UNIVERSIDAD CARLOS III DE MADRID

# TESIS DOCTORAL

# Information search and similarity based on Web 2.0 and semantic technologies

**Autor:**
**Damaris Fuentes Lorenzo**

**Directores:**
**Luis Sánchez Fernández**
**Norberto Fernández García**

DEPARTAMENTO DE INGENIERÍA TELEMÁTICA

Leganés, Abril, 2015

TESIS DOCTORAL

# Information search and similarity based on Web 2.0 and semantic technologies

Autor: Damaris Fuentes Lorenzo

Directores: Luis Sánchez Fernández, Norberto Fernández García

Tesis entregada a la Universidad Carlos III de Madrid, España.

Doctorado en Ingeniería Telemática

Leganés, Abril, 2015

# RESUMEN

Internet pone a disposición de la sociedad una enorme cantidad de información descrita en lenguaje natural. Los *buscadores web* nacieron de la necesidad de encontrar un fragmento de información entre tanto volumen de datos. Su facilidad de manejo y su utilidad los han convertido en herramientas de uso diario entre la población. Para realizar una consulta, el usuario sólo tiene que introducir varias palabras clave en lenguaje natural y el buscador responde con una lista de recursos que contienen dichas palabras, ordenados en base a *algoritmos de ranking*. Estos algoritmos usan dos tipos de factores básicos: factores dinámicos y estáticos. El *factor dinámico* tiene en cuenta la consulta en sí; es decir, aquellos documentos donde estén las palabras utilizadas para describir la consulta serán más relevantes para dicha consulta. La estructura de hiperenlaces en los documentos electrónicos es un ejemplo de *factor estático*. Por ejemplo, si muchos documentos enlazan a otro documento, éste último documento podrá ser más relevante que otros.

Si bien es cierto que actualmente hay consenso entre los buenos resultados de estos buscadores, todavía adolecen de ciertos problemas, destacando 1) la soledad en la que un usuario realiza una consulta; y 2) el modelo simple de recuperación, basado en ver si un documento contiene o no las palabras exactas usadas para describir la consulta.

Con respecto al primer problema, no hay duda de que navegar en busca de cierta información relevante es una práctica solitaria y que consume mucho tiempo. Hay miles de usuarios ahí fuera que repiten sin saberlo una misma consulta, y las decisiones que toman muchos de ellos, descartando la información irrelevante y quedándose con la que realmente es útil, podrían servir de guía para otros muchos.

Con respecto al segundo, el carácter textual de la Web actual hace que la capacidad de razonamiento en los buscadores se vea limitada, pues las consultas y los recursos están descritos en lenguaje natural que en ocasiones da origen a la ambigüedad. Los equipos informáticos no *comprenden* el texto que se incluye. Si se incorpora *semántica* al lenguaje, se incorpora *significado*, de forma que las consultas y los recursos electrónicos no son meros conjuntos de *términos*, sino una lista de *conceptos* claramente diferenciados.

La presente tesis desarrolla una capa semántica, *Itaca*, que dota de significado tanto a los recursos almacenados en la Web como a las consultas que pueden formular los usuarios para encontrar dichos recursos. Todo ello se consigue a través de *anotaciones colaborativas* y de *relevancia* realizadas por los propios usuarios, que describen tanto consultas como recursos electrónicos mediante conceptos extraídos de Wikipedia. Itaca extiende las características funcionales de los buscadores web actuales, aportando un nuevo modelo de ranking sin tener que prescindir de los modelos actualmente en uso. Los experimentos demuestran que aporta una mayor precisión en los resultados finales, manteniendo la simplicidad y usabilidad de los buscadores que se conocen hasta ahora. Su particular diseño, a modo de capa, hace que su incorporación a buscadores ya existentes sea posible y sencilla.

# ABSTRACT

The World Wide Web provides a huge amount of information described in natural language at the current society's disposal. *Web search engines* were born from the necessity of finding a particular piece of that information. Their ease of use and their utility have turned these engines into one of the most used web tools at a daily basis. To make a query, users just have to introduce a set of words - *keywords* - in natural language and the engine answers with a list of ordered resources which contain those words. The order is given by *ranking algorithms*. These algorithms use basically two types of features: dynamic and static factors. The *dynamic factor* has into account the query; that is, those documents which contain the keywords used to describe the query are more relevant for that query. The hyperlinks structure among documents is an example of a *static factor* of most current algorithms. For example, if most documents link to a particular document, this document may have more relevance than others because it is more popular.

Even though currently there is a wide consensus on the good results that the majority of web search engines provides, these tools still suffer from some limitations, basically 1) the loneliness of the searching activity itself; and 2) the simple recovery process, based mainly on offering the documents that contains the exact terms used to describe the query.

Considering the first problem, there is no doubt in the lonely and time-consuming process of searching relevant information in the World Wide Web. There are thousands of users out there that repeat previously executed queries, spending time in taking decisions of which documents are relevant or not; decisions that may have been taken previously and that may be do the job for similar or identical queries for other users.

Considering the second problem, the textual nature of the current Web makes the reasoning capability of web search engines quite restricted; queries and web resources are described in natural language that, in some cases, can lead to ambiguity or other semantic-related difficulties. Computers do not *know* text; however, if *semantics* is incorporated to the text, *meaning* and sense is incorporated too. This way, queries and web resources will not be mere sets of *terms*, but lists of well-defined *concepts*.

This thesis proposes a semantic layer, known as *Itaca*, which joins simplicity and effectiveness in order to endow with semantics both the resources stored in the World Wide Web and the queries used by users to find those resources. This is achieved through *collaborative annotations* and *relevance feedback* made by the users themselves, which describe both the queries and the web resources by means of Wikipedia concepts.

Itaca extends the functional capabilities of current web search engines, providing a new ranking algorithm without dispensing traditional ranking models. Experiments show that this new architecture offers more precision in the final results obtained, keeping the simplicity and usability of the web search engines existing so far. Its particular design as a layer makes feasible its inclusion to current engines in a simple way.

# TABLE OF CONTENTS

# LIST OF FIGURES

# List of Tables

# PART I. Introduction and State of the Art

# 1 INTRODUCTION

This chapter introduces the problems of current web search engines, which have motivated this thesis. I explain how existing techniques can solve those problems by adding semantic knowledge and relevance feedback in the existing information, and the reasons for the vocabulary selected to add these semantics. Specific goals are detailed and the planning tasks for the consecution of this dissertation are also listed.

## 1.1 MOTIVATION

Since its creation in 1989, the World Wide Web has become into one of the largest public information sources. In 2005, some reports pointed out that the indexable Web contained at least 11.5 billion pages (Gulli & Signorini, 2005); in 2009, the Web doubled the content to more than 25.21 billion pages (Worldwidewebsize.com, 2012).

Though the large amount of information available on the Web is one of its main positive aspects, it also has a negative side: the vast number of pages makes difficult for users to find the information they are looking for. Users need appropriate tools in order to take full advantage of the information stored, losing as less time as possible (Bates & Anderson, 2002).

*Web search engines,* like Google[1] or Yahoo[2], were born from this necessity and are well known examples of this kind of tools. Their ease of use and their utility have turned these engines into one of the most used web tools at a daily basis. To make a query, users just have to introduce a set of words - *keywords* - in natural language and the engine answers with a list of ordered resources which contain those words. These engines comprise 1) a web robot or crawler, also known as *spider*, to find web pages; 2) an indexer, where content is analysed and stored appropriately for later queries; 3) the interface to execute the final queries; and 4) algorithms to order results.

**Fig. 1. Collage of web search engines, retrieved from http://seotermglossary.com**

Which content is displayed and in which order are crucial for the effectiveness perceived by users. The order is given by *ranking algorithms*. Some ranking algorithms are very famous, like that used by Google, called *PageRank* (Page, Brin, Motwani, & Winograd, 1999). This iterative algorithm ranks web pages based on the number of other web pages that link there.

Engines success also depends on their easiness of use. Most of current web search engines have a simple web form as their graphical user interface. To execute a query, users normally type one or several words, *keywords*; then, the engine examines its index and provides a listing of best-matching web pages according to its ranking algorithm.

These features have made engines to achieve positive results in the web market. However, current web search engines still have some limitations.

[1] Google site (Spanish version): www.google.es

[2] Yahoo site (Spanish version): www.yahoo.es

First, navigating in a search for relevant information on the Web is one of the most lonely and time-consuming tasks (Jung, 2005). The performance of the overall searching process can be enhanced if users collaborate somehow in this task. Current algorithms in web search engines make use of both *static* and *dynamic features* that are independent of the final users. The *static features* do not take into account the query executed. An example of static feature is the hyperlinks structure among online documents. If most documents link to a particular document, this document may have more relevance than others because it is more popular. This document may be presented at the top of the results returned by a web search engine, just in case this document matches with the executed query. For this, a *dynamic feature* is needed, because it is query-dependent. With a dynamic factor, those documents which contain the keywords used to describe the query are more relevant for that query.

Traditional models for ranking algorithms pay attention to either the query or to the criteria of web creators - and what hyperlinks they inserted in their web pages -. Final users are relegated to merely write the keywords of the queries. However, given a query, previous users' opinions about similar or identical queries could improve the results of these algorithms.

