Revista Facultad de Ingeniería Universidad de Antioquia
ISSN: 0120-6230
revista.ingenieria@udea.edu.co
Universidad de Antioquia
Colombia

Suárez Barón, Marco; Salinas Valencia, Kathleen
An approach to semantic indexing and information retrieval
Revista Facultad de Ingeniería Universidad de Antioquia, núm. 48, junio, 2009, pp. 174-187
Universidad de Antioquia
Medellín, Colombia

An approach to semantic indexing and information retrieval

Una aproximación a la indexación semántica y a la recuperación de información

Marco Suárez Barón[*], *Kathleen Salinas Valencia*

Universidad Pedagógica y Tecnológica de Colombia, Apartado Aéreo 1094, Tunja, Colombia

(Recibido el 9 de julio de 2008. Aceptado el 12 de marzo de 2009)

Abstract

This paper presents an approach to the semantic indexes contained in particular topics as well as the most important models for Information Retrieval. They will be dynamically constructed using the annotations contained in the web resources and the definitions in ontologies of the annotation terms used. The indexes are envisioned as active agents that 'know' what topics they can handle (i.e. find content for) based upon their own ontologies.

----- *Keywords:* Information retrieval, ontologies, P2P, semantic, semantic indexing, taxonomy

Resumen

Este artículo presenta un acercamiento a los índices semánticos de tópicos particulares así como los más importantes modelos para la recuperación de información. Estos modelos serán construidos dinámicamente a través de anotaciones identificadas en recursos de la *web* y en las definiciones de anotaciones ontológicas de los términos utilizados. Los índices están proyectados a sus agentes activos que 'conocen' qué tópicos pueden manejar (e.j; encontrar el contenido para…) con base en sus propias ontologías.

----- *Palabras clave:* Recuperación de información, ontologías, P2P, semántica, indexación semántica, taxonomía

[*] Autor de correspondencia: teléfono móvil: 314 416 72 47, correo electrónico: marco.suarez@uptc.edu.co (M. Suárez).

Introduction

The aim of this paper is to propose an overwiew of different techniques about information retrieval and semantic indexing. It is designed to overcome a fundamental problem that plagues existing retrieval techniques that try to match words of queries with words of documents. The problem is that users want to retrieve on the basis of conceptual content, and individual words provide unreliable evidence about the conceptual topic or meaning of a document [1]. The discussions related to, and developments of, Semantic Web technologies indicate that there is an increasing interest in semantic description and structuring of content, the key applications of the technologies are in the areas information retrieval (e.g. Semantic Web), document abstraction, topic detection, and automatic classification [2,3]. There are usually many ways to express a given concept, so the literal terms in a user's query may not match those of a relevant document. In addition, most words have multiple meanings, so terms in a user's query will literally match terms in documents that are not of interest to the user [4]. The intended general architecture of the semantic indexes and semantic routers are queries received by an index for topics that are not handled by that index would then be semantically routed to a 'neighbour' index. In this context the closeness of a 'neighbour' is a function of the semantic distance between the topics covered by a pair of indexes. Determination of the semantic distance between indexes will involve comparison of the terms defined in their ontologies, and so will need to address the issue of semantic heterogeneity between different ontologies. The comparison of these ontology terms will utilise techniques for evaluating semantic similarity to identify the index that is best able to answer the query. Index agents might use specific techniques in order to come to an agreement on the semantics of a term. In order to operate within an open and dynamic environment such as the indexes need to exist in a distributed, decentralised and scalable architecture.

Information retrieval and semantic indexing

Information retrieval applications concerning textual documents use automatically generated free text index terms (post-coordinated), which are weighted by the statistical frequency of terms in documents and collections[5]. On the other hand, distinguishing features of a semantic index are that semantic relationships exist between controlled index terms, usually (but not necessarily) the result of manual cataloguing[5,6]. Semantically indexed hypermedia links are, by definition, computed corresponding to Intensional-Retrieval links; this allows the possibility of flexible query-based navigation tools.

Information retrieval

The subfield of computer science that deals with the automated storage and retrieval of documents is called information retrieval (IR), It has changed considerably in the last years with the expansion of the Web (World Wide Web) and the advent of modern and inexpensive graphical user interfaces and mass storage devices[6]. Automated IR systems were originally developed to help manage the huge scientific literature that has developed since the 1940's, and this is still the commonest use of IR systems. IR systems are in widespread use in university, corporate, and public libraries. IR techniques have also been found useful, however, in such disparate areas as office automation and software engineering [4]. For many authors the purpose of an information retrieval (IR) system is to process files of records and requests for information, and identify and retrieve from the files certain records in response to the information requests[7]. The retrieval of particular records depends on the similarity between records and the queries, which in turn is measured by comparing the values of certain attributes to records and information requests.

In order to be able to compare the similarity between the records (resources) and the queries, both need to be represented in a compatible way. Compatible representation makes it possible to automate the process of calculating the relevance between queries and resources. The term *indexing* has been widely used to refer to the process of building such representations. Indexing techniques have been developed in order to make possible the identification of the information content of documents (be they text documents, hypermedia or multimedia ones) [8]. In general, indexes permit the representation of knowledge about a domain in order to facilitate access to information. It simply means pointing to or indicating the content, meaning, purpose and features of messages, texts and documents [9]. Traditional indexing is based on the assignment of semantic labels or more formal typing to authored links. Typically the indexing of a textual document is obtained through the identification of a set of terms or keywords which characterise the document content that is terms which describe the topics dealt with in the document[8]. The terms included in this set have not only to be representative of the topics covered in the documents, but they also need to be distinguishing, in that they should make it possible to discriminate one document against the other documents in the collection covering the same or similar topics.

