in response to the user query. Conventional methods based on keywords or bag of words have very little to resolve the difficulty making ground for the development of automatic and intelligent agents for searching, retrieving, synthesizing and interpreting information. The idea had already been endorsed some thirty years ago by Rijsbergen [10] that "…it has been clear that further advances in the effectiveness of retrieval by such techniques (i.e. classical IR approaches) are not possible…" Some of the identified points endorsing the need of refined or complimentary IR models are:

- The machine readability of the contemporary Web is almost negligible since the Web content has been developed mostly using natural language text and HTML is a mark-up language used to address human readability issues with little scope to be processed by machines.
- The information about the relationships among the entities in the form of metadata is not available with the current Web resources and there is no standardization as such to be followed for the generation of metadata.
- There is little scope for generating the answers to complex questions using current search methodologies due to the absence of logical reasoning techniques.

1.3 The semantic search

Semantic search refers to the approach where semantic aspects of the Web resources and the meaning of the user query are considered in order to address the user's information needs [22]. There is a potential scope for the development of semantically aware search systems which can browse the semantic annotations of the available Web resources and query against them. The

name of the creator of a Web page or the URLs of people, organizations, locations and other referred entities can be marked up with the Web page as semantic annotation. As a mean of realization of semantically aware search systems, the Web resources are required to be identified by means of entities and concepts which can be termed as semantic features. This semantic feature-space can be represented with the help of reduced number of dimensions in comparison to traditional approach since similar terms like lecturer, teacher, instructor can refer to the same concept. Moreover, this modern feature space can be represented in the form of structured knowledge bases containing information about the entities mentioned in the documents.

1.3.1 The Semantic Web

Semantic Web [4] is not a separate Web but it is an approach to express information in a more meaningful manner in an effort to enable computers understand and process the information in more effective way. It is the idea facilitating the management, discovery, processing and integration of data available on Web for its more effective use and reuse across various applications. The objects of the Semantic Web will comprise of entities such as people, places, products, events etc. along with regular media objects such as Web pages, images, audio clips etc. Unlike the association existing among the Web pages in the contemporary Web with only a single sort of relationship (hyperlink), the Web objects are related with diversified set of relationships in case of Semantic Web.

The steps being taken in developing Semantic Web standards and technologies will help in making machines understand and process the data that is merely displayable at present. In the present scenario, the Web is more able to serve as a medium to provide information access to the people in the form of documents rather than having capability to process the information. For the realization of the Semantic Web, there is a requirement of effective information access methods to have access to structured collection of information. Alongside, the sets of inference rules for conducting automatic reasoning in order to take decisions and answer complex questions would help to offer more effective search capability.

Using Semantic Web approach, well defined machine interpretable semantics is used to share knowledge intended for publication by machines for machines-tools, Web services, software agents, information systems and so forth. Some of the popular Semantic Web service

composition methods have been reviewed in [11] and observations have been made on some of the frequently used Semantic Web languages for the purpose. A series of annotation and Knowledge representation standards were developed under W3C driven community projects such as Resource Description Framework (RDF) [19]. Metadata is represented by RDF as a language serving as a basic data model for the Semantic Web. It differs from the largely unstructured free text found on most of the Web pages and the highly structured information found in the databases. The domain of resource is virtually undefined where anything can be treated as a resource from a Web page, audio or video file to any real world entity like person, product or event. Literals can have concrete data values such as string, number or date. The information can be modeled in RDF statement as a triple <Subject, Predicate, Object> where Subject describes the resource, Predicate describes the relationship type or the attribute and the Object describes the resource or literal representing the value of the attribute. RDF and RDF Schema (RDFS) [20] introduce semantic features which allow for definition of new classes and properties. Web Ontology Language (OWL) [21] extends RDFS providing means for more comprehensive ontology definitions. The data representation is in the form of graph of literals and resources in RDF rather than a tree of nested elements as in ordinary XML, allowing users to define terms(for example, classes and properties),express relationships among them, and assert constraints and axioms that hold for well-formed data.

One of the advantages of the Semantic Web comes in the form of access to information available on it through semantic search systems. This will offer a more effective search capability in comparison to keyword based contemporary search systems. Semantic mark-up, standards and technologies can be used to enable semantic oriented search. Instead of just exploiting the syntactic element of the Web, the availability of large amount of structured, machine readable information offers a range of opportunities for improvement over classical search. Figure 1.5 shows one small chunk of the Semantic Web corresponding to the musician Michael Jackson. The illustration expresses a variety of aspects of the Semantic Web which are relevant to semantic oriented search. Semantic Web represents the Web of relations between real world objects such as persons, products, events and organizations rather than a Web of documents. In Figure 1.5, there are objects such as the city of “Gary, Indiana”, the musician “Michael Jackson”, the music album “History” etc. The Web objects are treated as resources rather than strings such as the resource “Michael Jackson” not the string Michael Jackson.

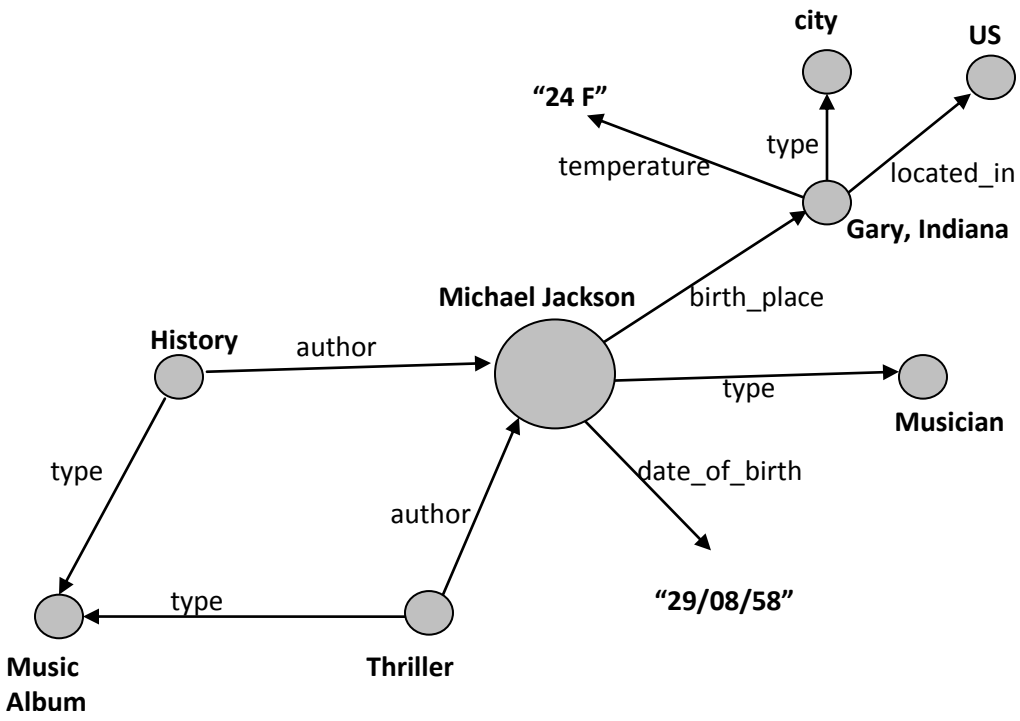

Figure 1.5: A segment of the RDF graph pertaining to *Michael Jackson*

There are many people with the name Michael Jackson. This segment denotes only one particular person with that name. The information contained in this segment is in the form of semantic mark-up with an ability to be processed by machines. The data in the Figure 1.5 is almost whole machine understandable. It states, in a machine understandable language, that Michael Jackson is a musician whose birth place is Gary, Indiana etc.

Contemporary search techniques use keywords as indexing terms with little scope for using semantic mark-up for indexing purposes. The vital effects of semantic mark-up simply remain hidden during the search process. Current Web search systems cannot use semantic mark-up to improve text retrieval. They typically rely on a basic and well-established mechanism of defining and satisfying an information need. A set of words is used as query to get in return a list of documents belonging to those words. Some approaches use the techniques of thesaurus expansion or blind relevance feedback for incorporating the content of inference to a humble extent but it seems that better results are possible with the use of semantic mark-up during the search process.

Now-a-days, semantic technologies have been evolved to such a state to be able to contribute to classical IR problems. Each Web page would be supposed to possess additional details about the page in the form of semantic annotation. The vocabulary for the annotations is usually expressed by mean of ontology. Such agreed-upon general purpose ontology can be exploited to match entity types based on known facts, relationships and other metadata. This will further give rise to rank the Web objects of various types, complexity and structure such as entities and relationships existing among those entities for dealing with more complex user queries. It seems very unlikely that contemporary ranking schemes for ranking Web documents may be applied to complex structures such as entities and relationships. Based on a peculiar set of parameters, a table of comparison between traditional Web based search and Semantic Web based search has been presented in Table 1.1.

Table 1.1: Comparison of features of traditional Web based search and Semantic Web based search

Parameter	**Traditional web based search**	**Semantic web based search**
Dataset	Documents	RDF Triples, Semantically annotated Documents
Data Organization	Unstructured	Semi- Structured
Search Orientation	Document – centric	Entity, Relationship and Semantic Document centric
Collection	Bag of Words	Bag of (RDF) assertions
Representation	Light weight syntax – centric Models	Ontology based better expressive models
Domain of satisfaction	Work well for topical searches	Complex queries are satisfied, More precise answers
Query processing approach	Matching and filtering	Not just matching and filtering but also joining
Scalability	Web scale	Not scale to massive and heterogeneous Web environment

1.3.2 Semantic search system

The emergence of conceptual frameworks for semantic based search on Web has a wide spectrum of precedents. One of the most appealing quotes from the work published by W. B. Croft [45] about three decades ago serves as a sound introductory factor for the present topic:

"*The systems that have been developed, such as those based on probabilistic models of relevance [46], capture 'domain knowledge' purely in the statistics of occurrence of individual words (or stems) in the documents and in statistical dependencies that exist between words. We define domain knowledge to mean information about the important topics or concepts in a particular domain and how they relate to each other. The statistical approach has many advantages and can achieve a reasonable level of effectiveness with techniques that are very efficient. However, it appears that to achieve significant improvements in retrieval effectiveness compared to current techniques, systems must be designed to acquire and use explicit domain knowledge."*

A system is referred to as a *semantic search system* if there is an involvement of semantic technologies at some stages of search process. The meaning of the Web resources as well as query is explicitly taken into consideration for the purpose of effective search. Meaning is established through semantic annotations built up using vocabulary available in general purpose ontology addressing the relationships existing between entities and their interpretations. Various semantic models work on incorporating the intent of the query and the meaning of the content possibly at different stages of search process such as query construction, query processing, result presentation and query refinement [25].

One of the approaches towards clear understanding and implementation of a semantic search system consists of a query interface where an SQL like query language for the semantic web is taken as input (such as SPARQL). The SPARQL like query is executed against a knowledge base (KB) returning a list of knowledge instances such as entities, relationships satisfying the conditions of the query in Boolean terms. The vision here is to provide exact matches of the query with instances of the information space termed as *data retrieval* in [2], "*A data retrieval language aims at retrieving all objects which satisfy clearly defined conditions … a single erroneous object among a thousand retrieved objects means total failure. For an information retrieval system, however, the retrieved objects might be inaccurate and small errors are likely to go unnoticed".* There is little consideration for approximate answer to an information need

expressed by the user in this approach. The semantic portals and ontology based QA systems are potential examples of this approach as available in literature. These systems return ontology instances rather than documents with little consideration for approximation. This Boolean approach can be judiciously realized in the presence of complete formal knowledge base. Noticing the voluminous unstructured information currently available on Web with insufficient presence of formal ontological knowledge and conversion systems for converting former to later, it seems difficult to fully realize the potential of such approach in present scenario.

Another strongly probable line of action in this direction is the domain modeling as a thesaurus of concepts with each concept has relation to some other concepts and a list of some ad hoc rules to recognize the concepts in a textual passage. This semantic knowledge in terms of named concepts/entities and relations between them can well serve as a mean to expand queries as well as documents towards effective search systems. The background resources like WordNet can be used for explicit conceptual descriptions of a wide variety of domains. They provide consistent vocabularies and word representations necessary for unambiguous communication within knowledge domains and facilitate to distinguish the meaning of words from free text sentences using domain specific contextual vocabulary. This domain related ontology based approach suffers from its own constraints of design and construction of ontologies [24]. At the same time, regular updation and maintenance of ontologies is a crucial issue in view of changing senses of the existing words and addition of new words.

There are linguistic models such as thesauri or corpora that capture relations between words and used for the expansion of query terms based on the mapping of query words to thesauri elements and its extension based on its relation to other terms in the thesaurus. Thesauri represent an approximation of conceptual spaces where thesauri terms approximate the concepts of the domain for which the thesauri is built. A significant quantum of research has been diverted to this direction where domain knowledge is used seeking fuller development of search systems towards greater improvement of results and its applications to different scenarios such as Web. In the database community, conceptual models and Entity Relationship diagrams are used to capture relations between data elements [23]. In the next section, the scope of semantic search paradigm has been elaborated with a view to give a reflection of its various diverse application areas.

1.4 Scope of semantic search paradigm

The application of semantic search has been observed in various environments like *desktop* [49], *controlled domain repositories* [48] and the *Web*. In the current form of the Web, information gathering is a task mainly performed by human using a Web browser. The Semantic Web shall provide an infrastructure so that the task can be performed by computer programs. To allow for processing of information on the Web by machines, information is annotated with machine interpretable data.

The incubation of search approaches in a *desktop environment* is a known phenomenon in information retrieval. Mostly available search approaches based on semantic technologies were originally meant for semantic desktop. Semantic information can be easily extracted from the semi-structured content like e-mails, folders and other XML content in this environment. User satisfaction is easier to achieve in view of explicit interaction and personalized environment. Opening up to broader environments, a significant number of works have been undertaken and tested over *controlled domain repositories* pertaining to popular domains like sports, travel, university, research, medicine and music. A novel domain of disaster management has been explored for integrating domain related heterogeneous data by a recent approach [47]. Available information is enclosed in the form of formal domain knowledge and conceptual meaning is extracted which is represented in the form of formal knowledge instances satisfying specific user needs. Most of the present web search engines are based on techniques originally developed for non-Web environments. The *Web* is an open system where information is spread across huge number of resources pertaining to hugely wide variety of domains. At the same time, the rate of its growth is extremely fast and is referred by a large number of users for a variety of information needs. In such scenario, it becomes difficult to obtain conceptualizations covering the meaning of all Web content under a uniform model.

Another form of semantic based search points towards the *question answering (QA)* that deals with natural language questions. The main objective is to find explicit and accurate answers to the questions rather than the whole documents. Open QA approaches over free text systems typically include a question classifier module that determines the type of question and the type of answer. A gradually reduced amount of text after question analysis is undergone through several modules that apply NLP techniques. Finally, an answer extraction module looks for further clues

in the text to determine if the candidate answer can serve as potential answer of the question. In order to identify the type of question, various systems have built hierarchies of question types based on the types of answers sought. QA approach can now-a-days be seen as integrated and tested in Web scale search systems like popular *Ask* search engine, Yahoo! SearchMonkey, IBM Watson.

Another area of significance for the application of semantic search directs towards finding out the relations between words which has cultivated rich results in a number of application domains. A particular approach of finding relatedness among words has been tested in the framework of a particular application. A number of such application domains are:

i) Word sense disambiguation
ii) Named Entity Recognition
iii) Clustering
iv) Information retrieval
v) Speech recognition
vi) Text summarization and annotation
vii) Automatic indexing
viii) Automatic correction of word errors in text
ix) Lexical selection

Word sense disambiguation: The task of resolving word sense points towards the appropriate use of the word having more than one meaning in a particular lexicon. A large number of words are available in literature having more than one meaning required to be disambiguated with respect to a certain context. Leacock and Chodorow [37] tested the use of WordNet based similarity as an aid to a local context classifier made up of large training sets for each sense of each polysemic word. At the same time, the problem of sparseness of training data was also proposed to be resolved by providing semantically similar words based on WordNet. E.g. if "apple" claims to be a good discriminator for a particular sense class of the verb "ripe" then so should be semantically similar words as "mango", banana" and "pineapple" even if they were not the part of training set.

Named Entity Recognition(NER): As the name self explains, the domain of Named Entity Recognition deals with the classification of text data objects into a set of pre-defined classes viz.

names of persons, products, events and locations. A framework to this effect has been presented in [38] for efficiently exploiting free-text annotations as a complementary resource to image classification. The entities occurring in the text data are mapped to a pre-defined set of concepts using Semantic Concept Mapping (SCM). The relationships existing between the entities as expressed in WordNet thesaurus are taken into account for classification purposes.

Clustering: Measuring semantic similarity between words is also noted to be significantly applied in the domain of document clustering. Semantic similarity measure as proposed by [39] significantly improves the accuracy in a named entity clustering task. Combining both, page counts and sentence-level contexts in snippets, the method is based on supervised classification trained on examples, which is uncommon in semantic relatedness and distributional similarity methods. A Support Vector Machine based classifier is trained using a set of synonym pairs as positive examples and a set of non-synonym pairs as negative examples, both of which are randomly selected from WordNet.

Information retrieval: With the consistent growth of Web, information retrieval has become one of the most widely used applications of Web. A large number of approaches depend on the link structure and the keywords indexed in their document repositories for the purpose of reaching to the desired information to the user. To find out the semantic similarity between the words play a significant role in reaching to the actual context of the user throwing a query. One such approach is presented by [42] for performing search in context automating the search process providing even the naïve users with highly relevant search results. The context driven information retrieval process involves semantic keyword extraction and clustering to automatically generate new, expanded queries.

Speech recognition: With the advent of technology, spoken audio documents are becoming increasing popular in order to reduce interaction time and quicker execution. A transcript generated through automatic speech recognition system typically contain many recognition errors making understanding quite difficult. The approaches for measuring semantic similarity between words facilitate to identify *semantic outliers* with respect to other words present in the transcript. The approach presented in [43] automatically identifies recognition errors and remove them from the transcript. The words which are not semantically related with other words in the

transcript termed as *semantic outliers* are identified and removed from the transcript in order to make the browsing of the audio content from the original file more efficient.

Text summarization: "Text summarization is the process of condensing a source text into a shorter version preserving its information content" [40]. This is observed to be of significant use in various fields like in a literature survey for high quality *informative summary* or in order to quickly decide whether a given piece of text is worth reading based on a high quality *indicative summary*. Such summaries which otherwise require thoughtful reading of the text; can be obtained by applying less powerful methods. One such method, deploying lexical chains is presented by [40].

Automatic spelling correction: Malapropisms are those syntactic and semantic errors that are close to their intended correction in spelling or sound, yet quite different and malapropos in meaning [44]. It seems that most of the word-spelling errors occur due to above two reasons. E.g. call ↔ all, wait ↔ weight, tale ↔ tail ↔ table. A proposal was given and tested for detection and correction of malapropisms [44] based on the hypothesis that the more a word is semantically distant from all the other words in a text, the higher the probability of the word being a malapropism. The program constructs lexical chains among the set of words that are semantically close and use those to find out the malapropism in a text.

Lexical selection: It is the task of choosing the most adequate translation among part-of-speech (POS) in the target language (TL), given several source-language (SL) translations with the same POS. The aim is to find the most adequate translation, not the most adequate sense unlike the task of word sense disambiguation. Having defined the similarity measure in a conceptual domain, [41] propose to define similarity between two verb meanings as a summation of weighted similarities between pairs of simpler concepts in each of the domains the two verbs are projected onto. A prototypical lexical selection system called UNICON is reported to have implemented where proposed measure of similarity play a leading role in finding the interpreted counterpart of a given verb in another language.

1.5 Thesis organization

This thesis is broadly divided into six chapters. **Chapter 1** focuses on the introductory background about the information retrieval on Web. After giving general idea about Web based

information retrieval, the limitations of the prevalent approaches have been introduced which serve as motivation for improvement. The concept of semantic search has been discussed in the light of semantic web based search and traditional web based search. Finally a number of other application areas have been explored where the aspect of semantic search can be judiciously used in terms of finding relatedness between two words.

Chapter 2: Literature Review: This chapter serves as a foundation for entire research work. In this line, it throws light on the existing approaches of semantic based search. Two broad classes have been identified as Conceptual models and Linguistic models. Comparative analysis among various approaches belonging to both the types has been given in the form of tables. A list of open issues which emerged as a result of literature review has also been presented with an intention to reflect their potential for future research directions in the field. Out of the open issues mined, one has been selected as foundation for problem formulation. Finally the prime objectives of the research work have been presented.

Chapter 3: Proposed Framework: QUery-context based Information retrieval using Corpus Knowledge (QUICK): The proposed framework is broadly divided into five steps towards the overall improvement of search results enhancing *Recall* and *Precision*. The five steps are: i) Data Acquisition ii) Corpus generation iii) Pre-processing iv) Candidate feature generation v) Feature selection. This chapter discusses about the theoretical foundation of the framework especially in terms of data acquisition and corpus generation.

Chapter 4: Design and Implementation of proposed framework: This chapter discusses the last three steps of the proposed framework in detail. Individual corpus is generated for each example category/query topic. After filtering out the insignificant content having little lexical value from the corpus, a peculiar aspect of conditional probability is applied for finding out the contextual features highly associated with the category/query topic.

Chapter 5: Testing and Validation: This chapter covers the testing and validation aspects of the proposed and implemented framework for semantic based search on Web. After reaching out to the contextual features which are highly associated with the category/query topic, the performance of the proposed system is tested against the existing system of search on Web. Two of the most frequently used evaluation parameters have been used for evaluating retrieval performance: *Precision* and *Recall*.

Chapter 6: Conclusion and Future Scope: With this chapter, the findings of the entire research are concluded along with potential scope for future directions in the domain.

1.6 Thesis Contributions

The major contributions of the thesis are as follows:

i) Various approaches for semantic search on Web have been studied and analyzed from conceptual perspective and linguistic perspective identifying fundamental limitations in the state of the art motivating for further research in the domain.

ii) A novel approach for semantic based search has been proposed which is inclined to use query dependent features for the purpose of efficient search semantically. The framework has been articulated with a relevant acronym :QUICK(QUery-context based Information retrieval using Corpus Knowledge)

iii) A Web mediated corpus has been generated which is used to fetch terms which are semantically related to query terms. A significant property of this domain independent corpus generation is its automatic updation without incurring high cost.

iv) A peculiar aspect of conditional probability has been exploited to find the semantic relatedness of original query terms with the associated terms generating the most probable context of the query.

v) Some crude solutions to the limitations found in the semantic search approaches have been described with an intention to provide potential areas for further improvement.

Chapter 2

Literature Review

In conventional Web search technology, information need is expressed in the form of a query whose words are expected to be lying in the documents of interest. The problem comes when two different words refer to the same concept and the search technology becomes unable to retrieve the complete set of relevant documents as a result of the query. At the same time, when the word having two or more different senses is used as query term, the search mechanism is unable to interpret the intention of the user's query. Although the problem of query ambiguity can substantially be resolved by careful choice of query words but it seems that the general user is not organized to do this. Another major limitation of the present day search technology comes when it becomes unable to answer a complex query requiring knowledge and data not available in a single document. Due to its limitations, there is a huge potential to carry out research activities for the improvement of retrieval techniques of the search systems.

2.1 Semantic search on Web

One of the predictable benefits of the Semantic Web oriented technologies comes in the form of semantic oriented search. The limitations of current search technology are expected to be overcome with the addition of explicit semantics. Semantic annotations can be added to the documents in order to describe it in a more predictable manner using languages like RDF. The language architecture allows expressing the relationships existing between two resources exclusively which helps to address the information need with knowledge and data not available in the documents. Although the notion of semantic search on the Web cannot be defined

exclusively, still it is commonly referred as the refined form of search on Web where both search queries and differently enriched Web content are used to extract meaning and structure. Semantic Web technologies are used to exploit such meaning and structure during Web search process with respect to one or more underlying ontology. The objective of Semantic Web vision is to facilitate the automation of tasks requiring machine understanding of the objects involved (e.g. information objects like entities, relationships). This will enable software programs to automatically find and combine information and resources in consistent manner. The ontologies provide the required vocabulary for describing background domain knowledge further facilitating the connection of the Web resources to semantic annotations or the extraction of semantic knowledge from Web resources.

Another significant perspective is to exploit lexical semantics towards more efficient semantic based search on Web. It can improve the performance of search systems using different kind of resources available for lexical semantic knowledge and using different methods to embed that knowledge into search process. Such knowledge is generally encoded in structured and semi-structured knowledge bases that form a graph of concepts interconnected with the help of edges or links representing a certain kind of relationship between two concepts.

Table 2.1: A comparison between conceptual perspective and linguistic perspective of semantic search system

Parameter	**Conceptual perspective**	**Linguistic perspective**
Relationship element	Models of relationships among objects	Models of relationships among words
Semantic backbone	Ontologies help to capture entities in the real word and their relationship	Taxonomies, thesauri, dictionaries, corpora for capturing entity names and relationships
Inference element	Inference along domain specific relations	Inference along linguistic relations, e.g. broader/narrower/ functionally related terms
Search basis	Knowledge based search	Natural language/keyword based search
Search elements	Entities, relationships, documents	Documents

Alternatively, a large corpus can also be used to fetch background information of the words in the form of their contexts in order to find the relatedness among the words.

2.2. Semantic search system: A conceptual perspective

In the present scenario of search technology, the resource descriptions and query processing techniques still leave a significant scope for capturing and exploiting the conceptualizations involved in user's information needs and resource meaning. Limitations include the inability to count for relations between search terms. There is nothing that can be thought in isolation in this great universe. Every single entity has got some sort of relationship with other entity. While submitting information need to a standard search system, it is treated obvious that the specific relations get hold between search terms. Take a query "Conferences or workshops on semantic search held in India in 2014-15" for instance. The user treats the relations in the query "conferences or workshops" *in_the_field_of* "semantic search" *held_in* "India" *in* "year 2014-15" to be included as a part of search in obvious manner. However, these semi-expressed details get lost when query was sent to search system mechanism. Traditional search mechanisms suffer with the problem of necessary infrastructure for exploiting relation based information that belongs to the semantic annotation of a Web document.

The Semantic Web [4] is an effort to find solution to this problem at architecture level. The semantics in terms of word sense or context is presented by the relations between entities and are recorded by RDF [19]. The relation is interpreted by OWL [21]. The ontology based search engines have gained a considerable momentum in the last decade due to ever growing amount of ontology based semantic mark-up in the Web. With the advent of vision of Semantic Web, there is equal interest and work in automatically extracting and representing the metadata as semantic annotation to the documents and services on the Web [50]. Semantic annotations based on classes of concepts and relations among them are recorded as additional details about the Web page. Ontology provide a reference framework in order to express the vocabulary of the annotation and play a crucial role in determining the relevance of a retrieved document based on known facts, relationships and other data.

Along with retrieval of documents, semantic search systems work for the retrieval of entities and relationships. Available techniques have been developed for entity oriented search of documents [51]. At the same time, relationships also play a vital role in the relevant information access as

the Web evolves continuously [52]. Semantics of relationships among entities are defined in schema ontologies (e.g. through the domain and range constructs in RDF(S) or OWL languages). It is increasingly possible to analyze metadata extracted from Web to discover interesting relationships. It seems that the retrieval of complex *relationships* would be a significant objective of upcoming semantic search systems as retrieval of *documents* in the present day search systems.

A system is referred to as a *semantic search* system if semantic technologies are involved in some stages of search process. Various semantic models work on incorporating the intent of the query and the meaning of the content possibly at different stages of search process. It seems logical to believe that semantics can be exploited at different steps of search process starting with knowledge management and representation. After management of data and knowledge in line with semantic requirements, the actual objective of information need satisfaction can be met following various stages viz. query interface, processing of query against the indices, presentation of results, finally the exploitation of user feedback. Hence it can be stated that **Semantic Search framework** comprises different components. There is a potential scope of working on each component exclusively in order to increase the search efficiency.

2.2.1 Knowledge Management

Ontology based knowledge management is one of the most sensitive requirement for semantically enhanced search system. Knowledgebase (KB) is required to be populated with KB schema in the form of domain ontology and data instances related to the domain. Related rules/axioms are also supposed to be stored in the KB. Some refer it as ABoxes: Assertion boxes and TBoxes: Term boxes.

Web documents can be visualized and represented in the form of semantic annotations for the purpose of extraction of data elements (triples, ontological sentences) pertaining to the document. KB is supposed to be populated with those entities/triples extracted from web documents along with domain ontology. *Entity disambiguation* comes out to be another area to be worked upon while populating KB with ontological elements. One or more of query interpretations is required to be run against the KB index in order to find relevant entities. Subsequently, entity based document search can be initiated to find documents relevant to entities. Entity ambiguity is required to be resolved beforehand.

