Chapter 2

Literature Survey

This chapter begins with the study of the earlier efforts made in Dawning of the Web Era, Information Retrieval, Query Expansion focusing on the area of Ontologies and Semantic Meta Data.

2.1 The Dawning of the Web Era

The Web is a system of interlinked, hypertext documents accessed via the Internet. The WWW is more and more used for application to application communication. The programmatic interfaces made available are referred to as web services. A web service is a software system designed to support computer to computer interaction over the Internet. Most people today can hardly conceive life without the internet. The web of documents has morphed into a web of data. The semantic wave embraces three stages of internet growth. The first stage, Web 1.0, was about connecting information and getting on the net. Web 2.0 is about connecting people putting the “I” in user interface and the “we” into a web of social participation. The next stage, Web 3.0, is starting now. It is about representing meanings, connecting knowledge and putting them to work in ways that make our experience of internet more relevant, useful and enjoyable.
2.1.1 Web 1.0

In Web 1.0, a small number of writers created web pages for a large number of readers. As a result, people could get information by going directly to the source. The WWW or Web 1.0 is a system of interlinked, hypertext documents accessed via the Internet.

Fig 2.1: WWW or Web 1.0

The first implementation of the web represents the Web 1.0, which, according to Berners-Lee, could be considered the “read-only web.” In other words, the early web allowed us to search for information and read it. There was very little in the way of user interaction or content contribution. However, this is exactly what most website owners wanted, their goal for a website was to establish an online presence and make their information available to anyone at any time [11].

2.1.2 Web 2.0

Currently, we are seeing the infancy of the Web 2.0 or the “read-write” web if we stick to Berners-Lee’s method of describing it. The newly introduced ability to contribute content and interact with other web users has dramatically changed the landscape of the web in a short time. Technologies such as web blogs, social bookmarking, wikis, podcasts, RSS feeds, social software, web APIs and online web
services such as eBay and Gmail provide enhancements over read-only websites.

![Fig 2.2: Web 2.0](image)

Web 2.0 makes use of latest technologies and concepts in order to make the user experience more interactive, useful and interconnecting. It has brought yet another way to interconnect the world by means of collecting information and allowing it to be shared affectively.

**Examples of Web 2.0 based websites**

1. Flickr: A photo sharing website which allows users to upload their photographs and share it with anyone and everyone.
2. Orkut: Social networking site which allows the users to send messages and communicate with other members.
3. YouTube: It allows the users to upload their videos and share it with everyone.
4. Blogs: Maintained by individuals or groups, they can be used to convey anything.
5. Google AD sense: Allows users to earn money through posting Google ads on their websites.

6. Wikipedia: Online encyclopedia where in the users contributes by writing the articles, definitions, etc. It is completely edited and maintained by the users.

7. Scribd: Users can upload any documents on the website where other users can either download or view those documents online.

### 2.1.3 Web 3.0

Web 3.0 is a term that is used to describe various evolutions of Web usage and interaction among several paths. These include transforming the Web into a database, a move towards making content accessible by multiple non-browser applications, the leveraging of artificial intelligence technologies, the Semantic web, the Geospatial Web or the 3D web. By extending Tim Berners-Lee’s explanations, the Web 3.0 would be something akin to a “read-write-execute” web. Web 3.0 is defined as the creation of high quality content and services produced by gifted individuals using web 2.0 technologies as an enabling platform [12].

Web 3.0 is a web where the concept of website or webpage disappears, where data isn’t owned but instead shared, where services show different views for the same web/data. Those services can be applications (like browsers, virtual worlds or anything else), devices and have to be focused on context and personalization and both will be reached by using vertical search [14].
The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation. The Semantic Web requires the use of a declarative ontological language like OWL to produce domain specific ontologies that machines can use to reason about information and make new conclusions, not simply match keywords.

Fig 2.3: Web 3.0

“The Semantic Web will bring structure to the meaningful content of Web pages, creating an environment where software agents roaming from page to page can readily carry out sophisticated tasks for users.’

With the Semantic Web, computers will scan and interpret information on Web pages using software agents. These agents could utilize metadata, ontologies, and logic to carry out its tasks. Agents are pieces of software that work autonomously and proactively on the Web to perform certain tasks.
The development of the Semantic Web is proceeding in a step-by-step approach building one layer on top of another as shown in figure 2.4 below.

We describe briefly these layers:

- **Unicode and URI**: Unicode, the standard for computer character representation, and URIs, the standard for identifying and locating resources (such as pages on the Web), provide a baseline for representing characters used in most of the languages in the world and for identifying resources.
- XML: XML and its related standards such as Namespaces and Schemas, form a common means for structuring data on the Web but without communicating the meaning of the data.

- Resource Description Framework: RDF is the proper first layer of the Semantic Web. RDF is a simple metadata representation framework, using URIs to identify Web based resources and a graph model for describing relationships between resources. Several syntactic representations are available, including a standard XML format.