Second, the retrieval model of current search engines is mainly based on looking whether keywords in a user query match the content of web documents; that is, by comparing text strings with text strings. As the possible results of this matching process are tied to the natural language in which both queries and web contents are defined, web resources obtained may be limited. As pointed out in (Telang, 2013), web search engines search in a "dumb" way. Whatever advances are made by Google or Bing[3], they still remain dumb. This fact can have a negative impact on the precision of results obtained.

For instance, the search engine may omit other documents referred to the same information if these documents have not the same keywords of the query. If I search the word "*buy*", I probably do not recover documents with the word "*purchase*". Another case where the keyword-matching approach is problematic, is that of ambiguous queries; the shorter the queries, the smaller the context to disambiguate them. Taking into account that, according to (Experian Hitwise, 2011), the most frequent query lengths are 1 or 2 words, this problem can affect to a large number of queries. If these documents are invisible to the engine recovery process, then they are also invisible to final users.

> *This dissertation focuses on the solution of these problems and is developed within the context of the Web 2.0 and semantic techniques, in order to improve the effectiveness of current web search engines.*

## 1.2 SEMANTIC ANNOTATIONS

The great majority of web search models use natural language for users to describe queries because web resources are also described with natural language. Even though this is the easier way for those users, it can lead to ambiguity or other semantic-related

[3] Bing engine site: http://www.bing.com

difficulties. However, if *semantics* is incorporated to the text, both of queries and resources, *meaning* is incorporated too.

There are two basic types of procedures to add meaning or *metadata* to the current web:

- Implement programs that automatically extract the semantics of the web content.
- Enrich the web content with annotations, in a declarative way, giving as a result information with machine-readable semantics.

> *This dissertation takes into consideration the second procedure, semantic annotations, to associate lists of well-defined concepts to queries and web resources. The annotation task is managed by the final users themselves, in a collaborative process.*

## 1.3 Collaborative feedback

A *collaborative* or *cooperative* work can be defined as a set of intentional processes of a group to reach specific goals, together with software tools that support these activities. A collaborative task can maximize the results and minimise costs, in benefit of the group objectives.

Vannevar Bush (1945) predicted the new vision of computer technologies, including hypertext, the Web and, in short, knowledge management systems with online cooperation. As Bush foresaw, the Web is indeed undergoing significant change with regards to how people communicate. A shift in the web content, where consumers turned into “prosumers”, is making the Web a means of conversation, cooperation and mass empowerment.

The most important cooperative social techniques implied in this dissertation are *collaborative filtering* and *collaborative tagging*.

*Collaborative filtering* is the process by which users help one another to perform filtering by annotating their reactions to documents they read. For example, users can annotate whether they find a particular document interesting or not - see the “I like” button on Facebook -. Even though this technique has grown in popularity in the last decade with the so-called web 2.0, there already exist collaborative filtering works dated on 1994, like GroupLens (Resnick, Iacovou, Suchak, Bergstrom, & Riedl, 1994), a system for searching news articles, or on 1992 with Tapestry (Goldberg, Nichols, Oki, & Terry, 1992), an e-mail organizer system.

> *For the dissertation presented here, collaborative filtering will serve as the basis to generate opinions about what resources users consider relevant to what queries. These suggestions will serve to future users asking for similar or even identical queries.*

*Collaborative tagging* is the process by which many users add metadata in the form of keywords to organize their content. This metadata is also known as *annotations*. Some of the well-known applications that allow this technique are Delicious[4], where the tagged resources are website bookmarks, or Flickr[5], where the target resources are photographs. Collaborative tagging can be seen as a form of collaborative filtering; in this case, users' reactions are the tags they relate to the resources.

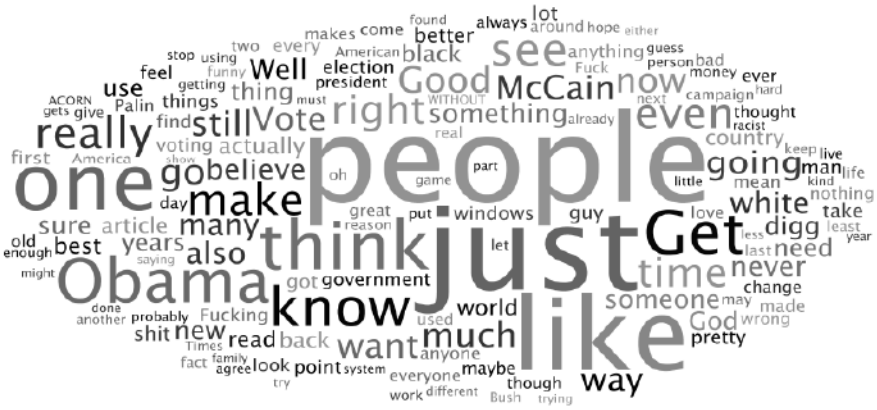


Fig. 2. An example of tag cloud, retrieved from http://www.outofthebrew.com

> *In the context of this dissertation, queries and web resources will be provided with semantics by adding annotations through collaborative tagging. One of the greatest benefits of social tagging applications is that there is not any predefined vocabulary for the tagging activity. First, this provides users with freedom to choose any keyword to use. Second, no expert knowledge is needed to define a domain vocabulary.*

One approach to collaborative filtering and tagging in a search engine consists on exploiting user queries' terms and activities, obtained from search engine logs.

### 1.3.1. Collaborative filtering: relevance

In the case of filtering, queries are used together with the links users click on the ranked results presented, in a process called *implicit feedback*. Queries and links selected are also called *click-through data*, and this information took relevance approximately one decade ago. In this area, (Hansen & Shriver, 2001) and (Joachims, 2002) are worth mentioning. The former proposed narrowing search results by observing the browsing patterns of users during search tasks. In the latter, Joachims used navigation data to improve the results in search engines by using classification techniques in conjunction with the click-through data of a meta-search engine.

Outcomes showed that the results obtained improved retrieval quality with respect to using the engine alone. However this approach makes assumptions that may have a negative impact in the obtained results. For example, the approach considers that the mere selection of a result implies this result is relevant to the query, which may not be true.

[4] Delicious site: www.delicious.com

[5] Flickr site: www.flickr.com

> *This dissertation considers a necessity to find the relevance of query results with explicit feedback.*

### 1.3.2. Collaborative tagging: annotation

In the case of tagging web resources, user queries can be considered as if they were textual tags. The terms used in a query can be considered as potential descriptions or tags of the URLs of the navigation data set obtained after a query execution. This is exactly the conclusion of several works, such as (Krause, Jäschke, Hotho, & Stumme, 2008), where it is demonstrated that the clicking behaviour of search engine users, based on the presented search results, and the tagging behaviour of social bookmarking users were driven by similar dynamics. Some of these works call the resulting network of keywords a *logsonomy*.

However, as explained in many studies like (Golder & Huberman, 2006; Motta & Specia, 2007; X. Wu, Zhang, & Yu, 2006), this apparent advantage leads to a number of weaknesses when using tags for information retrieval and search. Most of these problems can be grouped in the following sets:

- *Ambiguity*: When searching for documents with a word like “play”, related to a theatre piece, a search engine can return unrelated results such as, for example, a set of games for children.
- *Lack of synonym relations*: Words “irritated” and “annoyed” are very closely related; however, after searching for one of these words, found items will hardly contain the other word.
- *Lack of consensus*: To describe a particular item, different users may consider terms at different levels of generality/specificity. For example, a user can tag a photograph as “bird”, whereas another user can tag the same photo as “eagle”.

In (Heymann, Koutrika, & Garcia-Molina, 2008), authors demonstrated that *social tagging does not improve web search*.

The usage of formal annotation vocabularies, instead of plain text tags, may alleviate the aforementioned problems (Passant & Laublet, 2008). Ontologies are a type of formal vocabulary that can be used for this purpose. Appearing first in Philosophy, ontologies are grasped by Artificial Intelligence experts to represent needed parts of a particular domain (Gruber, 1993). Later on, the *Semantic Web* community started to make use of them. The basic principle of Semantic Web (Berners-Lee, Hendler, & Lassila, 2001; Shadbolt, Berners-Lee, & Hall, 2006) is that of adding further meaning to the current Web in such a way that the web content is not a set of simple data, but *knowledge*. The Semantic Web is not separated from the current Web; it is an extension where each piece of information is given a well-defined meaning. Having into account that metadata processing requires a controlled and well-defined vocabulary, Semantic Web acquired the ontology mechanism to represent, share and reuse the knowledge behind.

However, ontologies still lack of mass support, in contrast with the frequent use of tags in any Web 2.0 applications. The interaction between a user with no particular knowledge about semantics and a semantic web application is very limited; efforts to avoid this problem are still ongoing (Rico Almodóvar, 2012). The development of any ontology is still an activity addressed to knowledge experts - more if specifications from the Semantic Web technology stack have to be used, see Fig. 3 -, whereas users with no expertise can be involved in the creation of sets of tags with no effort.