2.2.2 Query Interface

Various approaches are available in literature through which a user is allowed to enter his information need in the form of a query. These can be classified as:

i. Keyword based

ii. Form based

iii. Natural language based

iv. Structure query language based(such as SPARQL)

Various state-of-the-art approaches to semantic search are observed in recent literature based on structured query languages [72, 80, 81, 82, 83, 84 and 85]. Query expressiveness can be enhanced to a great extent using structured query based approach but a general user may not be willing to learn structured query language. He is comfortable with keyword based approach or natural language based approach where it is not required to have familiarity with ad-hoc query language(s). Though the approach is easy to use and apply but it is not so expressive. A number of semantic search approaches reflect the use of keyword based queries [53, 68, 86, 87, 88, 89, 90, 91, 92] and natural language queries [93, 94, 95, 97, 98]. User intent may not be expressed so clearly using keyword based approach or natural language based approach. So a trade-off can be worked upon between easiness of keyword/natural language type query approach and expressiveness of structured query approach.

2.2.3 Indexing and query processing

External resources used for indexing and query processing consist of ontology and its corresponding KB. Unlike the traditional Web search systems where the inverted index generally contains keywords associated to the documents, in the conceptual perspective of semantic search approach, semantic entities associated to the documents are used for the purpose of finding inverted index where they appear rather than the keywords. The association between a semantic entity and a document is represented through annotation. Ontological languages like RDF and OWL can be used to represent metadata which can subsequently be used to index and retrieve related information. Indexing of the billion of entities, ontological elements of the KB is required to be done in a scalable manner so as to reduce response time and most appropriate matching of the KB elements with the refined query.

2.2.4 Result Presentation

How results are presented to the user finally so as to make it easy for him to reach the desired/intended result at the earliest. Ranking of documents with respect to query intention is observed to be a vital factor in this concern. It has been observed that the most of the semantic search approaches either do not find appropriate ranking strategy for the ranking of query results or use traditional ranking models based on conventional keyword based approaches [53]. A few ontology based approaches rank KB instances rather than documents with the support of available semantic information. In case of general conjunctive query, results are sets of tuples presented to the user in a structured, tabular manner.

2.2.5 Query expansion/refinement

Refinements to the query may be required for different reasons. Query input to the search system is required to be refined with augmentation of closely related entities, relationships etc with the original query terms so as to enhance the recall of the final search results. This would also help to more precisely address the user's information need addressing different query aspects in terms of sub-topics covered under main query topic. A large number of related terms can be generated out of which top k terms can be retained to be fired to the search system based on some ranking criteria. In other scenario, query results may not exactly match the information need due to poor description of the query. Here, semantic models like schema space may have been used describing different types, relations and attributes associated with the underlying resources. These so called facets provide a mean to the users for narrowing down or expanding the interesting resources as per their need.

Most of the surveyed systems exploit semantic technologies at one or more stages of the search process as described above. A brief description of most of such approaches is presented below which played a crucial role in achieving the next stage of research process leading to problem formulation:

SHOE*:* The semantic mark-up can be added to HTML pages using this tool called *SHOE knowledge annotator*. The instances, ontologies and the claims can be displayed through a special user interface as shown in Figure 2.1. There is a provision for user adding, editing or removing any of these objects. The mark-up added with the help of domain ontology can easily be crawled, indexed and queried. Ontology based complex queries can be constructed with the

help of a graphical user interface as provided by SHOE search [54]. The Web pages with SHOE annotation or mark-up are searched and crawled for extracting the mark-up. The mark-up is extracted and stored in a local knowledgebase with the help of a Web crawler called *Expos*. A form based tool facilitates for expressing additional search context through ontology navigation. A list of classes are generated and presented to the user to refine the search. User is able to issue the query filling the form whose values define the constraint rule for the resources to be retrieved. A conjunctive query is generated automatically and issued to the knowledge base. Manual semantic annotation becomes a bottleneck for the full realization of the approach though the presented architecture proves to be a good beginning in the direction of next generation semantic search system.

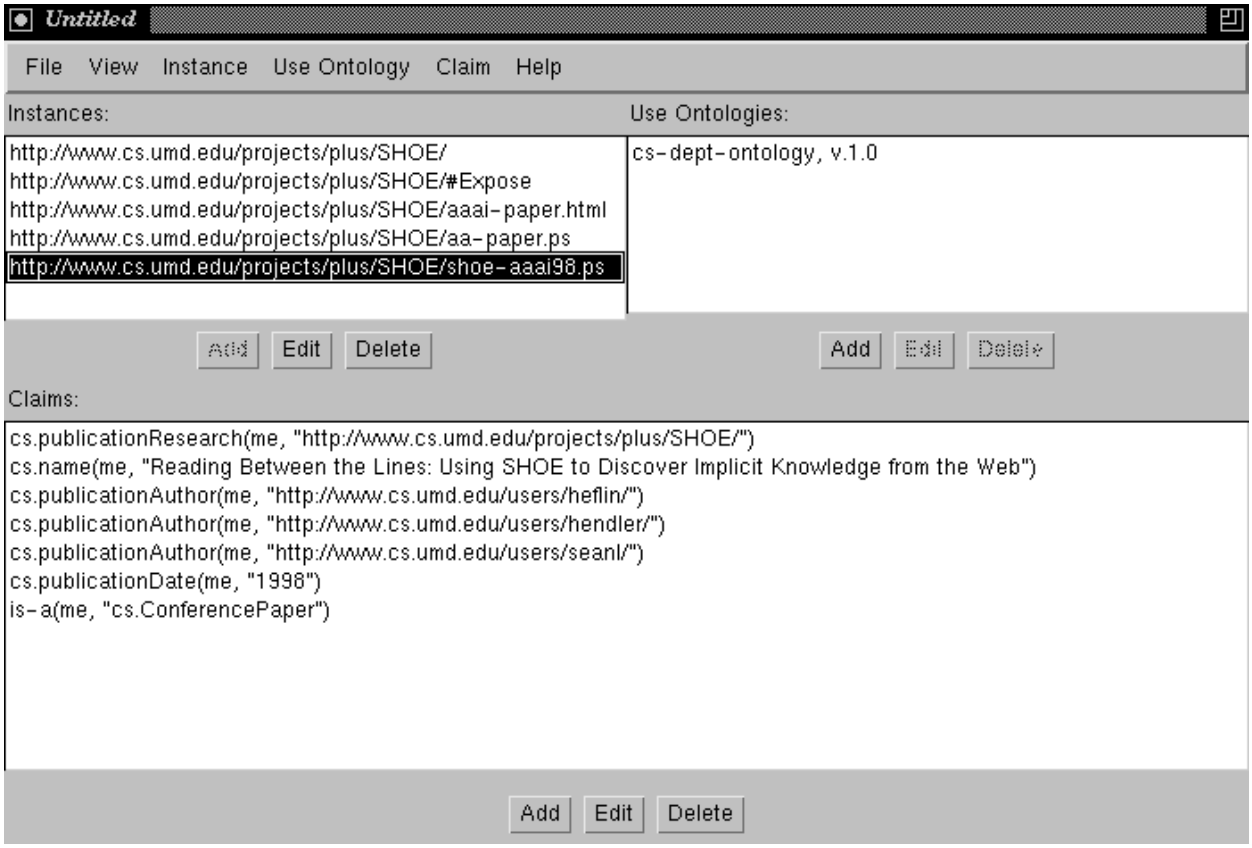

Figure 2.1: SHOE Knowledge annotator [54]

[Stuckenschmidt and Harmelen, 2001]: For Web-based information systems, metadata plays a significant role for searching, accessing and interpreting the information intelligently as observed by [55]. It has emphasized the specification of ontologies and metadata models in order to search for web resources with certain properties. It is a knowledge based approach for metadata

generation and validation. Search engine is supplied with metadata model as well as with the ontology which are used to filter the content based on the target concept specified by the user. Ontology provides the necessary vocabulary for mapping query concept to the concepts assigned to Web pages. Web pages are semi automatically classified based on their structure and are related to the existing ontology. The formal semantics of the ontology remains available for checking validity and filtering of Web pages. The approach relies on a specific ontology to which all the Web pages can be related. Hence its applicability on cross-domain arbitrary Web resources seems difficult to follow.

SEAL: The use of semantics had been explored in [56] for developing and maintaining the portal termed as SEmantic portAL (SEAL). The aim of this intranet application is the presentation of information to humans and software agents taking advantage of semantic composition. The main focus area of this framework is the ranking of search results motivated by the fact that one usually finds excess of information even with accurate semantic access. SEAL ranks the results of a search in such a manner that the results needing least number of inference steps from the original knowledge base are ranked highest. The method which is used for calculating the similarity measure between query results and the original KB (without axioms) does not provide clear justification and has not been validated experimentally.

OWLIR: OWLIR [50] is a system that retrieves documents containing both free text and semantic mark-up leveraging inference over the knowledge base. An integrated approach has been proposed combining the inference capability of Semantic Web and traditional information retrieval techniques. The element of inference is used at document indexing and query processing time leading to significant improvement in average precision in comparison to traditional full-text search engines as author reports. The system indexes RDF triples together with document content. OWLIR treats distinct RDF triples as indexing terms. RDF triples are generated by natural language processing techniques based on textual content which are used to enrich Web documents with semantic mark-up. These are also used by the inference mechanism for inferring additional semantic relations by reasoning over ontology instances and ontology hierarchy. This helps in the expansion of documents facilitating the answering of queries beyond the scope of only free text. Search can be performed in the form of query based on words and RDF triples with wild cards. The results retrieved by the system include semantic information as

in case of a query answering system and text documents as in IR system with no ranking of the final results.

SCORE: The classification and information extraction techniques have been used to extract metadata from textual sources in Semantic Content Organization and Retrieval Engine (SCORE) [57] for its use to do search semantically. There is provision for the user to specify the document category as well as one or more attribute values of the metadata while issuing the query.

QuizRDF: The searching and navigation through RDF(S) based annotations have been complimented with keyword based search in QuizRDF [58]. The approach overcomes the limitation of semantic incompleteness cropped up due to the absence of enough metadata by providing traditional free text search. User can refine queries using semantically related information by browsing over ontologies leading to better search precision and recall compared to traditional pure keyword based search. . RDF(S) is used to define and populate the domain ontologies in order to develop the search space. The resulting RDF annotations are also indexed along with full text of the annotated resources. The terms from the documents and the literals from RDF statements are referred to as *content descriptors* and used for indexing in QuizRDF. A query is formulated using terms. In addition the type of the intended resource can also be specified by allowing the user to navigate over the ontology. Search results can be filtered by property values of the resources. Simple ranking capability is reported with no evaluation.

PISTA: An approach to ranking of the relationships existing between two entities in the Semantic Web has been presented in [59]. The motive is to find and rank the semantic associations between the two entities in an RDF graph based on their importance. Relationships that span over several entities may be very important in domains such as national security enabling analysts to navigate the connections between disparate people, places and events. Based on this intuition, a test bed for querying semantic associations have been implemented in terms of an ontology for national security domain and a prototype named PISTA (Passenger Identification, Screening and Threat Analysis). Author expresses the need of a good ranking strategy in view of understanding the relevance of each of the semantic association as a result of query. Also user-oriented evaluation criterion needs to be incorporated in view of the subjective nature of user's interests.

TAP: TAP [53] is a Web based semantic search system where documents and concepts are treated as nodes of a big semantic network with resources correspond not only to media objects (like web documents, images) but also domain objects (like people, organizations, products and events) in the form of a Stanford KB developed by the authors. These resources and the relationships among resources are represented using RDF annotations linking resources to documents where they appear. It aims at enhancing search results from the WWW with data from the Semantic Web. It provides simple mechanism for web sites to publish data in semantic web form and allow applications to consume this data through a query interface called GetData. It performs a graph based search over RDF graph from the Web using a semi-structured query in order to extract the semantic information related to search results. It starts at one or more *anchor nodes* in the RDF graph which have to be mapped to the query terms. A breadth first search is performed in the RDF graph, collecting a predefined amount of triples. Optionally only links of a certain type are followed in traversing the RDF graph. This way, it attempts to augment the search results considering the search context and exploring closely related objects leading to better retrieval precision. The system also relies on conventional search system Google for carrying out keyword based search complementing the semantic search system. Semantic search capabilities are limited by size of queries leading to retrieval of considerable number of irrelevant results in the absence of specific information need. Both semantic search and conventional search are exclusive processes leading to the probability of redundancy of information.

[**Stojanovic et al., 2003]:** [60] presents an approach aimed at ranking search results of a semantic portal that exploits the explicitly shared semantics of the information supported by an ontology. There is a provision of defining the context of interest by the user in terms of a small subset of the concepts and relations of the whole RDF graph using an ad hoc language. Some "universal" and "user-defined" weights are assigned to each semantic relation based on the context as well as other parameters like *specificity* and *path length.* These weights along with some normalization coefficients are combined into a global formula. The value of coefficients is provided by the user and strictly query dependent. *Specificity* of the instance of a relation is higher, the less often the instances of the concepts in the relation are present in other instances of relations. In addition the inference process of the statements in terms of deduction from rules is taken into account for ranking results. Although approach seems to be promising for large

datasets related to any domain but it is shown to be tested on a small dataset of a particular domain. Also need for developing task-oriented strategies for calculating the relevance has been emphasized.

BioPatentMiner: An approach of ranking search results returned against the query entered on the Semantic Web has been presented in [61]. An adaptation of Kleinberg's HITS algorithm has been used for assigning weights to the nodes in the results graph. Instead of hub and authority scores, *subjectivity* and *objectivity* scores are calculated in order to rank the triples returned by an RDQL query. The type of relationships between resources is taken into account along with position in the class hierarchy for the purpose of assigning weights to the nodes. Inverse property frequency of a property is taken into account for the purpose of assigning weights to the edges in the results graph. Every result graph is evaluated based on the combined score attained after considering the weights assigned to all the nodes and edges in the result graph.

KIM: The task of automatic semantic annotation, indexing and retrieval of documents has been worked out through a Knowledge and Information Management framework and services as provided by KIM [62]. The textual content available in Web documents is enriched with metadata in the form of semantic annotation automatically. KIM extracts the information from the documents to map the words of the documents with ontology concepts. Documents are populated with identifiers for their corresponding mapping with the ontological concepts to serve the indexing purposes. Same identifier is used for the words having same meaning. Information need can be specified in terms of queries formed using entities/concepts and relations from the ontology that are expected in the documents. Scalability is one of the vital issues which needs to be addressed while working on the architecture of a knowledge management system like KIM to match the scale of the Web.

[Rocha et al., 2004]: A hybrid approach for searching in the Semantic Web [63] combines full text search with spreading activation search in ontology. Search starts with a keyword based query. Results to the full text search are instances from the ontology. Those instances are used to initiate a spreading activation search in the ontology to find additional instances. The process involves two main steps with first focused on generation of the search space where there is domain ontology, a set of weights defining the importance of ontological relations in that domain and an instance graph for each node comprising instance URI and all the values of its properties.

The second step is focused on the retrieval task. The query is expressed as initial set of keywords which have to be searched in the instance graph retrieving first set of instance satisfying the query. With the help of this initial set of instances along with initial activation values, spread activation techniques are used to find the related nodes in the ontology with no relative relevance in terms of ranking. As author claims, it is not possible to devise universal formula that proves to be the best for all application domains reflecting its domain specific nature. Also, all types of relations are considered to have same relative weight while relative importance of relation types based on context could have contributed towards expressing the contextual outlook of the answer set.

[Zhang et al., 2005]: [64] present an enhanced model for searching in semantic portals using text as well as semantic information. They integrate text based search with a fuzzy version of the description logic *ALC* and use ontology as background knowledge base. A search set to a query is represented as a class in the knowledgebase. A formal query modeled as a concept Q in Description Logic is used to search the portal KB. The retrieval status values (RSV) of the documents retrieved by text-based search for a query are used as the fuzziness degrees for the instances of this class. A textual representation is assigned to each node by considering their closest relationships in the graph in order to improve its applicability with IR techniques. The model enhances the capability of traditional search system by allowing searching instances of the ontology for keyword based queries too. For formal queries, only ontology instances are retrieves as answers. The model facilitates the different degrees of integration of keyword queries and formal queries.

Swoogle: A prototype of a semantic search engine is presented in [65] which help to search and rank Semantic Web Documents (SWDs). This follows a query independent approach for ranking purposes. Four types of semantic links between SWDs have been explored and different weights are assigned to them: i) Imports (A, B) : A imports all the contents of B ii) Uses-term (A, B) : Instead of importing all the terms of B, A uses some of the terms defined by B iii) Extends (A, B) : A extends the definitions of the terms defined by B iv) Asserts (A, B): A makes assertions about the terms defined by B. The ranking algorithm for the ranking of SWDs is termed as OntoRank [66]. The ranking score of an SWD is computed using an adaptation of PageRank algorithm. The approach uses domain-independent and query independent features for ranking. *Query context* remains unaddressed for ranking purposes.

SemRank: The issue of determining the relevance of relationships between entities in a knowledge base with respect to user's context is taken up in [67]. Different from contemporary views for ranking semantic associations or relationships that do not consider the situational aspect while approximating the relevance, it presents the view that relevance is situation dependent even for the same query and hence some flexibility should be built into the relevance models so that different orderings may be imposed on the same result set for the same query made in different situations. It presents an idea to rank search results based on how much information is conveyed by a result thereby giving a sense of how much information a user would gain by being informed about the existence of the result itself. To achieve their goal, authors define two measures, named "uniqueness" and "discrepancy" which allow accounting for the specificity or deviation of a particular result with respect to instances stored in the database. An added value of the SemRank is that in the computation of ranking, it exploits a so-called "Modulative Relevance Model" that is capable of taking into account the particular purpose for which a query has been submitted (e.g. conventional search or discovery search). Sample data is built on the schema involving university, banking, flight and organization domain. The query interface asks for the resources between which semantic associations have to be determined along with an additional sliding bar which can be used to adjust search mode without users having to manipulate different criteria values. Although it provides a flexible ranking approach that offers a variety of result orderings to choose as per the needs of the user, the empirical evaluation was done using a synthetically generated data set rather than existing RDF data collections. Query context has been taken into consideration for ranking purposes.

SemSearch: Semantic Web search from the point of view of the user's intent has been addressed in [68] where authors present two methodologies for capturing the user's information need by trying to formalize its mental model. It analyzes keywords provided during query definition, automatically associate related concepts and exploit the semantic knowledgebase to automatically formulate formal queries. The system hides the complexity of query processing by providing simple Google like interface and returning answers that are easy to understand. The various components of SemSearch are shown in Figure 2.2. The two important components are Semantic Entity Search Engine and Formal Query Construction Engine. Semantic Entity Search Engine is responsible for interpreting the keywords and matching those with ontological instances. Then the Formal Query Construction Engine takes as input the matched semantic

entities and provides a set of formal queries expressed as disjunction of number of RQL statements. The matching between keywords and semantic entities could have been improved by taking synsets of semantic entities into account which has been left in order to avoid computation cost overhead. As a result, mapping of keywords to class or relationship might lead to irrelevant information retrieval from the ontology instances.

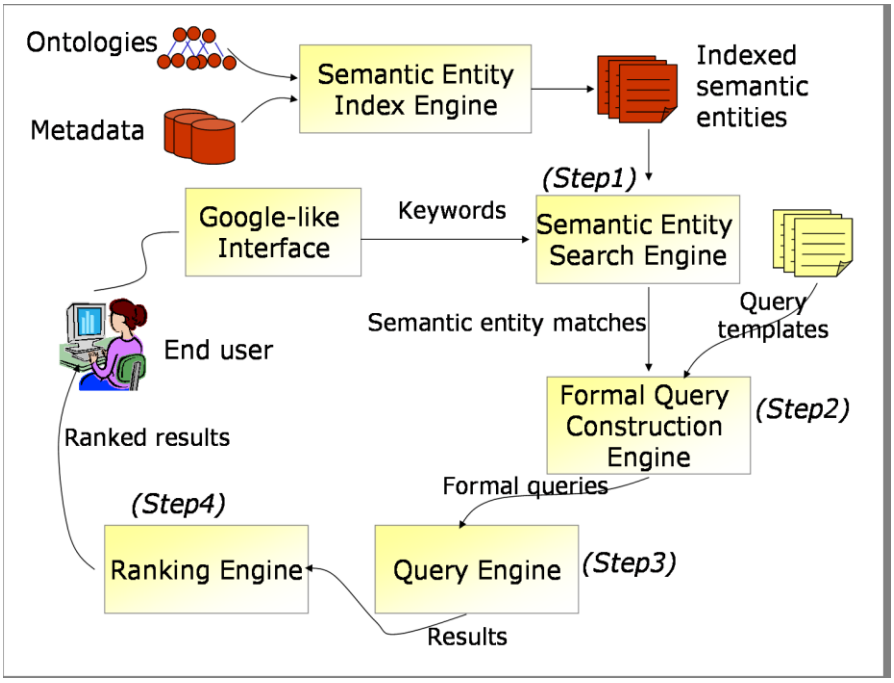

Figure 2.2: A comprehensive diagram of SemSearch search engine [68]

ObjectRank: [69] has idea of ranking objects in relational databases based on the principle inspired by PageRank. A number of parameters have been proposed to improve the relevance of search results to the keyword based user query. One is *specificity metric* to measure the keyword relatedness of the resulting objects. The other is *quality metric* to measure the global importance of the object independent of the query keyword. Two more parameters: i) importance of the results actually containing the query keywords ii) weight assigned to each query keyword have been explored although its implementation and validation is not shown. One interesting augmentation has been proposed to exploit the domain knowledge related to the given query. Domain based ontology graph is proposed to be integrated into their ObjectRank system which eventually enables to enhance the quality of query results. Although it seems to be a favorable approach but tested upon a relatively smaller dataset. It requires to be validated upon larger systems up to the scale of Web with ontology based approach.

[Castells et al., 2007]: Author proposes an ontology based information retrieval model using full-fledged domain ontologies and knowledge bases to support semantic search in document repositories [70]. Full documents rather than specific ontological instances are returned in response to a user query unlike Boolean semantic search systems. Once the list of documents is formed, the search engine computes a semantic similarity value between the query and each document. The added value of semantic information retrieval relies on additional explicit information-type, structure, relations, classification, and rules about the concepts referenced in the documents, represented in an ontology-based KB. The documents that are annotated with the returned instances are retrieved, ranked and presented to the user. The performance of the model significantly depends upon the amount and quality of information within the KB it runs upon. It is assumed by the author that annotations do not describe all the meaning conveyed by the document in a complete manner. Hence document retrieval phase leads to an approximate match. Further independently developed and maintained cross domain KBs can be integrated so as to deal with multiple heterogeneous data sets.

[Li et al, 2007]: The annotation of a Web page has been represented in the form of a graph where concepts and relationships are modeled as vertices and weighted edges respectively. There is a provision for representing multiple relationships existing between two concepts. The basic idea is to remove less relevant concepts from the graph with an intention to generate a candidate relation-keyword set (CRKS). This CRKS is submitted to an annotated database reducing the presence of uninteresting pages in the result set. The strategy behind prototypical system OntoLook [71] allows for empirically identifying relationships among concepts that are supposed to be less relevant with respect to user query. This information is used to reformulate the user query by including only a subset of all the possible relationships among concepts. The search scenario consists of user query, page annotation and underlying ontology leading to reduced cost of query answering as compared to whole semantic knowledgebase. Because of decentralized and heterogeneous nature of Web, it seems impossible that single ontology would serve for all the Web pages hence semantic communication among ontologies would be required. Moreover the weight of relations in forming the property – keyword candidate set also requires to be considered. It seems that the absence of an effective ranking strategy has greatly limited the scope of user satisfaction.

NAGA: [72] presents a semantic search system named NAGA. The back end consists of a knowledgebase YAGO [73] which is updated with facts extracted from the Web sources such as Wikipedia, Internet Movie Database (IMDB), World Facebook by exploiting different knowledge acquisition and information extraction techniques. A segment of knowledge graph YAGO is presented in Figure 2.3. A graph based query language is designed to support several different types of queries with additional semantic information named as *Discovery queries, Regular Expression queries and Relatedness queries*. An example of discovery query can be seen in Figure 2.4. Much of the syntax and some of the semantics for the query language have been derived from a structured query language named SPARQL. This new querying technique is simple to use yet more expressive than those provided by standard keyword based search systems. User has the choice of entering queries in text format through their user interface with results displayed in text format. Also, users can click on specific entities or facts in the retrieved results graphs and browse their neighborhood in the knowledgebase. A scoring model scores answer graphs based on certain parameters named *confidence, informativeness, compactness* which are integrated into a unified framework.

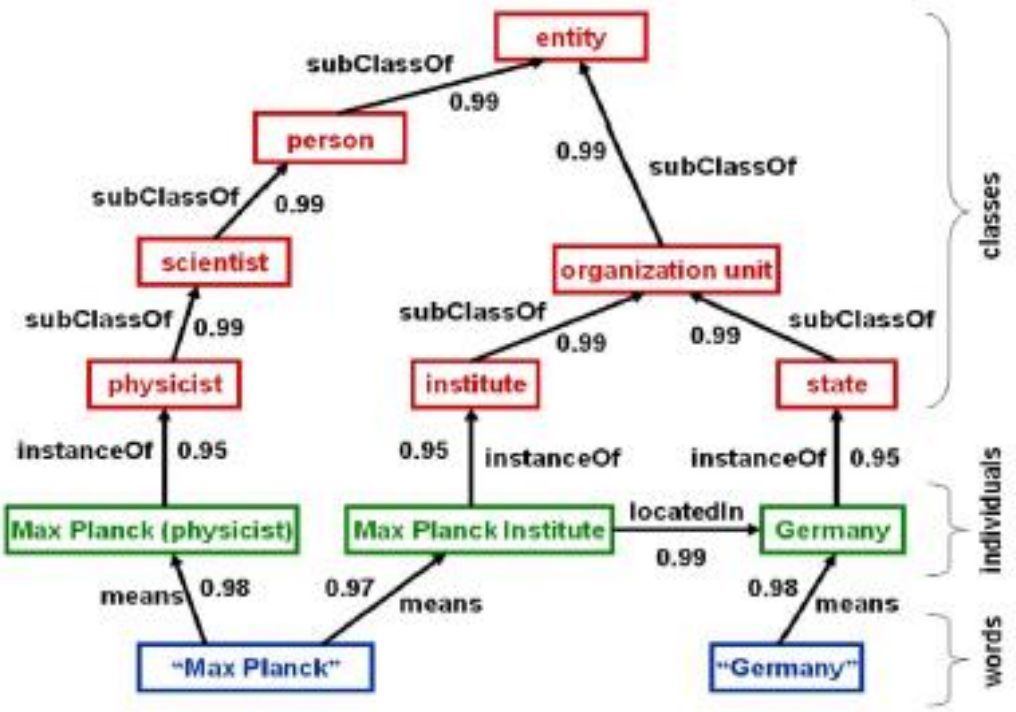

Figure 2.3: A segment of the knowledge graph YAGO [72]

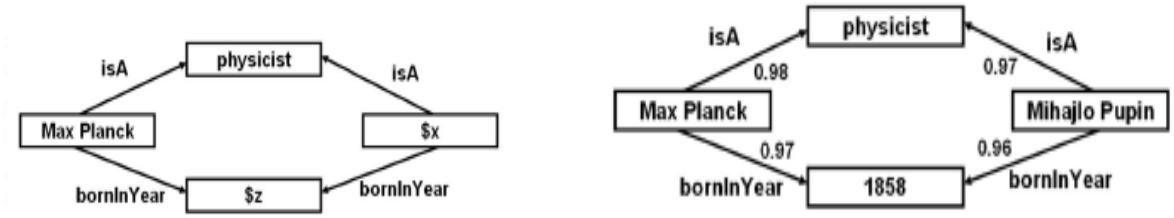

Figure 2.4: A discovery Query and answer to the query [72]

[Lamberti et al., 2009]: The approach works on defining a ranking criterion for Semantic Web pages returned against the query entered by the user. The basic idea [74] measures the probability that the relationships existing among the concepts within an annotated Web page are the same or similar to the ones in the user's mind while defining the query. Ontology, user query, annotated page containing the queried concepts are described in the form of ontology graph, query sub-graph and annotation graph respectively in order to effectively compute the probability. Ranking of the semantically annotated pages is realized based on the intuition that larger is the number of relations linking one concept with the other in a page, given the total number of relations among those concepts in the ontology, the higher is the probability that this page contains exactly the same relations as desired by the user. Hence it is treated as the most relevant page with respect to the user query. The cost of query answering seems to be reduced since only user query, the page annotation and underlying ontology are being used in the ranking process instead of whole semantic knowledgebase with billions of pages.