- RDF Schema: a simple type modeling language for describing classes of resources and properties between them in the basic RDF model. It provides a simple reasoning framework for inferring types of resources.

- Ontologies: a richer language for providing more complex constraints on the types of resources and their properties.

- Logic and Proof: an (automatic) reasoning system provided on top of the ontology structure to make new inferences. Thus, using such a system, a software agent can make deductions as to whether a particular resource satisfies its requirements (and vice versa).

- Trust: The final layer of the stack addresses issues of trust that the Semantic Web can support. This component has not progressed far beyond a vision of allowing people to ask questions of the trustworthiness of the information on the Web, in order to provide an assurance of its quality.
Three important technologies for developing the Semantic Web are eXtensible Markup Language (XML), Resource Description Framework (RDF) and Ontologies.

### 2.1.3.1 XML

The eXtensible Markup Language (XML) is a universal meta-language for defining markup. It provides a uniform framework for exchanging data between applications. It builds upon the original and most basic layer of the Web, Hypertext Markup Language (HTML). However, XML does not provide a mechanism to deal with the semantics (the meaning) of data.

XML lets everyone create their own tags, hidden labels such as `<zip code>` or `<alma mater>` that annotate Web pages or sections of text on a page. Scripts, or programs, can make use of these tags in sophisticated ways, but the script writer has to know what the page writer uses each tag for. Meaning is expressed by RDF, which encodes it in sets of triples, each triple being rather like the subject, verb and object of an elementary sentence.

### 2.1.3.2 RDF

Resource Description Framework (RDF) was developed by the World Wide Web Consortium (W3C) for Web based metadata in order to build and extend XML. The goal of RDF is to make work easier for
autonomous agents and automated services by supplying a rudimentary semantic capability.

The RDF is a format for data that uses a simple relational model that allows structured and semi-structured data to be mixed, exported, and shared across different applications. It is a data model for objects and relationships between them and is constructed with an object-attribute-value triple called a statement. While XML provides interoperability within one application (e.g., producing and exchanging bank statements) using a given schema, RDF provides interoperability across applications (e.g., importing bank statements into a tax calculating program).

It uses Universal Resource Identifiers (URI) as a mechanism for uniquely identifying the subject, predicate, and object of a statement. The subject, predicate, and object are each first-class citizens of the data model.

These triples can be written using XML tags. In RDF, a document makes assertions that particular things (people, Web pages or whatever) have properties (such as "is a sister of," "is the author of") with certain values (another person, another Web page). This structure turns out to be a natural way to describe the vast majority of the data processed by machines. Subject and object are each identified by a Universal Resource Identifier (URI), just as used in a link on a Web page. The verbs are also identified by URIs, which enables anyone to define a new concept, a new verb, just by defining a URI for it somewhere on the Web.
The triples of RDF form webs of information about related things. Because RDF uses URIs to encode this information in a document, the URIs ensure that concepts are not just words in a document but are tied to a unique definition that everyone can find on the Web. For example, imagine that we have access to a variety of databases with information about people, including their addresses. If we want to find people living in a specific zip code, we need to know which fields in each database represent names and which represent zip codes. RDF can specify that "(field 5 in database A) (is a field of type) (zip code)," using URIs rather than phrases for each term.

The RDF model is based on statements made about resources that can be anything with an associated URI (Universal Resource Identifier). The basic RDF model produces a triple, where a resource (the subject) is linked through an arc labeled with a property (the predicate) to a value (the object).

The RDF statements can be represented as:

A resource [subject] has a property [predicate] with a specific value [object]. This can be reduced to a triple: (subject, predicate, object).

2.1.3.3 Ontology

A program that wants to compare or combine information across the two databases has to know that these two terms are being used to mean the same thing. Ideally, the program must have a way to discover such common meanings for whatever databases it
encounters. A solution to this problem is provided by the third basic component of the Semantic Web, collections of information called ontologies.

In philosophy, ontology is a theory about the nature of existence, of what types of things exists, ontology as a discipline studies such theories. Artificial Intelligence and Web researchers have co-opted the term for their own jargon and for them ontology is a document or file that formally defines the relations among terms. The most typical kind of ontology for the Web has taxonomy and a set of inference rules. The taxonomy defines classes of objects and relations among them. For example, an address may be defined as a type of location and city codes may be defined to apply only to locations and so on. Classes, subclasses and relations among entities are a very powerful tool for Web use. We can express a large number of relations among entities by assigning properties to classes and allowing subclasses to inherit such properties. If city codes must be of type city and cities generally have Web sites, we can discuss the Web site associated with a city code even if no database links a city code directly to a Web site. Inference rules in ontologies supply further power. Ontology may express the rule "If a city code is associated with a state code, and an address uses that city code, then that address has the associated state code." A program could then readily deduce, for instance, that a Cornell University address, being in Ithaca, must be in New York State, which is in the U.S., and therefore should be formatted to U.S. standards. The computer doesn't truly "understand"
any of this information, but it can now manipulate the terms much more effectively in ways that are useful and meaningful to the human user. With ontology pages on the Web, solutions to terminology (and other) problems begin to emerge. The meaning of terms or XML codes used on a Web page can be defined by pointers from the page to ontology. Of course, the same problems as before now arise if I point to an ontology that defines addresses as containing a zip code and you point to one that uses postal code. This kind of confusion can be resolved if ontologies (or other Web services) provide equivalence relations: one or both of the ontologies may contain the information that zip code is equivalent to postal code.