Indexing systems can be categorised along three dimensions: [4,10].

- Index terms are automatically derived or manually assigned.
- Index terms belong to a controlled vocabulary or are uncontrolled.
- Terms can be combined as ordered strings representing a single concept when indexing (pre coordinated terms), e.g. “Association of Computing Machinery”, or must be post-coordinated on retrieval.

However, post-coordination might allow the possibility to return items that have no connection with the different terms in the string (false-positives). If the resources are hypertexts, however, indexes are determined by explicitly authored links, in addition to each information item indexed with descriptor terms (more than one term might be required) [11]. Index and document spaces might be separated in hypertext, as different conformations of those spaces permit different possibilities in automated reasoning. We will now concentrate on the first type of indexing in the list above; since it is the most commonly used and can allow for all types of resources. Indexing can be either manual or automatic. The former is based on human analysis whereas the latter depends on the use of some type of algorithm, typically machine learning. A study by [12] has compared the two approaches, taking into consideration the different aspects of indexing and concluded that there is no real motivation to prefer one approach over the other. However, there are some considerations concerning the domain, the type of documents to index and on the number of documents available. Manual indexing can be rather expensive to perform and it might become difficult to perform with very large collections of documents such as those stored in digital libraries. Furthermore, it is not sufficiently flexible to support different indexing strategies. On the other hand, automatic indexing is less expensive to perform and can easily support different indexing strategies, but it might result in less precise indexing, since it is based on some mathematical or statistical formulations and not on a real understanding of the semantics of the terms used for the indexing. A solution might be the approach proposed by [12], who propose to generally apply automatic indexing, and to reserve manual indexing for important documents, where the importance is evaluated by some rules of thumb, such as use, citation, etc. Another disadvantage of manual indexing that can be foreseen is that it cannot be performed dynamically, that is, each time a new resource is discovered, it needs to be indexed before it can be included in the collection used for retrieving information[13]. For this reason, in this review We will concentrate mainly on automatic indexing. Automatic methods have little or no-input from users, who might be requested for relevance feedback. In the following sections, we will review the literature relevant to the most important techniques on automatic indexing and retrieval.

Simple indexing techniques

The simplest automatic indexing technique is based on providing a count of every occurrence of a word (or term) in a document [14]. This raises the issue of what is to be considered a word for the purposes of indexing a document. Usually a word is defined as one or more characters separated by spaces or punctuation (at least for English and other western languages). However, such a definition is not sufficient, in that it does not explain how to treat punctuation marks, and in particular when a punctuation mark is to be considered part of a word and when it has to be considered as a word delimiter. The most problematic punctuation marks are hyphens and slashes, as they can be used to create a word out of two words, e.g. meta-property. There are various approaches concerning how to deal with these situations: some indexing algorithms treat the hyphen as a space (delimiting two words), some just ignore the hyphen and thus account for only one word, and others consider all the possible combinations (so, for instance, "on line", "online", and "on-line" would be counted as 3 occurrences of the same word). Other problems are related to considering numbers and single characters (such as the English pronoun "I") as words. So, a simple indexing algorithm consists of the principles for determining which sets of characters are words, the means to count their occurrences, which is the matching strategy, and the output display. The matching strategy is free-text, full text indexing, where every occurrence of any combination of characters, including the insignificant ones (also known as *stop words*) such as "the" or "and", are considered [15]. The output can be displayed to users in special formats (known as *permuted*), that is keyword in context, keyword out of context, keyword alongside of context, etc. More sophisticated indexing algorithms can include the removal of stop words, which can account for a significant proportion of text in the document [15]. Common stop words to be removed are articles, prepositions and conjunctions. However, some systems, such as the MEDical Literature Analysis and Retrieval System of the U.S. National Library of Medicine, have just a few stop words.

Another technique that is widely used to remove words which are not significant in a document is that of *stemming*. In fact, documents often present sets of related words, which have the same root, but perform different functions in a sentence or have slightly different meanings, for example "drink", "drinks", and "drinkable". The purpose of stemming is to merge many different words into a single form. The simple stemming removes the final "*s*" from plurals in English; however, more complex forms of stemming could be performed. The techniques described so far are quite rough, in that accounting for term frequency in a document does not help in distinguishing between documents belonging to the same collections. More refined automatic indexing techniques have been developed that take into account the relative importance of the terms.

Ranking the documents in terms of relevance

The weight associated with an index term indicates the relevance of the term for representing a certain domain and its ability to distinguish among documents. Weights are generated according to the following principles [16]:

- Terms that occur in few documents are more useful than those appearing in a large number of documents.
- The more a term occurs in a document the more it is likely that the term is relevant for that document.
- A term that occurs the same number of times in a short document and in a long document is probably a more relevant term for the short document.

These principles are reflected in the following approaches to compute the weight of a term in a document, which are the *vector-space model* and the *probabilistic model*.