RDFKB: [75] presents a relational database system for RDF datasets which support inference and knowledge management. This is a framework that supports inference at data storage time rather than as part of query processing unlike most of the contemporary approaches supporting inference queries. Ontology serves as backbone for defining rules governing inference for RDF datasets. For each RDF tuple, all possible additional RDF tuples are inferred and added to the knowledge base with the help of a global function *add (subject, property, subject).* Authors' concern of increasing query processing performance at the cost of performance of adding knowledge to the database is well addressed in this manner. The knowledge base so managed supports queries in its original form without any need of including knowledge of inference. It has

also tested and validated the performance of knowledge management technique against existing solutions.

[Aleman-Meza et al., 2010]: In this approach, semantic relationships are the exclusive resource for ranking documents [51]. Unlike many other approaches, this approach does not exploit any specific structure in a document or links between documents for the purpose of ranking documents. Relevance of documents is determined using relationships those are known to exist between the entities in a populated ontology. A measure of relevance is introduced based on the traversal and semantics of relationships that link entities in an ontology. Actually the relevance measure is calculated based on the subjective knowledge by a domain expert who assigns "low/medium/high" scores to the relationship sequences by referring to the schema of ontology. Based on this score, degree of relatedness of so called *match entity* with other entities existing in the ontology is found out. Finally the score of a document is determined depending on how many of its annotations belong to such related entity set. Although it is a novel approach of exploiting semantics of relationships for finding the relevancy of documents, a poorly populated ontology may greatly limit the effectiveness of semantic annotation step and in turn the retrieval step. In addition, a domain expert manually assigns the weights to relationship sequences. Although this is done for once and without the knowledge of end-user still it is felt that automation of this process is required.

RareRank: This is a domain specific and query specific approach for approximation of popularity of a page/resource. While classical IR models exclusively rank documents based on content (relevance) and link analysis based methods emphasize link structures (quality), [76] attempts to integrate the two scores: relevance score and quality score coherently and with proper tuning of parameters which is often tedious and generally missing. Following a so called Rational Research Model, it has taken a Knowledge base in Research domain (consisting of instances such as publication, author, and journal or conference) which is represented as a directed graph. Then domain topic ontology is plugged into the graph. The derived Ranking score integrates both relevance (using Domain topic Ontology) and quality (using citation links).The model generalizes the previous link analysis based methods which only aims at ranking documents. Hence it can be utilized by semantic search systems for the purpose of ranking Entities. By doing so, it attempts to bridge the gap of domain specificity which used to remain unaddressed while using PageRank – like algorithm for semantic based ranking. There is

a potential scope left in parameter tuning although tuning procedure adopted in the model is simple.

[Artz et al., 2007]: There are mechanisms for verifying the authenticity of the information sources with respect to the claims they made. The validity of the information brings the content of trust with respect to the mechanism followed. Another dimension to the trust adds when software agents and reasoners need to prioritize the one source of information over the other in case of alternative sources of the same information. The role of trust in order to visualize the vision of Semantic Web has been studied in [77]. In policy based trust scenario, reputation based policy concerns play a significant role in order to obtain trust. Policies are usually based on the verification of credentials e.g. degree holder of a university with certain credentials has a certain education level. In reputation based trust scenario, evaluation is based on evaluator's interactive experiences with the entity either directly or as reported by some other trusted source. Usually the problem occurs while considering the trust information from other sources recursively.

[P. Hasse, 2010]: The need for transition from linked document structure to linked data structure of the Web was felt in order to answer complex Web queries where the answer is not addressed by a single source.

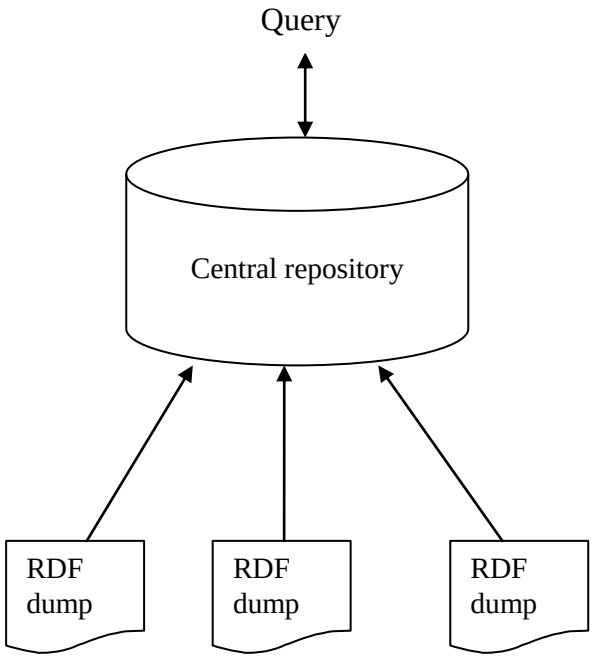

Figure 2.5: Integration in a central repository [78]

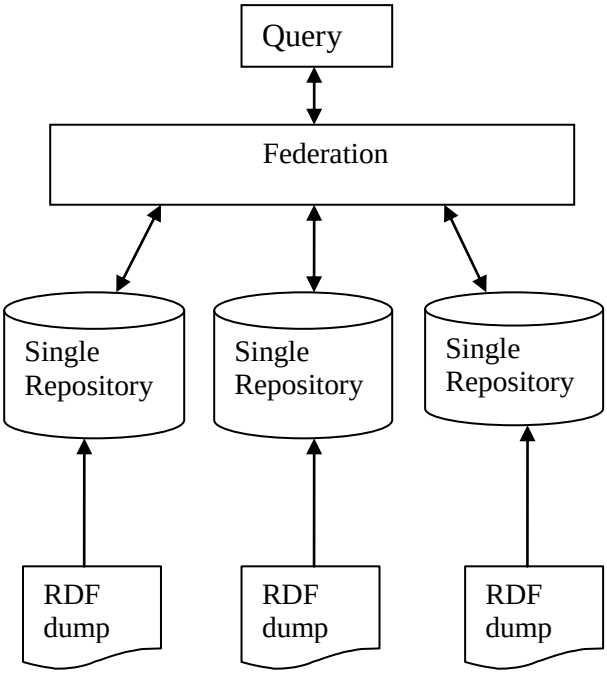

Figure 2.6: Federation over multiple single repositories [78]

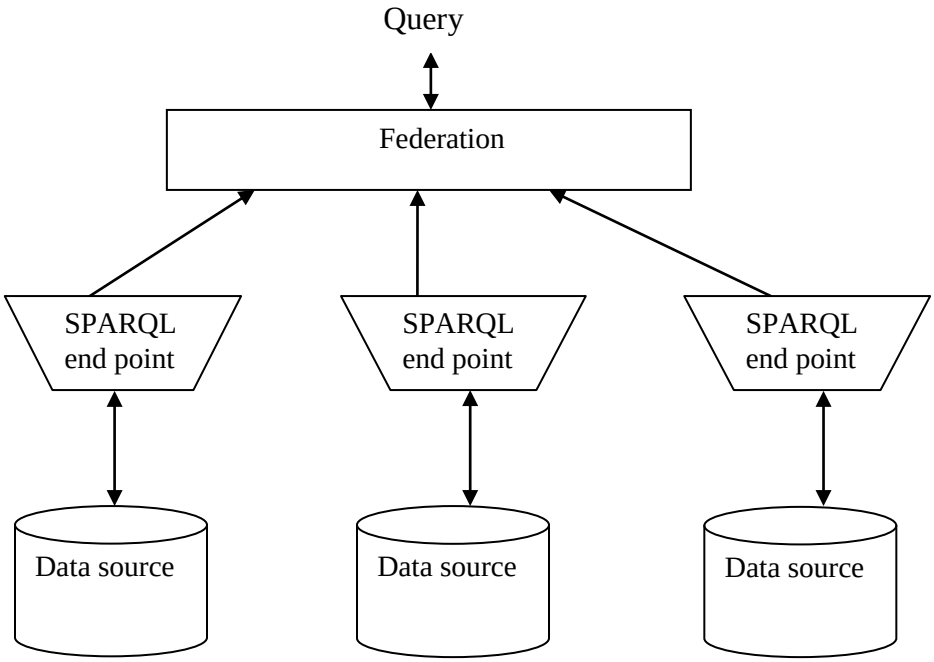

Figure 2.7: Federation over multiple SPARQL end points [78]

One contemporary solution comes in the form of populating the whole data from multiple, linked data sources into a single RDF graph repository and processing the queries in a centralized manner against this merged data set. This seems to be practically infeasible and less efficient for dealing with data on the scale of Web. This work [78] analyzes the applicability of the alternative approaches leading to federated query processing over linked data. The effect of different design alternatives on the performance and practicality of query processing have also been analyzed in terms of relative advantages and disadvantages. A benchmark has also been defined for evaluating the federated query processing. Data sources belonging to various diversified domains have been taken along with the corresponding set of representative queries. Various approaches to federated query processing can be seen in Figure 2.5, Figure 2.6 and Figure 2.7.

SEALS: One of the first effort to present a comprehensive evaluation methodology for semantic search tools has been presented in [79] as claimed by author. Tools are evaluated in 2 phases according to the methodology: i) Automated phase ii) user-in-the-loop phase. Outcome of these two phases allow benchmarking the tool both in terms of raw performance and also in terms of ease with which tool can be used. Different sets of *metrics, datasets,* and *test questions* are taken into consideration for evaluating different participating tools for two phases. Different tools participate in user-in-the-loop (UITL) phase and automated phase for evaluation based on *query expressiveness, usability, scalability* and *performance.*

In this section, a brief overview of semantic search on Web from conceptual perspective has been presented. It has been seen that there is a significant role of ontology to fulfill the aim of helping automate tasks where there is the requirement of a certain level of conceptual understanding of the involved objects (e.g. information objects) or the task itself. This intends to work towards enabling software programs to automatically find and combine information and resources in consistent ways. The Semantic Web vision was conceived with this objective in mind and the ontology was treated as key element to represent conceptual knowledge which is easy to understand, use and share among various applications and agents. Hence, in the direction of semantic search on Web, the search on the Semantic Web is integrated with the semantic search initiatives in order to add semantics to Web queries and the content while performing Web search. However it is understood that although the next generation Web architecture represented by the Semantic Web will provide adequate tools for improving the search

mechanisms in order to enhance the user satisfaction of information need but full potential of acquired knowledge, theoretical and technical achievements made through several years of work is yet to be exploited. The limitations of ontology based search systems play a crucial role in this direction viz. the automatic conversion of natural language queries into formal ontological queries, automatic addition of semantic annotations to Web content and automatic extraction of knowledge from Web, evolution and maintenance of underlying ontologies and mapping among ontologies. Another important issue is how to consider implicit and explicit contextual information to adapt the search results to the needs of the users. For instance, the needs and motivations of users may be defined in terms of ontology-based weighted constraints based on preferences (e.g., similar to [100]), which may then implicitly be expanded into the semantic search query and/or used in the computation of the ranking on objects and search results.

Some open issues related to ontology based search systems

1. Maintaining ontologies will be a real challenge. Since new words are constantly being created as well as new senses are assigned to existing words. To update ontologies regularly and consistently will be a real challenge for maintaining the completeness of the ontology.
2. Differently published ontologies pertaining to the same domain; to estimate the reliability of ontology and its publisher will be required before using it.
3. Issue of Ontological terms mapping will also crop up like mapping of class “car” in one ontology with “automobile” in the other.
4. In heterogeneous Web environment, there is a need for the system to move between ontologies without any need for domain specific reconfiguration, again a big challenge.
5. Evaluation benchmarks have not been standardized as yet in case of conceptual perspective of semantic search systems in terms of ontology based semantic search.

2.3 Semantic search system: A linguistic perspective

Numerous information retrieval and natural language processing applications exploit the knowledge of semantic relatedness among different words. For the purpose of semantically searching the desired information on the Web, finding relatedness among different words is a relatively novel aspect. Most of the available approaches use manually refined language

resources for the purpose of calculating semantic association between words. These resources serve as a backbone for measuring semantic association between words or concepts where information is encoded in structured, semi-structured or unstructured format. In one format, graph of concepts is formed which are interconnected by links or edges denoting a certain semantic relation occurring between the participating concepts. In taxonomical structure, hierarchy of nodes is formed in the form of generalization-specialization relationship. In ontological structure, semantic relationship space enriched with other diverse types of relations such as antonymy, synonymy, class attributes and axioms are organized in order to enhance reasoning capability over the information. In semantic graph or network structure, any types of concept graphs are connected by any semantic or loosely associative relations. It can be said that taxonomy is a special case of ontology and the both are specialized semantic graphs. In another scenario, context of a word can be derived in the form of finding functionally related terms to the original term with the help of a large corpus. In such cases, large documents are used to form a general purpose or special purpose corpus of words where words or concepts are not organized in structured format encoding their semantic association like semantic graph or ontology. Also sense-tagging of words is not provided in the documents of information and lexical semantic relations are defined in implicit manner.

WordNet is one of the language resources in the form of ontology of English words where nouns, verbs, adjectives and adverbs are grouped into *synsets*, each expressing a distinct concept. It has become one of the most popular background knowledge resources for conducting the studies of semantic relatedness. It intends to cover common English words of general use in a complete manner with little coverage of terms from specialized domains and very limited coverage of proper nouns. The richness of the resource in terms of vocabulary coverage is of great significance for efficiently finding the relatedness among words. It is not feasible to use these methods for the words which are not available in the knowledge resource repository. General purpose language resources tend to suffer from this problem of adequate term coverage due to which domain specific knowledge resources are constructed for specific domains. Such domain specific resources are generally preferred over general purpose resources in view of their comprehensive coverage of specific domain vocabulary. MeSH (Medical Subject Headings), GO (Gene Ontology) and UMLS (Unified Medical Language System) are some of the widely used knowledge bases for measuring semantic relatedness in biomedical domain. Recently, the Web

has been observed to be used as an ideal knowledge resource for harvesting semantic relatedness with unseen relationships in view of its heterogeneous nature. It covers a variety of words and concepts from general domains like news, sports etc. to highly specialized medical terminology. Also new words and senses of the existing words are added frequently and efficiently. These points motivate to use Web as a fertile ground for automatically acquiring semantic knowledge covering a large number of domains.

It is observed that the potential of query dependent features seems to have largely remained untapped for the purpose of semantic search. The role of lexical semantics in this direction seems to show promising results for effective search on Web. It can improve the performance of search systems using different kind of resources available for lexical semantic knowledge and using different methods to embed that knowledge into search process. Actually there exists a vocabulary gap between user query and relevant documents which can be reduced by adding the relevant terms to the query using lexical semantic knowledge [26]. Vocabulary coverage of such knowledge-bases is a crucial factor for effective expansion of query leading to desired information retrieval. Knowledge-bases can be classified into 3 broad categories: i) Semantic network/Ontology based ii) Corpus based iii) Web based.

2.3.1 Semantic network/ontology as a knowledge base

Ontology is a "specification of a conceptualization" which was first used in philosophy rather than artificial intelligence as explained in [27]. Ontologies provide consistent vocabularies and word representations which help in unambiguous communication within knowledge domains [28]. A word may have various meanings based on the context where it is used. Ontology helps to disambiguate the meaning of a word by providing a context for the vocabulary it contains.

A number of measures are available in literature where a hierarchical semantic network is used for the purpose of calculating semantic relatedness between a pair of words. Many of them have been experimented with WordNet as a knowledgebase. The noun network of the WordNet was the first to be richly developed and most of the surveyed works have restricted its experiments to this network. The central idea of this network is the subsumption hierarchy (hyponymy/hypernymy) covering maximum number of available relationships. At the top of hierarchy, there are 11 abstract concepts with maximum depth of the noun hierarchy is 16 nodes. Along with an implicit synonymy relationship associated with each node, there are nine types of

relationships on the noun sub-network viz. the hyponymy (*IS_A*) relation and its inverse; six meronymic (*PART_OF*) relations: *COMPONENT_OF*, *MEMBER_OF*, *SUBSTANCE_OF* and their inverses; and antonymy, the *COMPLEMENT_OF* relation. In most of the works, a pair of concepts is taken as input and relatedness between those concepts is returned quantitatively.

The *length* of the shortest path between two conceps/synsets c_i and c_j in terms of edges is denoted as $len(c_i, c_j)$. A root can be implanted above 11 unique concepts to materialize the path between any two nodes. The *depth* of a node is the length of the path from root to the node as $depth\ (c_i) = len\ (root, c_i)$. $lso(c_1, c_2)$ can be denoted for the *lowest super ordinate* or *most specific common subsumer* of c_1 and c_2. Irrespective of the method for calculating the relatedness between two concepts c_1 and c_2, the relatedness between two words w_1 and w_2 can be derived as $rel(w_1, w_2) = \max_{c_1 \in s(w_1), c_2 \in s(w_2)}[rel(c_1, c_2)]$ where $s(w_i)$ is "the set of concepts in the taxonomy that are senses of the word w_i" [101]. That is, the relatedness between two words can be represented as that of the most related pair of concepts they denote.

One of the first efforts for calculating the semantic relatedness between concepts adapted the semantic distance algorithm from *Roget's Thesaurus* to WordNet. Hirst and St-Onge's approach [44] works on the principle that two words are associated if there exists a so-called allowable path connecting a synset associated with each word. "The longer the path and more the changes of direction, the lower the weight of association" is the intuition behind it. For two WordNet concepts c1 and c2, the formula devised following the approach is: rel_{HS} (c1,c2) = C – len(c1, c2) – k * turns(c1, c2) where C and k are constants assigned values of 8 and 1 respectively in practice and turns(c1, c2) is the number of times the path between c1 and c2 changes direction. An illustration of path turns has been shown in Figure 2.8 where in two hierarchies, path from A to D has one turn in the left hand side and it has no turn in that on the right hand side. Such edge counting approaches assume that the edges in the taxonomy represent uniform distances while it may not be true in situations where certain sub-taxonomies are much denser than the others [101]. Leacock and Chodorow [37] proposed a formula for computing the scaled semantic similarity between two concepts c1 and c2 in WordNet:

$$Sim_{LC}(c1, c2) = -log \frac{len(c1,c2)}{2 \times \max\limits_{c \in WordNet} depth(c)} \quad (2.1)$$

where denominator includes the maximum depth of hierarchy.

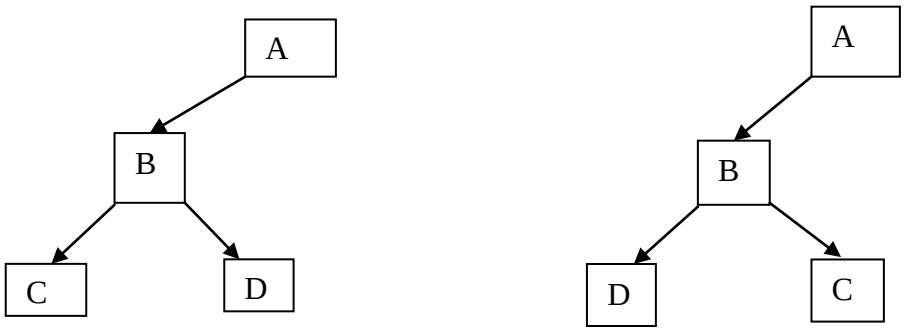

Figure 2.8: An illustration of path turns from A to D

Resnik's information-based approach [101] is based on the intuition that similarity between two concepts is the extent to which they share information in common which is defined as:

$Sim_R(c1, c2) = -\log p(lso(c1, c2))$,

lso → lowest common super – ordinate

The probability of a concept in the taxonomy was measured from noun frequencies collected from the Brown corpus of American English. The individual occurrence of any noun in the corpus contributes towards the occurrence of each taxonomic class containing it. Hence p(c), the probability of encountering an instance of concept c = $\frac{\sum_{w \in W(c)} count(w)}{N}$, where w(c) is the set of words(nouns) in the corpus whose senses are subsumed by the concept c, and N is the total number of word (noun) tokens in the corpus that are also present in the WordNet. The higher the position of lowest common super-ordinate for the given concepts in the taxonomy, the lower is their similarity. The approach deals with the problem of varying edge weights as seen in edge counting approach since number of edges is not counted in the formula for calculation of similarity. But the selective use of the taxonomical structure does not allow for distinguishing semantic distance between two different pairs of concepts with same lowest common super-ordinate. Reacting to the limitations of Resnik's method, Jiang and Conrath [102] presented an idea of combining edge and node based techniques resulting in the formula: $dist_{JC}(c1,c2)$ = IC(c1) + IC(c2) – 2 × IC(lso(c1,c2)) = 2log p(lso(c1, c2)) – (logp(c1) + log p(c2)).

Observing the nature of available similarity measures which were domain dependent in general, Lin [103] contributed towards defining a universal and theoretically justified measure of similarity. His measure of similarity between two concepts in a taxonomy is a corollary of a similarity theorem: " The similarity between A and B is measured by the ratio between the

amount of information needed to state their commonality and the information needed to fully describe what they are", that is $Sim_L(A,B) = \frac{logp(comm(A,B))}{logp(descr(A,B)}$. Hence similarity between two concepts c1 and c2, $Sim_L(c1, c2) = \frac{2 \times log\ p(lso(c1,c2))}{logp(c1)+logp(c2)}$.

Taking query context as one of the dominating criterion for the purpose of semantic search on Web, the words which are semantically related to the query words can be found out using ontology or semantic network as a knowledge base as described above. This will help the user to reformulate the query in an appropriate manner in order to yield good search performance satisfying user's need. First experiments on query expansion by [29] using lexical semantic relations extracted from an ontological knowledgebase namely WordNet[30] showed significant improvement in performance only for manually selected expansion terms. It was seen by [31] on several test collections that the combination of WordNet and thesauri built from the underlying document collections improved the performance. It also concluded that while using only WordNet as knowledge base, performance cannot be improved to that extent due to missing relationships especially cross part of speech relationships and insufficient lexical coverage.

Another issue realized while using ontologies for semantically expanding the query is vocabulary mismatch between query terms and ontological concepts. An additional effort is required to be put up for mapping query terms to ontological concepts. A feature set of ontology to justify its lower utility as a knowledge-base for semantic search purposes is presented below:

Nature of semantic association

Semantic Relatedness indicates the degree to which words are associated via any type (such as synonymy, meronymy, hyponymy, hypernymy, functional, associative and other types) of semantic relationships. *Semantic similarity* is a special case of relatedness and takes into consideration only hyponymy/hypernymy/meronymy relations. A measure of semantic similarity uses only hierarchical and equivalence relations of the semantic network e.g. Teacher and Examination are *not very similar* since very few features of entity *Teacher* and *Examination* are in common but *closely related* in a functional context namely that teacher *evaluates_through* examination. Whereas for the purpose of semantic search on Web, semantic relatedness is supposed to be of greater importance than semantic similarity, semantic network/ontology based measures are not suitable for measuring relatedness between words. They are able to identify the

amount of relatedness only if such relations are inherent in ontology. If agent-instrument relation does not link concepts in a semantic network (as in WordNet), ontology measures will not be able to identify relations like Teacher – Examination. Moreover, they usually include noun relations like names of persons, places and organizations. Cross part of speech relations like noun-verb or noun-adjective relations remain missing from ontological knowledge-bases.

Creation and updation of ontology

A lot of effort is required to construct ontology from scratch. In addition to the technical issues involved in its creation, the process of knowledge extraction from domain experts and arriving at a consensus view is very difficult and time intensive. Also, there usually remains a lag between current state of language usage/comprehension and semantic network representing it. Usually it becomes quite difficult to make changes in language.

Domain specificity

Semantic models trained on feature sets containing graph based features reflect greater inconsistencies than those of text based features when tested on different domains [32]. Usually, ontologies represent specific domains so semantic similarity is measured with respect to domain data. Hence the use of ontologies and similar semantic networks for domain independent applications is not so useful. Two words may be semantically very similar with respect to a certain domain, not otherwise e.g. *Time* and *Space* are closely related in the domain of quantum mechanics but not so much in other cases.

2.3.2 Corpus as a knowledgebase

Here the corpus has been referred to be a large set of document collections which can be used to find semantic relatedness between words applying certain methods. The reference knowledge is collected at the level of words rather than concepts in the absence of sense-tagging of words. Unlike ontology or knowledgebase, lexical semantic relations between words are provided implicitly. The central idea is to find the textual context of the words based on their co-occurring behaviors referring a sufficiently large document resource to the order of millions of words belonging to various topics. A large number of such document resources have been compiled for use in various applications of semantic relatedness and searching. Some of the most widely used are the Brown corpus [104], the British National Corpus [105], the Penn Treebank corpus [106],

the Reuters corpus [107] and Wall Street Journal (WSJ); some of these corpora have been archived by Herman and Liberman [108].

Given a text corpus, context of a word is composed of words co-occurring with it within a certain window around it. *Distributional Measures* use statistics acquired from a large text corpora to determine how similar the contexts of two words are. These measures give a fair idea of semantic similarity as words found in similar contexts tend to be semantically similar as per distributional hypothesis [110]. Such measures have originally been referred to as measures of distributional similarity which can be fairly used to mimic lexical semantic relatedness occurring between two words. Whereas semantic relatedness is concerned with relations on the concepts, distributional similarity focuses on the relations between words. Also the former uses semantic network or ontology as knowledgebase whereas the later relies on a corpus as a knowledgebase [109]. A table of comparison between ontology based measures of semantic relatedness and corpus based measures of semantic relatedness is shown in Table 2.2.

Table 2.2: A comparison between ontology based measures of semantic relatedness and corpus based measures of semantic relatedness

Parameter	**Ontology based measures of semantic relatedness**	**Corpus based measures of semantic relatedness**
Goal	Ability to mimic the human judgment of semantic relatedness	
Background Knowledgebase	Semantic Network/Ontology(e.g. WordNet)	A large text corpus (e.g. Wall Street Journal or American Printing House for the Blind(APHB) corpus)
Capability (to identify Semantic association)	Able to identify semantic similarity, generally.	Able to identify both Semantic Similarity and Semantic Relatedness.
Advantage	**Powerful knowledge source**: A rich source of information with various concepts linked together by powerful relations: hyponymy and meronymy.	**Wider applicability:** can be effectively applied in Resource-poor languages also.

Limitation(s)	• Creating/updating ontology is very expensive, time intensive. • Goodness of a measure depends on richness of the resource used.	Clearly related word pairs may be assigned low similarity values due to data sparseness.
Word Sense Approach	Applied to particular sense of word (concept). E.g. semantic relatedness of bank (financial institution sense) with interest(interest rate sense)	Applied to word pairs irrespective of their word sense (or nature of polysemy) or particular sense they have been used in.

The sense that a word takes in one context is defined by its neighboring words. Therefore a corpus can help in collecting the statistical information that can be used to define the senses that a word takes at each occurrence using distributional similarity methods as in [114, 115]. The sentences within a corpus define the relations between their words comprising the sentences. E.g. the frequent occurrences of some specific words in several sentences will reflect a stronger relation between these words.

Using corpora over thesaurus or semantic network would help in defining more complete relations between words. This is because of the fact that this would include any type of relations which may exist between two words rather than usual hyponymy/synonymy relations which is usually found in ontology/semantic networks. Large corpora of more formal writing are available publically e.g. Wall Street Journal (WSJ) or American Printing House for the Blind (APHB) corpus. Another approach for building domain specific corpora was investigated during late 1960s and 70s by many researchers. Based on the assumption that similar documents are relevant to the same requests, similar documents are placed in one cluster. The terms from the specific cluster(s) were used for finding additional relevant terms to the query if the query terms mapped onto one or more clusters. It is noted that poor cluster formation due to small document collections [33] and insufficient differences in vocabulary between relevant and non-relevant documents [34] were observed as limiting factors of this approach.

The lexical richness of the corpus is a significant factor for finding out the diverse relations between different words. Sometimes clearly related word pairs may be assigned low similarity values due to data sparseness. Although it may be true that building corpora require lesser time

and efforts in comparison to building ontology but it seems very cumbersome to build or find content-rich corpora pertaining to every domain.

2.3.3 Web as a knowledgebase

The Web is a multilingual repository having temporal and location character as well. Due to the addition of new words frequently and efficiently, it reflects the maximum coverage of possible interpretations of the words. The Web covers a huge set of domains ranging from news articles and blogs to highly specialize medical terminology. Despite less number of domain expert users, average Web content hides meaningful implicit semantics making it a fertile ground for automatic semantic knowledge acquisition. Consequently, it can be used as an ideal knowledge source for mining semantic relationships between a pair of words.