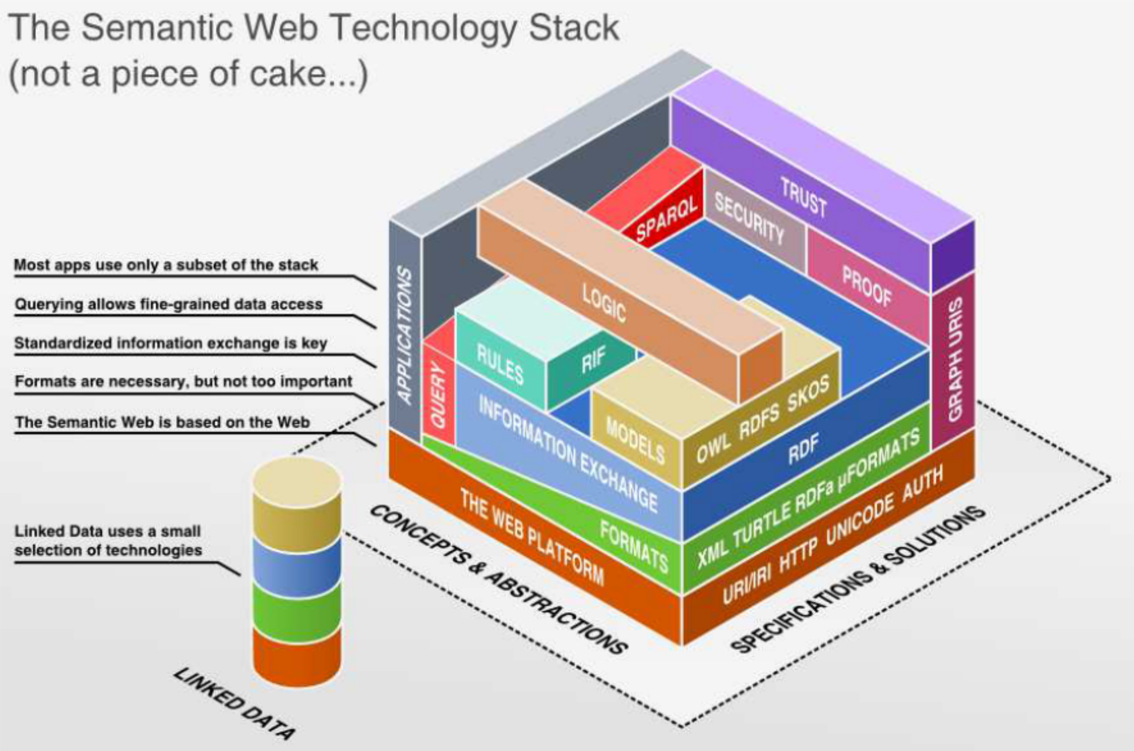

The Semantic Web Technology Stack
(not a piece of cake...)
Most apps use only a subset of the stack
Querying allows fine-grained data access
Standardized information exchange is key
Formats are necessary, but not too important
The Semantic Web is based on the Web
Linked Data uses a small selection of technologies
LINKED DATA
APPLICATIONS
QUERY
SPARQL
SECURITY
TRUST
PROOF
GRAPH URIS
LOGIC
RULES
RIF
MODELS
OWL RDFS SKOS
INFORMATION EXCHANGE
RDF
FORMATS
XML TURTLE RDFa µFORMATS
THE WEB PLATFORM
URI/IRI HTTP UNICODE AUTH
CONCEPTS & ABSTRACTIONS
SPECIFICATIONS & SOLUTIONS

**Fig. 3 The Semantic Web technology stack, retrieved from http://bnode.org/blog**

For this reason, from several years up to now, online taxonomies and encyclopaedias like Wordnet or Wikipedia are being presented as a good alternative to semantically annotate resources in applications where word sense disambiguation is crucial.

## 1.4 WordNet, Wikipedia and DBpedia

WordNet[6] (Miller, 1995) is an English lexical database elaborated in the Princeton University. In this semantic lexicon, nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms, *synsets*, each expressing a distinct concept; that is, a set of words that share one sense is a synset, and words with multiple meanings belong to multiple synsets. Its original version is implemented in English and has about 117.000 synsets[7]. Its extensive scope, its well-structured taxonomy and its free availability has fostered its use in many applications for processing natural language and retrieving information.

[6] WordNet home page: http://wordnet.princeton.edu/

[7] The WordNet version used in this dissertation is 3.1

Wikipedia[8] is a free online semi-structured encyclopaedia basically composed of *articles*, which define and describe conceptual entities. It was created on 2001 and has 21 million articles[9] (over 3.8 million in English alone). It is collaboratively written by volunteers around the world. Wikipedia articles are identified by unique identifiers (URIs), which can be used as reliable and consensual identifiers to represent *concepts* of the real world; these concepts, represented in a single page, are updated constantly by a large community. Articles can be assigned to one or more *categories*, providing an additional taxonomy

**Fig. 4. Wikipedia logo**

DBpedia[10] is a project where the main goal is extracting and structuring the information in Wikipedia. The content obtained is represented through Resource Description Framework (RDF). DBpedia extracts factual information from Wikipedia article, allowing users to find answers to questions where the information is spread across many different Wikipedia articles.

Among these knowledge sources, Wikipedia has more coverage of information than WordNet or domain-specific taxonomies, offering objects in a great variety of domains - science, geography, history, etc. -. WordNet does not include information about named entities - “Barak Obama” - or specific nouns - “hyperpolarization” - (Miller, 1995), and DBpedia is a step back from Wikipedia in terms of up-to-date issues, mainly because the former depends on the latter.

> *Due to these characteristics, this dissertation considers Wikipedia a valid vocabulary for the semantic annotations. Every item involved in a searching process - queries and documents - can then be related to the particular Wikipedia concepts they are referring to.*

## 1.5 PURPOSE AND GOALS

The main *purpose* of this dissertation is to develop an infrastructure, called *Itaca*, which, taking benefit from semantic and social annotations, obtains more relevant web pages than large-scale, current web search engines.

Semantics have to be gathered by means of collaborative usage of information generated by users, obtained through explicit relevance feedback techniques and annotations extracted from the searching process.

[8] Wikipedia home page: http://www.wikipedia.org

[9] The Wikipedia version used in this dissertation is that of January 2012.

[10] DBpedia home page: http://dbpedia.org

In Itaca, explicit feedback will provide the relevance of a web document with respect to the query executed. Annotations will refer to queries and documents; that is, every query and web document will be unambiguously described. The process of disambiguating a piece of data consists on selecting the most suitable sense for that information in a specific context from a well-defined vocabulary. In this dissertation, this vocabulary is Wikipedia.

Itaca will respond to user queries with a ranked list of the most relevant resources for a query. To elaborate this list, Itaca must take advantage of features from current web search engines, along with new ranking factors based on the gathered semantic annotations.

Annotations allow grapple with the semantic problems exposed on section 1.3.2, like ambiguity or polysemy. Users often attempt to address these problems by manually refining a query; however, semantics will allow applying *query expansion* automatically (Manning, Raghavan, & Schütze, 2008). In query expansion, given a query with its query terms, additional query terms are suggested. This is known as a *global method* to adjust the query, because it is independent of the query results. Relevance feedback is an example of a *local method*, because queries after the current query are adjusted in relation with the documents that have been selected as relevant. Itaca must find, for a given query, semantically similar concepts to the concepts of that query. If a query has been disambiguated with Wikipedia concepts, other similar concepts can be also taken into account to retrieve relevant documents. Query expansion with controlled vocabularies has been proved to improve recall in search engines (Williams, 2013).

With all this, the specific goals attained in the development of Itaca are the following:

> ***Goal 1:*** *The design and implementation of a data flow that allows collaborative 1) semantic annotations of resources without expertise knowledge about ontologies or other semantic techniques; and 2) filtering by explicit relevance feedback.*
>
> ***Goal 2:*** *The design and implementation of a ranking algorithm that, along with traditional static and dynamic features existing in current web search algorithms, uses semantic annotations and social feedback information to provide more relevant results.*
>
> ***Goal 3:*** *The design and implementation of a semantic and domain-independent similarity algorithm that, given two semantic concepts, automatically determines a score that indicates their similarity at semantic level, in order to provide query expansion.*

## 1.6 METHODOLOGY

Attainment of the goals presented in the previous section needs the execution of clear different tasks, mainly analysis, design, and validation. The design activities will pay special attention to the analysis tasks, because algorithms to be implemented should take benefit of current techniques, modifying existing methods - instead of working in new ones - if good results are proved:

***Task 1*: Analysis of existing ranking algorithms**: *In this task other ranking algorithms will be analysed, in order to see the features they take into account, how they gather the needed information, and the results obtained.*

***Task 2*: Analysis of existing semantic similarity techniques**: *It comprises the study of existing semantic similarity techniques, both with Wikipedia and with other knowledge sources.*

***Task 3: Architectural design of Itaca and hypothesis validation***: *This task designs the general structure of Itaca and sets the hypotheses that are to prove with this dissertation.*

***Task 4*: Development of a collaborative data flow**: *This task completes Goal 1, implementing the whole process that collects data from users by semantic annotations and explicit relevance feedback.*

***Task 5*: Development of a ranking algorithm**: *This task completes Goal 2, and uses the data collection gathered from user searching processes. It may use or modify existing techniques seen in Task 1.*

***Task 6*: Development of a semantic similarity algorithm applied to Wikipedia**: *This task completes Goal 3. It may use or modify existing techniques seen in Task 2.*

***Task 7*: Hypothesis validation**: *This task proves as valid the hypotheses formulated in Task 3, implementing experiments to evaluate them.*

***Task 8*: Documentation and conclusions**: *The aim of this task is to document this dissertation, paying special attention to the context, existing works and the final design, implementation and evaluation of the specific goals and hypotheses. Conclusions and future work will be also elaborated.*