Vector-Space model

The *vector-space model* is based on the representation of both documents and queries as weighted vectors in the space of the index terms, whose dimensionality is determined by the size of the vocabulary used in the indexing process. A *similarity measure* is used to cluster together documents which show the higher degree of similarity. A vector-based information retrieval method represents both documents and queries with high-dimensional vectors, while computing their similarities by the vector inner product. When the vectors are normalized to unit lengths, the inner product measures the cosine of the angle between the two vectors in the vector space [17]. Once the terms have been associated with weights, documents can be represented by *term vectors* that take the form: *Vd=(v1, v2, …., vn)* where the elements *vi* corresponds to the weight of the term *i* and *d* refers to the document.

Probabilistic model

A probabilistic model is a novel approach to automated document indexing which is based on a statistical latent class model for factor analysis of count data [16].

Fitted from a training corpus of text documents by a generalization of the Expectation Maximization algorithm, the utilized model is able to deal with domain-specific synonymy as well as with polysemous words [18]. In *probabilistic models* documents in a collection are ranked according to their estimated probability of relevance to a query. The probabilities are estimated on the frequency of query terms in each document, and they make use of user determined *relevance judgements*, which play the role of training data [19]. A probabilistic model is based on the following notions : Term frequency: the frequency of a term *ti* in a document *dj* is *Tfi,j*, that is, the number of occurrences of *ti* in *dj*. Length of a document dj: the total number of terms occurring in a document *dj* is denoted by *DLj*.

Similarity and dissimilarity measures and clustering methods

The terms and the weights associated with them are then used rank the documents according to their relevance, thus enabling classification of the documents in the correct categories so that they can be retrieved [3]. All indexing is based on clustering documents together on the grounds of their similarity in characterising features. Automatic clustering techniques could be used to attempt to compute degrees of associations, either among terms or among documents. The former is known as term clustering, while the latter is known as document clustering [20]. Document clustering could be used to organise documents or to present them to users in an effective manner. Various clustering techniques have been developed during the years. Clustering is based on the hypothesis that closely associated documents tend to belong to the same cluster and to be relevant to the same type of requests. In clustering methods the objects to be clustered are described in terms of characterising features (*attributes*) and clusters are built by grouping together objects that have similar attributes. In document clustering the role of attributes is usually played by keywords and their weights, which are obtained from the indexing process [13]. Other features, such as citations, could be used to evaluate the degree of similarity among documents.

The greater *Sim(X, Y)* is, the more similar the two vectors, and therefore the two documents, are. This permits us to assign a new document to a category by comparing it to a pre-categorised document [21].

Latent s Semantic Indexing (LSI)

In more recent years a sophisticated technique for automatic indexing has been proposed. This technique, known as Latent Semantic Indexing (LSI), this topic is an advanced information retrieval (IR) technology that was developed by research scientists at Telcordia Technologies over ten years ago [22]. LSI is a variant of the vector retrieval method or SVD[12], that exploits dependencies or “semantic similarity” between terms. It is assumed that there exists some underlying or “latent” structure in the pattern of word usage across documents, and that this structure can be discovered statistically [10]. One significant benefit of this approach is that, once a suitable reduced vector space is computed for a collection of documents, a query can retrieve documents similar in meaning or concepts [23]. The other side LSI is a well-known information retrieval algorithm; this subject has been applied to a wide variety of learning tasks, such as search and retrieval, classication and filltering [4]. LSI is a vector space approach for modeling documents, and many have claimed that the technique brings out the `latent’ semantics in a collection of documents [24].

LSI became famous as one of the first IR techniques exhibiting effectiveness in dealing with the problems of synonymy and polysemy [4]. The basic idea of LSI is that if two document vectors represent the same topic, they will share many associated words with a keyword and they will have very close semantic structures after dimension reduction via truncated SVD [25].

Several studies have reported that Latent Semantic Indexing (LSI) based on truncated Singular Value Decomposition (SVD) could be compared favorably with other Information Retrieval (IR) techniques in terms of retrieval accuracy [26]. Many researchers in LSI have devoted a lot of time to testing the effectiveness of SVD in solving the problems of synonymy and polysemy. Empirical results show a general increase in the retrieval quality, but to the best of our knowledge, such algorithms come with no guarantee regarding the quality of the approximation produced [4]. Most of the rank reductions achieved via truncated SVD concern some properly chosen characteristic matrices. This fact led to a common practice where the computation of SVD is first carried out before rank reduction is accomplished [1]. In a recent project sponsored by the National Institute of Standards and Technology, LSI was compared with a large number of other research prototypes and commercial retrieval schemes. Direct quantitative comparisons among the many systems were some what muddied by the use of varying amounts of preprocessing-things like getting rid of typographical errors, identifying proper nouns as special, differences in stop lists, and the amount of tuning that systems were given before the final test runs. Nevertheless, the results appeared to be quite similar to earlier ones. Compared to the standard SVD method (essentially LSI without dimension reductions) LSI was a 16% improvement [1,10]. LSI has also been used successfully to match reviewers with papers to be reviewed based on samples of the reviewers own papers , and to select papers for researchers to read based on other papers they have liked [27].

Semantic indexing

The goal of semantic indexing is to use semantic information (within the objects being indexed) to improve the quality of information retrieval. Compared to traditional indexing methods, based on keyword matching, the use of semantic indexes means that objects are indexed by the concepts they contain rather than just the terms used to represent them.