It is seen in the recent past that corpora compiled over the Internet has been experimented easily. Projects at Linguistic Data Consortium (LDC) and The European Language Resources Association (ELRA) have significantly contributed to the availability of such resources. An increasing number of approaches [35, 112, and 113] have been observed exploring the Web as background knowledgebase for measuring distributional similarity between a pair of words. Web based metrics use Web search engine content in order to find out relationships between a pair of words. Such approaches use the Web search results returned by a search engine considering the number of times words co-occur in the documents indexed by a search engine like Normalized Google Distance (NGD) [35] and PMI-IR [36]. In another approach, the distributional similarity between two words w1 and w2 is calculated in [112] using statistics collected from search engine results. Number of pages containing w1 is counted as *freq (w1)* after entering w1 as query to a standard search engine and number of pages containing w2 is counted as *freq (w2)* after entering w2 as query. Then, number of pages for a query concatenating both the words is counted as *freq (w1, w2)*. These figures are used to measure Point-wise Mutual Information (PMI), the strength of association between two words considering the probability of occurrence of each word in the corpus and probability of occurrence of both the words together in the corpus. Apparently the whole web page has been taken as context of the target word. Another interesting dimension can be explored in terms of finding interesting patterns or trends by extracting association rules similar to [111] from voluminous Web data.

Using Web as a knowledgebase does not confine to a specific domain for the purpose of calculating semantic relatedness making it universally workable without concerning the final purpose. On the other hand, it warrants the maximum coverage of possible interpretations of the existing words in the highly dynamic context of the Web. However, most of the approaches seem to suffer from the problems inculcated due to search engine behavior like limited query syntax and little weight to relative position and frequency of words in a page [116]. A comparison among three approaches of using background knowledgebase for the purpose of semantic search on Web is presented in Table 2.3.

Table 2.3: A comparison of knowledge-bases with respect to distinctive set of characteristics

Parameter/Characteristic	**Approach for measuring query word semantic association**		
	Case I: Semantic Network as a Knowledgebase	**Case II: Corpus as a knowledgebase**	**Case III: Web as a knowledgebase**
Vocabulary coverage	Lack domain-specific vocabulary	High coverage of general and domain-specific terms	High coverage of general and domain-specific terms
Scope	Usually Domain-specific	Can be either domain-specific or domain-independent	Usually Domain-independent
Updation	Regular updation not feasible	Not feasible	Automatic
Nature of semantic association	Semantic similarity	Semantic relatedness	Semantic relatedness
Nature of relations	Mainly noun relations	Cross POS(part of speech) relations	Cross POS (part of speech) relations
Computation cost	High	Moderate	Less
Language dependency	Language dependent	Language independent	Language independent
Temporal sensitivity	No	No	Yes
Location sensitivity	No	No	Yes
Relevance criteria	Domain Experts assessment	Expert assessment	Highly sophisticated Ranking algorithms like PageRank

2.4 Problem formulation

2.4.1 Research gaps

In this section, some open issues have been summarized which are considered to be relevant with respect to efficient semantic search. It is thought that a concise discussion on these issues will facilitate to catalyze further research in this field.

Query interface

Four modes of user interaction with the system have been observed for expressing its intent of search. Those are: i) Keyword based ii) Form based iii) Natural language based iv) Structure query language based (e.g. SPARQL). Query expressiveness can be enhanced to a great extent using structured query based approach but a general user may not be willing to learn structured query language. He is comfortable with keyword based approach which is easy to use but not so expressive. User intent may not be expressed so clearly using keyword based approach. So a trade-off is required between easiness of keyword type query approach and expressiveness of structured query approach. One of the solutions may be to offer a range of different modes of search formulation, to allow users to pick the method that best suits their task.

Heterogeneity

Semantic search systems are supposed to find answers to user queries by directly returning information or knowledge on entities in an efficient manner. Many a times, multiple ontologies are likely to be referred to satisfy the needs of complex user queries. The search system must be able to search several different domains at the same time. For example, the term 'apple' may occur in fruit ontology and computer ontology. The search system would need to investigate all of these cases and eliminate irrelevant ones. If necessary, it might also have to integrate the knowledge structured according to different ontologies together. However, it is observed that many of the surveyed approaches have relied upon an ontology related to a single domain for finding the relevant resources related to the user query. The contemporary major solution to this problem is proposed in the form of "Linked Open Data". An extended discussion of related issues regarding open semantic environments has also been presented by [99].

Query Context

It is noted that in many of the surveyed approaches, query specific features have been paid little attention for the purpose of approximating the relevance of Web resources. Although global importance of resources play a vital role in approximating its relevance to user query but it is asserted that query context has to be treated as one of the dominating criterion for ranking resources semantically. However, it seems to be an open issue how to map query context to ontologies with unknown structure for the purpose of semantic search on Web with conceptual perspective. In most of the approaches with linguistic perspective also, the context has been seen to be treated in a broader sense leaving further scope of improvement in addressing this significant criterion. As one other dimension, the aspect of *personalization* can be explored from efficient *context retrieval* point of view. In this line, an effective approach for identifying user's interests based on the mining of user's search logs has been presented in [96]. History of a particular user browsing patterns, information demands etc, combined with ad-hoc/instant context expression by the user, can be exploited to reach to its intention to the highest closeness.

Portability

The available ontologies often exhibit different conceptualizations of similar or overlapping domains. One of the challenging tasks for efficient semantic search on Web is the integration of ontologies with the purpose of building a common ontology for all Web sources and consumers in a domain. This will facilitate the system to move across ontologies without any need for domain-specific reconfiguration. This can be done by detecting semantic relations between concepts, properties or instances of two ontologies, i.e. ontology matching. This is not only important concerning the portability across ontologies related to a domain but also regards as an important step towards domain-independent heterogeneous knowledge bases.

Scalability

Efficient implementation of semantic search systems from the point of view of indexing time, index space, and response time is required to compete with contemporary search engines. Only a little overhead may be introduced as compared to standard search systems. Only a few works have been found out on the performance of semantic search systems on corpora as large as Web.

Evaluation Benchmarks

Semantic search systems have started taking shape which ultimately aims at a human-like interface to the knowledge and services available on the Web. Despite this fact, SW community is still a long way to go for defining standard evaluation benchmarks to judge the quality of semantic based search methods. Systematic evaluation of semantic search tools involve appropriate test collection of data and queries, standard performance criteria and independent judgments of performance, thus, supporting performance comparisons between systems. Present approaches for semantic search evaluation are mostly based on user-centric methods, small scale and difficult to repeat. SEALS project [79] seems to be a good initiative in the direction of providing such benchmarks.

2.4.2 Problem identification

On-line search engines facilitate the general user to locate and access the desired information from huge number of sources of information available on Web. The dynamic expansion of the Web, visible both in the growth of its resources and the increase in the number of its users, generates a demand for new techniques to access the desired information. One of the key aspects these techniques need to address is the fact that the majority of today's Internet users do not have such technical background as was typical for users a few decades ago during its inception. Nowadays the Web is accessed in various routine life activities and its average user does not wish to learn new advanced information access techniques nor dedicate his time and attention to express his current interests in a precise way. For instance, most of the users are totally unfamiliar with the Boolean operators which can often help achieve the desired precision.

Recently a new discipline, called *semantic search*, has emerged which aims at developing new intelligent methods for wide-ranging management and access of Web resources. Information processing and management methods constitute a subfield of the discipline, which attains particular attention. Because of the fact that Web resources are mainly text documents, most of the carried research is a continuation of the decades-long studies in Information Retrieval (IR) [46]. There is a widely recognized necessity for taking into account the specific aspects of processing documents in order to satisfy the needs of typical Web users. On the other hand, studies [8, 117] show that typical queries tend to be imprecise – they are short (with the average of 2 or 3 keywords) and are often expressed in ambiguous terms. This results in the usually overwhelming number of returned documents. It has been reported that more than 50% of search

engine users consult no more than the first 2 screens of results [117]. If they do not find relevant information among the first 20 documents they either get discouraged and give up their search or try to reformulate the query in the hope of better expression of information need and desired result.

In such scenario, query context has been emerged as a significant criterion to be considered while semantically searching for desired information on Web. Semantic relationships which are supposed to be occurring between a pair of words need to be explored for the purpose of judicious reformulation of the query. This semantically enhanced query would be better able to produce more probable set of search results satisfying user need. Information modeling paradigms in general and query modeling paradigms in particular such as relational data models or ontology-based models have greater expressive power but have the limitation of direct application to unstructured information objects currently available on Web. Hence, it is asserted that a novel shift is required from already existing search processing techniques based on pure keywords having limited expressivity, without any reliance on ontological knowledge bases in the process.

2.4.3 Objectives

Based on the literature review of already existing perspectives and approaches pertaining to semantic based search on Web, some open issues have been identified which provide a clear ground for formalizing the problem to be worked upon. One of such open issues has been taken as the potential problem for this thesis work. In order to achieve a valid solution of this problem, following objectives have been laid out for this work:

1. To study and analyze the finer aspects of managing semantic content for the Web.
2. To propose an efficient framework for semantic based search on the Web.
3. To design and implement the proposed framework.
4. To test and validate the framework in terms of parameter like more relevant search results with respect to specific domain.

Chapter 3

Proposed Framework: QUery-context based Information retrieval using Corpus Knowledge (QUICK)

In chapter 2, a review of the various perspectives of the semantic search paradigm has been presented. After careful survey of existing approaches for semantic based search, it is perceived that a lot of work has been done during last decade in the field of semantic based search in terms of background knowledge management, query interfaces, query processing and indexing aspects, presentation of results and query refinement. Conceptual perspective treats the information objects as concepts and intends to facilitate the automation of tasks requiring a certain level of conceptual understanding of the objects involved or the task itself. On one side, ontology serves as background domain knowledge source used for linking semantic annotations to Web documents or other resources. On the other side, semantic knowledge is extracted from Web resources to populate ontologies. Search system having a great dependency on ontological knowledge base can work well under the situation where whole information space can be fully adapted as per the ontological requirements of the system. Huge amount of information currently available to search systems is in unstructured format contrary to the requirement of semi-structured ontology based systems. At the same time, it is quite a difficult job to update the whole information space as per the requirement of ontology based systems, that too in a cost

effective manner. Hence, it seems quite logical to think of a novel search system which is implementable considering current architectural set-up of the information space.

Through the process of reviewing and critically analyzing the selected approaches, a series of open issues have been mined which reflect a great deal of scope for further improvement in semantic based search systems at different dimensions. It can be inferred that there is potential scope for a novel approach for further refinement in search results retrieval semantically.

3.1 Motivation for query context

Context is a growing area which researchers are working on for providing improvements in query refinement for subsequent improvement in Web search process. There are many dimensions of context with no standard definition. One of the literal meaning of context is ‘‘the parts of a discourse that surround a word or passage and can throw light on its meaning’’. The second is a more general one based on circumstances ‘‘the interrelated conditions in which something exists or occurs”. The first one has been accepted as being appropriate for information retrieval by many researchers. Hence it is asserted that the sense that a word takes in one context is defined by its neighboring words.

The kind of search in which user’s information needs are addressed by considering the meaning of user’s query as well as available resources is referred to as Semantic Search [22]. One of the significant issues in the field of semantic search is to exploit the query dependent features for effective search. Query context has been emphasized to be treated as one of the dominating criteria for the purpose of efficient semantic search [119].

One of the workable approaches to find the real context of query can be through addition of relevant terms to the initial query. The main aim is to add new meaningful and relevant terms to the initial query in view of the difficulty in expressing an information need using exact query terms. It would facilitate the user with the addition of morphological/semantic variations of original query terms in the search process. The present approaches available in literature have shown this being done in three different ways: manually, semi-automatically or automatically. Manually, query expansion relies on user expertise to make decisions on which terms to include in the new query, which requires reasonable expertise of the user in order to specify its information need precisely. In semi-automatic approach, the process moves a step further in facilitating the user. It generates possible query expansion terms and the user selects which of

these to include. In automatic approach, weights are calculated for all terms and the terms which have the highest weighting are added to the initial query without user intervention. Different weight functions produce different results hence deciding the retrieval performance. The new terms resulting from the chosen term selection method should provide contextual information for the initial query with the purpose of improving retrieval results. The contextual information can be acquired from different knowledge-bases like a standard corpus, semantic network like ontology or the Web. Semantic enhancement of the query has been successful to a certain extent but there is still a scope of improving the techniques, interfaces or algorithms used to infer context more accurately and efficiently in order to improve the results and processes even further. Efforts can be put up in identifying techniques about the timing of expansion like during the search process or before the search process starts. Another potential area can belong to how to deal with document collections which do not have a controlled vocabulary and are not written consistently like Web pages.

3.2 Proposed approach

A search mechanism should be able to automatically identify the features of a query according to its *most likely intent* so that the search intent of the user could be taken into account in the retrieval process. A better match between the query intent and the search results increases user satisfaction. Our approach is motivated by the intuition that closer the topic of the query is to the category identified by the user, least are the chances for induction of ambiguity and thus greater is the probability for reaching closest to the query context. Further, more the query terms are appended by its contextual features while searching the information space, greater are the chances to obtain more precise and supportive documents satisfying the user's information needs. Hence, a step-by-step procedure has been proposed for the purpose of efficient search of desired information semantically with enhanced relevance of search results in comparison to classic keyword based models. The framework encompassing the whole arrangement has been broadly divided into five phases:

i) Data acquisition
ii) Pre-processing
iii) Corpus knowledge generation
iv) Candidate context feature generation

v) Feature selection

3.2.1 Data acquisition

The task of data acquisition for the purpose of semantically enhancing the query can be linked to various dimensions. One of the investigated dimensions in this concern is personalizing the user models referring the saved history of queries entered and documents viewed by the user over a period of time. Also, voting patterns of different users are looked for similarity and used to provide automatic recommendations in query context based on the intuition that the users having similar tastes and interests in the past tend to follow same trend in future also. Such collaborative effort for formulating queries helps in reducing time to reach to user intent in an effective manner [118]. But effort to create individual user profiles in such manner seems to produce little result in view of changing preferences and diversified needs even for the same user. Another approach for acquiring data for semantic enhancement of the query can occur in terms of hyperlinked Web pages pertaining to the original query related Web pages. According to T. Tran et al. [89], content and the structure of information space surrounding the documents give a fair idea about the context of the documents. Although linkages among documents may indicate the importance of linked document based on the logic that only important documents are pointed by many documents but that importance would be global in nature having little weight with respect to the query at hand.

The domain specific knowledge resources have also been seen to be used as a prime source for acquiring data in order to reach to the context of user query. To use these knowledge resources to their full potential, it is better to be aware about the embedded context of the resource which will make it eligible to be used for contextualizing the queries related to that domain. The vocabulary richness of the resource is of prime importance for reaping the desired benefits from this mode of data acquisition.

Web has also been tested for the purpose of acquiring data used for reaching to semantic features of a query in recent approaches. It is almost impossible to analyze this huge source of data document-by-document in view of its size as well as high growth rate of the Web. Hence, Web search engine can be used as an efficient interface to this vast source of data. Data acquired in such manner for facilitating the generation of Web mediated corpus has a set of properties which further enhance the appeal of this approach. Unlike knowledge models such as ontologies and

thesauri which provide a mean to paraphrasing the user's query in local context, this global technique is data-driven which do not always have simple linguistic interpretation of the query words. Additionally, such models use features which are language-dependent whereas the present approach uses the features of the Web document collection which suit for arguably all language collections. The use of robust and tested algorithm like PageRank makes the initial collection of documents more relevant to the topic at hand using Web as the source of data. The tokens for the corpus which would be extracted from top-ranked documents would be having a geo-temporal characteristic as well by the very nature of a standard Web search engine mechanism. The geo-location sensitivity would help in extracting the features which are relevant to the query fired at a specific geographic location. Another significant factor is the temporal sensitivity in the sense that different pages are returned by a standard search engine with changed senses of the existing words at different points of time. E.g. a few years ago, Web pages pertaining to the term *Tablet* were better be linked to the sense as agent to cure a disease, now-a-days, it seems to be more popular being used to the sense as hand-held computing device. The approach is able to generate basic corpus for invariably every domain or category, which is small in size and enriched in lexical content with the feasibility of its re-generation with respect to time and location. A summary of peculiar set of features of data acquisition through Web search engine based techniques is presented below:

Location-sensitive: The standard search engine Google is used here to generate raw information set. By its very nature, the Web pages acquired by search mechanism are dependent on the geographic location where query is fired. Hence it will support the generation of relative set of context features as prevalent in that particular location.

Time-sensitive: The results generated by a standard search engine are refreshed periodically as per crawling architecture. It is not uncommon that different set of top ranked pages are returned every refresh cycle. This will facilitate the generation of time specific relevant context features in view of addition of new words and changing senses of existing words.

Robust relevance criterion: The standard relevance criterion PageRank as used by the search mechanism helps to provide related documents with high global importance.

Domain-independent: The most appealing feature of the approach is its domain-independent nature. Without the use of any domain-specific one or more ontology, it provides a mean to get a set of terms which are semantically related to the original term.

Unlike semantic network/ontology approach, here it is observed that the words which are associated via any type of semantic relationship with the target word are achieved.

Language-independent: There do not seem any language constraints so far as acquisition of semantically related terms is concerned.

Cost-effective: Without the requisition of any significant increase in processing power or storage requirement, this approach produces a set of such semantically related terms which when augmented with original query terms produces a set of results with high degree of probability meeting user requirement.

All the above factors motivate us to use the Web as an ideal source for harvesting implicit and useful relations between words which are otherwise not possible to be discovered using other handcrafted language resources such as ontology. As an initial exercise of the proposed framework, various categories reflecting different domains are identified with an intention to cover a large set of query areas. One of such category specific user query is entered to a standard search engine with an intention to receive a relevant set of documents pertaining to the query category. Although these documents may not be having desired capability to satisfy user requirements but these documents are supposed to be enriched with the content having strong association with the entered query category. The top-ranked documents coming out as a result of the query serve as a fertile ground for finding the most probable context features of the query category.

3.2.2 Pre-processing

This phase is concerned with the refinement of the top-ranked Web documents generated in the previous phase in such a manner that the data available in the documents could be transformed into more useful knowledge for its effective utilization by next phases. During this phase, data objects are extracted from raw data of the documents which serve as potential source of context features required to be generated from acquired data. The phase of pre-processing a data corpus which results into refinement of data into knowledge is independent of specific user query that has to be semantically enhanced.

The document corpus generated using top-ranked Web documents in the previous phase is dissolved with unwanted material having very negligible lexical value with little role in accessing and manipulation of context features. The information contained in the top-ranked

Web documents is required to be extracted in the form of tokens. The collection usually consists of documents in the form of HTML, PDF, MS-WORD etc. from which text is extracted using the process of *tokenization.* Most of the times, these documents being in HTML format are required to be filtered for removing unwanted material concerning site navigation and related matter. The individual words are extracted ignoring case, punctuation marks and other special symbols. The actual content having lexical value is separated out using various techniques. Firstly, *stop-words*, the common words like articles and prepositions usually having little lexical value are removed from the text. Then, non-alphabetic tokens having some special symbols or numeric figures are filtered. It is assumed that usually non-alphabetic data items have little lexical significance. Finally a stemming algorithm is used to reduce different words to the same stem. A stemmer may reduce all the infected or derived words to their root form that strictly follow the language syntax (e.g. Singular/plural of nouns or forms of verbs). In case of derived words, reduction may not be in morphological root of the word. E.g. the words "generalizations", "generalization", "generalize", "general" would be reduced to stem "gener" as per Porter's derivational stemming algorithm [120]. Porter Stemmer is used to strip the affixes for supporting search with words having alternative forms e.g. word *lying* is mapped to word *lie,* although there are chances of error incorporation due to over generalization. Each top ranked Web page is passed through these phases resulting in actual tokens having significant lexical value.

Each document is represented as a set of lexical tokens which can serve as potential source for the identification of context features with respect to original query. This is the strict pre-processing phase required to be implemented irrespective of the category-specific original query at hand.

3.2.3 Corpus knowledge generation

One of the crucial phases of the proposed framework is to generate Web mediated corpus knowledge which is subsequently used to generate knowledge in terms of context features of the query entered by the user. Many of the present approaches use page counts and snippets for the purpose of measuring semantic relatedness between words. Page count of a query is an approximation of the number of pages containing query words based on the result set returned by a search engine. The whole set of documents as returned by search engine serves as corpus knowledge in that case. For a query K1 and K2, the number of pages containing both K1 and K2

reflect the semantic similarity between both the terms which can further be compared with the page count for K1 and K3. If page count for K1 and K2 is more than that of K1 and K3, it can be inferred that K1 is more semantically related to K2 than K3. Such global measure of co-occurrence apparently lacks to count word frequency since there are chances that query word appears many times on a single page. Also, the method does not consider the relative position of words on the page since two words lying farther on a page might not be related to each other. Moreover, page count of a word having multiple senses would be counted for every sense under consideration giving wrong impression of the similarity. Also, sometimes words co-occur on pages without actually being related in view of scale of Web and noise factor. These factors dilute the utility of this method when used in isolation for the purpose of calculating the semantic relatedness between words.

Another approach is to use the snippets as a local context of the query term. It is a brief textual idea about the query term present in the document as extracted by a search engine. Although snippet processing is efficient mode in comparison to full Web page processing for deriving the refined context of a term, but the textual idea about the query term as generated by search engine mechanism in the form of snippet is unique to the underlying search engine. This might not be enriched with actual data content which can be used to calculate the semantic similarity between a pair of words.

In our approach, a Web mediated corpus is generated based on some basic information on specifically selected categories belonging to diversified domains. The category words are treated as query words and entered using a standard search engine. The Web documents returned as a result of the query are taken as base data for the corpus generation. The contents of top-ranked documents are appended with the intention to get a consolidated list of tokens which are having a high probability of containing the words semantically related to the query words. At this stage, rather than considering the whole web document as the context of the query word, the words which are occurring in very close proximity of the query words lying in the document are considered as the context windows of the word. E.g. if a query word occurs 5 times in a document, then 5 context windows would be generated reflecting the most probable context of the query word in that document. The refined data received within the corpus after implementation of this phase has been termed as *corpus knowledge*.

In this manner, where on one hand, link structure based techniques have been exploited with the use of robust ranking algorithms of standard search engine for reaching to the close context of the query words, on the other hand, due consideration has been given to the relative position of the words in the document for the purpose of identifying the most probable context of the query word. Nonetheless, this approach of generating context using query specific features make it more effective than standard-corpus specific techniques where context features are selected based on their frequency in the collection which may otherwise be irrelevant for the query at hand. The periodic regeneration of these category-specific corpora intact its potential for generating ever relevant context features pertaining to specified categories at low cost.

3.2.4 Candidate context feature generation

Based on the category specific original query and preprocessed Web mediated corpus, a set of context features are generated in this phase which are having a close association with the category. The information contained in the top-ranked Web documents retrieved in response to the actual query is required to be indexed. All the data belonging to the Web mediated corpus is stored in a specially designed data structure such as *list*. All word occurrences lying in the Web mediated corpus are indexed. This also includes the actual category words which had been entered as original query and lying in the Web mediated corpus. The context of the category word(s) is identified in the form of context windows which are the words lying in the close proximity of every occurrence of the category specific original query word(s). The words belonging to these context windows are stored in a list with an intention to consolidate the candidate context features of the category. The process results in a refined list comprising the feature set having greater probability of contextualizing the original query.

This is to emphasize again that the words that occur in close proximity of a word are closely associated with it in terms of semantic relatedness. In other words, given a text corpus, context of a word is composed of words co-occurring with it within a certain window around it. Such contextual words having close association with a specified word by way of non-similar syntactic relationships reflect distributional relatedness among the words. The phenomenon of distributional relatedness is of greater significance than that of distributional similarity while measuring the association of the words with respect to semantics based search. This is because of the fact that all such words are associated by cross part of speech relations whereas in case of

distributional similarity, the association between co-occurring words happens to be by same syntactic relations. The five words on the left side of the target word (Category word) and right side of the target word are taken to be considered as the context of the target word. This has been referred to as *context window* of size 10. These sets of 10 words are taken for each occurrence of the target word in the corpus. The conditional probability of the occurrence of i^{th} word w_i, when target word w_t is already there, is calculated as

$$\text{Conditional probability, } P(w_i|w_t) = \frac{P(w_i \cap w_t)}{P(w_t)} \quad (3.1)$$

where w_i is the ith word in the list

w_t is the target word

$$P(w_i \cap w_t) = \frac{|w_i, w_t|}{|w|} \quad (3.2)$$

$$P(w_t) = \frac{|w_t|}{|w|} \quad (3.3)$$

(3.1) can be expanded to (3.4) using (3.2) and (3.3)

$$P(w_i|w_t) = \frac{|w_i, w_t| / |w|}{|w_t| / |w|} \quad (3.4)$$

Where

$|w_i, w_t|$ is the number of occurrences of w_i when w_t is already there in the corpus(with context window of size 10)

$|w|$ is the total number of words in the corpus

$|w_t|$ is the number of occurrences of the target word in the corpus

The more the value of P, the more closely the word w_i is associated with the target word w_t and hence have a closer semantic relation with the target word. The set of such identified words have been referred to as context features of the category. Several features having high conditional probability values would be emerged out in this process.

In this novel approach for candidate feature generation, in first stage, a set of words is generated in the form of corpus knowledge having greater probability of association with the target word/category. This is due to the fact that instead of using any standard corpus, the top-ranked documents of result set returned by a standard search engine are used for the generation of corpus knowledge which are supposed to be having significant presence of target word(s) with global importance. Such corpus of words although small in size but has full potential to give a fair context of the target word supportive in the generation of required context features of the category/target word. In the second stage, a novel aspect of conditional probability has been exploited where in the frequency of occurrence of a word has been assigned importance in terms of its presence in close proximity of the target word. The occurrence of a word is counted only if target word is already there close-by. This probabilistic feature of the approach helps to assign due importance to the relative positioning of the words with respect to the target word which in turn ensures to have greater probability in reaching to the context of the target word/category.

3.2.5. Feature selection

The candidate features generated in the previous phase are selected for possible augmentation of the original query. The original query is augmented with selected features of the category/domain as part of the original query. The mutual dependence among the generated features is not taken into consideration while selecting the features based on some experimental results suggesting the justification of context feature independence assumption [121]. Some studies have suggested the selection of limited number of features for expansion ranging from 5-10 context features [122, 123] to a few hundred [124 - 126]. The typical choice can be 10-30 features to be selected. Another alternative is to select only the terms having probability greater than a certain threshold where context feature scores can be interpreted as probabilities as taken in [127]. Nonetheless, it has been observed that the impact of sub-optimal number of context features on performance is not very significant [133] where several studies have inferred that the number of context features does not have reasonable significance with respect to performance degradation.

One idea behind selecting a limited number of context features for expansion is the processing efficiency; secondly retrieval effectiveness also remains intact for a small set of context features than addition of all context features due to noise reduction [128, 129]. Another approach can be

the adoption of more informed selection policies rather than finding an optimal number of context features. It has been observed that different queries may have different number of optimal context features for expansion [130, 131]; also sometimes many expanded context features are harmful to retrieval effectiveness [131]. It would be an ideal situation where one can be able to find out accurately the best features for each query in order to achieve the level of retrieval effectiveness with significant improvement [131, 132].

In the proposed approach, a set of top-ranked context features having highest probability of semantic relatedness with the category are selected. This will facilitate to depict a more accurate model of the query topic filtering those pages which are tagged as relevant due to shorter description of the query. A pictorial representation of the flow of processes is presented in Figure 3.1.