Ontologies can enhance the functioning of the Web in many ways. They can be used in a simple fashion to improve the accuracy of Web searches, the search program can look for only those pages that refer to a precise concept instead of all the ones using ambiguous keywords.

2.1.3.4 OWL

The main requirements of an ontology language are a well-defined syntax, a formal semantics, convenience of expression, an efficient reasoning support system and sufficient expressive power. A necessary condition for machine-processing is a well-define syntax. As a result, OWL has been built upon RDF and RDFS and has the same XML based syntax. A formal semantics precisely describes the meaning of knowledge and allows users to reason about knowledge.
Semantics is a prerequisite for reasoning support that allows consistency checks, checks for unintended relationships and automatic classification of instances in classes. A formal semantics and reasoning support can be provided through mapping an ontology language to a known logical formalism.

The OWL is partially mapped on description logic which is a subset of predicate logic for which efficient reasoning support is possible. Both RDF and RDF Schema allow some ontology knowledge representation. The main modeling primitives of RDF–RDFS are concerned with the organization of vocabularies in typed hierarchies. However, some of the missing necessary features are local scope of properties, disjointness of classes, Boolean combinations of classes, cardinality of restrictions and special characteristics of properties. Therefore, an ontology language that is richer than RDF Schema, with respect to these additional features.

The layered architecture of the Semantic Web would suggest that one way to develop the necessary ontology language is to extend RDF Schema by using the RDF meaning of classes and properties (rdfs:classes, etc.) and adding primitives to support richer expressiveness. However, simply extending RDF Schema would fail to achieve the best combination of expressive power and efficient reasoning.

The W3C has defined OWL to include three different sublanguages (OWL Full, OWL DL, OWL Lite) in order to offer different balances of expressive power and efficient reasoning.
**OWL Full:** The entire language is called OWL Full and it uses all the primitives and allows their combination with RDF and RDFS. The OWL Full supports maximum expressiveness and the syntactic freedom of RDF, but has no computational guarantees. For example, in OWL Full, a class can be treated simultaneously as a collection of individuals and as an individual in its own right. It is intended to be used in situations where very high expressiveness is more important than being able to guarantee the decidability or computational completeness of the language. Therefore it may be impossible to perform automated reasoning on OWL Full ontologies.

The advantage of OWL Full is that it is fully compatible with RDF syntax and semantics. Any legal RDF document is also a legal OWL Full document. Any valid RDF–RDFS conclusion is also a valid OWL Full conclusion. The disadvantage of OWL Full is that the language is undecidable and therefore cannot provide complete (or efficient) reasoning support.

**OWL DL:** Ontology Web Language DL (Descriptive Logic) is a sublanguage of OWL Full that restricts how the constructors from OWL and RDF can be used. This ensures that the language is related to description logic. Description Logics are a decidable fragment of First-Order Logic (FOL), and are therefore amenable to automated reasoning. The OWL DL supports strong expressiveness while retaining computational completeness and decidability. It is therefore possible to automatically compute the classification hierarchy and check for inconsistencies in an ontology that conforms to OWL DL.
The advantage of this sublanguage is efficient reasoning support and the loss of full compatibility with RDF. However, every legal OWL DL document is a legal RDF document.

**OWL Lite:** Further restricting OWL DL produces a subset of the language called OWL Lite, which excludes enumerated classes, disjointness statements, and arbitrary cardinality. The OWL Lite supports a classification hierarchy and simple constraints. It is simpler for tool support and provides a quick migration path for taxonomies. While this language is easier to comprehend and is less expressive. The choice between OWL Lite and OWL DL depends on the extent of requirements for expressive constructs. All sublanguages of OWL use RDF for their syntax and instances are declared as in RDF.

The layered architecture of the Semantic Web promotes the downward compatibility and reuse of software. It is only achieved with OWL Full, but at the expense of computational tractability.