## 1.7 DOCUMENT STRUCTURE

Chapters in this document are presented within four main parts:

***Part I*: Introduction and State of the Art**: *This part comprises chapters 1 to 3. After the introduction elaborated in this chapter, chapters 2 and 3 present scenarios, works and techniques related to this dissertation. More specifically, chapter 2 elaborates a review of search and ranking algorithms existing in the state of the art, some of them including semantics. Chapter 3 focuses on semantic similarity techniques with Wikipedia and with other knowledge sources.*

***Part II: Itaca layer***: *This part comprises chapters 4 to 7 and explains the inner details of the Itaca layer developed for this thesis. Chapter 4 introduces a brief explanation of the solution and lists the hypotheses to be proved. Chapters 5, 6 and 7 explain the data gathering process, the ranking algorithm and the semantic similarity measure implemented, respectively.*

***Part III: Evaluation and Conclusions**: This part is mainly devoted to the validation of the hypothesis listed in chapter 4, and comprises chapters 8 and 9. Chapter 8 shows the web application developed on top of a very well-known web search engine, proving Hypothesis 1. Afterwards, experiments to prove both Hypothesis 2 and Hypothesis 3 are detailed. Finally, conclusions and future research lines are exposed in Chapter 9.*

***Part IV: Appendices and References**. This part includes appendixes for further information, such as acronyms and definitions used throughout this document, the dissemination of results obtained with this dissertation, and the references used.*

# 2 RANKING ALGORITHMS AND SEMANTIC SEARCH

This chapter presents a review with different approaches for ranking documents in web search engines. First, the chapter analyses dynamic algorithms - those which take the user query into consideration for ranking. Second, this chapter describes static algorithms - those which measure the relevance of documents independently of queries. Third, descriptions of social characteristics from Web 2.0 applications are also presented, due to the fact that they have been presented as a possible enhancement for ranking algorithms. Finally, models involving semantic search are broached.

## 2.1 Dynamic algorithms

Dynamic models study the problem of identifying the best documents for a user query. In contrast to the static algorithms explained in 2.2, dynamic algorithms do take into account the terms involved in the query to select the documents.

### 2.1.1. Boolean model

This model is based on the Boolean logic, and views the documents to be searched and the user's query as sets of words or terms. Retrieval is based on whether or not the documents contain the query terms, and queries can be defined with Boolean expressions like *AND*, OR, and *NOT* (Manning et al., 2008).

The group of documents is also called *collection* or *corpus*, and they are usually *indexed* before the actual retrieval task starts. These indexes (also called *inverted indexes*, *inverted files* or *lexicon*) map the terms with the documents they appear on. The list of terms is also called *dictionary*, and the list of every term with the documents in which that term occurs is called *posting* (Fig. 5).

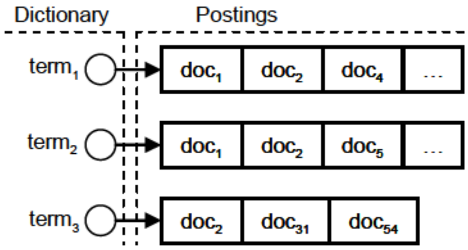


**Fig. 5. Structure of an inverted index**

This is a very simple model, where the queries are formulated with free text (plus the Boolean operators); no special language is required. Some extended versions have appeared, incorporating additional operators such as term proximity, where proximity can be declared with particular measure units like "within 6 words" or "within the same paragraph." Basically, the rest of the search algorithms are initially constructed with the principle of this Boolean model.

### 2.1.2. Vector-space model

In the Boolean model, a search process consists on looking whether a document matches a query or not. In the case of large document collections, the number of matching documents can far exceed that a human user could possibly shift through (Manning et al., 2008). In this case, the search engine has to re-order the documents matching a query. To do this, for each matching document, the search engine computes a score related to the query.

In the vector space model (VSM), every term of a document is given a score, based on the statistics of occurrence of the term in that document. In this model, a document is represented as a vector of such scores (Salton, 1971).

The simplest approach is to use as score the number of occurrences of the term *t* in a document *d*, known as *term frequency* or *tf(t,d)*:

$$tf(t,d) = \frac{f(t,d)}{max\{f(w,d) | w \in d\}}$$

**Equation 1. Term Frecuency**

Where the raw frequency of *t* in *d*, *f(t,d)*, is divided by the maximum raw frequency of any term, *w*, in the document *d*, to avoid a bias towards longer documents. However, this frequency considers all the words equally important for the measure, and this is not true. For example, in a collection of documents on the vehicle domain, the term "car" may be mentioned in almost every document, so its power in determining relevance is low. Thus, the *tf(t,d)* is combined with the *inverse document frequency* of a term *t*, *idf(t)*, defined as:

$$idf(t) = \log \frac{N}{df(t)}$$

**Equation 2. Inverse document frequency**

Where *df(t)* is the number of documents that contain the term *t*, and *N* is the total number of documents in the collection.

A composite weight is then defined as a combination of *tf* and *idf*, called *tf x idf*.

$$tf \times idf(t,d) = tf(t,d) \times idf(t)$$

**Equation 3. Term frequency - Inverse document frequency**

Finally, the relevance or score of a document *d* for a query *q* is the sum of the *tf x idf's* of each of the query terms:

$$score(q,d) = \sum_{t \in q} tf \times idf(t,d)$$

**Equation 4. Score of a document over a query in the vector space model**

Since the main problem in web search is to select a few relevant documents from many non-relevant ones, the general objective of the VSM weighted scheme is to assign high values to discriminating terms.

### 2.1.3. Probabilistic model

A probabilistic model measures the probability that a document belongs to the set of relevant documents in a corpus for a particular query. With this model, the document ranking is obtained estimating the probability of relevance with respect to the query, as stated by the *probability ranking principle* (Van Rijsbergen, 1979):

> *If a reference retrieval system's response to each request is a ranking of the documents in the collection in order of decreasing probability of relevance to the user who submitted the request, where the probabilities are estimated as accurately as possible on the basis of whatever data have been made available to the system for this purpose, the overall effectiveness of the system to its user will be the best that is obtainable on the basis of those data.*

Statistics about the actual document collection are used to estimate the probabilities of relevance or irrelevance of a document.

## 2.2 STATIC ALGORITHMS

A query-independent ranking, also called *static* ranking, is very important in a search engine. A good static ranking algorithm provides numerous benefits (L. Richardson & Ruby, 2007):

- *Relevance*: The static rank of a page provides a general indicator to the overall page quality. This is a useful input to the dynamic ranking algorithm.
- *Efficiency*: Typically, the search engine's index is ordered by static rank. By traversing the index from high-quality to low-quality pages, the dynamic ranker may abort the search when it determines that no later page will have as high of a dynamic rank as those already found.
- *Crawl priority*: The Web grows and changes very quickly. Search engines need a way to prioritize their crawl and, among other factors, the static rank of a page is used to determine this prioritization.

These algorithms are not used alone; the ordering of pages in a web search result list depends on the query executed. For this reason, these static methods are one of the multiple factors in scoring a web page given a query; static algorithms are usually applied to the set of relevant pages discovered using dynamic algorithms (query-dependent models), in order to rank the pages.

### 2.2.1. LINK-BASED FEATURES

Basic ranking algorithms of search engines, like PageRank (Brin & Page, 1998; Page et al., 1999) or HITS (Kleinberg, 1999), are based on the link structure of their indexed web pages. These algorithms focus on the quality of web pages by means of their inner and outer hyperlinks. In general, hyperlinks are defined by people. As such, they are indicative of the quality of the pages which they point to - when creating a page, a designer is supposed to link to pages of good quality -.

PageRank metric measures the intrinsic quality of a web page by the sum of the importance of the pages that do point to it. Consider a user who randomly surfs the Web, beginning at a web page, *p*. At each time, the user goes from the current page *p* to a randomly chosen web page that *p* hyperlinks to. As the user proceeds in this random walk, he visits some nodes more often than others. The most visited nodes are those with many links coming in from other frequently visited nodes. The basic idea behind PageRank is that pages visited more often in this walk are more important. The user will occasionally jump to a random page with some small probability, $\alpha$, or when on a page with no outer links. Then, the PageRank of a page *j*, scoring between 0 and 1, is the probability the user is on that page *j* at some point in time:

$$P(j) = (1-\alpha) + \alpha \sum_{i \in J} \frac{P(i)}{|I|}$$

**Equation 5. PageRank algorithm**

Where *I* is the set of outer links of page *i*, and *J* is the set of pages that link to page *j*. One of the problems of this method is that popular pages appear in the top ranking and, therefore, are more visible than others and became still more popular, failing in identifying new high-quality pages.

In HITS algorithm (*Hypertext Induced Topic Selection*), every web page is given two values, the *authority* number and the *hubness* number. The authority number indicates how good the page is in terms of its informational content. The calculation is obtained by a weighted sum of the hubness values of the pages that links to that page. The hubness number indicates, given a page, how good the information that links to is. The calculation is obtained by a weighted sum of the authority values of its outer links.

Because these algorithms are recursive, they must be iteratively evaluated until they converge. For this reason, they are computationally expensive. This is especially bad in HITS, which relies on query-time processing to deduce the hubs and authorities that exist in a subset of the Web, consisting of both the results to a query and the neighbourhood of these results. In the case of PageRank, this problem is less relevant, because it is calculated offline. However, PageRank assume statements that may not be true. Basically, it is based on two hypotheses:

- The number of visits to a particular page within a time interval is proportional to the relevance of the page.
- All web users will visit a particular page with equal probability.