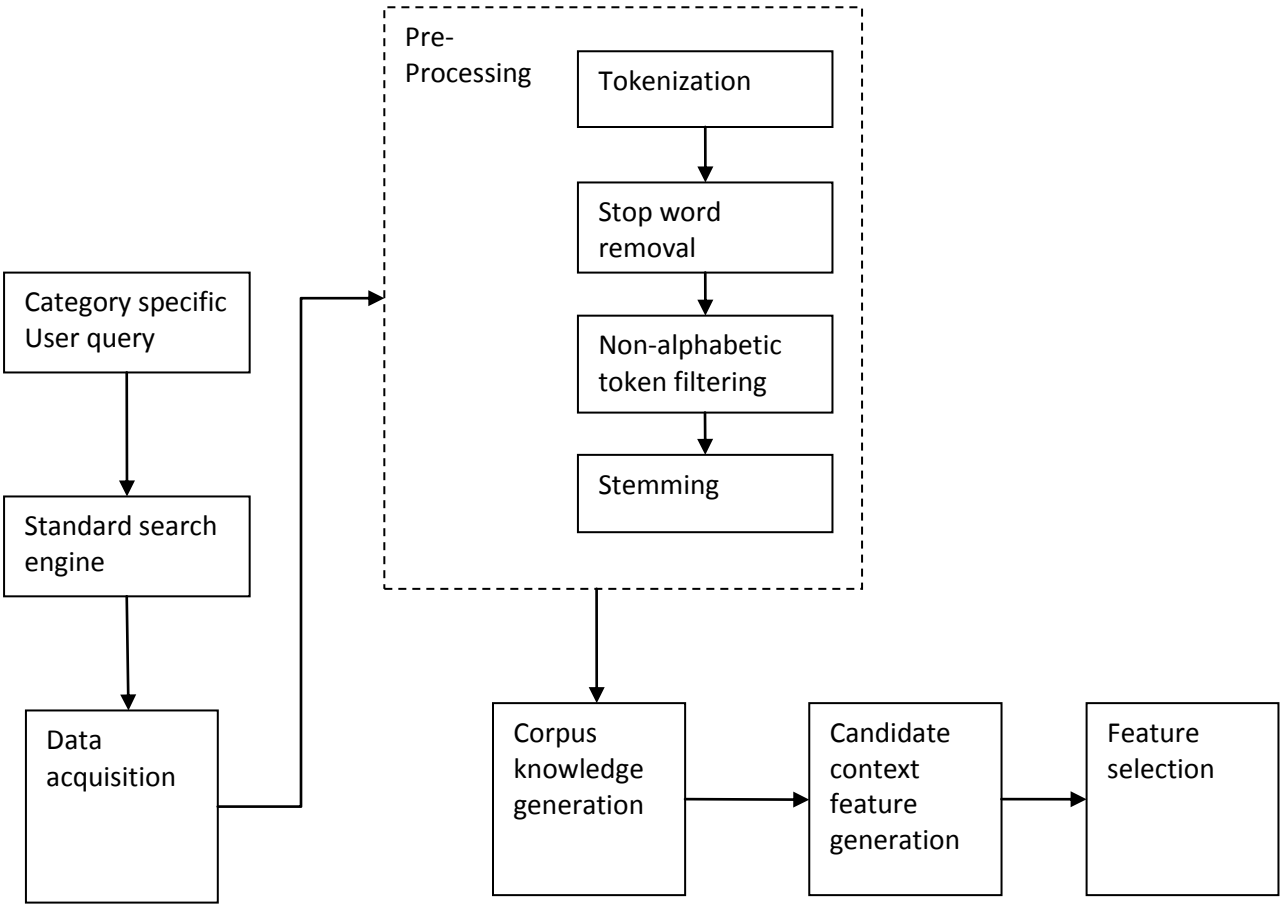

Figure 3.1: Pictorial representation of flow of processes from category specific user query to context feature selection

3.2.6. Efficiency

There are different levels where the proposed framework is thought to witness the aspect of efficiency. Firstly, while generating the corpus knowledge, the latest and most current terms and most current senses of the existing terms are taken into consideration because the Web is such a mean of information space which is continuously updating and augmented. Hence it is supposed to improve the precision of the results by reaching to the accurate set of results satisfying user's information need, also it will look after the recall of the results by covering the maximum number of aspects underlying the query topic at hand.

At the same time, the proposed approach facilitates the generation of context features before the submission of actual user query. The selected context features depicting a more accurate model of the query are already clustered under the query topic. These context features are also referred as part of the query while searching for the desired information from the Web. The process of context feature generation being off-line does not contribute in the degradation of the performance of the search system while searching for information. Hence it is asserted that the improved set of results is achieved underlying semantic based search at almost no extra cost in terms of time response.

Another aspect of efficiency reflects at the stage of accession of corpus knowledge. The knowledge which is required to be accessed for reaching to the closest context of the query is from a small size Web mediated corpus made up of features with highest probability of defining the context of the query topic. The small size of Web mediated corpus generated in such a fashion facilitates to access the most probable set of context features efficiently in contrast to the process of feature generation using ontology based systems or big size standard corpus. The size of the standard corpus may be of the order of thousand bigger than that of Web mediated corpus.

3.3 The proposed framework

In view of the various phases and procedures adopted from data acquisition to feature selection, the framework has been articulated with title: **QU**ery-context based **I**nformation retrieval using **C**orpus **K**nowledge (**QUICK)**. A sequence of steps required to be executed from data acquisition to context features selection for reaching to the overall improvement of search results is shown in Figure 3.2 in the form of an algorithm.

A sequence of steps of the proposed approach

Off-line/Pre Query Submission

i) Categories are to be manually stored to the search system.

ii) Features pertaining to a category have to be clustered under that category as proposed by algorithm 1.

Algorithm 1: The whole process involves a number of computational steps towards the overall improvement of search results.

Step 1: Data acquisition

Step 2: Preprocessing

- Step 2.1: Tokenization
- Step 2.2: Stop words removal
- Step 2.3: Lexical token filtering
- Step 2.4: Stemming

Step 3: Corpus knowledge generation

Step 4: Candidate context feature generation

Step 5: Feature selection

Figure 3.2: Algorithm for steps from data acquisition to context feature selection

On-line/Post Query Submission

i) Broad category/domain has to be selected by the user involving removal of ambiguity.

ii) The features of the respective category would also be referred whenever the words pertaining to a category are entered by the user as part of query.

iii) The result set would present semantically-related query-dependent results with greater probability to address the most likely intent of the user query.

In this approach, categories are proposed to be manually fed to the system which would serve as a base data source for identifying broad domains taking into consideration the diversity factor of Web queries. It is observed that the common issue affecting the precision of the result set is the out-of- context match of query term with document. Hence it is proposed to present a set of domains to the user based on the query terms entered by the user to remove any possible cases of

ambiguity e.g. a user may ask for information related to *bank* in his query, he has to be given the option to select the broad domain/category whether *Financial Bank* or *River Bank*. This would help to avoid the adverse effect which could have otherwise been caused to the *precision* of the result set due to the ambiguous nature of the term. Based on his selected category, the features which have already been generated pertaining to that category would also be augmented with original query terms as final query. A schematic diagram of the proposed framework has been presented in Figure 3.3.

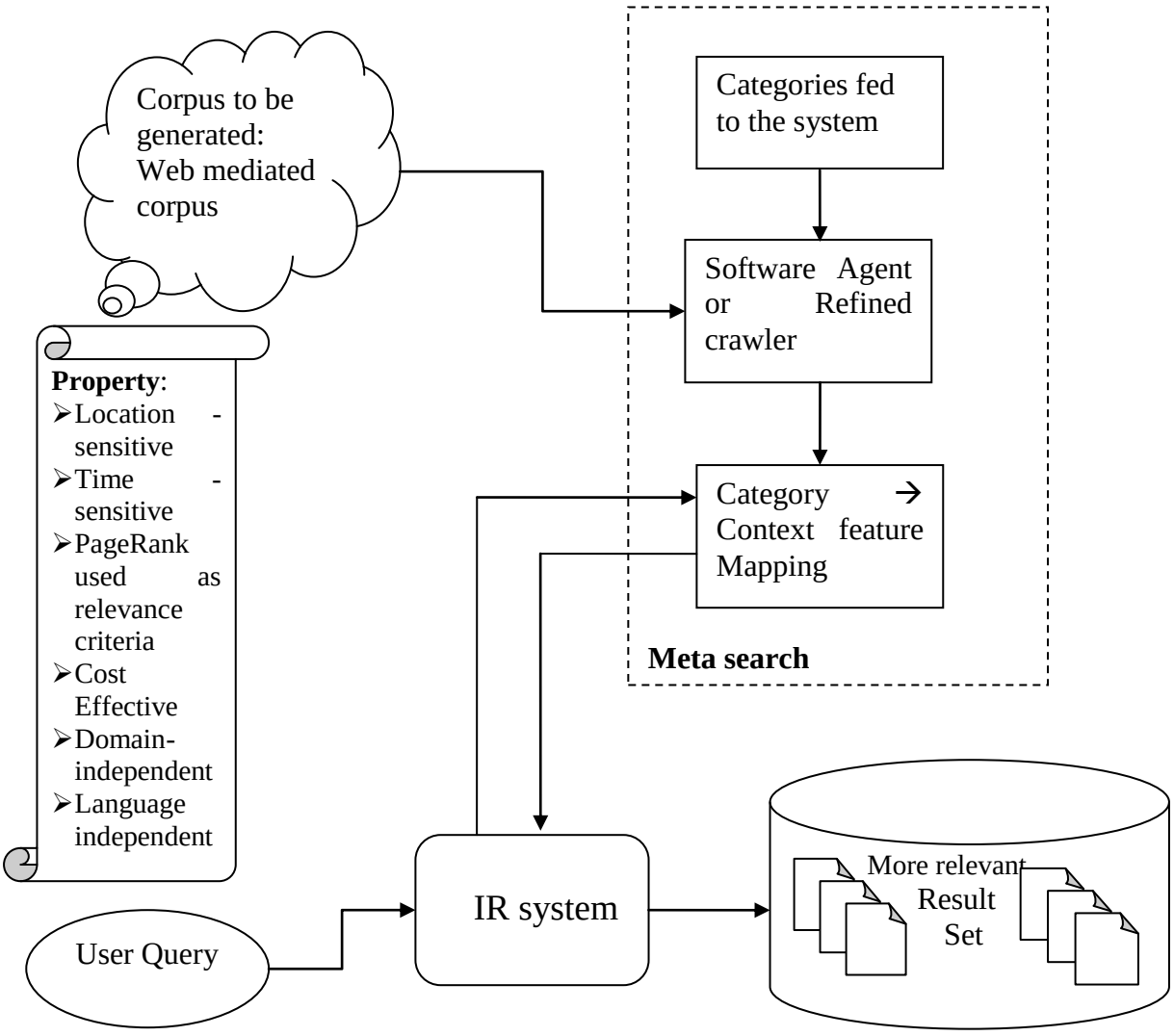

Figure 3.3: Proposed framework for semantic based search on Web: QUICK

The sequence of steps involved in the process of mapping query with context features is referred to as *Meta-search* in the framework. The overall process reflects the possibility of generating a set of contextual features in an efficient manner that produces a more probable model of the query intention.

Chapter 4

Design and Implementation of QUICK

After careful analysis of already existing approaches for semantic based search on Web, an efficient and novel approach was proposed which could be of good significance in terms of improvement in search result retrieval. In this chapter, the design and implementation aspects of the proposed framework, QUICK are discussed with emphasis on the applicability of this approach. The chapter starts with the refined introduction of the software tool, NLTK which was instrumental in implementation of QUICK. Then, the design of the scenario has been put in place in order to realize the implementation of QUICK. It covers a case based scenario where in different categories have been selected broadly encompassing different domains which are used while searching for information related to various domains. Category specific user query is entered to a standard search engine in order to retrieve most relevant documents pertaining to that domain with respect to search engine logic. The returned documents are stored and useful data is extracted from those documents in order to generate a corpus which is subsequently used for the knowledge extraction from the corpus. The knowledge generated in such a fashion is used to facilitate the generation of a context vector which is quite specific to the identified category with respect to the corpus knowledge generated in the proposed approach. The context vector serves as input to the similarity calculation where the strength of association of various features to the category is calculated. Finally a set of features having best strength of association to the category are selected and treated as the context features of the category used for the expansion of query pertaining to that category.

4.1 NLTK

Although there are number of software suites available for semantic analysis of the text documents still there exists implicit merit in each of such tool with respect to its usability. One such Python-based software tool [134] is popularly available for computational linguistics called Natural Language Tool Kit (NLTK). Python as a programming language provides ease of use in accomplishing interesting results with only a small learning in contrast of other widely available programming languages like Java and Prolog. Python can be easily understood even having little programming experience earlier and the desired can be accomplished due to its suitability for rapid prototyping. These features make Python a vital resource in researcher's repository of resources. NLTK is a decently developed open source project with broad coverage of language processing features. A uniform computational framework has been provided for language processing with respect to lexical, syntactic and semantic aspects. This makes it easier to use NLTK for precisely focusing on computational linguistics in natural language analysis domain.

NLTK also comes with a significant collection of corpora and easy to use corpus readers. Various corpora available with NLTK include parsed, POS-tagged, plain text, categorized text and lexicons. This makes it easy to experiment with complexities of semantic analysis of realistic bodies of text. It is very convenient to build an experimental set-up for exploring data and testing a hypothesis using NLTK. NLTK is a leading platform for building Python programs to work with human language data. It provides easy to use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing and semantic reasoning. It is available for Windows, MacOSX, and Linux. It is a free, open source and community driven project.

There is also literature available for natural language text processing using easy-to-use programming languages like Python. Such content throws a clear light on the use of language features and guiding the reader through the fundamentals of writing Python programs for working with different corpora, text categorizing, linguistic structure analysis etc.

4.1.1 Implementation steps

Following steps are implemented with the help of open source library NLTK integrated in Python:

i) The top ranked documents of the result set generated after entering the query pertaining to the manually identified category are extracted.

ii) Data objects with significant lexical value are extracted from raw data of the documents with the help of various techniques. The raw data dissolved with unwanted material having very negligible lexical value is filtered out resulting in the formation of corpus knowledge corpus.

iii) The lexical tokens of Web mediated corpus are stored and indexed using data structure *list* available in Python programming language.

iv) The words lying in the close proximity of the category word are stored in a separate list to be termed as context vector.

v) The words of the context vector having highest probability of occurrence with the query category are extracted to form the context features of the category.

4.2 Category selection

In our approach, categories are proposed to be manually fed to the system which would serve as a base data source for identifying broad domains taking into consideration the diversity factor of Web queries. It is observed that the common issue affecting the precision of the result set is the out-of- context match of query term with document. Hence it is proposed to offer a set of domains to the user based on the query terms entered by the user to remove any possible cases of ambiguity. E.g. a user may ask for information related to *bank* in his query, he is given the option to select the broad domain whether *Financial Bank* or *River Bank*. The step would help to avoid the damage which could have otherwise been caused to the *precision* of the result set due to the ambiguous nature of the term. For experimental purposes, the categories *Financial Bank* and *Apple Fruit* have been selected.

4.3 Corpus selection

For the purpose of corpus, out of the various options which are available in the form of ontology, standard corpus or Web based corpus, the last option was preferred over the other two. Web mediated corpus where on one hand, facilitate to access context features without any domain

constraints as there is no need to establish in advance which ontology to be accessed for the purpose of reaching to the context of the query. On the other hand, this method of corpus generation supports the maximum coverage of available interpretations of the existing words, whereas, relying on a particular set of ontologies or WordNet will confine the scope of applications. Moreover, Web serves as huge corpus of unstructured text data which can be very easily and efficiently searched in order to generate relevant linguistic examples with the help of prevalent search engines.

Since Web has been argued to be the most significant source of text data, in order to acquire basic data from the Web for preparing required corpus, a standard search engine Google has been used. Some very common categories have been identified and used for the purpose of data acquisition. For the purpose of experimentation, query containing category word(s) is entered to a standard search engine in order to get a result set of the Web documents which are most relevant to the query as per the search engine logic. The top documents of the result set generated after entering the query with these words/phrases are taken. The set of words belonging to the top documents in the format of HTML, PDF etc. will serve as the raw data corpus for the generation of corpus knowledge.

4.4 Data acquisition

Most of the text available on Web is in the form of HTML documents. One way of accessing the HTML content for the purpose of text processing is to save a page as text file and then reading the file using common code available in Python. Another way of doing the same is to get Python access the Web page directly and saving the HTML content including meta tags, an image map, Java script, forms and tables. Getting text out of the exhaustive HTML content has been made extremely simple in NLTK. There is a function available called `nltk.clean_html( )` which takes an HTML string as input and returns raw text which can be tokenized to further get the desired text structure for experimentation.

In Figure 4.1, the result set returned by Google in response to the query *Financial Bank* has been shown. The top two documents have been taken for the generation of corpus in order to maintain the simplicity of the process. The documents being in PDF format are converted to .txt files which are opened using command `f=open('work406.txt')` and `f=open('wp10-11bk.txt')` in Python. Further these files are read using command `raw=f.read( )`. The `read ( )` method

facilitates in the generation of a string containing the contents of entire file. The data pertaining to the query which is stored in these files is purely in raw form containing content with very little lexical value. This includes some uninteresting details such as white spaces, lines breaks and blank lines. For the purpose of desired text processing and analysis, it is required to be converted to a familiar structure breaking the large string into a list of words and punctuation. This process is called *Tokenization* which is facilitated by NLTK for the accomplishment of this task. The command used for the same is tokens=nltk.word_tokenize(raw).

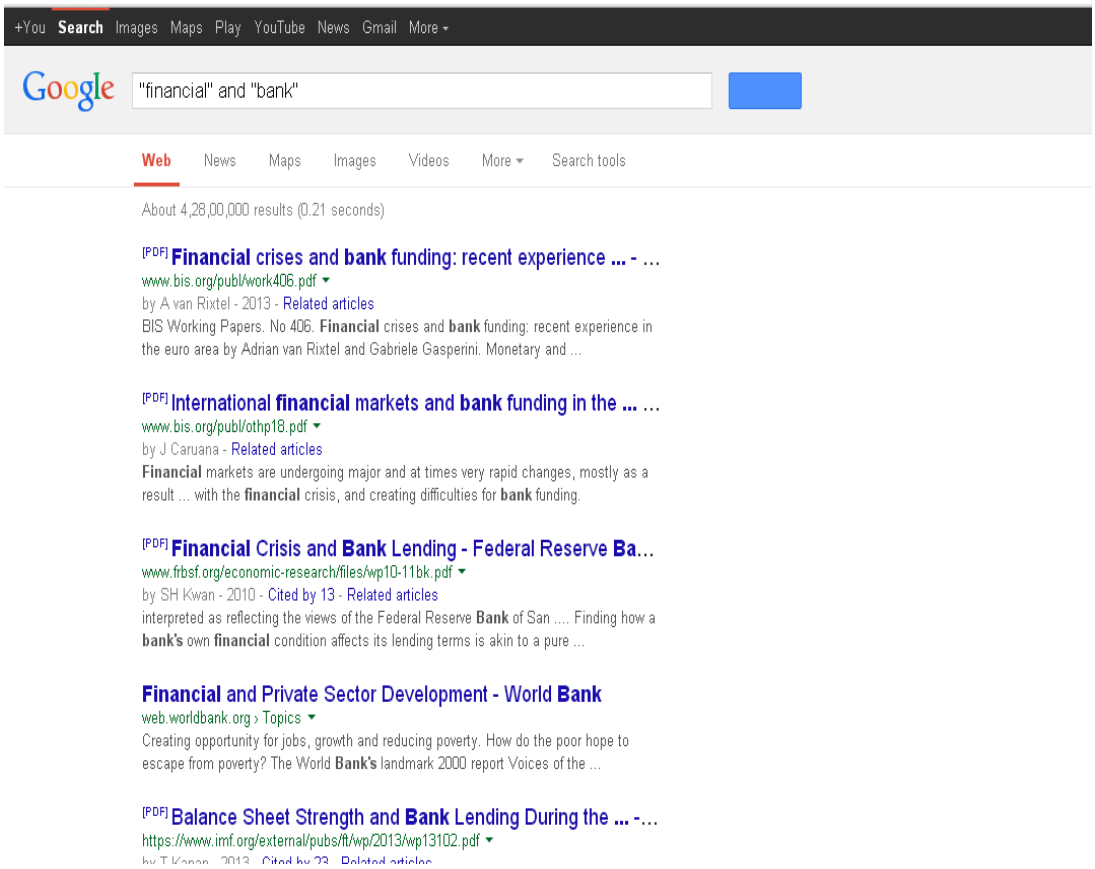

Figure 4.1: The top results returned by Google in response to the query "Financial" and "bank" (as of September 2014)

This list of tokens still contain uninteresting content in the form of site navigation and other related matter which further needs to be filtered. Following steps have been executed using relevant techniques in order to separate out actual content having significant lexical value with respect to context finding.

Lexical token filter: Any punctuation marks, special symbols or numbers have also been eliminated from the list of legitimate tokens in order to maintain the lexical value of the text content. s.isalpha( ) is the function to test whether all the characters of the string s are alphabetic.

Double-count word filter: Further the frequency count of the words has also been rationalized by eliminating the double-count of the words like *That* and *that* removing case distinction and ignoring punctuation. The respective helper function s.lower( ) has been used which tests whether all cased characters in s are lowercase.

Stop-word filter: There is also a corpus of *stop words* which are usually the most common words in a language. Such high frequency words like *the, to, also, is, at, which* are required to be filtered out of the document content in view of its very little lexical value with respect to Web search. Although there is no single universal list of stop words used by all language processing tools but some good tool will provide a broad coverage of such high frequency and low lexical value words. Following Python script integrated with NLTK functions was used to perform this task.

```
>>> from nltk.corpus import stopwords
>>> stopwords=nltk.corpus.stopwords.words('english')
>>> tokens=[w for w in tokens if w.lower() not in stopwords]
```

Stemming: Finally a stemmer, that is normally application specific, has been used to strip-off affixes. Porter Stemmer had been considered a good choice where indexing of the words has to be performed and search applications have to be facilitated using some alternative forms of words. The whole process leaves the content with tokens having highest lexical value in order to find the most associated context of the query word(s).

The similar process is repeated for the top results of another query *Apple fruit* as shown in Figure 4.2. The documents being in HTML format are required to be accessed for its content using script in Python:

```
>>> from urllib import urlopen
>>> url=http://en.wikipedia.org/wiki/Apple
```

```
>>> raw=urlopen(url).read()
>>> raw=nltk.clean_html(raw)
```

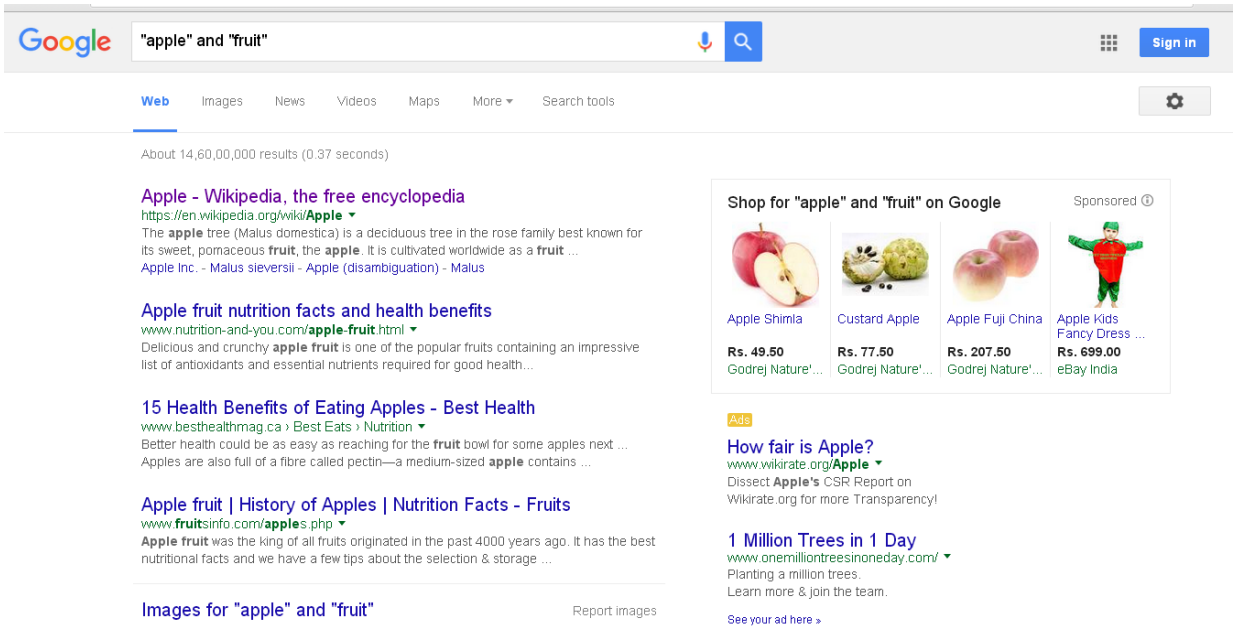

Figure 4.2: The top results returned by Google in response to the query “Apple” and “fruit” (as of September 2014)

The above Python code will open the mentioned URL and read its contents for further storing it in a variable *raw*. Then the HTML content as read from the URL is stripped off the meta tags, forms, tables, Java script etc. The content is further filtered in different stages in order to get the word tokens having significant lexical value facilitating to reach the most associated context of the query word(s).

4.5 Context vector formation

This phase deals with the formation of context vector pertaining to each query with respect to the category it addresses. The vector contains the list of features which are considered most relevant to the category context. Unlike other approaches of context consideration as mentioned in the literature where whole Web page is taken as the context of the query word(s), the present approach considers only the words lying in very close proximity of the query word(s) for the purpose of finding its context.

The lexical tokens generated in the previous phase pertaining to a query are in the form of two lists where each list corresponds to the tokens of one top Web document which is returned by Google search engine in response to the query. These lists are required to be concatenated in order to get a consolidated list of tokens useful for finding the context vector tokens having significant association with the query. The command used for the purpose is `tokens.extend(tokens1)` where tokens of one list named *tokens1* are simply added to the tokens of another list named *tokens*. This list serves as the *Web mediated corpus* which is achieved after refining raw data with the help of implementing various steps described above.

The corpus contains a significant number of occurrences of the query word(s) along with the words of the top-ranked Web documents supposed to be the most relevant to the query as per underlying search engine logic. Such corpus of words although small in size but has full potential for the formation of required context vector of the query word(s)/category supportive to give a fair context of the target query word(s). The words that occur in close proximity of a target word are considered to be semantically related to it in view of its close association with the target word. In other words, given a text corpus, context of a word is composed of words co-occurring with it within a certain window around it.

With the objective of finding the context of the query word(s), all the lexical tokens of Web mediated corpus are indexed. These lexical tokens of the corpus also include a significant number of occurrences of the query word(s). The indices of the query word are identified and are stored in a list variable. The Python script for finding the indices of query word *bank* in the respective corpus is

```
>>> indices=[i for i,x in enumerate(tokens) if x=='bank']
```

The context vector of the query word(s) is identified in the form of context windows which are the words lying in the close proximity of every occurrence of the category specific query word(s). The five words on the left side and right side of the query word are taken to be considered as the potential context of the query word. This has been referred to as *context window* of size 10. These sets of 10 words are taken for each occurrence of the query word in the corpus. The words lying in these context windows are stored in a list with an intention to consolidate the candidate context features of the query. The Python script used for the same is as follows:

```
>>> newlist=[]
>>> newlist1=[]
>>> for i in range(len(indices)):
        a=tokens[indices[i]-5:indices[i]]
        newlist.extend(a)
        b=tokens[indices[i]+1:indices[i]+6]
        newlist1.extend(b)

>>> newlist.extend(newlist1)
```

The process results in a refined list comprising the feature set having greater probability of contextualizing the original query. This list is termed as the *context vector* of the query.

4.6 Context feature generation

The context vector of the query contains the words having highest probability of defining the context of the query in comparison to all other words which are lying in the top-ranked Web documents returned on firing the query. This is because of the fact that the words of context vector have been identified based on their relative position in the document. Being closer to the query word, these words have higher probability of strong relatedness with the query word than that of the words lying farther from the query word.

4.6.1 Strength of Relatedness

The model reflects the probability of occurrence of a word in a document when query word is already there close by. This in turn will reflect the relatedness of the identified word with the query word. The smaller the context window, greater is the probability of reaching to the closely related words to the query word. The window size which was usually taken to be of the size of the page has been reduced to a great extent. Within the page itself, the attempt is to find the conditional probability of the word when the query word is already there. This is to mention that lexical tokens of n top-ranked pages are combined in order to enhance the coverage of relevant

word collection. This will help in reflecting the real picture of relatedness of the words with query word.

All the words of context vector though have strong relatedness with the query word but this is also true that strength of relatedness in all the cases is not same. Hence, in order to measure the strength of relatedness of a word of context vector with query word, a peculiar aspect of conditional probability has been explored. Strength of relatedness of a word of context vector with the query word has been calculated in terms of the conditional probability of occurrence of the word with the query word. Let the word of context vector be w_i and the query word be w_t the conditional probability has been calculated as follows:

Conditional probability, $P(w_i|w_t) = \frac{P(w_i \cap w_t)}{P(w_t)}$

Where $P(w_i \cap w_t) = \frac{|w_i, w_t|}{|w|}$

$|w_i, w_t|$ → number of instances where context vector word w_i and query word w_t are occurring together in Web mediated corpus

$|w|$→ total number of word instances in Web mediated corpus

And $P(w_t) = \frac{|w_t|}{|w|}$

$|w_t|$ → number of instances of query words in Web mediated corpus

$|w|$→ total number of word instances in Web mediated corpus

The more the number of instances where context vector words and query words are occurring together in Web mediated corpus, greater is the probability of strong relatedness of the word with query word. In order to implement this for the case of query word *Bank*, Python script used is as follows:

```
>>> fdist=FreqDist(newlist)
>>> fdist["bank"]
>>> from __future__ import division
>>> for word in fdist:
```

```
print word,'-->',fdist[word]/tokens.count("bank"),'\n',
```

4.6.2 Output in terms of context features

As an output, the Python interpreter will return the words of context vector in the order having highest probability of occurrence together with the query word first. The first 50 words of the context vector having highest strength of relatedness for query *Financial Bank* and query *Apple fruit* have been shown in Table 4.1 and Table 4.2 respectively. These probabilistic values of the words are said to help in defining the context of the query and the words with highest probability values be termed as context features of the query.