### 2.2 Information Retrieval

The ability to search and retrieve information from the Web efficiently and effectively is an enabling technology for realizing its full potential. IR focuses on retrieving documents based on the content of their unstructured components. An IR request (typically called a “query”) may specify desired characteristics of both the structured and unstructured components of the documents to be retrieved. IR typically seeks to find documents in a given collection that are “about” a given topic or that satisfy a given information need. The topic or
information need is expressed by a query, generated from the user. Documents that satisfy the given query in the judgment of the user are said to be “relevant”. Documents that are not about the given topic are said to be “non-relevant”. An IR engine may use the query to classify the documents in a collection (or in an incoming stream), returning to the user a subset of documents that satisfy some classification criterion.

2.2.1 Indexing Web Documents

We can view effective Web searches as an information retrieval problem. IR problems are characterized by a collection of documents and a set of users who perform queries on the collection to find a particular subset of it. This differs from database problems, for example, where the search and retrieval terms are precisely structured. In the IR context, indexing is the process of developing a document representation by assigning content descriptors or terms to the document. These terms are used in assessing the relevance of a document to a user query. They contribute directly to the retrieval effectiveness of an IR system.

IR systems include two types of terms: objective and nonobjective. Objective terms are extrinsic to semantic content and there is generally no disagreement about how to assign them. Examples include author name, document URL and date of publication. Nonobjective terms, on the other hand, are intended to reflect the information manifested in the document and there is no
agreement about the choice or degree of applicability of these terms. Thus, they are also known as content terms. Indexing in general is concerned with assigning nonobjective terms to documents. The assignment may optionally include a weight, indicating the extent to which the term represents or reflects the information content.

The effectiveness of an indexing system is controlled by two main parameters. Indexing exhaustively reflects the degree to which all the subject matter manifested in a document is actually recognized by the indexing system. When the indexing system is exhaustive, it generates a large number of terms to reflect all aspects of the subject matter present in the document and when it is nonexhaustive, it generates fewer terms, corresponding to the major subjects in the document. Term specificity refers to the breadth of the terms used for indexing. Broad terms retrieve many useful documents along with a significant number of irrelevant ones, narrow terms retrieve fewer documents and may miss some relevant items.

The effect of indexing exhaustivity and term specificity on retrieval effectiveness can be explained by two parameters used for many years in IR problems: Recall is the ratio of the number of relevant documents retrieved to the total number of relevant documents in the collection. Precision is the ratio of the number of relevant documents retrieved to the total number of documents retrieved.

Ideally, you would like to achieve both high recall and high precision. Indexing terms that are specific yields higher precision at
the expense of recall. Indexing terms that are broad yields higher recall at the cost of precision. For this reason, an IR system’s effectiveness is measured by the precision parameter at various recall levels. Indexing can be performed either manually or automatically. The sheer size of the Web together with the diversity of subject matter make manual indexing impractical. Automatic indexing does not require the tightly controlled vocabularies that manual indexers use and it offers the potential to represent many more aspects of a document.

### 2.2.2 Information Retrieval Models

An IR model is characterized by four parameters:

- Representations for documents and queries.
- Matching strategies for assessing the relevance of documents to a user query.
- Methods for ranking query output.
- Mechanisms for acquiring user relevance feedback.

IR models can be classed into four types: set theoretic, algebraic, probabilistic and hybrid models. In the following sections, we describe instances of each type in the context of the IR model parameters.

#### 2.2.2.1 Set Theoretic Models

The Boolean model represents documents by a set of index terms, each of which is viewed as a Boolean variable and valued as
True if it is present in a document. No term weighting is allowed. Queries are specified as arbitrary Boolean expressions formed by linking terms through the standard logical operators: AND, OR and NOT. Retrieval status value (RSV) is a measure of the query document similarity.

In the Boolean model, RSV equals 1 if the query expression evaluates to True, RSV is 0 otherwise. All documents whose RSV evaluates to 1 are considered relevant to the query. This model is simple to implement and many commercial systems are based on it. User queries can employ arbitrarily complex expressions, but retrieval performance tends to be poor. It is not possible to rank the output since all retrieved documents do not have the same RSV nor can weights be assigned to query terms. The results are often counter-intuitive. For example, if the user query specifies 10 terms linked by the logical connective AND, a document that has nine of these terms is not retrieved. User relevance feedback is often used in IR systems to improve retrieval effectiveness. Typically, a user is asked to indicate the relevance or irrelevance of a few documents placed at the top of the output.

Since the output is not ranked, however, the selection of documents for relevance feedback elicitation is difficult. The fuzzy set model is based on fuzzy set theory, which allows partial membership in a set, as compared with conventional set theory, which does not. It redefines logical operators appropriately to include partial set membership and processes user queries in a manner similar to the
case of the Boolean model. Nevertheless, IR systems based on the fuzzy set model have proved nearly as incapable of discriminating among the retrieved output as systems based on the Boolean model.

The strict Boolean and fuzzy set models are preferable to other models in terms of computational requirements, which are low in terms of both the disk space required for storing document representations and the algorithmic complexity of indexing and computing query document similarities.