Some works have tried to elaborate metrics to obtain unbiased web rankings. In (Cho, Roy, & Adams, 2005), authors study which the ideal way to measure the intrinsic quality of a page is, measuring the general probability that users will like a page when they look at it. Then, they propose an estimator that predicts the quality value of a page based on the evolution of the link structure of the Web. They define the quality of a page as the conditional probability that an average user will like the page when the user sees that page for the first time. Their main ideas are that 1) the creation of a link often indicates that a user likes the page and 2) a high quality page will be liked by most of its visitors, so its popularity may increase more rapidly than others. Basically, they consider not just the current link structure, but also the evolution and change in that link structure. Their experiments are done with a small subset of the Web and, even though their results indicate improvement over PageRank metric, they do not prove their efficiency for a larger dataset.

A problem with these link-based techniques is that the quality is implicitly stated by the web designer - the person who defines the hyperlinks in the web documents -, and not by the final user who reads the documents.

### 2.2.2. NON LINK-BASED FEATURES

The metrics of (M. Richardson, Prakash, & Brill, 2006) takes into account a number of simple page-based features that do not have into account the link structure of the Web. They explore the use of PageRank and other features for the direct task of statically ranking web pages, combined in a ranking machine learning algorithm, called fRank, based on a neural network of two layers. Authors propose four different sets of features, apart from PageRank:

- *Popularity*: It is measured as the number of times that a page has been visited by uses over some period of time. This data can be obtained by tools in users web browsers (if users are willing to provide this information). Here, as opposed to link-based algorithms, popularity is biased towards pages that web users, rather than web authors, visit.
- *Anchor text and incoming links*: These features are based on the information associated with links to a particular page. It includes features like the total amount of text in links pointing to the page (*anchor text*) or the number of unique words in that text.
- *Page*: This set consists of features which may be determined by looking at the page alone, such as the number of words in the body or the frequency of the most common term.
- *Domain*: This set contains features that are computed as averages across all pages in the domain, like the average number of outer links on any page or the average PageRank.

Their results outperform PageRank, implying that other non-linked based features contain useful information regarding the overall quality of a page.

## 2.3 EXPLOITING USER INFORMATION FROM SEARCH PROCESS

Using the information of users' searching sessions took relevance more than one decade ago; previous works have explored the idea of exploiting the information obtained from users in their searching process to improve the results offered by search engines.

### 2.3.1. CLICK-THROUGH DATA

The *click-through data* technique takes into account both the queries users execute in a search engine and the links users select afterwards, from the ranked results presented. This selection can be used to obtain *implicit relevance feedback* over a set of web resources.

In this area, (Hansen & Shriver, 2001) propose narrowing search results by observing the browsing patterns of users during search tasks. From users' logs, they first extract the search path that a user follows. Then, they make implicit query clustering, combining similar search terms on the basis of the web pages visited during a search session, because they observed that semantically related query terms often draw users to the same sets of URLs. In Table 1, there are three search sessions (initiated by different users) related to wedding dresses, and all produce responses of the same web pages.

**Table 1. Three search sessions initiated by 3 different users (Hansen & Shriver, 2001)**

| Query | "bridal + dresses" | "bridesmaid + dress" | "flower + girl + dresses" |
|---|---|---|---|
| URLs | www.priscillaofboston.com<br>www.bestbuybridal.com<br>weddingworld.net<br>www.ldswedddings.com<br>www.usedweddingdres… | www.martasbridal.com<br>weddingworld.net | www.bestbuybridal.com<br>www.martasbridal.com |

They also include pages that may not be listed by the search engine, but are visited by the user in the query session (and therefore registered in the proxy logs). Their presented algorithm is sufficient for the small data set involved, but they assume it does not scale well as either the number of queries or the number of query clusters increases.

(Joachims, 2002) use navigation data to improve the results in search engines by using classification techniques in conjunction with the click-through data of a meta-search engine. In particular, the author develops a method based on an SVM approach that uses click-through data for training, namely the query log of the search engine and the log of links the users clicked on in the presented ranking. The process used in this work, also called *learning to rank*, is depicted in Fig. 6:

1. Each query is assigned a unique identifier, which is stored in the query log along with the query words and the presented ranking.
2. The links on the results page presented to the user do not lead directly to the suggested document, but point to a proxy server. These links encode the query identifier and the URL of the suggested document.
3. When the user clicks on the link, the proxy records the URL and the query identifier in the click log. The proxy then uses the *HTTP Location* command to forward the user to the target URL.

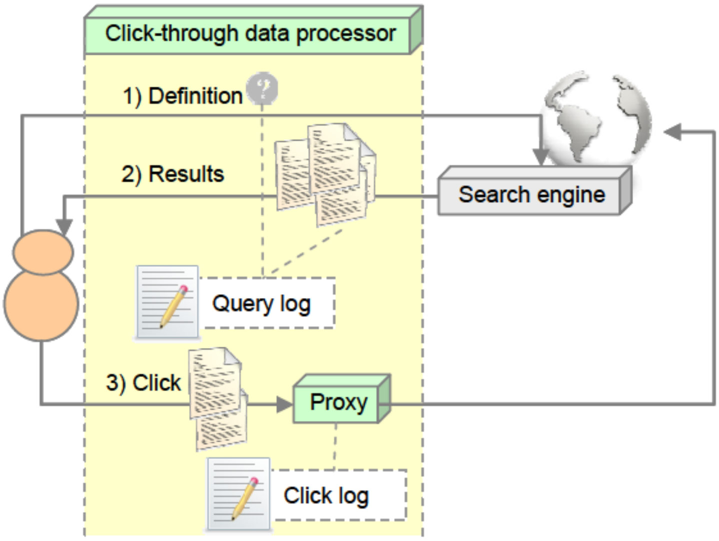


**Fig. 6. Process followed in (Joachims, 2002)**

Table 2 shows the first ten ranked results for the query “support vector machine” in Joachims’ work. The links underlined are the links the user clicked on (some links are abbreviated for space purposes).

**Table 2. Ranking presented for the query "support vector machine" (Joachims, 2002)**

| **Query** | **“support vector machine”** |
|---|---|
| **Ranking** | 1. Kernel Machines, svm.first.gmd.de/ |
| | 2. Support Vector Machine, jbolivar.freeservers.com |
| | 3. SVM-Light Support Vector Machine, ais.gmd.de/~thorsten/svm light |
| | 4. An introduction to Support Vector Machine, www.support-vector.net/ |
| | 5. Support Vector Machine and Kernel Methods References, svm.research.bell-labs.com/… |
| | 6. Archives of Support-vector-machines, www.jiscmail.ac.uk/lists/… |
| | 7. Lucent Technologies: SVM demo applet, svm.research.bell-labs.com/… |
| | 8. Royal Holloway Support Vector Machine, svm.dcs.rhbnc.ac.uk/ |
| | 9. Support Vector Machine - The Software, www.support-vector.net/software.html |
| | 10. Lagrangian Support Vector Machine Home Page, www.cs.wisc.edu/dmi/lsvm |

Considering this example, it is not possible to infer that links 1, 3 and 7 are relevant on an absolute scale. However, it is more plausible to infer that link 3 is more relevant than link 2 with probability higher than random. Author assumes that the user scans the ranking from top to bottom, and this user must have observed link 2 before clicking link 3, making a decision to not click on it. Outcomes showed that results obtained improved retrieval quality with respect to using the search engine alone.

A distinct approach was proposed on (Baeza-Yates & Tiberi, 2007). In this work, the authors extract semantic relations between queries from a query-click bipartite graph where nodes are queries and an edge between nodes exists when at least one equal URL has been clicked after showing the list of results. The goal of extracting relations from the logs is to create a tag-like structure with the queries, and to recommend URLs for similar queries. The structure is not a taxonomy based on queries, but a taxonomy of queries - a *logsonomy*, where queries are used as tags for web resources.

Even though the click-through technique appears to show good results and does not require any additional steps for users in their searching process, the nature of the click-through records does not allow capturing any real information about users’ activities or opinions beyond their selections; that is, this approach makes assumptions that may have a negative impact on the final results. For example, the approach considers that the mere selection of a result implies this result is relevant somehow to a particular query, which may not be the case for several reasons:

- Users are less likely to click on a link low in the ranking, independent of how relevant it is.
- Users might click on a link of the results of a query because it is interesting to them for other reasons than the query itself.
- Users might click on a link just to check if the result is interesting and then decide that it is not.

Information collected with this technique should be pre-processed somehow before its direct use, or combined with other techniques, in order to improve results with reliability.

### 2.3.2. Users Profiles

Some works explained in section 2.2.2, such as (D. Zhang & Dong, 2002) or (M. Richardson et al., 2006), use users’ preferences for certain queries and documents to generate a ranking. In fact, (D. Zhang & Dong, 2002) was one of the first works where the consideration of the tripartite structure of query logs appeared. In these models, the algorithm ranks resources based on the relationships among users, queries and resources of a search engine's log.