The semantic index approach employs a set of semantic relationships between index terms, determined by means of thesauri such as the Medical Subject Headings [28]. Classification systems, such as Dewey Decimal or Library of Congress, focus on hierarchical relationships.

Both classification systems and thesauri determine the controlled vocabularies that are used in standard cataloguing practice in libraries and are now also applied to digital hypertexts (by means of thematic keywords in metadata descriptors of resources) [29]. An example is given by the Dublin Core standard metadata set which includes elements describing document characteristics such as “Title”, or “Date”, “Format”, etc. in addition to notions concerning the *content* of the document, such as the “Topic”. The “Topic” element usually refers to a controlled vocabulary, and semantic relations in thesauri can be used to determine the links among concepts in the topic domain. The main relationships used are:

- Equivalence - synonym terms.
- Hierarchical - broader and narrower terms.
- Associative - more loosely related terms.

The document collection is navigated by querying the semantic index space, rather than following explicit links. The queries can be simple or complex. Furthermore, a semantic index space can be seen as an organised set of browsable concept descriptors, where users can browse the index space.

In this way a user can focus on specific items of interest, or, conversely, consider more general items. Additionally, when index terms are combined, users have the ability to browse around each term, broadening and narrowing the specificity of description and seeing the effect on likely 'hits'. If a user enters a set of query terms instead of simply browsing the index space, then synonyms are also considered for retrieval purposes (by means of equivalence relationships between the terms), with no need for the user to specify the exact term employed for indexing. As a simple example, we might consider the ACM Computing Classification [9] used in the ACM Digital Library pages, where explicit hypertext links can be navigated.

The inclusion of semantic information in the index space provides the opportunity for knowledge-based hypermedia systems that provide intelligent navigation support and retrieval, with the system taking a more active role in the navigation process than relying on manual browsing alone. For example, rules governing permitted combinations of terms can filter a user's possible navigation options.

The notion of a semantic index key can be extended to the use of arbitrarily structured concepts such as those found in description logics representations. In such an index there are two properties of description logic based systems that play an important role: the ability to handle any degree of partial information in conjunction with an open world assumption and the ability to describe objects using complex concepts and to use these descriptions for query answering [30].

By utilising potentially complex concepts, which might be linked by subsumption and disjointness to index objects, this approach has the following characteristics:

- A semantic index is inherently multidimensional, since any combination of properties cast into a DL concept can serve as an indexing element.
- As a structured concept the indexing elements are not just attribute values, but can be based on complex descriptions of related objects.
- A semantic index as a whole is highly adaptable to patterns of usage. Indexing concepts can be added or removed at will, making it very dense and precise with respect to interesting sets of individuals, or very sparse in other less interesting areas.
- Since the index is actually a set of partial descriptions for the indexed objects, lots of information can be drawn from the index alone without accessing individual descriptions at all.

These kinds of index improve retrieval efficiency in large and heterogeneous collections of documents.

Semantic similarity and heterogeneity

From the peer-to-peer area emerges that a key capability of routing agents (or indexes) is identifying their semantic neighbour that is the index which deals with concepts that are semantically similar to those dealt with by the routing agent. Assessing semantic similarity among concepts is not a trivial task, which can be made more complex by the fact that the resources might be *heterogeneous*. Indeed, differences often occur between independently developed knowledge resources and their underlying ontologies even when they regard the same (or similar) domains of knowledge [31].

Heterogeneity Affecting Resources

When dealing with heterogeneous knowledge resources, one key issue is understanding what forms of heterogeneity exist between the knowledge sources and what are the mismatches they can cause. The vast amount of literature on the integration of heterogeneous information sources is sometimes confusing regarding the kinds of heterogeneity and the mismatches that can arise, especially where the knowledge engineering and data modelling fields meet. This makes it less easy to compare the different approaches. An attempt to reconcile and compare the different definitions presented in the literature and to find commonalities is given by [32] and [33]. We used these works and those by [14,34,35] as a starting point for reviewing the different types of heterogeneity that might affect resources. It has to be pointed out that not all the types of heterogeneity will be relevant in the context of this paper, but they will be presented for a matter of completeness. The importance of dealing with heterogeneity is that it causes mismatches that need at least to be taken into account [35], if not reconciled, when knowledge needs to be aggregated in order to obtain added value. We can broadly distinguish between mismatches caused by non-semantic and semantic heterogeneity [34]. The former type of heterogeneity is also known as syntactic or language heterogeneity in [32], while the latter is also called ontology heterogeneity by [14] and [34]. Syntactic heterogeneity denotes the differences in the language primitives that are used to specify ontologies, while semantic heterogeneity denotes differences in the way the domain is conceptualised and modelled.

Syntactic Heterogeneity

Syntactic heterogeneity occurs when resources and their underlying ontologies that are written in different ontology languages are combined. Four types of mismatch due to language heterogeneity are recognised [17]:

- Syntax: Different ontology languages are often characterised by different syntaxes. Differences in the language syntax give rise to mismatches that can be resolved by means of rewrite rules.
- Logical representations: This kind of mismatch is caused by differences in the representation of logical notions, and more precisely, differences in the language constructs that are used to express something.
- Semantics of primitives: This is, to a certain extent, a more subtle kind of mismatch deriving from non-semantic heterogeneity. Indeed, it is caused by differences in the semantics of the language statements. These differences can be sometimes quite difficult to detect, since two languages can use constructs with the same name, but slightly different interpretations, or sometimes the same interpretation might be associated with constructs with different names.
- Language expressivity: Mismatches due to differences in the expressivity between two languages are those which have the most impact on the problem of integrating/merging ontologies. Differences in the expressive power of the languages imply that one language can express something that the other language cannot express. For example, some languages support negation while others do not.