Table 4.1: Strength of relatedness of the candidate context features for query *Financial Bank*

Sr No	Context vector word	Strength of Relatedness
1.	loan	0.292173913043
2.	fund	0.28
3.	financi	0.206956521739
4.	euro	0.189565217391
5.	area	0.186086956522
6.	crise	0.126956521739
7.	larg	0.125217391304
8.	tighten	0.107826086957
9.	small	0.104347826087
10.	crisi	0.100869565217
11.	rate	0.095652173913
12.	recent	0.0921739130435
13.	effect	0.0817391304348
14.	experi	0.0713043478261
15.	capit	0.0660869565217
16.	sovereign	0.0608695652174
17.	market	0.0573913043478
18.	debt	0.055652173913
19.	graph	0.0539130434783
20.	medium	0.0504347826087
21.	paper	0.0504347826087
22.	time	0.0504347826087
23.	characterist	0.0486956521739
24.	credit	0.0486956521739
25.	econom	0.0469565217391
26.	bond	0.0452173913043
27.	countri	0.0434782608696
28.	journal	0.0417391304348
29.	liquid	0.0417391304348

30.	mean	0.0417391304348
31.	risk	0.0417391304348
32.	spread	0.0417391304348
33.	asset	0.04
34.	wholesal	0.04
35.	differ	0.0382608695652
36.	figur	0.0382608695652
37.	increas	0.0382608695652
38.	sever	0.0382608695652
39.	ecb	0.0365217391304
40.	global	0.0365217391304
41.	issuanc	0.0365217391304
42.	lend	0.0365217391304
43.	feder	0.0347826086957
44.	section	0.0347826086957
45.	sourc	0.0347826086957
46.	averag	0.0330434782609
47.	central	0.0313043478261
48.	chang	0.0313043478261
49.	financ	0.0313043478261
50.	reserv	0.0313043478261

Table 4.2: Strength of relatedness of the candidate context features for query *Apple Fruit*

Sr No	**Context vector word**	**Strength of Relatedness**
1.	appl	0.736677115987
2.	fruit	0.15987460815
3.	malu	0.119122257053
4.	tree	0.115987460815
5.	cook	0.0658307210031
6.	cider	0.0626959247649
7.	cultivar	0.0626959247649
8.	aphid	0.0532915360502
9.	product	0.0532915360502
10.	seed	0.0532915360502
11.	domestica	0.0470219435737
12.	origin	0.0470219435737
13.	varieti	0.0470219435737
14.	dessert	0.0438871473354
15.	juic	0.0438871473354
16.	new	0.0407523510972
17.	see	0.0407523510972
18.	common	0.0376175548589
19.	fresh	0.0376175548589
20.	gener	0.0376175548589

21.	genom	0.0376175548589
22.	grow	0.0376175548589
23.	list	0.0376175548589
24.	may	0.0376175548589
25.	rootstock	0.0376175548589
26.	state	0.0376175548589
27.	develop	0.0344827586207
28.	form	0.0344827586207
29.	archiv	0.0313479623824
30.	b	0.0313479623824
31.	countri	0.0313479623824
32.	crisp	0.0313479623824
33.	cultiv	0.0313479623824
34.	day	0.0313479623824
35.	eaten	0.0313479623824
36.	grown	0.0313479623824
37.	import	0.0313479623824
38.	north	0.0313479623824
39.	wild	0.0313479623824
40.	brown	0.0282131661442
41.	cake	0.0282131661442
42.	diseas	0.0282131661442
43.	doctor	0.0282131661442
44.	eat	0.0282131661442
45.	europ	0.0282131661442
46.	isbn	0.0282131661442
47.	nutrit	0.0282131661442
48.	Often	0.0282131661442
49.	Pie	0.0282131661442
50.	sauc	0.0282131661442

The original query as entered by the user pertaining to the selected category is augmented with the selected number of context vector words with highest strength of relatedness termed as *context features*. Although a number of considerations have been observed in literature for selecting the context features of the query but studies have suggested the selection of a limited number of context vector words ranging from 5 to 10 as appropriate [122, 123]. At the same time, it has also been observed that the impact of sub-optimal number of context features on performance degradation is not worth counting [133]. First 10 words of context vector having highest strength of relatedness with the query word have been taken as context features in the present scenario. Addition of a limited number of context features will lead to processing efficiency [128]. At the same time, this helps in noise reduction which could have otherwise

been incorporated due to addition of all or a large number of context features [129]. Table 4.3 shows the context features generated after implementing the whole process for the query *Financial Bank* and *Apple fruit*.

Table 4.3: Context features for the query *Financial Bank* and *Apple Fruit*

Original query	**Context features**
"Financial" AND "Bank"	Loan(0.355), fund(0.28), financial(0.207), euro(0.187), area(0.186), crisis(0.127), large(0.125), tighten(0.108), small(0.104), rate(0.096)
"Apple" AND "Fruit"	apple (.737), fruit(.160), malus (.119), tree (.116), cook (.066), product(.066), cider (.063), cultivar(.063), aphid(.053), seed(.053)

The user's interaction with the system depicting the relationship between the user and the different use cases has been shown in Figure 4.3 below:

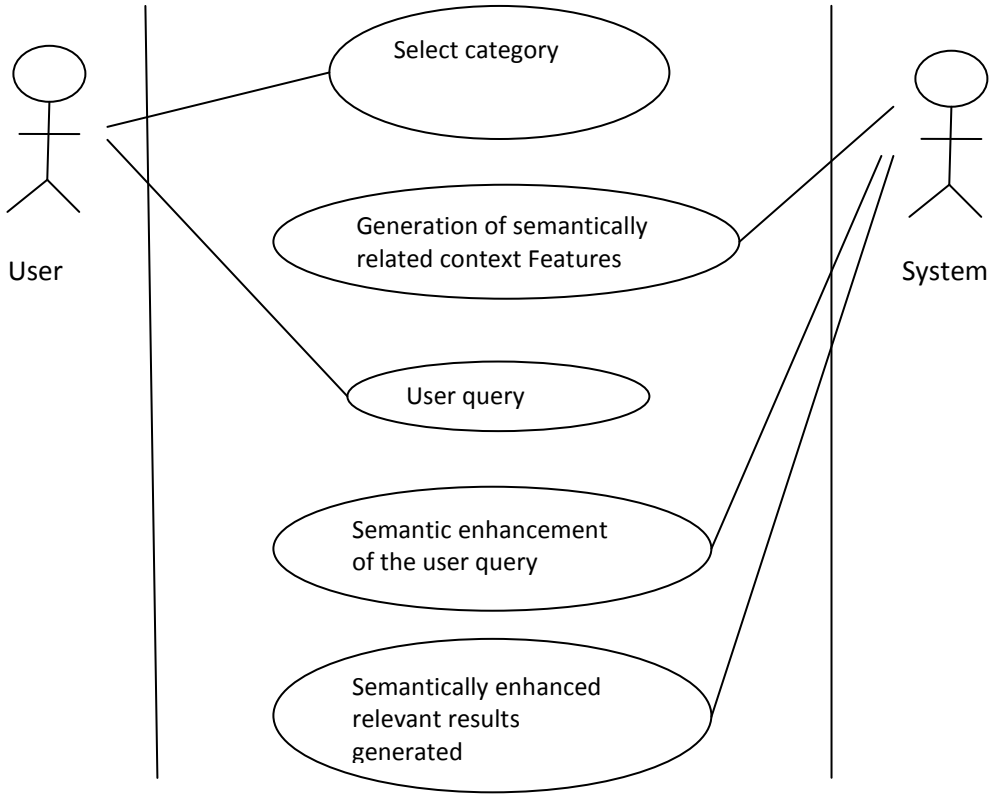

Figure 4.3: Use case diagram representing the user's interaction with the system

Based on user's selection, the features which have been generated pertaining to his domain of selection would also be augmented with original query terms as final query. The overall process reflects the possibility of generating a set of expansion features in a very efficient manner that produces a more accurate model of the query topic, thus filtering out those documents that would have otherwise matched the shorter description.

Chapter 5

Testing and Validation

5.1 Introduction

On-line search systems play a significant role in locating relevant and desired information from the huge information source in the form of Web. In the present scenario of competition and new age development of tools and techniques, it becomes a necessity for the designers, developers and vendors of search tools to test and validate the performance of the search systems in order to witness its competitive advantage. The objective of this chapter is to test and validate the performance reflected by the proposed and implemented approach for semantic based search on Web as discussed and demonstrated in previous chapter. The evaluation of search systems have been categorized as in three different forms [2]. In the first scenario, specific functional aspects of the system are tested and validated individually. The second one deals with the performance evaluation of the system in terms of time and space parameters. Lesser the response time, shorter the space used by the system algorithms, the better the system is considered. The third form of evaluating the search system is related to test the retrieval performance in terms of parameters like more relevant search results. Relevance is a subjective judgment and may include:

- Being on the proper subject.
- Being timely (recent information).
- Being authoritative (from a trusted source).
- Satisfying the goals of the user and his/her intended use of the information (*information need*)

Major relevance criterion a search system should fulfill is user's information need. This form of evaluation assesses the probability of satisfaction of the system user with respect to its information need. This has further been classified as i) user based ii) system based. The first one focuses on the aspect that how well the system is capable of satisfying user's information need based on comparison with human judgments for relevance while the second one measures the document ranking capability of the system. Although the user-centric evaluation strategy is more informative and useful considering human judgment for relevance deemed to be correct by definition, but the strategy is quite cumbersome. Its main drawback comes with the difficulty in obtaining a large set of reliable subject-independent judgments for comparison, designing a psycholinguistic experiment, validating its result and so on. Whereas the system based approach for evaluation focuses on evaluating the proposed measure with respect to its performance in the framework of a particular application. The strategy allows the experiments to control and fix some of the aspects of system that influence retrieval performance while different measures can be compared by analyzing the effectiveness of the system with respect to each of the measures, thereby enhancing the capability for comparative analysis.

Given a particular search approach, for each query, the evaluation measure quantifies the similarity between the set of documents retrieved by the search approach and the set of pre-identified relevant documents. The evaluation measures in the form of *Precision* and *Recall* are used to determine the validity of the approach in terms of its performance.

5.2 Evaluation measures

A large number of evaluation measures have been developed for evaluating the performance of search systems. None of such measures can be termed as absolute in nature since these measures refer user-centric judgments in order to compare the performance. At the same time, there are multiple dimensions involved which can affect the retrieval performance while the result of the measures comes out as a single value without absolute consideration of all the dimensions. The judgment is usually binary in nature (either relevant or non-relevant).

Two of the most frequently used evaluation measures have been used for evaluating retrieval performance: *Precision* and *Recall*.

Suppose there is an example query q and R is the set of documents which have been adjudged as relevant to the query needs. Let A be the set of documents returned for q by a retrieval

mechanism under evaluation, and let R_a be the set of documents which are both in R as well as A, i.e. the relevant documents in the answer set. Recall and precision are defined as follows:

Recall – is the fraction of the relevant documents which has been retrieved, a measure of the ability of a system to present all the relevant documents.

$$Recall = \frac{number\ of\ relevant\ documents\ retrieved}{number\ of\ relevant\ documents\ in\ the\ collection} = \frac{|R_a|}{|R|} \qquad (5.1)$$

Precision – is the fraction of the retrieved documents which are relevant, a measure of the ability of the system to present only relevant documents.

$$Precision = \frac{number\ of\ relevant\ documents\ retrieved}{total\ number\ of\ documents\ retreived} = \frac{|R_a|}{|A|} \qquad (5.2)$$

Both of the above measures are set based measures considering the list of documents as an unordered set for the purpose of relevance evaluation. In order to measure the relevance of ranked list of documents, *precision* can be plotted against the *recall* for each retrieved document. Here precision values are interpolated to a set of standard recall levels (from 0 to 1in increments of 0.1). This is based on the idea that the highest precision obtained for any real recall level greater than or equal to i, has to be used as an interpolated precision at standard recall level i. The method assigns an interpolated value at recall level 0.0 also, unlike in a non-interpolated case. Based on an illustration given below, tabular representation for recall and precision has been depicted in Table 5.1. The whole idea has been demonstrated in Recall-Precision graph in Figure 5.1 where real precision values have been plotted with bubbles connected by solid lines and interpolated precision values have been highlighted with dashed lines.

Illustration

Assume a document collection has 20 documents, four of which are relevant to topic t. Further assume a retrieval system ranks the relevant documents first, second, fourth, and fifteenth. The exact recall points are 0.25, 0.5, 0.75, and 1.0. Using the interpolation rule, the interpolated precision for all standard recall levels up to .5 is 1, the interpolated precision for recall levels .6 and .7 is .75, and the interpolated precision for recall levels .8 or greater is .27.

Table 5.1: Tabular representation of Recall-Precision illustration

Document Rank	Relevance	Recall	Precision
1	R	0.25	1
2	R	0.5	1
3	IR	0.5	0.67
4	R	0.75	0.75
5	IR	0.75	0.60
6	IR	0.75	0.50
7	IR	0.75	0.43
8	IR	0.75	0.38
9	IR	0.75	0.33
10	IR	0.75	0.30
11	IR	0.75	0.27
12	IR	0.75	0.25
13	IR	0.75	0.23
14	IR	0.75	0.21
15	R	1	0.27
16	IR	1	0.25
17	IR	1	0.24
18	IR	1	0.22
19	IR	1	0.21
20	IR	1	0.20
R – Relevant, IR – IR-Relevant			

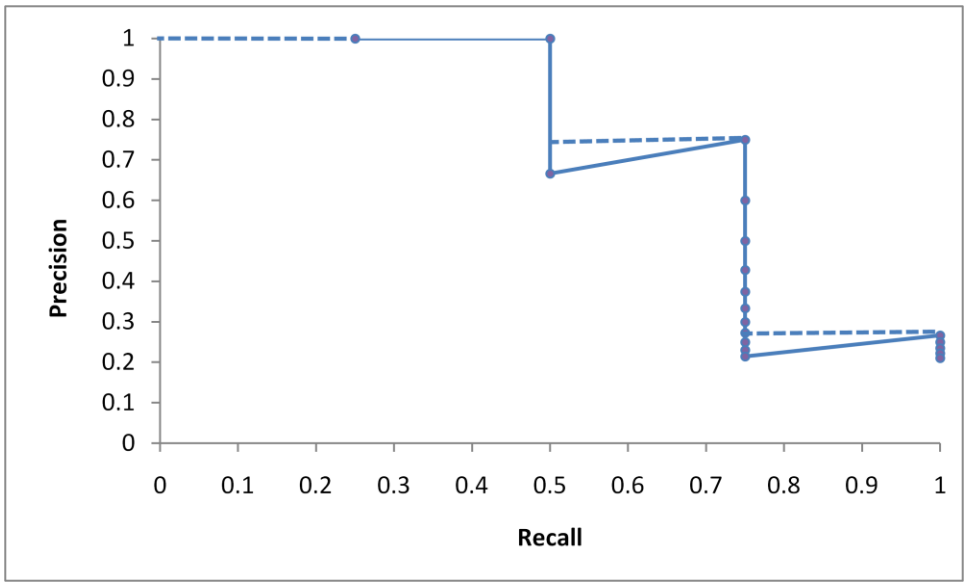

Figure 5.1: Recall-Precision graph demonstrating the interpolation of precision values to a set of standard recall levels

Precision at 11 standard recall levels: The performance of different search systems can be compared in terms of ranking efficiency based on precision averages at 11 standard recall levels which can be visually depicted in the form of recall-precision graph. The precision average at a particular recall level is calculated by adding the interpolated precisions at that level for each query topic in consideration and dividing it by the number of query topics.

Let P_{ik} denote the interpolated precision at recall level i for query topic k,

N denote the number of query topics in consideration,

$\sum_{k=1}^{N} P_{ik}$ denote the sum of interpolated precisions at recall level i for each query topic in consideration,

The precision average at recall level i, is calculated as:

$$\frac{\sum_{k=1}^{N} P_{ik}}{N} \quad \text{where i} \in \{0.0, 0.1, 0.2, 0.3, \ldots.., 1.0\}$$

Average precision over all relevant documents, non-interpolated: This measure reflects a single value based on the precision value obtained after each relevant document is retrieved. Subsequently, the average precision value is calculated by adding the individual precision value for each relevant document and dividing the sum by the number of relevant documents. As an illustration, let there be a query topic with 4 relevant documents which are ranked at position 1, 3, 7, 10 by the search system. The real precision values obtained for each relevant document are 1, 0.67, 0.43 and 0.4 respectively; the average precision value over all relevant documents for this query comes out to be 0.625. More quickly the search system retrieves the relevant documents, better the system is considered.

Precision at n document cut-off values: Rather than considering the precision at specified recall cut-off values, this measure takes into consideration the precision computed after the retrieval of a given number of documents. This seems to be more appealing to the user to know how many relevant documents are coming as a part of the result set on the retrieval of a specified number of documents. The average precision at a particular document cut-off value is obtained by adding the precisions for each of the query topic in consideration at that cut-off value and dividing the sum by the number of topics.

5.3 Example scenario: Financial Bank

Here experiments have been performed based on an ambiguous category identified as *"bank"*. It is assumed that the word *bank* can exist with two different senses namely *financial bank* and *river bank*. A set of context features has been generated which are related to *financial bank* sense of *bank* with the help of following steps:

- A query string *Financial Bank* is entered to a standard search engine Google.
- The top ranked documents are extracted from the result set. (The number of such documents has been taken as two in the scenario to keep the process simple.)
- The documents are filtered for removing unwanted material concerning site navigation and related matter (If in HTML format.)
- The actual content having lexical value is separated out using various techniques viz. stop word removal, stemming and non-alphabetic token removal.
- A small corpus called Web mediated corpus is constructed by appending the lexical tokens of top n documents (n =2) in order to generate corpus knowledge.
- The words lying in the close proximity of the term *financial bank* are separated out in the form of context vector. The close proximity has been considered in the window of size 10 surrounding the category. These sets of 10 words are taken for each occurrence of the category (word) in the corpus.
- The conditional probability of the occurrence of i^{th} word w_i, when category word is already there, is calculated in order to find the words having highest semantic association with the category.
- The set of such identified words having high conditional probability values have been referred to as context features of the category.

The original query is augmented with selected features of the category/domain as part of the original query with an intention to produce a better match between the query intent and the search results, increasing user satisfaction.

5.3.1 Context feature generation: The process glimpse

Python programming integrated with open source library Natural Language Tool Kit (NLTK) have been shown in the snapshots given below with an intention to catch a glimpse of context

feature generation. Top ranked documents pertaining to the query *financial bank* are extracted to generate corpus knowledge (Figure 5.2).

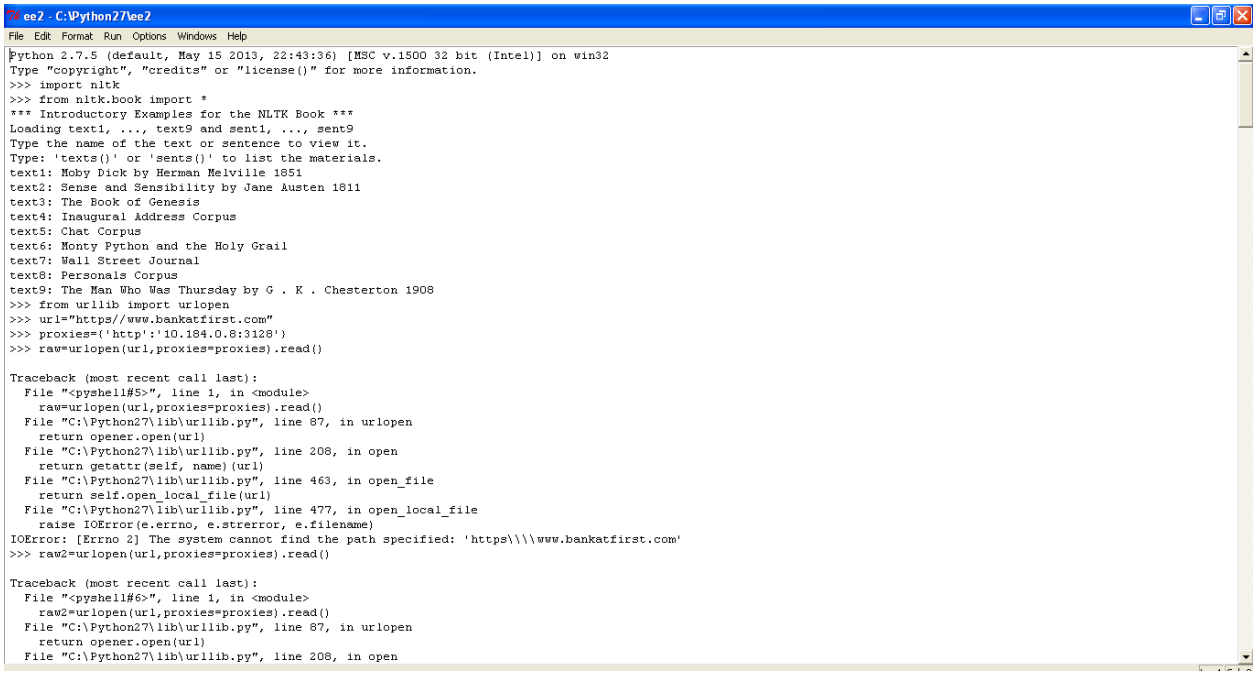

Figure 5.2: The process of corpus knowledge generation: Category *Financial Bank*

A small corpus called Web mediated corpus is constructed by appending the lexical tokens of top n documents (n=2). The words lying in the close proximity of the term *financial bank* are separated out in the form of context vector (Figure 5.3). The context windows of size 10 are taken for each occurrence of the category (word) in the corpus. The conditional probability of the occurrence of i^{th} word w_i, when category word is already there, is calculated in order to find the words having highest strength of relatedness with the category (Figure 5.4). The context features having highest strength of relatedness are shown in Table 5.2. Selection of a small set of good features rather than adding all candidate context features is attributed to maintain *search effectiveness* due to noise reduction: addition of small set of good features is not necessarily less useful than that of all candidate context features and *search efficiency* due to the size of the resulting query: formulated query with less features can be processed more rapidly [128, 129].

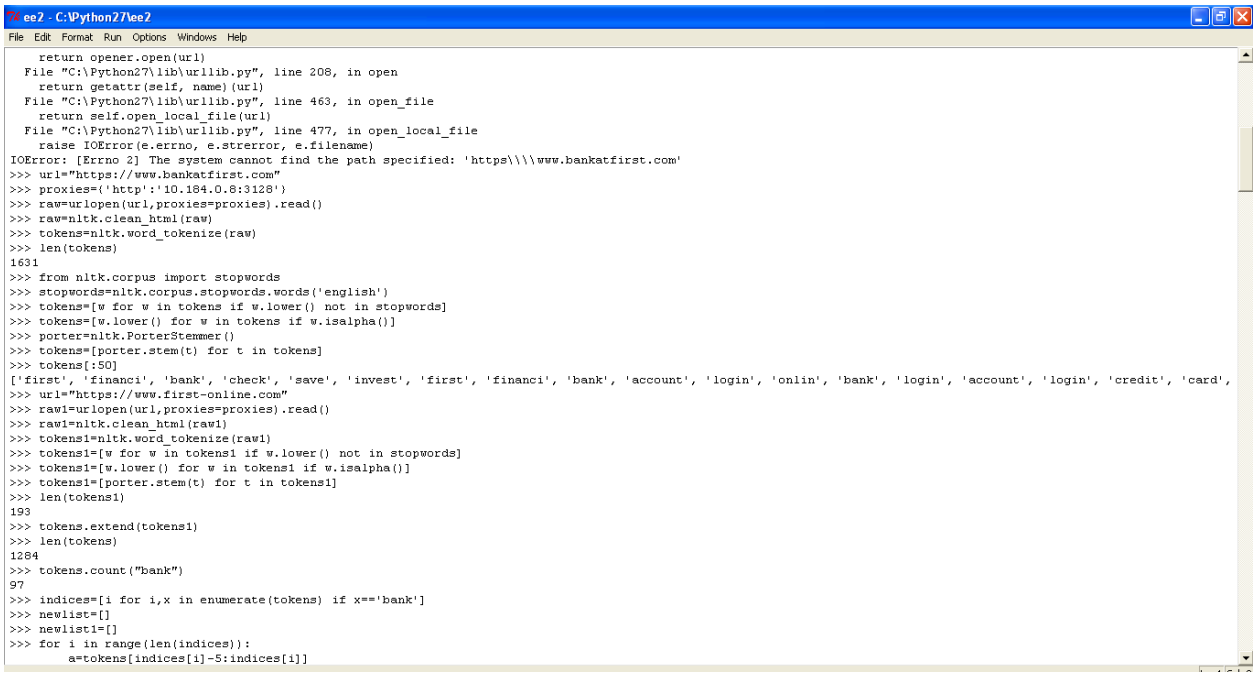

Figure 5.3: The process of context vector formation: Category *Financial Bank*

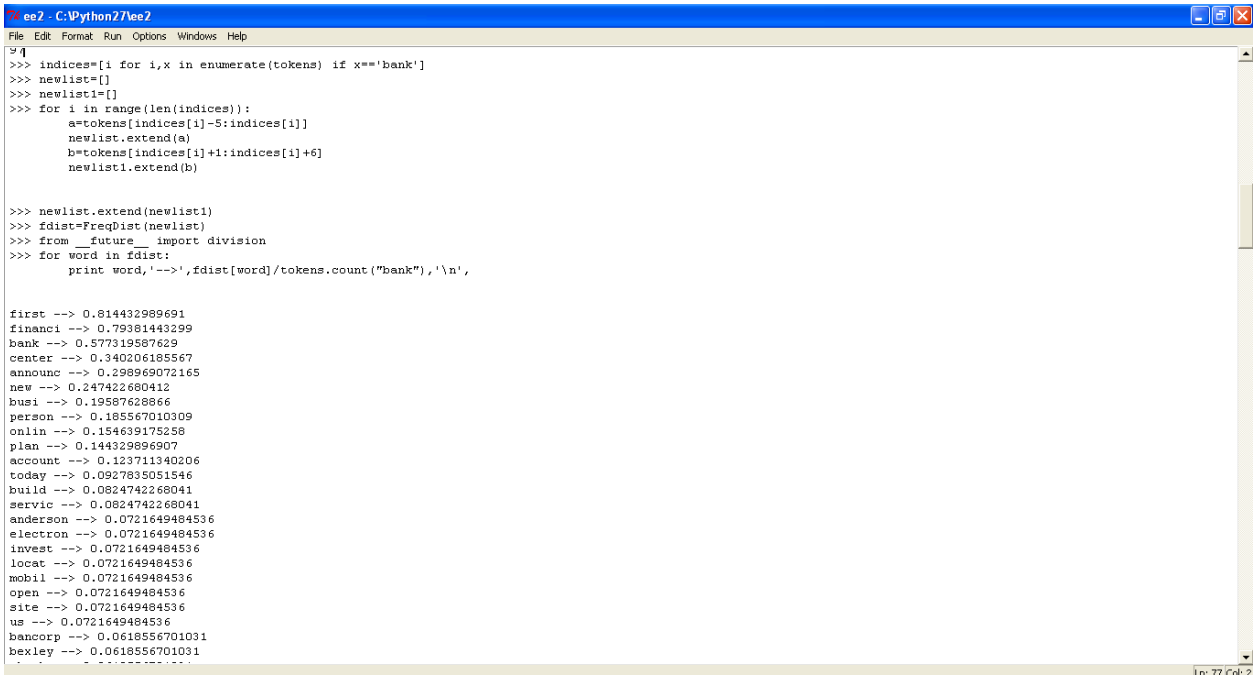

Figure 5.4: The process of calculation of Strength of Relatedness: Category *Financial Bank*

Table 5.2: Context features for original query *Financial Bank*

Context feature	Strength of Relatedness
First	0.814
Financial	0.794
Bank	0.577
Center	0.340
Announce	0.299
New	0.247
Business	0.196
Person	0.186
Online	0.155
Plan	0.144
Account	0.124

5.3.2 Testing and Validation

Precision at 11 standard recall levels: The interpolated precision at 11 standard recall levels have been calculated for the results returned by keyword based search and QUICK based semantic search for the query *Financial Bank* and shown in Table 5.3. The graphical comparison of the performance of two approaches has been shown in Figure 5.5.