2.2.2 Algebraic Models

The vector space model is based on the premise that documents in a collection can be represented by a set of vectors in a space spanned by a set of normalized term vectors. If the vector space is spanned by \( n \) normalized term vectors, then each document will be represented by an \( n \) dimensional vector. The value of the first component in this vector reflects the weight of the term in the document corresponding to the first dimension of the vector space and so forth. A user query is similarly represented by an \( n \) dimensional vector. A query document’s RSV is given by the scalar product of the query and document vectors. The higher the RSV, the greater is the document’s relevance to the query. The strength of this model lies in its simplicity. Relevance feedback can be easily incorporated into it. However, the rich expressiveness of query specification inherent in the Boolean model is sacrificed.
2.2.2.3 Probabilistic Models

The vector space model assumes that the term vectors spanning the space are orthogonal and that existing term relationships need not be taken into account. Furthermore, the model does not specify the query document similarity, which must be chosen somewhat arbitrarily. The probabilistic model takes these term dependencies and relationships into account and in fact, specifies major parameters such as the weights of the query terms and the form of the query document similarity.

The model is based on two main parameters: Pr(rel) and Pr(nonrel), the probabilities of relevance and nonrelevance of a document to a user query, which are computed using the probabilistic term weights and the actual terms present in the document.

Relevance is assumed to be a binary property so that Pr(rel) = 1 - Pr(nonrel). In addition, the model uses two cost parameters, a1 and a2, to represent the loss associated with the retrieval of an irrelevant document and nonretrieval of a relevant document, respectively. The model requires term occurrence probabilities in the relevant and irrelevant parts of the document collection, which are difficult to estimate. However, this model serves an important function for characterizing retrieval processes and provides a theoretical justification for practices previously used on an empirical basis.
### 2.2.2.4 Hybrid Models

As in the case of the vector space model, the extended Boolean model represents a document as a vector in a space spanned by a set of orthonormal term vectors. However, the extended Boolean (or $p$-norm) model measures query document similarity by using a generalized scalar product between the corresponding vectors in the document space. This generalization uses the well known $L^p$ norm defined for an $n$-dimensional vector, $d$, where the length of $d$ is given by
\[
\|d\|_p = \left(\sum_{i=1}^{n} w_i^p\right)^{1/p}
\]
where $1 \leq p \leq \infty$, and $w_1, w_2, \ldots, w_n$ are the components of the vector $d$.

Generalized Boolean OR and AND operators are defined for the $p$ norm model. The interpretation of a query can be altered by using different values for $p$ in computing query document similarity. When $p = 1$, the distinction between the Boolean operators AND and OR disappears, as in the case of the vector space model.

When the query terms are all equally weighted and $p = \infty$, the interpretation of the query is the same as that in the fuzzy set model. On the other hand, when the query terms are not weighted and $p = \infty$, the $p$ norm model behaves like the strict Boolean model. Varying the value of $p$ from 1 to $\infty$ offers a retrieval model whose behavior corresponds to a point on the continuum spanning from the vector space model to the fuzzy and strict Boolean models. The best value for $p$ is determined empirically for a collection, but is generally in the range $2 \leq p \leq 5$. 
Many recent IR systems are built on the popular vector space model, which is capable of producing a ranking of the retrieved documents. The vector space models for information retrieval are just one subclass of retrieval techniques that have been studied in recent years.

A main disadvantage of all term vector models is that terms are assumed to be independent (i.e. no relation exists between the terms). Often this is not the case.

Terms can be related by:

1. **Polysemy**: Terms can be used to express different things in different contexts (e.g. driving a car and driving results). Thus, some irrelevant documents may have high similarities because they may share some words from the query. This affects precision.

2. **Synonymity**: Terms can be used to express the same thing (e.g. car insurance and auto insurance). Thus, the similarity of some relevant documents with the query can be low just because they do not share the same terms. This affects recall.

3. **Long documents** are considered poor representatives of the Vector Space Model because they had poor similarity values (a small scalar product and a large dimensionality).

4. Documents with similar context but different term vocabulary ("False negative match").
5. The search keywords were being typed during the search in an inappropriate manner giving poorer results e.g. key + ing, para + meter ("False positive match").

Models based on and extending the vector space model include:

*Topic based vector space model* (TVSM) [16]: Extends the vector space model by removing the constraint that the term vectors be orthogonal. In contrast to the generalized vector space model the topic based vector space model does not depend on co-occurrence based similarities between terms.

*Latent semantic analysis* [17]: The Latent Semantic Indexing information retrieval model builds upon the prior research in information retrieval and using the singular value decomposition (SVD) to reduce the dimensions of the term document space, attempts to solve the synonymy and polysemy problems that plague automatic information retrieval systems. LSI explicitly represents terms and documents in a rich, high dimensional space, allowing the underlying ("latent"), semantic relationships between terms and documents to be exploited during searching.