In (D. Zhang & Dong, 2002), authors propose MASEL, an algorithm that uses the search engine's log to exploit the relationships among users, queries and documents. In this model, the relevant documents retrieved must be also of the highest quality. Here, *quality* means both *authority* and *freshness*. The documents being frequently and recently accessed by experienced users have high quality. This is especially crucial in documents that have no hyperlinks (multimedia, images, etc.), and where static link-based models cannot be directly applied. Beginning with an initial query, their process is as follows:

1. The algorithm looks for the set of all users who have issued the query recently.
2. The set of all queries these users have issued recently is constructed.
3. The set of all resources relevant to these queries can be constructed.
4. Finally, the numerical quality is estimated by an iterative procedure, where a user is *good* if he/she issues many good queries, while a query is *good* if it can retrieve many good resources, while a resource is *good* if it is accessed by many good users.

Their initial experiments show that MASEL provides good search results for a wide range of queries. Besides, *query expansion* implicitly occurs. For example, the query "car" can return documents related to "BMW" or "Toyota" because they are often queried by users with similar interests recently. However, the iterative process makes the algorithm time consuming.

### 2.3.3. Tags

*Tags* are arbitrary words used to label resources, especially in social applications of the Web 2.0. The users of these applications make use of these annotations to organize their content. Popular sites that apply this technique are Delicious, where the tagged resources are website bookmarks, or Flickr, where the target resources are photographs. Though this way of classifying documents is not new, the collaborative process of doing it gained popularity on the Web several years ago. *Collaborative tagging* is the practice of allowing a group of users to freely attach keywords or tags to content. This process is useful when there is nobody in the librarian role or there is simply too much content for a single authority to classify (Golder & Huberman, 2006).

Initially, searching is performed over the text of tags and resources' descriptions, but no ranking is elaborated apart from ordering the hits in reverse chronological order or by the counts of tags. Furthermore, as the documents consist of short text snippets, or even photographs, basic techniques like *tf x idf* are not feasible. When the functionality of explicitly tagging appeared, several works started to take advantage of the tagging information for retrieval purposes. FolkRank (Hotho, Jäschke, Schmitz, & Stumme, 2006) or algorithms in (Bao et al., 2007) are examples of iterative methods with social annotations.

FolkRank is based on the PageRank algorithm. The original formulation of PageRank reflects the idea that a page is important if there are many pages linking to it, and if those pages are important themselves. The basic notion in FolkRank is that a resource which is tagged with important tags by important users becomes important itself. The same holds for tags and users (e.g. users are considered important if they tag important resources with important tags). Thus, FolkRank has a graph of vertices which are mutually reinforcing each other by spreading their weights. Its real application, though, is limited to a small-scale system which is not proved to be ready to use in large-scale web search engines.

Authors in (Bao et al., 2007) propose both static and dynamic algorithms for page ranking in web search (see Fig. 7):

- *SocialPageRank (SPR)*: A static algorithm which captures the quality of web pages, measured by their popularity; that is, the number of times they have been tagged.
- *SocialSimRank (SSR)*: A dynamic algorithm which calculates the similarity between social tagging and web queries.

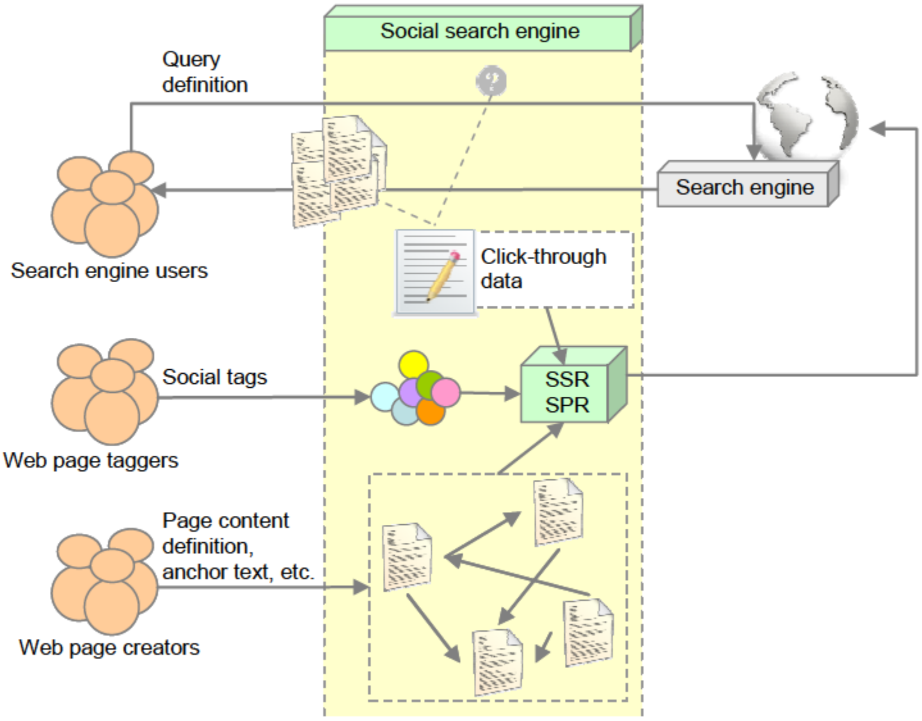


**Fig. 7. Social search with SocialSimRank and SocialPageRank (Bao et al., 2007)**

In the figure, the *web page creators* provide the web pages and anchor texts especially for the static ranking. The interaction log of the *search engine users* also benefits web search by providing the click-through data, which can be used in both static and dynamic rankings. Finally, *web page taggers* provide cleaner data that serve as brief reviews of the web documents. However, the iterative nature of both algorithms makes them inefficient when applied to a large number of resources, and no evidence of enhancing the retrieval quality of resources is shown in their experiments.

In general, traditional social tagging systems rank their results according to one of these main methods:

- *Naïve approach*: This technique ranks the pages according to the number of tagging actions. It locates the most popular pages at the top in the ranking. This is the case of SocialPageRank algorithm.
- *Co-occurrence approach*: Tags used to describe a single page are related somehow. In this case, when searching for resources related with a tag, resources of its related tags are also returned.
- *Adaptive approach*: this is a combination of the co-occurrence approach with the time factor of tags; a resource tagged more times recently is more relevant than another tagged more times in the past.

Some works (Jie et al., 2008; Michlmayr & Cayzer, 2007) made experiments applying these techniques and found that, even though the adaptive approach gives better results, it is more computationally expensive than the previous ones.

Most of the co-occurrence algorithms are based on *clustering* techniques to improve search and, thus, the user experience and the success of collaborative tagging. In the clustering step, tags are automatically clustered without putting the burden in final users (Hruschka, Campello, Freitas, & De Carvalho, 2009). Some approaches use semi-automated techniques for tagging using a controlled vocabulary. Other approaches are based on the probability that certain tags appear together in the same document. A graphical example of clustering can be seen in Fig. 8 (Begelman, Keller, & Smadja, 2006).

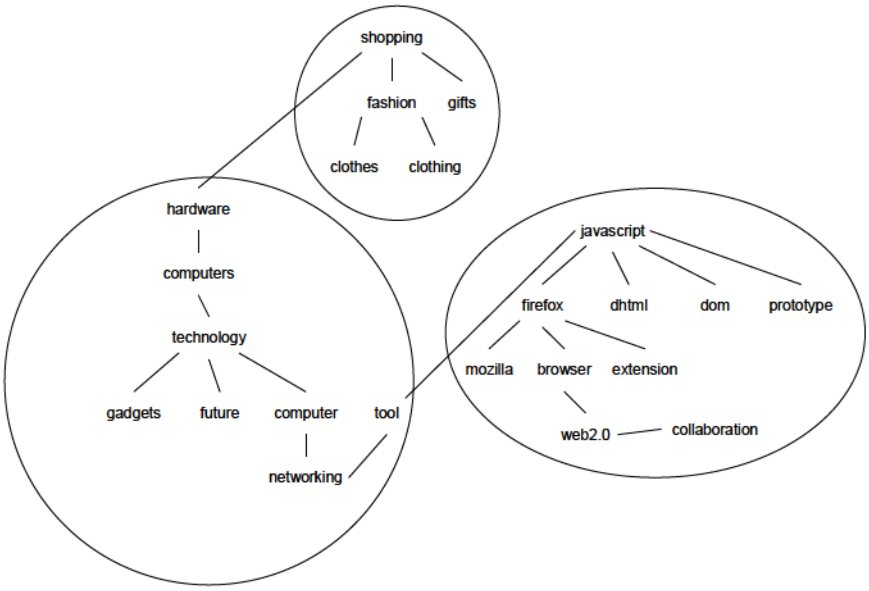


**Fig. 8. Example of a clustered graph of items, adapted from (Begelman et al., 2006)**

Nevertheless, as explained in section 1.3.2, several works such as (Golder & Huberman, 2006; Motta & Specia, 2007; X. Wu et al., 2006) conclude with a number of weaknesses when using tags for information retrieval and search. Most of these problems can be grouped in the following categories:

- *Ambiguity*: An ambiguous word has more than one meaning. When searching for documents with a word like “play”, related to a theatre piece, a search engine can return unrelated results such as, for example, a set of games for children.

- *Lack of synonym relations*: Words are synonymous if they have the same meaning. Words “irritated” and “annoyed” are very closely related; however, after searching for one of these words, found items will hardly contain the other word. Documents about television may be tagged either with tag “television” or with tag “tv”. This fact produces the perception that, given a query, not all the relevant items have been found.
- *Lack of consensus*: The lack of consensus in the use of tags, especially as granularity is concerned, makes a traditional tagging system quite inefficient. To describe a particular item, different users may consider terms at different levels of generality/specificity. For example, a user can tag a photograph as “bird”, whereas another user can tag the same photo as “eagle”. The example of the previous item about “television” or “tv” can be seen as another problem of lack of convention.