We have listed here the four types of syntactic heterogeneity; however, we should point out that mismatches due to syntactic heterogeneity can be overcome by wrapping the resources and by providing the means to translate ontologies into different ontology languages in an automatic fashion. Facilities of this kind are offered by various ontology editors such as WebOde or OWL, which permits the editing of ontologies in a language independent representation and their automatic translation at a later stage.

Semantic Heterogeneity

Mismatches caused by *semantic heterogeneity* occur when different ontological assumptions are made about the same domain. This kind of mismatch also becomes evident when combining ontologies which describe domains that partially overlap.

In particular, mismatches due to ontology heterogeneity can occur while *conceptualising* and/or *explicating* the domain. [14] and colleagues use these terms to refer to the definition of ontology given: *"An ontology is the explicit specification of a conceptualisation*" [14]. That is, the process of designing the ontology is comprised of two main stages, the *conceptualisation* of the domain and the subsequent *explication* of this conceptualisation, and the idea is that ontology heterogeneity can be introduced in both stages of the design.

Mismatches due to ontology heterogeneity can, therefore, be subdivided into *conceptualisation* and *explication mismatches*. Conceptualisation mismatches are semantic differences arising from different conceptualisations of the concepts and the relations between them in the ontology domain. Conceptualisation mismatches can be caused by the following types of heterogeneity:

- Model coverage and granularity: This type of ontology heterogeneity occurs when different conceptualisations, and thus different ontologies, model the same part of domain differently both with respect to model coverage and granularity.
- Scope: This mismatch occurs when two concepts or relations in the ontologies seem to be the same but their extensions (that is the set of their instances) are not the same although they are not disjoint. Relations mismatches also include mismatches concerning the assignment of attributes to concepts, since those represent relations between conceptual entities.

Explication mismatches arise because of differences in the specification of the domain conceptualisation. During the conceptualisation phase the concepts describing the domain are selected. In the explication phase these concepts are made explicit, usually by labelling each of them with a *term* (which is one or more words in natural language) and associating a *definition* with each term, which could be expressed in natural language or in a formal ontology language [36]. We distinguish six types of mismatches, in which the first three concern the modelling choices, the following two concern the choice of terms that are used to label a concept in the ontology [16], whilst the last type of mismatch concerns the way in which concepts are encoded:

- Representation paradigm: This type of mismatch depends on different representation paradigms used to model the same domain. It can become apparent with concepts such as time, actions, plans, causality, etc.
- Top-level concepts: Top-level concept mismatches arise because ontologies differ in the top-level ontologies they refer to.
- Modelling conventions (Also known as *concept description*): Modelling convention mismatches depend on modelling decisions made while designing the ontology. For instance, it is often the case that an ontology designer has to decide whether to model a certain distinction by introducing a separate class or by introducing a qualifying attribute relation.
- Synonym terms: This type of mismatch is called *term mismatch*. It occurs when the same concept, attribute, or relation is referred to by different terms and/or described by different definitions, which are semantically equivalent.
- Homonym terms: This type of mismatch occurs when a term can refer to different concepts depending on the context. It is mainly due to the existence of homonyms in natural language, such as the English word *wood,* which can mean a collection of trees or the material that forms the main substance of the trunk and branches of a tree. Homonym terms can appear in different ontologies *concerning the same domain* if these 'operationali-

se' the term in different ways. For example the concept "Year" might be described as *a period of time divided into 12 months* in two different ontologies, *O*1 and *O*2. If the first ontology considers a month as a period of time of 30 days, whereas the second ontology considers a month as a period of time that can have a number of days between 28 and 31, then the term "Year" in *O*1 is a homonym of the analogous term in *O*2.

- Encoding: This is maybe the easiest mismatch to resolve. It occurs when different ontologies encode values in different ways. Heterogeneity, and especially ontology heterogeneity, can seriously hinder attempts to share and reuse knowledge automatically. In fact, in order to recognise whether two concepts from heterogeneous knowledge source are similar, we cannot only rely on the terms denoting them and on their descriptions, and we need to have a full understanding of the concepts in order to decide whether they are semantically related or not.

Evaluating similarity among concepts

There is extensive literature on measuring similarity in general and on word similarity in particular. Tversky's work is based upon a psychological view of similarity, where similarity is treated as a property characterized by human perception and intuition.

Several similarity measures or semantic distance functions have been developed in Artificial Intelligence [37]. Many of these have been provided to evaluate similarity between simple objects, in which the objects are represented as vectors of attribute values and similarity measures are defined in terms of those vectors. More recently, there has been some work, such as in [36] that deals with similarity between complex objects. However, these measures can only account for structural similarity, but can say very little on the similarity in meaning, that is similarity between concepts rather than objects.