### 2.3 Query Expansion

The query expansion techniques have been investigated from the dawn of information retrieval. Query expansion is the process of adding additional search terms to a user’s search. The intent is to improve precision and/or recall. The additional terms may be taken
from a thesaurus or calculated in a statistical or probabilistic way. The query expansion techniques developed can be classified as user assisted and automatic from IR point of view.

2.3.1 User-assisted Techniques

One well known user assisted technique is relevance feedback [18], which requires a user to iteratively judge the relevance of a set of retrieved documents. The documents identified as relevant are used to refine the original query and the search process continues until the user is satisfied with the retrieval results. Relevance feedback is a powerful technique, with an improvement of up to 90% for single search iteration being reported [18]. One major advantage of this method is that it lessens the burden on the user to reformulate a query. One major disadvantage is that it does not improve the retrieval performance of the original query. The user has to invest time and effort judging the relevance of retrieved documents before any improvements can be made.

Another user assisted technique used in most commercial IR systems [19] is to incorporate a browsable thesaurus. Given a set of search terms, a user is presented with a list of similar terms which he/she can choose to replace some existing terms or use as additional terms. Once again, this technique relies on the user's knowledge about the problem domain and the user's ability to judge what is an effective search term.
2.3.2 Automatic Techniques

The automatic techniques for query expansion do not rely on users to make relevance judgments and are often based on language analysis [20] and term co-occurrences [21].

2.3.2.1 Language Analysis Approaches

Language analysis [20] approaches require a deep understanding of queries and documents, usually at high computational costs. These techniques have also been shown to have only small improvements in retrieval performance.

2.3.2.2 Co-occurrence Approaches

The co-occurrence approaches [21] can be grouped into four categories:

1. Term classification
2. Document classification
3. Syntactic context and
4. Relevance information

Term Classification: The term classification methods place a term into a class based on its similarity with other terms in the class. Expansion is done by matching a search term with a similar class and adding the terms from the class to the query. This process is employed after the indexing of the documents.
Document Classification: The document classification is similar to the term classification except that it is used during the indexing phase. Documents are first classified using a document classification algorithm. Infrequent terms found in a document class are considered similar and clustered in the same term class [22]. The indexing of documents and queries is enhanced either by replacing a term by a thesaurus class or by adding a thesaurus class to the index data. The number of documents is much larger than the number of terms in the database. Consequently, document classification is much more expensive and has to be done more often than the simple term classification.

Syntactic Context: The syntactic context methods make use of linguistic knowledge to enhance search terms. The term relations are generated in the basis of linguistic knowledge and co-occurrence statistics [23] [24]. The method uses a grammar and a dictionary to extract for each term t a list of terms. This list consists of all the terms that modifies it. The similarities between terms are then calculated by using these modifiers from the list. Subsequently, a query is expanded by adding those terms most similar to any of the query terms. This produces only slightly better results than using the original queries [23].

Relevance Information: The relevance information [18] methods include the relevance feedback technique without the user involvement. The documents which are judged to be relevant are further used in the expansion process till the user is satisfied.
2.3.3 Thesauri in Query Expansion

Besides term co-occurrence based statistics another way to improve search effectiveness is to incorporate background knowledge into the search process. The IR community concentrated so far on using background knowledge expressed in the form of thesauri.

Thesauri define a set of standard terms that can be used to index and search a document collection (controlled vocabulary) and a set of linguistic relations between those terms, thus promise a solution for the vagueness of natural language and partially for the problem of high level concepts. Unfortunately, while intuitively one would expect to see significant gains in retrieval effectiveness with the use of thesauri, experience shows that this is usually not true. One of the major cause is the noise of thesaurus relations between thesaurus terms. Linguistic relations, such as synonyms are normally valid only between a specific meaning of two words, but thesauri represent those relations on a syntactic level, which usually results in false positives in the search result.

Another big problem is that the manual creation of thesauri and the annotation of documents with thesaurus terms are very expensive. As a result, annotations often are incomplete or erroneous, resulting in decreased search performance.

Using lexical aid as a source of related terms has met with some success in small experiments. Salton and Lesk found that expansion by synonyms improved performance but expansion by broader or narrower terms selected from a hierarchical thesaurus was too
inconsistent [25]. Wang et.al found that a variety of lexical semantic relations improved retrieval performance [26]. Voorhees examines the utility of query expansion by lexical semantic relations in a large collection that spans several domains [27].

All the efforts discussed above take into consideration the syntactic relations existing between them like synonymy and do not talk anything about the semantics of the terms.