Table 5.3: Precision at 11 standard recall levels for QUICK based semantic search and keyword based search: Category *Financial Bank*

Recall	Precision: QUICK based semantic search	Precision: Keyword based search
0.0	1.0000	1.0000
0.1	1.0000	1.0000
0.2	1.0000	0.7500
0.3	0.7500	0.7500
0.4	0.7143	0.5000
0.5	0.7143	0.4545
0.6	0.6667	0.4000
0.7	0.6667	0.2857
0.8	0.6667	0.2857
0.9	0.6667	0.2250
1.0	0.5882	0.1786

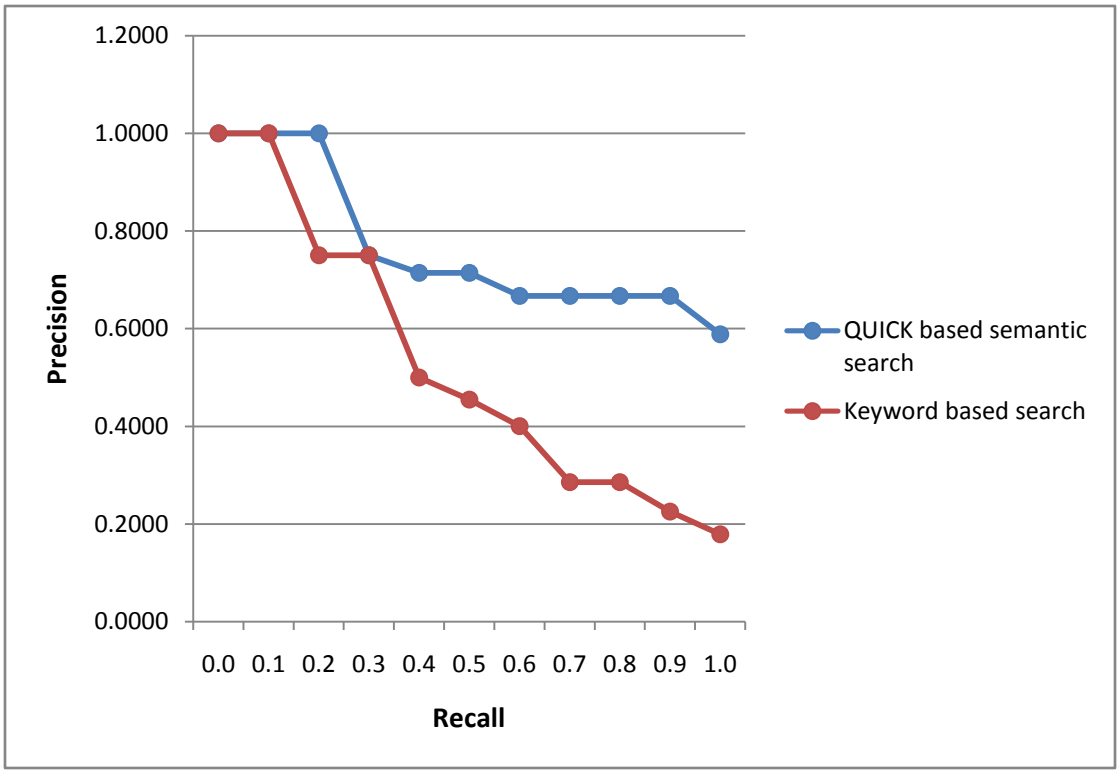

Figure 5.5: Comparison of keyword based search and QUICK based semantic search for precision at 11 standard recall levels: Category *Financial bank*

Precision at 5 document cut-off values: The precision computed after a given number of documents have been retrieved seems to be more relevant in terms of user idea of system performance. It gives an idea about how many relevant results the system is able to return after the retrieval of n documents.

Table 5.4: Precision at 5 document cut-off values for QUICK based semantic search and keyword based search: Category *Financial Bank*

	Precision: QUICK based semantic search	**Precision: Keyword based search**
At 5 docs	0.6000	0.6000
At 10 docs	0.6000	0.4000
At 15 docs	0.6000	0.4000
At 20 docs	0.5000	0.3000
At 30 docs	0.4333	0.2667

Precision at 5 document cut-off values for the results returned by QUICK based semantic search and keyword based search for the category *Financial Bank* have been shown in Table 5.4. The graphical comparison of the performance of two approaches has been shown in Figure 5.6.

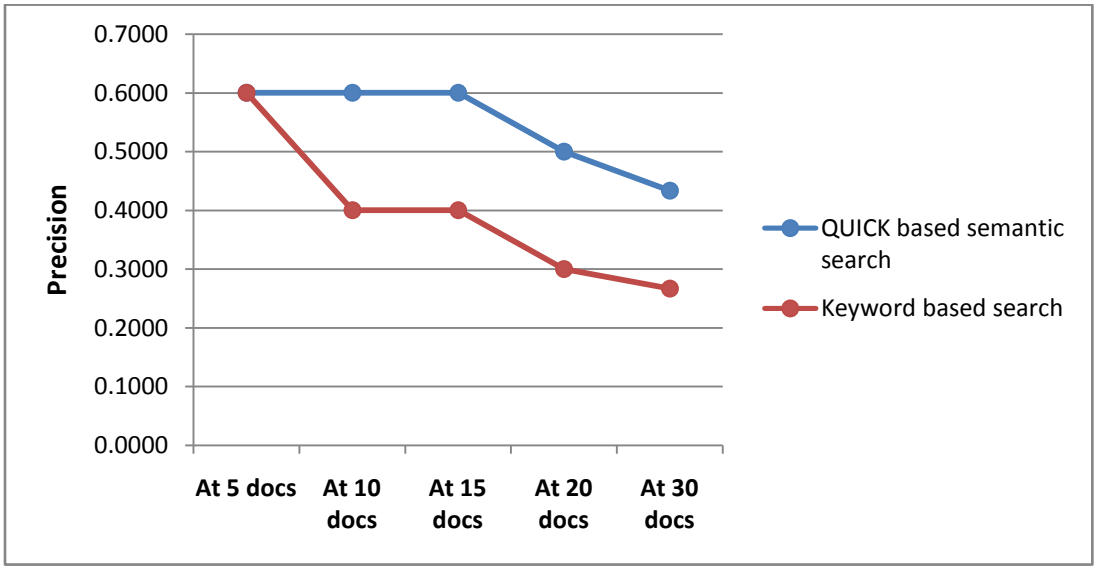

Figure 5.6: Comparison of keyword based search and QUICK based semantic search for precision at 5 document cut-off values: Category *Financial bank*

5.4 Example scenario: Apple Fruit

Another ambiguous category has been identified as "*Apple"* with two senses namely *Apple fruit* and *Apple computer*. Experiments have been performed after generating a set of context features related to *Apple fruit* sense of the Apple. A series of steps have been followed starting from data acquisition to the generation of context features as discussed in the previous section. A set of selected context features is augmented with the original query pertaining to the category *Apple fruit* in order to produce a better match between the query intent and the search results, increasing user satisfaction.

5.4.1 Context feature generation: The process glimpse

A glimpse of the process of context feature generation is given in the form of snapshots demonstrating Python programming integrated with open source library NLTK. Data is acquired from the top ranked documents returned as result on entering query *Apple fruit* to a standard search engine. The resultant set of data is passed through various phases in order to filter out unwanted material having negligible lexical value, generating corpus knowledge (Figure 5.7).

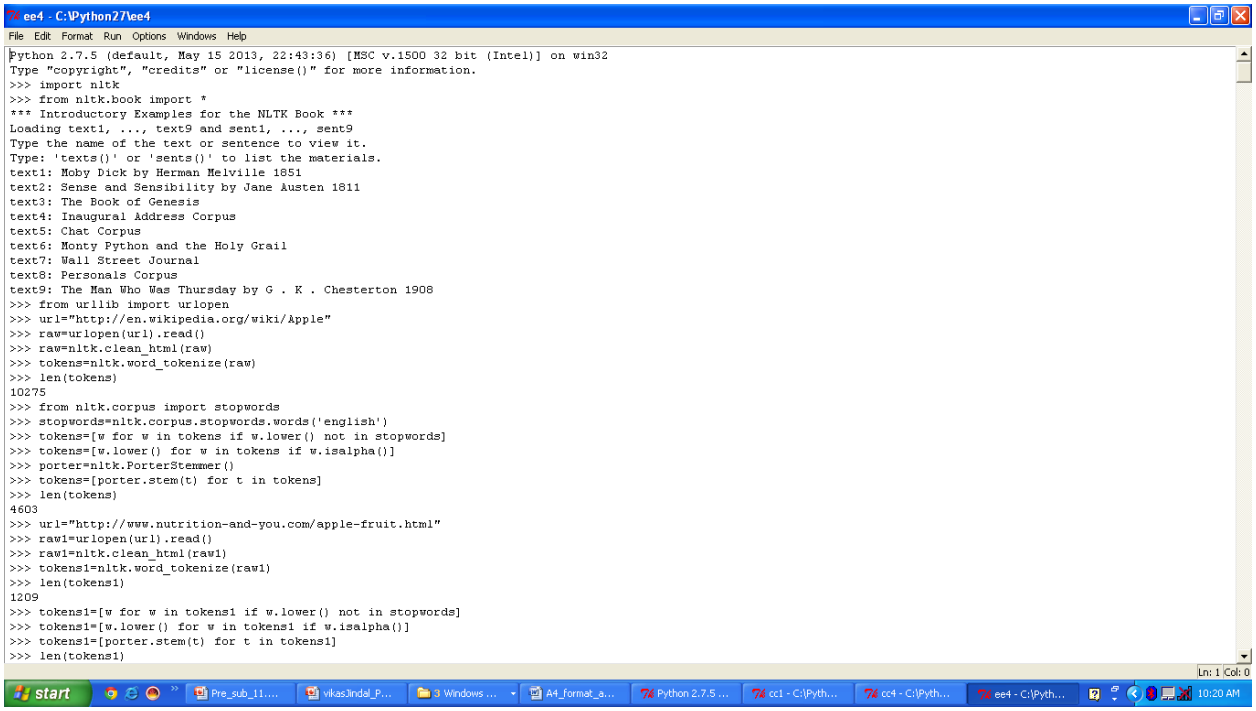

Figure 5.7: The process of corpus knowledge generation: Category *Apple fruit*

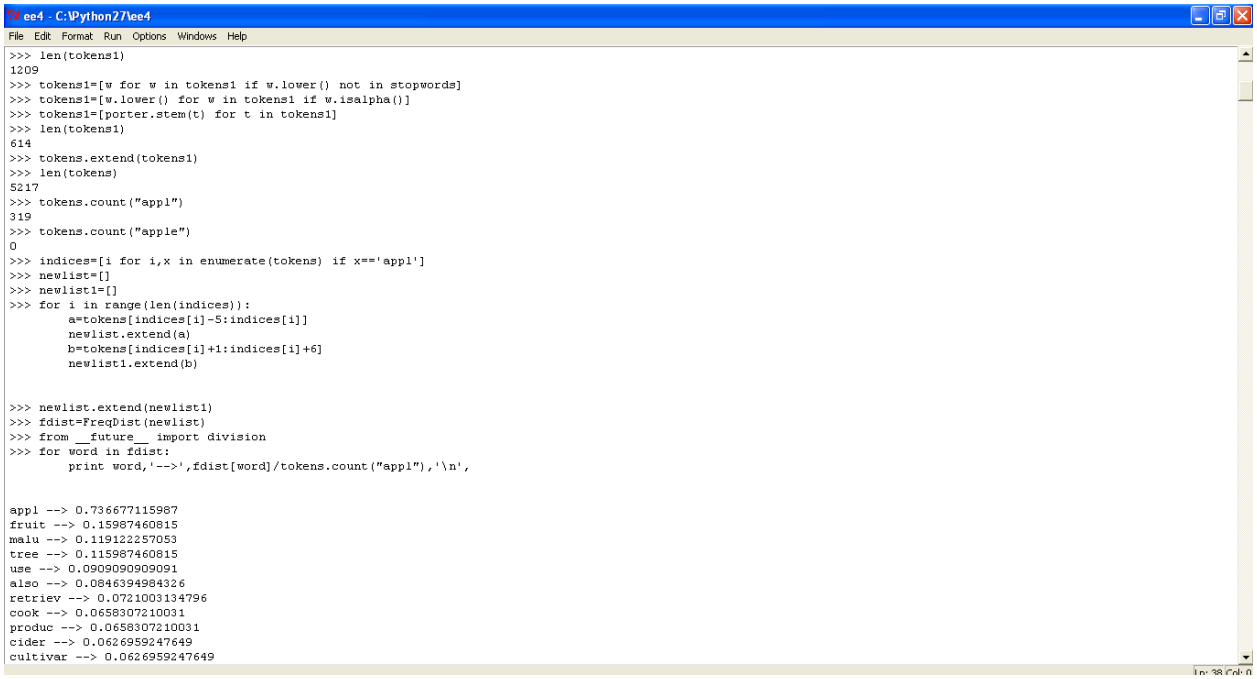

Figure 5.8: The process of context vector formation and calculation of Strength of Relatedness: Category *Apple fruit*

The words lying in the close proximity of the term *Apple fruit* are identified in the form of context vector. The strength of relatedness of context vector words with *Apple fruit* is calculated using conditional probability aspect (Figure 5.8).

The context features having highest strength of relatedness with the category *Apple fruit* are selected so as to augment with the original query pertaining to *Apple fruit*. Using only a small number of context features rather than taking the whole set of features will increase the query processing time quite tolerably. At the same time, retrieval effectiveness is not necessarily hampered with addition of small number of good features in comparison of addition of all the features. Applying the proposed approach, the context features with highest strength of relatedness with *Apple fruit* are presented in Table 5.5.

Table 5.5: Context features for original query *Apple fruit*

Context feature	**Strength of Relatedness**
apple	0.737
Fruit	0.160
Malus	0.119
Tree	0.116
Cook	0.066
Product	0.066
Cider	0.063
Cultivar	0.063
Aphid	0.053

5.4.2 Testing and Validation

Precision at 11 standard recall levels: The interpolated precision at 11 standard recall levels have been calculated for the results returned by keyword based search and QUICK based semantic search for the query *Apple fruit* and shown in Table 5.6. The graphical comparison of the performance of two approaches has been shown in Figure 5.9.

Table 5.6: Precision at 11 standard recall levels for QUICK based semantic search and keyword based search: Category *Apple fruit*

Recall	**Precision: QUICK based semantic search**	**Precision: Keyword based search**
0.0	1.0000	1.0000
0.1	1.0000	1.0000
0.2	1.0000	1.0000
0.3	0.7500	0.7500
0.4	0.7500	0.6000
0.5	0.7500	0.6000
0.6	0.7500	0.6000
0.7	0.7000	0.4000
0.8	0.6923	0.4000
0.9	0.6923	0.3750
1.0	0.5555	0.2941

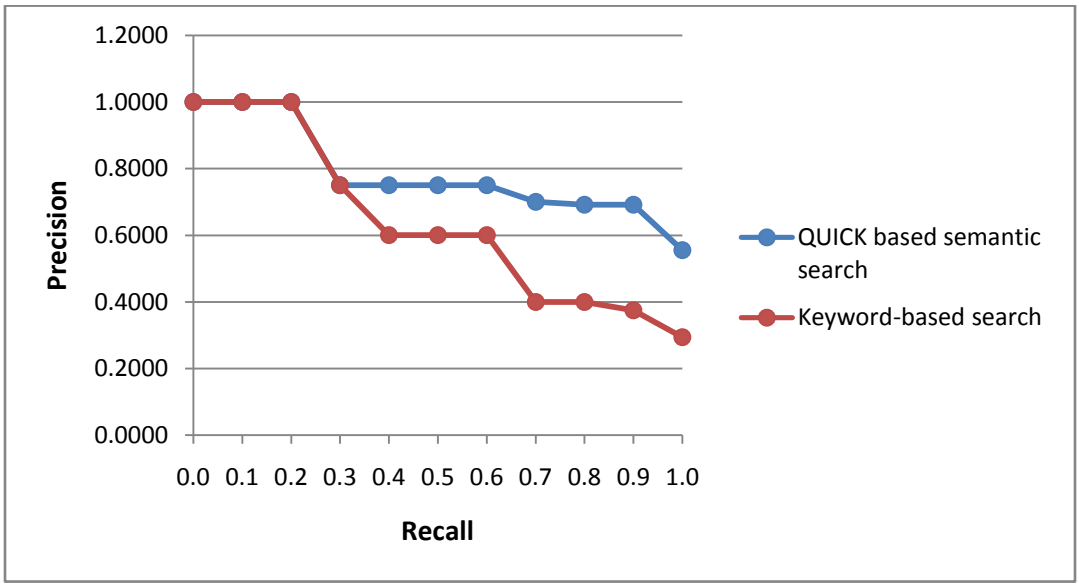

Figure 5.9: Comparison of keyword based search and QUICK based semantic search for precision at 11 standard recall levels: Category *Apple fruit*

Precision at 5 document cut-off values: Precision at 5 document cut-off values for the results returned by QUICK based semantic search and keyword based search for the category *Apple fruit* have been shown in Table 5.7. The graphical comparison of the performance of two approaches has been shown in Figure 5.10.

Table 5.7: Precision at 5 document cut-off values for QUICK based semantic search and keyword based search: Category *Apple fruit*

	Precision: QUICK based semantic search	**Precision: Keyword based search**
At 5 docs	0.6000	0.6000
At 10 docs	0.7000	0.6000
At 15 docs	0.6000	0.4000
At 20 docs	0.5500	0.4000
At 30 docs	0.5333	0.3000

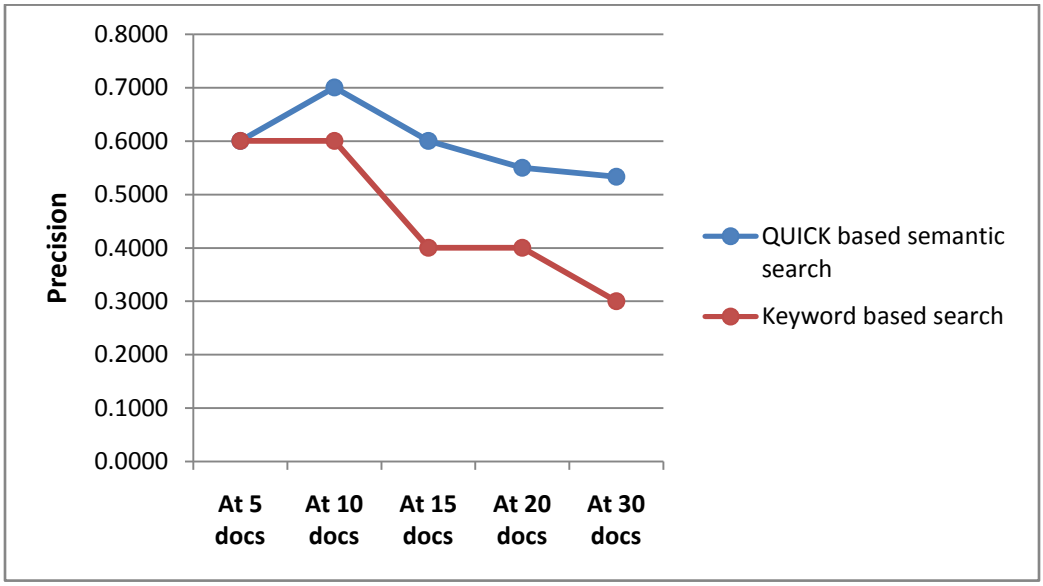

Figure 5.10: Comparison of keyword based search and QUICK based semantic search for precision at 5 document cut-off values: Category *Apple fruit*

Standard recall level precision average: Each recall level precision average has been computed by summing the interpolated precisions at the specified recall cut-off value for both of the categories taken above and dividing the sum by two (Table 5.8). This has been used to compare

the performance of the two systems: keyword based search and QUICK based semantic search and as the input for plotting the recall-precision graph (Figure 5.11). Although the precision of QUICK based semantic search is seen to be dropping down at some recall level in the beginning but its overall average performance is seen to be higher. The idea of precision calculation at various different recall levels is to achieve a near accurate assessment of the participating systems' behavior in terms of their performance.

Table 5.8: Precision average at 11 standard recall levels for QUICK based semantic search and keyword based search

Recall	**QUICK based semantic search**	**Keyword based search**
0.0	1.0000	1.0000
0.1	1.0000	1.0000
0.2	1.0000	0.8750
0.3	0.7500	0.7500
0.4	0.7322	0.5500
0.5	0.7322	0.5273
0.6	0.7084	0.5000
0.7	0.6834	0.3429
0.8	0.6795	0.3429
0.9	0.6795	0.3000
1.0	0.5719	0.2364

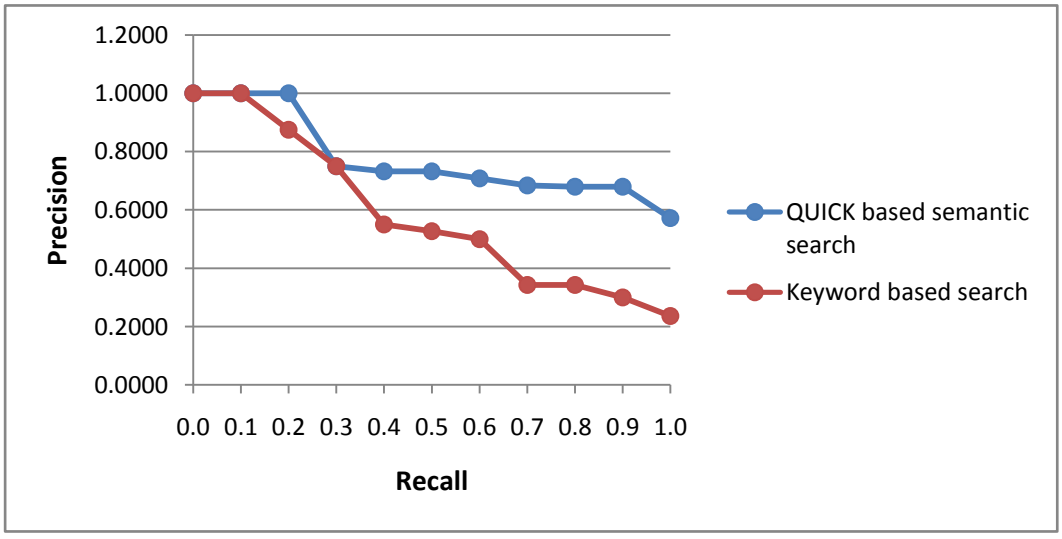

Figure 5.11: Comparison of keyword based search and QUICK based semantic search for average precision at 11 standard recall levels

Document level averages: Each document precision average has been computed by summing the precisions at the specified document cut-off value and dividing by the number of topics that is two (Table 5.9). The graphical comparison of the performance of two approaches has been shown in Figure 5.12.

Table 5.9: Average Precision at 5 document cut-off values for QUICK based semantic search and keyword based search

	QUICK based semantic search	**Keyword based search**
At 5 docs	0.6000	0.6000
At 10 docs	0.6500	0.5000
At 15 docs	0.6000	0.4000
At 20 docs	0.5250	0.3500
At 30 docs	0.4833	0.2833

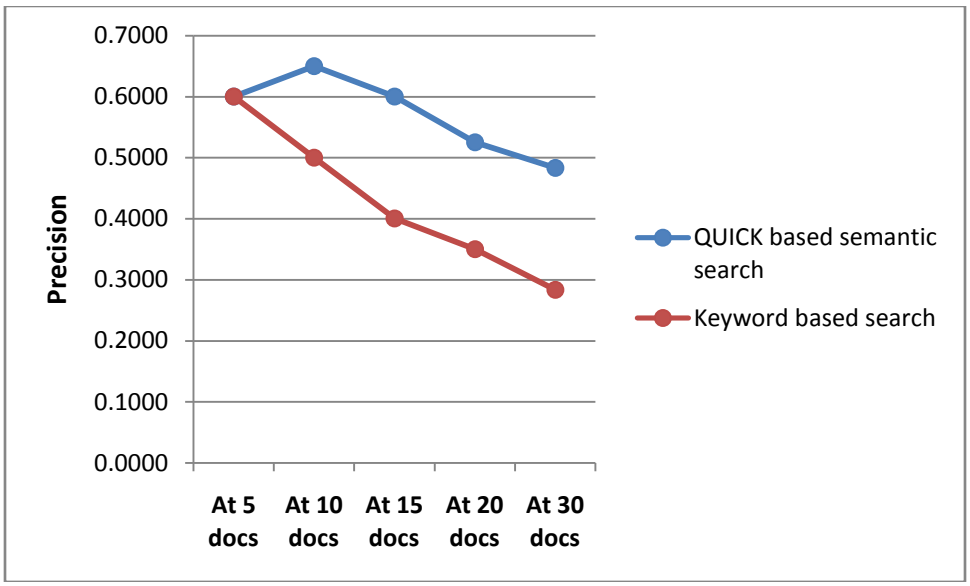

Figure 5.12: Comparison of keyword based search and QUICK based semantic search for average precision at 5 document cut-off values

Chapter 6

Conclusions and Future Scope

With this chapter, the findings of the entire thesis are concluded along with potential scope for future directions in the said domain. The semantic search on Web which has been related to finding the search elements on the Semantic Web is getting a gradual overlap to adding semantics to Web search queries for the purpose of efficient Web search. As a result, the reasoning capabilities envisioned in the Semantic Web paradigm are being imported to Web search and the Web.

6.1 Conclusions

A number of approaches are available and reported in the literature proposing to contribute effectively towards Web search with the help of various semantic (web) techniques. A distinctive characteristic of a semantic search system is the explicit use of semantics in view of the meaning of the resources made available for search. Meaning is established through a semantic model, which essentially captures interrelationships between syntactic elements and their interpretations. Various semantic models work on incorporating the intent of the query and the meaning of the content possibly at different stages of search process such as query construction, query processing, result presentation and query refinement. There are linguistic models such as thesauri [37, 44] or corpora [102, 105] that capture relations between words. The lexical richness of the corpora plays a significant role in finding the implicit relations between words. Sometimes clearly related word pairs may be assigned low similarity value due to data sparseness. It has been observed that ontologies are also being used since they provide consistent vocabularies and

word representations necessary for unambiguous communication within knowledge domains. They facilitate to distinguish the meaning of words from free text sentences using domain specific contextual vocabulary. But where, on one hand, the design and construction of domain ontologies is labour intensive, time consuming and difficult [24], on the other hand, regular updation and maintenance of ontologies is a crucial issue in view of changing senses of the existing words and addition of new words.

Unlike knowledge models such as ontologies and thesauri which provide a mean to find the context of user's query in local context, the Web covers a huge set of domains ranging from news articles and blogs to highly specialize medical terminology. A peculiar set of features is identified endorsing the suitability of this data-driven global technique from different angles. Web based approach uses the features of the Web document collections which suit for arguably all language collections. Using Web as the source of data, the corpus of initial set of documents automatically becomes more relevant to the query topic due to the use of robust algorithms like PageRank. The approach is able to generate basic corpus for invariably every domain or category; it is small in size and enriched with lexical content. An important feature is the feasibility of its re-generation with respect to time and location. The periodic regeneration of Web mediated corpora preserves its potential for generating ever-relevant feature set pertaining to specified categories at low cost. The findings based on the literature review of existing approaches provide a clear ground for formalizing the problem and proposing a framework for semantic based search on Web.

A framework called QUICK: **QU**ery context based **I**nformation Retrieval using **C**orpus **K**nowledge has been proposed. The proposed technique has the potential to overcome one of the main limitations of present day search systems that is the difficulty faced in providing more complete and precise description of information need. The entire framework has been divided into five phases: i) Data acquisition ii) Preprocessing iii) Corpus knowledge generation iv) Candidate context feature generation v) Feature selection. The implementation has been done using a variety of language processing features available in an open source library NLTK by integrating it with Python language interpreter. Implementations details have been elaborated in the chapter on design and implementation of QUICK. Various categories covering different domains have been identified and category specific user query is entered to a standard search engine in order to receive a relevant set of documents as part of the data acquisition. The top

ranked documents are preprocessed for filtering out useless content in order to generate corpus knowledge. A potential set of context features semantically related to the category has been generated using some peculiar aspect of conditional probability. The features having highest strength of relatedness with the category are treated as the most probable context of the category and are referred while dealing with the query pertaining to the category.