Semantic similarity is a form of semantic relatedness using network representation [8,31], a problem that has received much attention in the artificial intelligence field suggest that similarity in semantic networks can be assessed solely on the basis of the *IS-A taxonomy*, without considering other types of links. One of the easiest way to evaluate semantic similarity in taxonomies is to measure the distance between the nodes corresponding to the items being compared, that is the shorter the path between the nodes, the more similar they are. In [36] this idea is the basis of some definitions of dissimilarities defined for cluster analysis, namely ultrametrics, tree distances and strong Robinsonian dissimilarities. More precisely, an ultrametric dissimilarity fulfils the ultrametric inequality: d(a,b) ¡Ü max{d(a,c),d(c,b)}.

Where *a, b* and *c* are nodes of the taxonomy in our case. It can be shown that an ultrametric results from a hierarchical classification (dendrogram) of individuals, and conversely by putting *d (a, b)* = "the lowest level at which the objects 'a and b' meet in the dendrogram". This *bijection theorem* provides a unique characterisation of hierarchical classifications by ultrametrics. Alternatively, a tree distance characterises an additive tree, i.e. a tree T with *n* vertices and n-1 weighted edges. The dissimilarity *d (a, b)* is then reproduced as the sum of the weights of all edges of the (unique) path connecting two given vertices *a* and *b* in T. Additive trees are widely used in the reconstruction of phylogenetic evolutions, and might be useful in the context of ontology clustering only if a weight is associated with each concept inheritance, in order to take into account the amount of inherited properties. Finally, Netchesian dissimilarities [31] characterise a pyramidal classification of individuals. The pyramidal model generalises hierarchies by allowing non-disjoint classes at each given level, therefore Robinsonian dissimilarities should be considered only in the case of overlapping concepts in ontologies.

The concepts in the top-level ontology provide the basis for evaluating semantic similarity between concepts; indeed, the idea is to calculate the paths as those connecting each concept to its nearest ancestor in the top-level ontology. There are two principal problems with similarity measurements based upon evaluation of an *IS-A* taxonomy [38]:

- It assumes that taxonomic links represent uniform distances, whereas in real taxonomies there is a wide variation in the 'distance' covered by a single link.
- It is based upon the assumption that the two concepts being compared have a common ancestor within the taxonomy.

The first problem has been addressed in a number of ways, in particular by the use of weighted path measures. The weighting calculation for each link can be based on many different factors, such as: the types of links present, the depth of a link in the taxonomy, and the density of concepts in the immediate neighbourhood of the link.

The second problem limits the applicability of this sort of measure to those concepts that have an ancestor that is common to both of them. Semantic similarity functions have been studied in the fields of information retrieval and data integration to compare concepts both within and between ontologies. Early approaches to computing semantic similarity operated between concepts in a single ontology, but more recent work enables comparison between concepts in different ontologies. 'A similarity function determines similar entity classes by using a matching process over synonym sets, semantic neighbourhoods and distinguishing features' [39]. Maedche and Staab have investigated ontology similarity measures based on a two layer view of ontologies [34]. This view separates the ontology into lexical and conceptual levels, both of which are utilised in the evaluation of similarity. Many existing systems utilise lexical matching and synonym sets using terminological taxonomies, such as WordNet [35], and semantic neighbourhoods to compute semantic similarity, for example, SymOntos (developed during the IST project *Harmonise*) and OBSERVER. Anchor-PROMPT assesses both lexical and semantic matches exploiting the content and structure of the source ontologies [19]. Chimaera partially considers the ontology structure in that it assesses similarity between concepts on the grounds of the subclass-superclass relationship and the attributes attached to the concept. Some approaches also use additional information encoded into the ontology concepts, such as the mereology (part-whole relations) or typical and distinguishing features of concepts. Other features of ontological structure and concept definitions that are utilised in calculations of semantic similarity include [40]:

- Path distance measures.
- Weighting of path measures based on concept density/depth in taxonomy.
- Intersection of concept instances.
- Information content.
- Typical and distinguishing features.

Meaning negotiation

Meaning negotiation is a relatively new field that provides a potential mechanism for evaluating the similarity of concepts. The mechanism is for agents that commit to the semantics of the two concepts to engage in a process of negotiation to determine to what extent the semantics of the two concepts overlap [41]. This is achieved by a sequence of communications between the agents, during which they gradually come to an agreement on the shared meaning of a concept. Some meaning negotiation techniques have been employed in ontology-based information systems to determine matches between concepts in different ontologies.

Meaning negotiation techniques have been applied in Natural Language Processing [42]. Using such techniques the concepts under negotiation are evaluated in terms of:

- The linguistic meaning of the term used to denote the concept. This is determined using the synonym sets defined in a terminological taxonomy such as WordNet [32].
- The contextualization of the term, which is computed by combining its linguistic me-

aning with the linguistic meaning of (some of) the other terms in its taxonomic neighbourhood [31,41].

One research area that may be relevant in terms of negotiating about the meaning of terms is that of *argumentation.* Argumentation refers to the process by which one agent tries to convince another of the truth (or falsity) of a state-of-affairs. The process involves agents putting forward arguments for and against propositions, together with justifications for the acceptability of these arguments. Argumentation is useful in those cases where the agents need to give reasons for believing a certain statement, therefore it could be used to compare and contrast concepts' definitions (where agents commit to the overlapping concepts and find an agreement on what differs). Semantic similarity techniques enable the parties involved to determine which of their concepts are similar to each other. Argumentation can then be applied for the agents to come to an agreement about the meaning (possibly in terms of definition) of shared concepts.