### 2.3.4 Ontologies in Query Expansion

The use of ontologies to overcome the limitations of keyword search engines and syntactic based approaches have been put forward as one of the motivations of the Semantic Web since its emergence. A way to improve the IR effectiveness is using the semantics encoded in Ontology. Most of the efforts [28] [29] use the existing vector space model as the underlying model while others have their own information model. Few efforts that incorporate background knowledge stored in the form of ontologies are discussed below:

The effort made in [30] examines how ontology can be exploited during the information retrieval process. This approach is based on the philosophy that it is possible to extend the query syntactically based on the information stored in ontologies so that a simple, syntax-based similarity measure will yield semantically correct results. Ontology is used in query extension and query formulation. The content of a resource is represented by a set of instances from a
domain ontology which is identical with the information model used by classical IR engines built on vector space model [31].

This model has the advantage that the retrieval algorithms, index structures or even complete IR engines can be reused with their model. During query formulation ontology is used to disambiguate the queries specified in the natural text. As soon as the query is presented, the system provides the set of instances of the concept, so that the information seeker can correctly interpret the query. Several ontology based heuristics [32] [33] are used to extend the IR query. Various ontology based heuristics are applied one by one to create separate queries which are executed independently using a traditional full text engine.

A semantic search engine based on query refinements powered by ontology navigation is presented in [28] in which the components of classical vector space model [31]: keywords, documents and queries are re-defined to form the concept level vector model.

In the concept level vector model documents are represented as vectors located in a hyperspace defined by the set of all ontology concepts called “Conceptual Vector”. Documents are semantically described by annotations composed of RDF triples by using DOSE [34]. The components of annotation are an XPointer for resource identification, a weight that plays the role of \( w_{ij} \) in the classical vector space model and a concept identifier. The similarity between the query i.e. conceptual vector and the document is defined exactly as in the
vector space model by the cosine formula. The query is refined using a query expansion technique powered by ontology navigation.

The query refinement uses the semantic relationships between the concepts to find co-related terms for the query in the taxonomic ontology model. Depending upon the relevance threshold specified by an external application and the number of documents required, the navigation of the ontology is triggered. If the number of relevant resources retrieved by the basic search are too low, the query is iteratively expanded concept by concept either by traversing towards its ancestors (generalization) or its descendants (focalization) defined by an internal ontology navigation ratio followed by a search using basic search. The output is a set of URI, XPointer pairs associated to each resource. Richardet.al [29] explores the possibility of extending traditional information retrieval systems with knowledge based approaches (artificial intelligence techniques) to automatically expand natural language queries. Two types of knowledge bases, a domain specific and general world knowledge [35], both represented as semantic networks are used in the expansion process. Conventional vector space model for retrieving information is used. The search strategies include searching either in terms of the broader terms or narrower terms or in both directions in isolated type of search. In correlated type of search a path between the words is discovered and the path is used in the search.

The effort made in [36] proposed a probabilistic query expansion method based on a similarity thesaurus, which was built
automatically by determining the similarities between all of the term pairs. The similarity thesaurus captures the knowledge of a domain. The richness of the thesaurus depends on the document collection used during its construction. The queries are expanded by adding the most similar terms to the concept of the whole query rather than for each term in the query. Assumption is that the documents and queries are represented by the set of keywords and weighted indices. The similarity between the query and the document is calculated by mapping the query from term vector space into the document vector space i.e., by the scalar vector product. The terms with higher similarity ratio are considered for query expansion.

Another semantic searcher has been presented in [37]. It relies on Semantic Web infrastructure and aims at improving traditional web searches by integrating relevant results with data extracted from distributed sources. The user query is also mapped onto a set of ontology concepts and ontology navigation is performed. However, such process only provides other concept instances that are strongly related to the query ones, by means of a breadth first search. The issue of performing semantic inference by means of graph navigation is not addressed and moreover, the system does not work on annotations.

Several concept based search services [38] [39], proposed in the Semantic Web community rely, to a certain degree, on logic inference to extract resources from a given domain in response to a user query.
Basically, they use a reasoning engine to verify an input clause built from the user query and they subsequently rank retrieved results.

The paper [40] proposes an ontology based retrieval model meant for the exploitation of full-fledged domain ontologies and knowledge bases, to support semantic search in document repositories. The search system takes advantage of both detailed instance level knowledge available in the KB and topic taxonomies for classification. The retrieval model is based on an adaptation of the classic vector space model, including an annotation weighting algorithm and a ranking algorithm. Semantic search is combined with keyword based search to achieve tolerance to KB incompleteness where the keyword based index is replaced by a semantic knowledge base.

### 2.4 Semantic Meta Data

Metadata is structured information that describes, explains, locates or otherwise makes it easier to retrieve, use or manage an information resource. Metadata is often called data about data or information about information. Metadata can describe resources at any level of aggregation. Describing a resource with metadata allows it to be understood by both humans and machines in ways that promote interoperability.