The addition of the context features as part of the query helps in addressing various query aspects in terms of sub-topics covered underlying the query topic which in turn would be having greater probability of reaching to the query intention. The experiments for the comparison of result set precision for QUICK based semantic search and standard keyword base search have been performed and the findings are reported in chapter on testing and validation. The proposed semantic based search approach has shown significant improvement in terms of more satisfying result set to the user.

Hence, it is concluded that although there is no standard technique to address the vocabulary problem through semantic search, the proposed technique has the potential to overcome difficulty faced in providing more complete and precise description of information need which in turn produces more relevant search results satisfying the user's information need.

6.2 Future Scope

Query context has been exhibited as one of the dominating criterion for semantic based search on Web in the proposed approach still there is a reasonable potential for further improvement in the processes and activities involved in reaching to the most probable context of the query. One of the potential improvement areas can be the identification of categories in case of homonym words. Rather than using a pre-defined list of senses in terms of categories, it may be more convenient to use a corpus as evidence to perform word sense induction. Hence, better methods can be worked out for automatically inducing the word senses replacing the process of manually inducing the word sense. One possibility can be worked out in terms of finding the context of every occurrence of a word and similar contexts can be used as deciding factor for determining the word sense in an efficient manner. One more aspect for further research can be related to finding the timing of feature generation which can also be tested for various possibilities with an intention to further improve the search effectiveness with reasonable response time.

In broader sense, although semantically oriented search engines and specifically that use ontologies as enabling technologies have gained considerable interest in the last decade, there is still a huge potential for working towards human like interface to the knowledge and services available on the Web. Many a times, multiple ontologies are likely to be referred to satisfy the needs of complex queries. The search system must be able to search several different domains at the same time. Ontology mapping emerges as one of the potential research problems to overcome issues of interoperability by detecting semantic relationships between concepts, properties and instances of two ontologies. The need for standardized evaluation benchmarks has also been felt in order to judge the quality and performance of semantic based search systems. Systematic evaluation of semantic search tools involve appropriate test collection of data and queries, standard performance criteria and independent judgments of performance, thus, supporting performance comparisons between systems. Present approaches for semantic search evaluation are mostly based on user-centric methods, small scale and difficult to repeat.

Bibliography

[1] C. Mooers, "The theory of digital handling of non-numerical information and its implications to machine economics," in *Proceedings of the meeting of the Association for Computing Machinery at Rutgers University*, March 1950.

[2] R. Baeza Yates and B. Ribeiro Neto, *Modern Information Retrieval*. Harlow, UK: Addison- Wesley, 1999.

[3] W. B. Croft and D. J. Harper, "Knowledge-based and statistical approaches to text retrieval," *IEEE Expert: Intelligent Systems and their Applications*, vol. 8, no. 2, pp. 8-12, 1993.

[4] T. burners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," *Scientific American*, pp. 34-43, May 2001.

[5] C. D. Manning, P. Raghavan and H. Schütze, *An Introduction to Information Retrieval*. Cambridge, England: Cambridge University Press, 2007.

[6] H. Chen, "Machine Learning for Information Retrieval: Neural Networks, Symbolic Learning, and Genetic Algorithms," *Journal of the American Society for Information Science*, vol. 46, no. 3, pp.194-216, 1995.

[7] D. C. Blair and M. E. Maron, "An evaluation of retrieval effectiveness for a full-text Document retrieval system," *Communications of ACM*, vol. 28, no. 3, pp. 289-299, Mar. 1985.

[8] T. Lau and E. Horvitz, "Patterns of search: Analyzing and modeling Web query refinement," in *Proceedings of the 7th International Conference on User Modeling* (New York: Springer Wien), pp. 119–128, June 1999.

[9] A. Broder, " A taxonomy of web search," *ACM SIGIR Forum,* vol. 36, no. 2, pp. 3–10, 2002.

[10] C.J. Rijsbergen, "A new theoretical framework for information retrieval", in *Proc. of the ACM Conf. on Research and Dev. in IR*, pp. 194-200, 1986.

[11] S. Kumar and R. B. Mishra, "Semantic Web Service Composition," *IETE Technical Review,* vol. 25, no. 3, May-June 2008.

[12] S. E. Robertson and K. Sparck Jones, "Simple, Proven Approaches to Text Retrieval," J*ournal of the American Society for Information Science*, pp. 129-146, 1976.

[13] K. Sparck Jones, S. Walker and S. E. Robertson, "A probabilistic model of information retrieval: development and comparative experiments," *Information Processing and Management,* vol. 36, no. 6, pp. 779 – 808, 2000.

[14] **S.** Deerwester et al., "Indexing by latent semantic analysis," *Journal of the Society for Information Science,* vol. 41, no. 6, pp. 391–407, 1990.

[15] Tom M. Mitchell, *Machine Learning*, McGraw-Hill, International Edition, 1997.

[16] Jon M. Kleinberg, "Authoritative sources in a hyperlinked environment", *Journal of the ACM (JACM)*, Vol 46, Issue 5, pp.604–632, 1999.

[17] S. Brin and L. Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine", in *Proceedings of the seventh international conference on World Wide Web 7*, Elsevier Science Publishers B. V. Amsterdam, The Netherlands, The Netherlands ©1998, pp. 107—117, 1998.

[18] P. Baldi, P. Frasconi and P. Smyth, *Modeling the Internet and the Web: Probabilistic Method and Algorithm*s, John Wiley, 2003.

[19] G. Klyne and J.J. Corroll, "Resource Description Framework (RDF*): Concepts and Abstract syntax*," http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/., 2004.

[20] D. Brickley and R. V. Guha, "*RDF Vocabulary Description Language* 1.0: RDF Schema," http://www.w3.org/TR/2004/REC-rdf-scema-20040210/, Feb. 2004.

[21] M. Dean and G. Schreiber, "OWL Web Ontology Language Reference," http://www.w3.org/TR/2004/REC-owl-ref-20040210/, Feb 2004.

[22] T. Tran, P. Haase and R. Studer, "Semantic Search – Using Graph-Structured Semantic Models for Supporting the Search Process," in *Conceptual Structures: Leveraging Semantic Technologies*, S. Rudolph, F. Dau and S. O. Kuznetsov, Eds. Lecture Notes in Computer Science, 5662, pp. 48-65, 2009.

[23] P. P. Chen, "The entity-relationship model - toward a unified view of data," ACM Trans. Database Syst., vol. 1, no. 1, pp. 9–36, 1976.

[24] V. Kashyap, "Design and creation of ontologies for environmental information retrieval," in *AOS Workshop*, Rome, Nov. 2001.

[25] T. Tran, D. M. Herzig and G. Ladwig, "SemSearchPro – Using semantics throughout the search process," *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 9, no. 4, pp. 349–364, 2011.

[26] C. Muller and I. Gurevych, "A Study on the Semantic Relatedness of Query and Document Terms in Information Retrieval," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP '09),* pp. 1338–1347, Aug. 2009.

[27] T. Gruber, "A translation approach to portable ontologies," *Knowledge Acquisition*, vol. 5, no. 2, pp. 199–220, 1993.

[28] G. Leroy et al., "Customizable and ontology-enhanced medical information retrieval interfaces," *Methods of Info in Medicine*, 2000.

[29] E. M. Voorhees, "Query expansion using lexical semantic relations," in *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '94),* New York, NY, USA, pp. 61–69, 1994.

[30] C. Fellbaum, WordNet: An Electronic Lexical Database, Cambridge, MA: MIT Press, 1998.

[31] R. Mandala, T. Tokunaga and H. Tanaka, "The Use of WordNet in Information Retrieval," in *Proceedings of the COLING-ACL workshop on Usage of WordNet in Natural Language Processing*, Somerset, New Jersey, pp. 31-37, 1998.

[32] L. Dali and B. Fortuna, "Learning to rank for semantic search", in *Proc. of fourth international Semantic Search workshop* located at 20th international World Wide Web Conference WWW2011, 2011.

[33] M. Lesk, "Word–word associations in document retrieval systems," *American Documentation*, vol. 20, no. 1, pp. 27–38, 1969.

[34] K. Sparck-Jones, "An evaluation of query expansion by addition of clustered terms for a document retrieval system," *Information Storage and Retrieval*, vol. 9, no. 6, p. 339, 1973.

[35] R. Cilibrasi and P. Vitanyi, "The Google Similarity Distance," *IEEE Trans. Knowledge and Data Eng*., vol. 19, no. 3, pp. 370-383, Mar. 2007.

[36] P. Turney, "Mining the web for synonyms: PMI-IR versus LSA on TOEFL," in *Proceedings of the Twelfth European Conference on Machine Learning (ECML-2001),* pp. 491-502, 2001.

[37] C. Leacock and M. Chodorow, "Combining local context and WordNet similarity for word sense identification," in *WordNet: An Electronic Lexical Database,* C. Fellbaum, Ed. Cambridge: The MIT Press, pp. 265–283, 1998.

[38] T. Kliegr et al., "Combining image captions and visual analysis for image concept classification" in *proceedings of the 9th International Workshop on Multimedia Data Mining*, pp. 8-17, 2008.

[39] D. Bollegala, Y. Matsuo, M. Ishizuka, "An integrated approach to measuring semantic similarity between words using information available on the web," in *proceedings of NAACL HLT*, pp. 340-347, 2007.

[40] R. Barzilay and M. Elhadad, "Using lexical chains for text summarization," in *Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization*, pp. 10-17, Madrid, Spain, July 1997.

[41] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," in *Proceedings of ACL'94*, pp. 133-138, 1994.

[42] L. Finkelstein et al., "Placing search in context: the concept revisited" in *ACM Transactions on Information Systems*, vol. 20, no. 1, pp. 116 – 131, 2002.

[43] D. Inkpen and A. Desilets, "Semantic Similarity for Detecting Recognition Errors in Automatic Speech Transcripts," in *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP),* pages 49–56, Vancouver, October 2005 © 2005 Association for Computational Linguistics.

[44] G. Hirst and D. St-Onge, "Lexical chains as representations of context for the detection and correction of malapropisms" in *WordNet: An Electronic Lexical Database*, ch. 13, C. Fellbaum, Ed., Cambridge: The MIT Press, pp. 305–332, 1998.

[45] W. B. Croft, "User-specified domain knowledge for document retrieval," in *Proceedings of the 9th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '86)*, Pisa, Italy, pp. 201-206, 1986.

[46] C. J. Rijsbergen, *Information Retrieval.* Newton, MA: Butterworth-Heinemann, 1979.

[47] P. Apisakmontri, E. Nantajeewarawat, M. Buranarach and M. Ikeda, " Towards the use of upper ontologies for refugee emergencies in disaster management," in *Proceedings of the Second Asian Conference on Information Systems (ACIS 2013)*, 31 October – 2 November 2013, Phuket, Thailand, pp. 153-160, 2013.

[48] B. Popov et al., "KIM – A Semantic Platform for Information Extraction and Retrieval," *Journal of Natural Language Engineering*, vol. 10, no. 3-4, Cambridge University Press New York, NY, USA, pp. 375-392, Sept. 2004.

[49] P. A. Chirita et al., "Activity based metadata for semantic desktop search," in *Proceedings of 2nd European Semantic Web Conference (ESWC '05),* Springer-Verlag Berlin, Heidelberg ©2005, Heraklion, Greece, pp. 439-454, 2005.

[50] U. Shah et al., "Information Retrieval on the Semantic Web," in *Proceedings of Eleventh International Conference on Information and Knowledge Management (CIKM '02)*, 4-9 November 2002, McLean, VA, USA, 2002.

[51] Aleman-Meza et al., "Ranking Documents Semantically Using Ontological Relationships" in *Proceedings of IEEE Fourth International Conference on Semantic Computing (ICSC)*, Pittsburgh, PA, pp. 299-304, 2010.

[52] A. P. Sheth and C. Ramakrishnan, "Relationship Web: Blazing Semantic Trails between Web Resources," *IEEE Internet Computing* vol. 11, no. 4, pp. 77-81, 2007.

[53] R. V. Guha, R. McCool, and E. Miller, "Semantic search," in *Proc. WWW-2003*, ACM Press, pp. 700–709, 2003.

[54] J. Heflin and J. Hendler, "Searching the Web with SHOE," *Artificial Intelligence for Web Search Papers from the AAAI Workshop*, WS-00-01, 2000.

[55] H. Stuckenschmidt and F. Van Harmelen, "Ontology based Metadata Generation from Semi-Structured Information" i*n Proc. Of K-CAP 01*, 2001.

[56] N. Stojanovic, et al., "SEAL: a framework for developing semantic portals," in *Proc. of the 1st int. conf. on knowledge capture (K-CAP)*, 2001.

[57] A. Sheth et al., " Managing Semantic Content for the Web," *IEEE Internet Computing*, vol. 6, pp.80-87, 2002.

[58] J. Davies, U. Krohn, and R. Weeks, "Quizrdf: search technology for the Semantic Web," in *WWW2002 workshop on RDF and Semantic Web Applications*, 11th int. WWW Conf, 2002.

[59] B. Aleman-meza et al., " Context aware semantic association ranking," in *Proc. Of SWDB 03,1st Int. Workshop on Semantic Web and Databases*, 2003.

[60] N. Stojanovic, R. Studer and L. Stojanovic, "An approach for the ranking of query results in the semantic Web," in *Proceedings of second International Semantic Web Conference, ISWC'03*, pp. 500-516, 2003.

[61] B. Bamba and S. Mukherjea, "Utilizing resource importance for ranking semantic web query results," in *Semantic Web and Databases, 2nd Int. workshop, SWDB 2004*, 2004.

[62] A. Kiryakov et al., "Semantic annotation, indexing and retrieval," *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 2, no.1, pp. 49-79, 2004.

[63] C. Rocha, D. Schwabe and M. P. Aragao, "A Hybrid Approach for searching in the Semantic Web," in *Proceedings of the 13th international conference on World Wide Web (WWW2004),* 17-22 May, 2004, NY, USA, 2004.

[64] L.Zhang et al., "An enhanced model for searching in semantic portals," in *WWW 05:Proc. of the 14th Int. Conf. on World Wide Web,* 2005.

[65] L. Ding et al., "Swoogle: a search and metadata engine for the semantic web," in *Proceedings of ACM thirteenth conference on Information and Knowledge Management, CIKM 2004,* New York, NY, USA, pp. 652-659, 2004.

[66] L. Ding et al., "Finding and ranking knowledge on the semantic web," in *Proceedings of the 4th International Semantic Web Conference*, LNCS, Springer-Verlag, vol. 3729, pp. 156–170, 2005.

[67] K. Anyanwu, A. Maduko and A. Sheth, "SemRank: Ranking complex relation search results on the Semantic Web" in *Proceedings 14th International Conference on World Wide Web (WWW 05),* pp. 117-127, 2005.

[68] Y. Lei, V. S. Uren, and E. Motta, "SemSearch: A search engine for the Semantic Web," in *Proc. EKAW-2006*, Springer: LNCS 4248, pp. 238–245, 2006.

[69] H. Hwang, V. Hristidis and Y. Papakonstantinou, "Objectrank: A system for authority-based search on databases," in *Proceedings of the ACM SIGMOD International Conference on Management of Data,* June 27-29 2006, Chicago, Illinois, USA, pp. 796–798, 2006.

[70] P. Castells, M. Fernández and D. Vallet, "An adaptation of the vector space model for ontology-based information retrieval," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 2, pp. 261–272, 2007.

[71] Y. Li, Y. Wang and X. Huang, "A Relation based search engine in Semantic Web," *IEEE Trans. on Knowledge and Data Engineering,* vol. 19, no. 2, pp. 273-282, 2007.

[72] G. Kasneci et al., "NAGA: Searching and ranking knowledge" in *Proc. ICDE-2008*, IEEE Computer Society, pp. 953–962, 2008.

[73] F. M. Suchanek, G. Kasneci, and G. Weikum, "YAGO: A core of semantic knowledge*"* in *Proc. of WWW 07*, 2007.

[74] F. Lamberti, A. Sanna and C. Demartini, "A Relation based Page Rank Algorithm for Semantic Web Search Engines," *IEEE Transactions On Knowledge and Data Engineering*, vol. 21, no. 2, pp. 123-136, 2009.

[75] J. P. McGlothin and L. R. Khan, "RDFKB: Efficient support for RDF inference queries and Knowledge Management" in *Proc of IDEAS 09*, 2009.

[76] W. Wei, P. Barnaghi and A. Bargiela, "Rational Research model for ranking semantic entities," *Information Sciences*, vol. 181, no. 13, pp. 2823–2840, 2011.

[77] D. Artz and Y. Gil, "A survey of trust in computer science and the Semantic Web," *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 5, no. 2, pp. 58-71, 2007.

[78] P. Hasse, T. Mathäß and M. Ziller, "An evaluation of approaches to federated query processing over linked data," in *Proceedings of the 6th International Conference on Semantic Systems (I-SEMANTICS '10)*, New York, NY, USA, 2010 © ACM.

[79] S. N. Wrigley et al., "Evaluating semantic search tools using SEALS platform," in *Proceedings of the International Workshop on Evaluation of Semantic Technologies (IWEST 2010)*, Full day workshop at 9th International Semantic Web Conference(ISWC2010), Shanghai, China, 2010.

[80] O. Corby, R. Dieng-Kuntz, and C. Faron-Zucker, "Querying the SemanticWeb with Corese search engine," in *Proc. of ECAI-2004*, IOS Press, pp. 705–709. 2004.

[81] T. W. Finin et al., "Swoogle: Searching for knowledge on the Semantic Web," in *Proc. AAAI-2005*, AAAI Press / MIT Press, pp. 1682–1683, 2005.

[82] J. Heflin, J. A. Hendler, and S. Luke, "SHOE: A blueprint for the Semantic Web," in *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*, D. Fensel, W. Wahlster, and H. Lieberman, Eds., MIT Press, pp. 29–63, 2003.

[83] V. Novácek, T. Groza, and S. Handschuh, "CORAAL — Towards deep exploitation of textual resources in life sciences," in *Proc. AIME-2009*, Springer: LNCS 5651, pp. 206–215, 2009.

[84] E. Oren, C. Guéret, and S. Schlobach, "Anytime query answering in RDF through evolutionary algorithms," in *Proc. ISWC- 2008*, Springer: LNCS 5318, pp. 98–113, 2008.

[85] E. Thomas, J. Z. Pan and D. H. Sleeman, "ONTOSEARCH2: Searching ontologies semantically" in *Proc. OWLED-2007*, CEUR Workshop Proceedings 258, CEUR-WS.org, 2007.

[86] P. Buitelaar, T. Eigner, and T. Declerck, "OntoSelect: A dynamic ontology library with support for ontology selection," in *Proc. Demo Session at ISWC-2004*, 2004.

[87] G. Cheng, W. Ge and Y. Qu, "Falcons: Searching and browsing entities on the Semantic Web," in *Proc. WWW-2008*, ACM Press, pp. 1101– 1102, 2008.

[88] A. Harth et al., "SWSE: Answers before links!," in *Proc. Semantic Web Challenge 2007*, CEUR Workshop Proceedings 295, CEUR-WS.org, 2007.

[89] T. Tran et al., "Ontology based interpretation of keywords for semantic search," in *Proc. ISWC/ASWC-2007*, Springer: LNCS 4825, pp. 523–536, 2007.

[90] G. Tummarello et al., " Sig.ma: Live views on the Web of data," in *Proc. WWW-2010*, ACM Press, pp. 1301–1304, 2010.

[91] *Yahoo!SearchMonkey* Available: http://developer.yahoo.com/searchmonkey.

[92] G. Zenz et al., "From keywords to semantic queries - Incremental query construction on the Semantic Web," *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 7, no. 3, pp. 166–176, 2009.

[93] P. Cimiano et al., "Towards portable natural language interfaces to knowledge bases - The case of the ORAKEL system," *Data and Knowedge Engineering*, vol. 65, no. 2, pp. 325–354, 2008.

[94] D. Damljanovic, M. Agatonovic, and H. Cunningham, "Natural language interface to ontologies: Combining syntactic analysis and ontology-based lookup through the user interaction," in *Proc. ESWC-2010, Part I*, Springer: LNCS 6088, pp. 106–120, 2010.

[95] M. Fernández et al., "Semantic search meets the Web," in *Proc. ICSC-2008*, IEEE Computer Society, pp. 253–260, 2008.

[96] R. Rawat, R. Nayak and Y. Li, "Identifying interests of web users for effective recommendations," *International Journal of Innovation, Management and Technology*, vol. 2, no. 1, 2011.

[97] V. Lopez, M. Pasin, and E. Motta, "AquaLog: An ontology portable question answering system for the Semantic Web," in *Proc. ESWC-2005*, Springer: LNCS 3532, pp. 546–562, 2005.

[98] V. Lopez, M. Sabou, and E. Motta, "PowerMap: Mapping the real Semantic Web on the fly," in *Proc. ISWC-2006*, Springer: LNCS 4273, pp. 414–427, 2006.

[99] E. Motta and M. Sabou, "Next Generation Semantic Web Applications," in *Proceedings of 1st Asian Semantic Web Conference*, Beijing, China, 3-7 September 2006.

[100] T. Lukasiewicz and J. Schellhase, "Variable-strength conditional preferences for ranking objects in ontologies," *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 5, no. 3, pp. 180–194, 2007.

[101] P. Resnik, "Using information content to evaluate semantic similarity," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, Montreal, Canada, pp. 448–453, 1995.

[102] J. J. Jiang and D.W. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy," in *Proceedings of International Conference on Research in Computational Linguistics (ROCLING X)*, Taiwan, pp. 19–33, 1997.

[103] D. Lin, "An information-theoretic definition of similarity," in *Proceedings of the 15th International Conference on Machine Learning*, Madison, WI, pp. 296-304, July, 1998.

[104] H. Kucera and W. Francis, *Computational analysis of present-day American English*. Brown University Press, 1967.

[105] *The British National Corpus, version 3 (BNC XML Edition).* Distributed by Oxford University Computing Services on behalf of the BNC Consortium, Available: http: //www.natcorp.ox.ac.uk/, 2007.

[106] M. Marcus, B. Santorini and M. Marcinkiewicz, "Building a large annotated corpus of English: the Penn Treebank," *Journal of Computational Linguistics - Special issue on using large corpora*, vol. 19, no. 2, pp. 313 – 330, 1993.

[107] T. Rose, M. Stevenson and M. Whitehead, "The reuters corpus volume 1 - from yesterday's news to tomorrow's language resources," in *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, pp. 29 – 31, 2002.

[108] D. Harman and, M. Liberman, *TIPSTER*, vol. 1, Linguistic Data Consortium, Philadelphia, 1993.

[109] A. Budanitsky and G. Hirst, "Evaluating WordNet based measures of semantic distance," *Computational Linguistics*, vol. 32, no. 1, pp. 13-47, 2006.

[110] J. R. Firth, "A synopsis of linguistic theory 1930-1955," in *Studies in linguistic analysis*, Oxford: Philological Society, pp. 1-32, 1957; reprinted in *Selected Papers of J.R. Firth*, F. Palmer, Ed. Harlow: Longman, 1968.

[111] P. Mitra and C. Chaudhari, "Efficient algorithm for the extraction of association rules in data mining," in *Proceedings of Computational Science and Its Applications*, LNCS, Springer-Verlag, vol. 3981, pp. 1 – 10, 2006.

[112] Y. Matsuo et al., "Graph-based word clustering using a web search engine," in *Proceedings of the 2006 Conference on Empirical Method in Natural Language Processing (EMNLP '06)*, pp. 542-550, 2006.

[113] J. Gracia and E. Mena, "Web-Based Measure of Semantic Relatedness," in *proceeding of the 9th international conference on Web Information Systems Engineering(WISE '08),* pp. 136-150, 2008.

[114] L. Lee, "Measures of distributional similarity," in *proceedings of ACL'99*, pp. 25–32, 1999.

[115] J. Curran and M. Moens, "Improvements in Automatic Thesaurus Extraction," in *Proceedings of the Workshop of the ACL Special Interest Group on the Lexicon*, pp. 59-66, 2002.

[116] A. Kilgarrif, "Googleology is bad science," in *Computational Linguistics,* vol. 33, no. 1, pp. 147-151, 2007.

[117] A. Spink et al., " Searching the web: The public and their queries," *Journal of the American Society for Information Science,* vol. 52, no. 3, pp. 226–234, 2001.

[118] L. Fu, D. H. Goh and S. S. Foo, "Evaluating the effectiveness of a collaborative querying environment," in *Proceedings of the 8th International Conference on Asian Digital Libraries: implementing strategies and sharing experiences (ICADL '05*), Lecture Notes in Computer Science, Springer-Verlag Berlin, Heidelberg ©2005, pp. 342–351, 2005.

[119] V. Jindal, S. Bawa and S. Batra, "A Review of ranking approaches for semantic search on Web," *Information Processing & Management,* vol. 50, no. 2, pp. 416-425, Mar. 2014.

[120] M. F. Porter, "An algorithm for suffix stripping," in *Readings in Information Retrieval*, K. S. Jones and P. Willett, Eds. San Francisco, CA: Morgan Kaufmann, pp. 313–316, 1997.

[121] J. Lin and G. C. Murray, "Assessing the term independence assumption in blind relevance feedback," in *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '05),* 15-19 August 2005 Salvador, Brazil, pp. 635-636, 2005.

[122] G. Amati, "Probabilistic models for information retrieval based on divergence from randomness," Ph.D. thesis, Department of Computing Science, University of Glasgow, Glasgow, Scotland, UK, 2003.

[123] Y. Chang, I. Ounis and M. Kim, "Query reformulation using automatically generated query concepts from a document space," *Information Processing and Management,* vol. 42, no. 2, pp. 453–468, March 2006.

[124] A. Bernardini and C. Carpineto, "FUB at TREC 2008 relevance feedback track: extending rocchio with distributional term analysis," in *Proceedings of TREC-2008,* National Institute of Standards and Technology, Gaithersburg, MD, USA, 2008.

[125] C. Buckley et al., "Automatic query expansion using SMART: TREC3," in *Proceedings of the 3rd Text REtrieval Conference (TREC-3),* National Institute of Standards and Technology (NIST), Gaithersburg, MD, pp. 69–80, 1995.

[126] W. S. Wong et al., "Re-examining the effects of adding relevance information in a relevance feedback environment," *Information Processing and Management,* vol. 44, no. 3, pp. 1086–1116, May, 2008.

[127] C. Zhai and J. Lafferty, "Model-based feedback in the language modeling approach to information retrieval," in *Proceedings of the 10th International Conference on Information and Knowledge Management (CIKM '01),* ACM New York, NY, USA ©2001, pp. 403–410, 2001.

[128] G. Salton and C. Buckley, "Improving retrieval performance by relevance feedback," *Journal of the American Society for Information Science,* vol. 41, no. 4, pp. 288–297, 1990.

[129] D. Harman, "Relevance feedback and other query modification techniques," in *Information Retrieval – Data Structures and Algorithms*, W. B. Frakes and R. Baeza-Yates, Eds. Englewood Cliffs, N. J.: Prentice Hall, pp. 241–263, 1992.

[130] C. Buckley and D. K. Harman, "Reliable information access final workshop report," in *Proceedings of the Reliable Information Access Workshop (RIA),* NRRC, pp. 1–30, 2004.

[131] G. Cao et al., "Selecting good expansion terms for pseudo-relevance feedback," in *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '08),* ACM New York, NY, USA ©2008, pp. 243–250, 2008.

[132] C. Carpineto, G. Romano and V. Giannini, "Improving retrieval feedback with multiple term-ranking function combination," *ACM Transactions on Information Systems.,* vol. 20, no. 3, pp. 259–290, July, 2002.

[133] C. Carpineto, R. D. Mori, G. Romano and B. Bigi, "An information theoretic approach to automatic query expansion," *ACM Transactions on Information Systems.,* vol. 19, no. 1, pp. 1–27, Jan. 2001.

[134] S. Bird, E. Klein and E. Loper, *Natural Language Processing with Python*. Sebastopol, CA: O'Reilly Media Inc., 2009.

List of publications

1. V. Jindal, S. Bawa and S. Batra, "A Review of ranking approaches for semantic search on web," *Information Processing & Management,* vol. 50, no. 2, pp. 416-425, Mar. 2014. (SCI indexed journal, Impact Factor: 1.069, 5-Year Impact Factor: 1.481)
2. V. Jindal, S. Bawa and S. Batra, "A query-context oriented approach to semantic search on web," *International Journal of Artificial Intelligence and Knowledge Discovery*, vol. 5, no. 1, pp. 14-20, Jan. 2015.