There has not been a great deal of research into argumentation in knowledge sharing, and argumentation techniques have not been applied to the process of determining an agreed upon meaning of concepts. The approach by Bailin and Truszkowski seems to make use of argumentation notions, though in a socially based setting rather than in Computer Science. In this work they consider the sorts of discussions that go on between humans, such as arranging a party. However, the argumentation notions used are not defined in any formal way. The process of argumentation offers good prospects for our purposes. It may be used to enable agents to come to agreements about similar concepts, which may themselves be determined using the various semantic similarity techniques. Such agreements would consist of a shared meaning that each of the agents agree to associate with their own local concept.

Conclusions

Based upon this approach in semantic indexing, routing techniques and semantic similarity techniques we can conclude that there is a wide-ranging body of research in each of these areas, much of which will bear further investigation in the context of Semantic Indexation web and management knowledge. In section 2 of this paper it has been reviewed the most used automatic indexing techniques. We have reviewed the classical indexing techniques which have been developed in information retrieval. These make use of statistical information, such as term frequency, to identify the topic treated in a document and cluster together documents concerning a same topic. However, these techniques are not sufficient to deal with large collection of heterogeneous documents, and more sophisticated techniques have been developed. Indexing of large collection of documents is obtained by determining keywords in the documents, but also by considering groups of equivalent keywords, or of terms related to the keywords found. Synonym relationship and other kind or relationships are determined by thesauri or ontologies describing the domain of interest. This type of indexing is also known as *semantic indexing*. Similarity measurement has been studied widely in many research areas, such as psychology, cognitive science and mathematics, as well as in many areas of computer science and AI. Approaches to semantic similarity can be drawn from many fields, such as distributed databases, information retrieval, data integration and natural language processing. Specific techniques for evaluating semantic similarity that might be applicable in this context are:

- Use of hierarchical taxonomic links and weighting of the values assigned to these links on the basis of the link's depth in the taxonomy and on the basis of the local concept density.
- Use of synonym sets to address the use of different terms to describe the same concept.
- Use of the lexical context of a concept term, including the evaluation of the semantic effect of terms in the same context upon the term under consideration.
- Overlap of concept instances.
- Typical and distinguishing features.

References

1. E. Voorhees." The cluster hypothesis revisited". *SIGIR*.1995. pp. 188-196.

2. S. Deerwester, S. T. Dumais, T. K. Landauer. "Indexing by latent semantic analysis". *Journal of the American Society of Information Science*. Vol. 41. 1990. pp.391-407.

3. S. Weng, C. Lin. "Using text classification and multiple concepts to answer e-mails". *Experts systems with applications*. Vol 26. 2004. pp. 529-543.

4. W. Frakes, R. Baeza-Yates. *Information Retrieval Data Structures and Algorithms*. Ed. Prentice Hall. New Jersey.1992. pp. 504.

5. T. Hofmann. "Latent class models for collaborative filtering". *Proceedings of the 16th International Joint Conference on Artificial Intelligence IJCAI.* Stockholm. Sweden. July 31 - August 6. 1999 pp. 10-20.

6. S. Deerwester, S. Dumais, R. Harshman. "Indexing by Latent Semantic Analysis". *Journal of the American Society for Information Science*. Vol. 41. 1990. pp. 391-407.

7. R. Baeza-Yates, C. Castillo, F. Saint-Jean: "Web Dynamics, Structure and Page Quality". In M. Levene and A. Poulovassilis (eds.) *Web Dynamics*. Ed. Springer, New York. 2004. pp. 93-109.

8. D. Bassu, Cl. Behrens. "Distributed LSI: Scalable Concept-based Information Retrieval with High Semantic Resolution". *Proceedings of the 3rd SIAM International Conference on Data Mining (Text Mining Workshop).* San Francisco. 2003. pp. 72-82.

9. URL http://www.acm.org/class/2000/. Consultada el 20 de agosto de 2005.

10. S. Dumais. "Latent Semantic Indexing (LSI) and TREC-2". D. Harman (ed*.). The Second Text Retrieval Conference (TREC2), NIST Special Publication 500-215*. Ed. Morgan Kaufmann Publishing Co. San Mateo, California. 1994. pp. 105-116.

11. G. Salton, M. J. McGuill. *Introduction to modern information retrieval*. Ed. McGraw Hill. New York. 1986. pp. 400.

12. J. D. Anderson, J. Pérez-Carballo. "The Nature of Indexing: how Humans and Machines Analyse Messages and Texts for Retrieval - Part I: Research, and the Nature of Human Indexing". *Information Processing and Management.* Vol. 37. 2001. pp. 231-254.

13. M. W. Berry, P. G. Young, "Using latent semantic indexing for multilingual information retrieval". *Computers and the Humanities Journal*. Vol.29. 1995. pp. 413-429.

14. M. Visser. "Semantic Indexing Based on Description Logics". *Proceedings of Knowledge Representation Meets Databases (KRMD).* Saarbruecken. Germany. 2004. pp. 456-463.

15. C. Tang, S. Dwarkadas, Z. Xu. "On scaling latent semantic indexing for large peer-to-peer systems". *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* 2004. pp. 112–121.