It can describe a collection, a single resource or a component part of a larger resource (for example, a photograph in an article). Just as catalogers make decisions about whether a catalog record should
be created for a whole set of volumes or for each particular volume in the set, so the metadata creator makes similar decisions. For example, a metadata record could describe a report, a particular edition of the report or a specific copy of that edition of the report. Metadata can be embedded in a digital object or it can be stored separately. Metadata is often embedded in HTML documents and in the headers of image files. Storing metadata with the object it describes ensures the metadata will not be lost, obviates problems of linking between data and metadata, and helps ensure that the metadata and object will be updated together.

2.4.1 What Does Metadata Do?

An important reason for creating descriptive metadata is to facilitate discovery of relevant information. In addition to resource discovery, metadata can help organize electronic resources, facilitate interoperability and legacy resource integration, provide digital identification and support archiving and preservation.

2.4.2 Resource Discovery

Metadata serves the same functions in resource discovery as good cataloging does by:

- Allowing resources to be found by relevant criteria.
- Identifying resources.
- Bringing similar resources together.
- Distinguishing dissimilar resources.
2.4.3 Organizing Electronic Resources

As the number of Web based resources grows exponentially, aggregate sites or portals are increasingly useful in organizing links to resources based on audience or topic. Such lists can be built as static web pages, with the names and locations of the resources “hardcoded” in the HTML. However, it is more efficient and increasingly more common to build these pages dynamically from metadata stored in databases.

2.4.4 Interoperability

Describing a resource with metadata allows it to be understood by both humans and machines in ways that promote interoperability. Interoperability is the ability of multiple systems with different hardware and software platforms, data structures and interfaces to exchange data with minimal loss of content and functionality. Using defined metadata schemes, shared transfer protocols and crosswalks between schemes, resources across the network can be searched more seamlessly.

Two approaches to interoperability are cross system search and metadata harvesting. The Z39.50 protocol is commonly used for cross system search. Z39.50 implementers do not share metadata but map their own search capabilities to a common set of search attributes. A contrasting approach taken by the Open Archives Initiative is for all
data providers to translate their native metadata to a common core set of elements and expose this for harvesting. A search service provider then gathers the metadata into a consistent central index to allow cross repository searching regardless of the metadata formats used by participating repositories.

### 2.4.5 Digital Identification

Most metadata schemes include elements such as standard numbers to uniquely identify the work or object to which the metadata refers. The location of a digital object may also be given using a file name, URL (Uniform Resource Locator) or some more persistent identifier such as a PURL (Persistent URL) or DOI (Digital Object Identifier). Persistent identifiers are preferred because object locations often change, making the standard URL (and therefore the metadata record) invalid. In addition to the actual elements that point to the object, the metadata can be combined to act as a set of identifying data, differentiating one object from another for validation purposes.

### 2.4.6 Archiving and Preservation

Most current metadata efforts center on the discovery of recently created resources. However, there is a growing concern that digital resources will not survive in usable form into the future. Digital information is fragile, it can be corrupted or altered, intentionally or unintentionally. It may become unusable as storage media and hardware and software technologies change. Format migration and
perhaps emulation of current hardware and software behavior in future hardware and software platforms are strategies for overcoming these challenges.

Metadata is key to ensuring that resources will survive and continue to be accessible into the future. Archiving and preservation require special elements to track the lineage of a digital object (where it came from and how it has changed over time), to detail its physical characteristics, and to document its behavior in order to emulate it on future technologies.

Semantic metadata refers to the metadata that are formally modeled based on their context, thus giving them meaning. The way that current service oriented infrastructure handles and manages services metadata is not adequate and effective for metadata to help services discovery and knowledge sharing. There are no problems for humans to understand XML based metadata because we know the meaning of these English words, the question is: “can machines understand and consume them?”, so that they can perform automatic processing with regards to the use of Web/Grid services. Clearly without further assumptions, the answer will be no. The Semantic Web/Grid are extensions of the current Web/Grid in which information and services are given well defined meaning, better enabling computers and people to work in cooperation. We believe that the first step towards the Semantic Web/Grid is to make the Web/Grid full of rich SMD, in other words, metadata with semantics.
2.5 Conclusion

The internet has evolved rapidly over the last few decades. Effective extraction of query relevant information present within documents on the web is a nontrivial task. Despite its explosive growth over the last decade, the Web remains essentially a tool to allow humans to access information. The movement from Web 1.0 to Web 3.0 shows the drastic changes, the internet has undergone to make life simpler and easier for man. The next generation Web, will extend the Web’s capability through the increased availability machine processable information. These machine processable descriptions of Web information resources are called metadata and are associated with ontologies or conceptualizations of the domain of application. Metadata and associated ontologies then allow more intelligent software systems to be written, automating the analysis and exploitation of Web based information.

The way that current service oriented infrastructure handles and manages services metadata is not adequate and effective for metadata to help services discovery and knowledge sharing.

As seen from the efforts presented above, there were efforts to statistically build the ontology [41]. Unfortunately there exist no predefined techniques to build a rich knowledge representation of a domain capturing the real world. Thus there is a need to build domain ontology.