16. E. H. Han, G. Karpis. "Fast supervised dimensionality reduction algorithm with applications to document categorization and Retrieval". *Proceedings of the ACM CIKM.* 2000. pp. 12-19.

17. R. B Yates, B. R. Neto. *Modern Information Retrieval*. Ed. Pearson Education. 1999. pp.240.

18. T. Hofmann. "Probabilistic Latent Semantic Indexing". *Proceedings of the Twenty-Second Annual International SIGIR Conference on Research and Development in Information Retrieval.* Berkeley. California. 1999. pp. 140-160.

19. P. Bennet, S. T. Dummais, E. Horvitz. "Probabilistic combination of text classifiers using reliability indicators". *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* Tampere. Florida. 2002. pp. 207-214.

20. S. Yuan, C. Cheng. "Ontology-Based couple clustering for heterogeneous product recommendation in mobile marketing". *Experts systems with applications*. Vol 26. 2004. pp. 461-476.

21. D. Lin. "An information-theoric definition of similarity". *Proceedings of the 15th International Conference on Machine Learning.* San Francisco CA. 1998. pp. 296-304.

22. J. Nielsen, S. Dumais. "Automating the assignment of submitted manuscripts to viewers". *Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* Copenhagen, Denmark. 1992. pp. 233-244.

23. T. K. Landauer, P. W. Foltz, D. Laham. "An Introduction to Latent Semantic Analysis". *Discourse Processes*. Vol 25. 1998. pp. 259-284.

24. A. Kontostathis, W. M. Pottenger. "A Framework for Understanding Latent Semantic Indexing (LSI) Performance". *Information Processing and Management.* Vol. 42. 2006. pp. 56-73.

25. C. Aswani-Kumar, S. Srinivas. “An Information Retrieval Model Based on Latent Semantic Indexing with Intelligent Preprocessing”. *Journal of Information and Knowledge Management.* Vol 4. 2005. pp. 279-285.

26. C. Aswani-Kumar, S. Srinivas. ”Latent Semantic Indexing Using Eigenvalue Analysis for Efficient Information Retrieval”. *International Journal in AMSC.* Vol. 16. 2006. pp. 551-558.

27. J. Nielsen, S. T. Dumais. “Automating the assignment of submitted manuscripts to viewers”. N. Belkin, P. Ingwesen, & A. M. Pejtersen (Eds.) *Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* New. York. 2000. pp. 209-228.

28. E. Brill. “Some advances in rule-based part of speech tagging”, *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*. Seattle. WA. 1994. pp. 722–727.

29. H. H. Chen, C. J. Lin. “A multilingual news summarizer”. *Proceedings of 18th International Conference on Computational Linguistics.* Saarbrücken. 2000. pp. 159-165.

30. C. Wei, T. H. Cheng, Y. C. Pai. “Semantic enrichment in knowledge repositories: annotating reply semantic relationships between discussion documents”. *Journal of Database Management*. Vol 17. 2006. pp. 49-66.

31. R. Neches, R. E. Fikes. “Enabling Technology for Knowledge Sharing”. *AI Magazine.* Vol. 12. 2001. pp. 36-56.

32. G. Miller, N. Antonakis. “Nouns in WordNet: A Lexical Inheritance System”. *International Journal of Lexicography*. Vol. 3. 2000. pp. 245-264.

33. H. Kitakami, M. Arikawa. “An Intelligent System for Integrating Autonomous Nomenclature Databases in Semantic Heterogeneity”. R. R. Wagner, H. Thoma (Eds.), *Proceedings of Database and Expert System Applications (DEXA)*. Zurich. Switzerland. 1996. pp. 187-196.

34. A. Maedche, S. Staab. “Measuring Similarity between Ontologies”. *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management.* EKAW. Siguenza. Spain. 2002. pp.34-45.

35. J. Euzenat, P. Shvaivo. *Ontology Matching*. Ed. Springer. Heidelberg (DE). 2007. pp. 61-72.

36. W. Nejdl, A. Loser. “Super-Peer-Based Routing and Clustering Strategies for RDF-Based Peer-to-Peer Networks”. *Proceedings of WWW.* Budapest. Hungary. 2003. pp. 123-130.

37. A. Voutilainen. “A detector of english noun phrases”. *Proceedings of Workshop on Very Large Corpora.* Ohio State University. Columbus (OH). 1993. pp. 48-57.

38. S. Casteleyn, P. Plessers. “Generating semantic annotations during the web design process”, Proceedings *of the 6th international conference on Web engineering*. Menlo Park. California. 2006. pp. 91-92.

39. H. Han, L. Reeve, “Survey of semantic annotation platforms”. *Proceedings of the 2005 ACM symposium on Applied computing*. Santa Fe. New Mexico. 2005. pp. 1634-1638.

40. T. Hofmann. “Latent class models for collaborative filtering”. *In Proceedings of the 16th International Joint Conference on Artifcial Intelligence.* IJCAI. San Jose. CA. 2004. pp. 1234-1250.

41. D. Boley, M. Gini, R. Gross, E. Han, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, J. Moore, “Partitioning-based clustering for web document categorization”. *Journal of Decision Support Systems*. Vol 27. 1999. pp. 329-341.

42. D. Roussinov, H. Chen. “Information navigation on the web by clustering and summarizing query results”. *Journal of Information Processing & Management.* Vol 37. 2001. pp. 789-816.