### 2.3.4. Recommendations

Other approaches for ranking in collaborative systems focus on *recommendation*, which suggest a list of resources that are unknown to a particular user. The recommendations are based on the opinions of users; any user providing information (usually rating or any other general filtering information) becomes a *recommender*. Based on this additional information, the two basic techniques for recommendation are:

- *User-based*: This technique explores the relationship among users. Here the recommendations are generated by considering solely the opinions of users on resources, which are then compared with similarity techniques, like the cosine metric.
- *Item-based*: This technique appeared when collaborative tagging was getting more widely used, and explores the relationship among resources – also called *items* - in order to give a certain recommendation for a particular resource.

Several works (Begelman et al., 2006; Sarwar, Karypis, Konstan, & Reidl, 2001) have demonstrated that item-based algorithms provide better results than user-based algorithms. Collaborative filtering works by building a repository where users set their preferences for items. The bottleneck in systems with user-based recommendations is the search for a set of neighbours among a large user population of potential neighbours. This set must have a history of agreeing with the target user (i.e., they either rate different items similarly or they tend to select similar items).

In item-based algorithms, though, recommendations for users are computed by finding resources that are similar to other resources the user has liked. Because the relationships between items are relatively static, less online computation is required. Algorithms in this category take a probabilistic approach and compute the expected value of users’ prediction given their ratings on other items. The efficiency of item-based algorithms in contrast to user-based ones was later confirmed in (Lathia, Hailes, & Capra, 2008).

However, this item-based approach has limitations. As seen in 2.3.3, using a combination of tags and the times they have been linked together for a same resource, can lead to anomalies for the search process. An example can be seen on Fig. 9. The terms “china” and “censorship” do not semantically relate to the term “google”. However, they were grouped in the samples of (Begelman et al., 2006) because of the hype around the story of Google’s censorship in China.

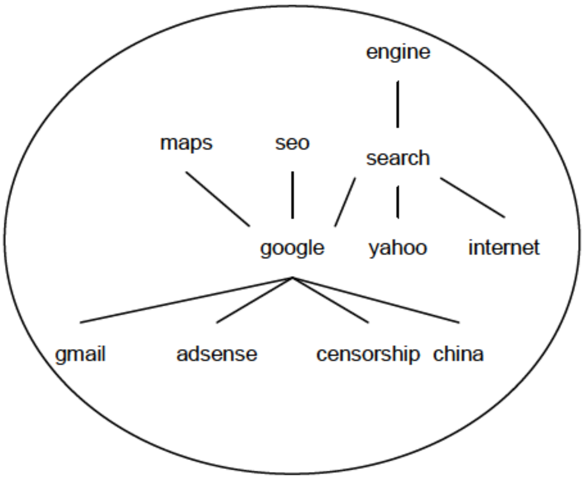


**Fig. 9. Cluster around "google", retrieved from (Begelman, 2006)**

## 2.4 SEMANTIC SEARCH

Leaving aside static methods, dynamic retrieval models are mainly based on looking whether keywords in a query match the content of web documents; that is, they compare text strings with text strings. Final results may omit documents referred to the piece of information stated in the query if these documents have not the same keywords as the query. As expressed by (Telang, 2013), web search engines still remain dumb. One of the main troubles is related with synonymy. If looking for the word “buy”, engine probably will not recover documents with the word “purchase”; if searching “motor vehicles”, engine will not recover documents with the word “car”. Another important case where the keyword-matching approach is problematic, is that of ambiguous queries or polysemy; the shorter the queries, the smaller the context to disambiguate them. As the most frequent query length is of 1 or 2 words (Experian Hitwise, 2011), ambiguity can affect to a large number of queries.

*Semantic search* is understood as the search by word senses, rather than literal strings or keywords. The Semantic Web paradigm fostered the importance of general semantics in the development of web search engines, even though conceptual search has been studied in Information Retrieval in general. This section reviews some of the engines or algorithms which apply semantics to search and ranking documents on the World Wide Web. This section does not focus on architectures for semantic repositories (that is, models developed for the retrieval of semantic documents), such as KIM (Kiryakov, Popov, Terziev, Manov, & Ognyanoff, 2004), or other web engines which locate ontologies and semantic documents online, like Watson (d'Aquin & Motta, 2011) or Swoogle[11]. The section also skips specific semantic search engines, like GoPubMed[12], a large-scale biomedical semantic indexing

[11] Swoogle home page: http://swoogle.umbc.edu/

[12] GoPubMed home page: http://www.gopubmed.org

and retrieval engine, or Yummly[13], a semantic web search engine for food, cooking and recipes.

### 2.4.1. Web directories

An old approximation to semantic search can be found on *web directories*; that is, categories to which web pages are somehow assigned. In the manual version, given a query, search results are organized by category, because pages are previously assigned to those categories. This basic approach needed manual updates to cover new pages. For this reason, methods for the automatic classification of web documents were proposed (Xue, Xing, Yang, & Yu, 2008).

Another approximation can be found on (Haveliwala, 2002), with Topic Sensitive PageRank. This method allows the query to influence the link-based score of simple PageRank, but it is still computed offline, requiring minimal query-time processing. During the offline crawling process, 16 topic-sensitive PageRank vectors are computed, using the top-level category from the Open Directory Project[14], to create for each page a set of importance scores with respect to those particular topics. At query time, the similarity of the query is compared to each of these topics. Then, instead of using a single global ranking vector, the metric takes the linear combination of the topic-sensitive vectors, weighted using the similarity of the query to the topics. This method yields a very accurate set of results relevant to the context of the particular query, because pages considered important in some subject domains may not be considered important in others, regardless of what keywords may appear either in the page or in anchor text referring to the page.

These approaches, though, suffer from relying on a predefined taxonomy of coarse categories.

### 2.4.2. Diversification

Another meaning-related approach focuses on *diversification*, which aims to rank top search results based on criteria which maximize their diversity.

SenseBot[15] is an example of this group. SenseBot generates a text summary of a list of web pages on the topic of the search query. It uses text mining and multi-document summarization to extract sense from web pages. However, its list of results is quite limited, and the semantic cloud it offers does not clarify the different meanings of the query. Besides, the average response time for every query is of 10 seconds.

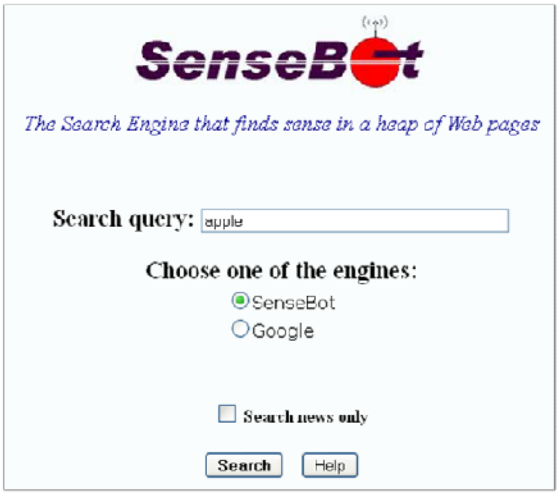
[13] Yummly home page: http://www.yummly.com/

[14] The Open Directory Project: http://www.dmoz.org/

[15] SenseBot main page: http://www.sensebot.net/

Fig. 10 shows the entry page for query definitions. In the sample of the figure, the search query is “apple”.

Fig. 11 depicts the results (summary page) listed with a generated semantic cloud (right). This semantic cloud, along with the different senses of the topic listed in Fig. 12, shows that those senses are not really different meanings, but different grouped websites.


SenseBot

The Search Engine that finds sense in a heap of Web pages

Search query: apple

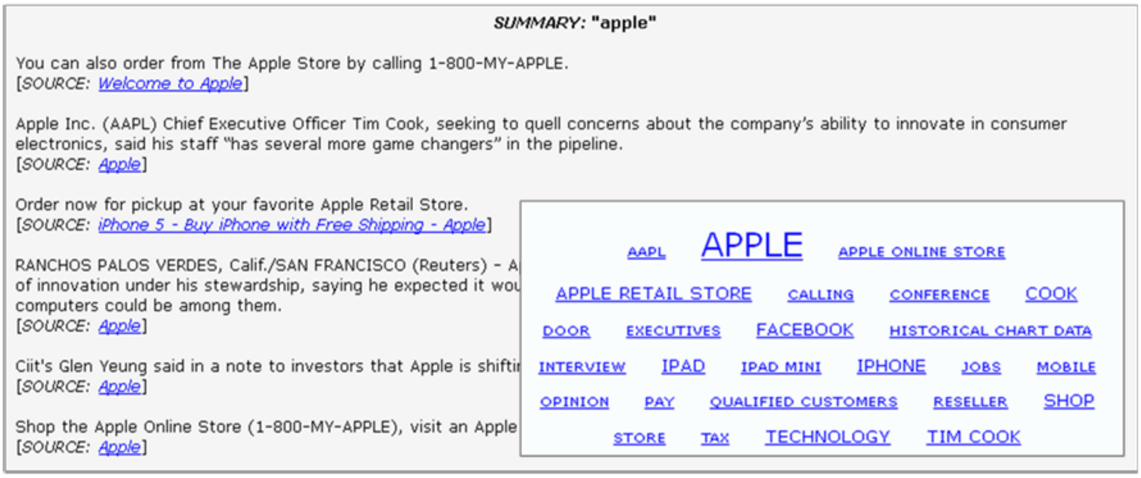
Choose one of the engines:

● SenseBot

○ Google

□ Search news only

**Fig. 10. Query definition in SenseBot, screenshot**


*SUMMARY:* "apple"

You can also order from The Apple Store by calling 1-800-MY-APPLE.
[*SOURCE: Welcome to Apple*]

Apple Inc. (AAPL) Chief Executive Officer Tim Cook, seeking to quell concerns about the company’s ability to innovate in consumer electronics, said his staff “has several more game changers” in the pipeline.
[*SOURCE: Apple*]

Order now for pickup at your favorite Apple Retail Store.
[*SOURCE: iPhone 5 - Buy iPhone with Free Shipping - Apple*]

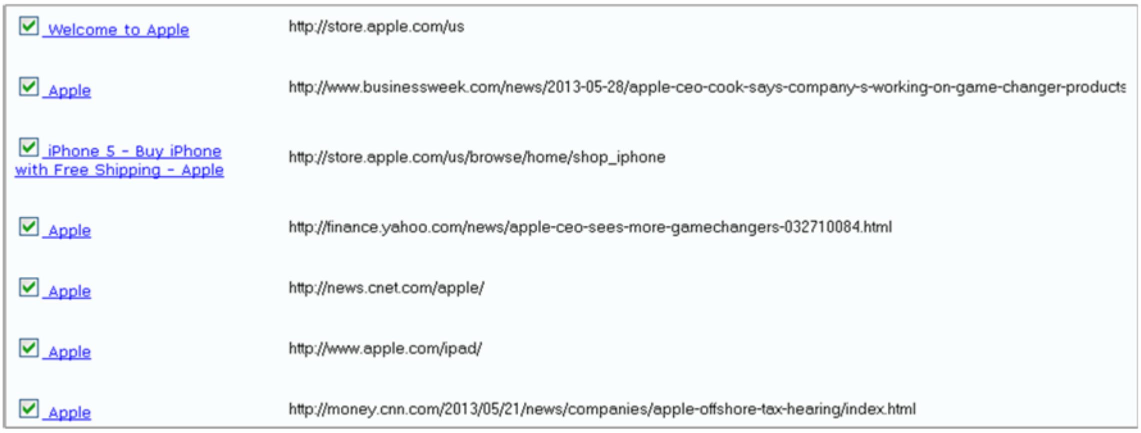
RANCHOS PALOS VERDES, Calif./SAN FRANCISCO (Reuters) - A
of innovation under his stewardship, saying he expected it wou
computers could be among them.
[*SOURCE: Apple*]

Ciit's Glen Yeung said in a note to investors that Apple is shifti
[*SOURCE: Apple*]

Shop the Apple Online Store (1-800-MY-APPLE), visit an Apple
[*SOURCE: Apple*]

AAPL APPLE APPLE ONLINE STORE
APPLE RETAIL STORE CALLING CONFERENCE COOK
DOOR EXECUTIVES FACEBOOK HISTORICAL CHART DATA
INTERVIEW IPAD IPAD MINI IPHONE JOBS MOBILE
OPINION PAY QUALIFIED CUSTOMERS RESELLER SHOP
STORE TAX TECHNOLOGY TIM COOK

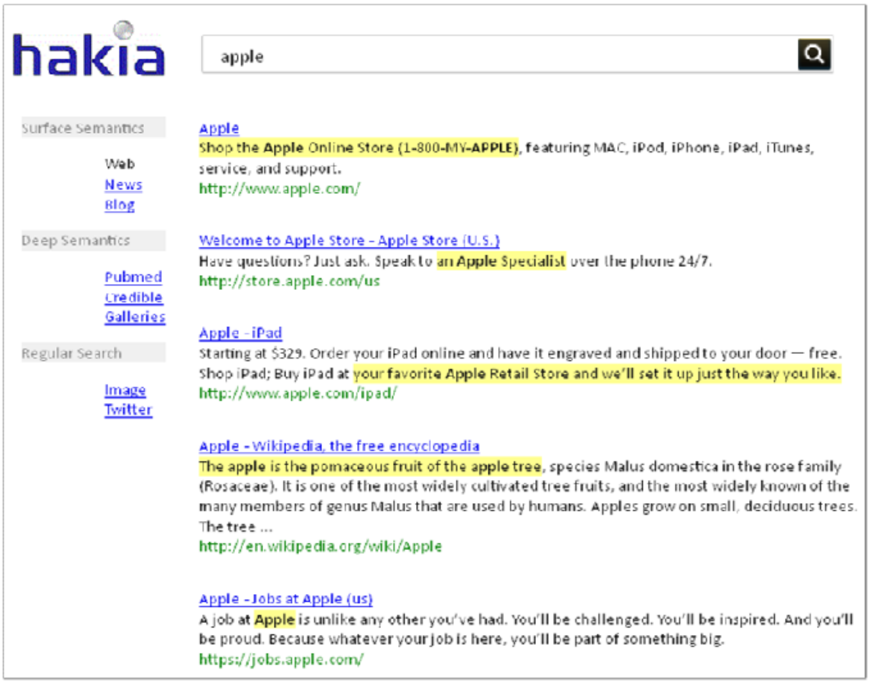
**Fig. 11. SenseBot screenshots**



| | |
|---|---|
| ☑ Welcome to Apple | http://store.apple.com/us |
| ☑ Apple | http://www.businessweek.com/news/2013-05-28/apple-ceo-cook-says-company-s-working-on-game-changer-products |
| ☑ iPhone 5 - Buy iPhone with Free Shipping - Apple | http://store.apple.com/us/browse/home/shop_iphone |
| ☑ Apple | http://finance.yahoo.com/news/apple-ceo-sees-more-gamechangers-032710084.html |
| ☑ Apple | http://news.cnet.com/apple/ |
| ☑ Apple | http://www.apple.com/ipad/ |
| ☑ Apple | http://money.cnn.com/2013/05/21/news/companies/apple-offshore-tax-hearing/index.html |

**Fig. 12. Different senses of the search query in SenseBot, screenshot**

Hakia[16] has a high level of diversification, returning a list of results for different source types (blog, Wikipedia, news, etc.). However, as in SenseBot, it does not offer a set of resources of a particular meaning either (see example for the search query “apple” in Fig. 13).



Apple
Shop the Apple Online Store (1-800-MY-APPLE), featuring MAC, iPod, iPhone, iPad, iTunes, service, and support.
http://www.apple.com/

Welcome to Apple Store - Apple Store (U.S.)
Have questions? Just ask. Speak to an Apple Specialist over the phone 24/7.
http://store.apple.com/us

Apple - iPad
Starting at $329. Order your iPad online and have it engraved and shipped to your door — free. Shop iPad; Buy iPad at your favorite Apple Retail Store and we'll set it up just the way you like.
http://www.apple.com/ipad/

Apple - Wikipedia, the free encyclopedia
The apple is the pomaceous fruit of the apple tree, species Malus domestica in the rose family (Rosaceae). It is one of the most widely cultivated tree fruits, and the most widely known of the many members of genus Malus that are used by humans. Apples grow on small, deciduous trees. The tree ...
http://en.wikipedia.org/wiki/Apple

Apple - Jobs at Apple (us)
A job at Apple is unlike any other you've had. You'll be challenged. You'll be inspired. And you'll be proud. Because whatever your job is here, you'll be part of something big.
https://jobs.apple.com/

**Fig. 13. Hakia web search sample, screenshot**

### 2.4.3. Search result clustering

Another approach to conceptual search is the *web search result clustering technique*. This technique consists of partitioning the results obtained in response to a query into a set of labelled clusters that reflect the different meanings of the query. In (Bernardini, Carpineto, & D'Amico, 2009), a query is first executed in the search engine and then, results are grouped by the different senses of the query with a clustering algorithm. If the documents that relate to a same subtopic have been correctly placed within the same cluster and if the user is able to choose the right cluster from the cluster labels, such items can be accessed in logarithmic rather than linear time. The algorithm is based on extracting and analyzing keyphrases contained in the snippets of the search results, through a combination of natural language processing and clustering techniques. In (Bernardini et al., 2009; Navigli & Crisafulli, 2010), authors first acquire the senses of a query, from a text corpus, and then cluster the search results based on their semantic similarity to those word senses.

Although interesting for certain tasks, this technique may return irrelevant results if users are interested in just one particular meaning of the query. This is mainly due to the reason that the top results of the search engine, the ones used for the clustering task, are considered relevant, which is not always the case.

[16] Hakia main page: http://www.hakia.com/

### 2.4.4. Semantic information retrieval

An ultimate model consists in associating explicit concepts to queries and documents, performing *word sense disambiguation*. One of its implementations assumes an existing ontology-based repository, where the instances of an ontology are used as *semantic annotations* for documents. In (Castells, Fernández, & Vallet, 2007), authors propose an adaptation of the vector space model, enriched with annotations, to elaborate a ranking algorithm. They address further challenges in the enhanced model proposed in (M. Fernández et al., 2011) (see Fig. 14).

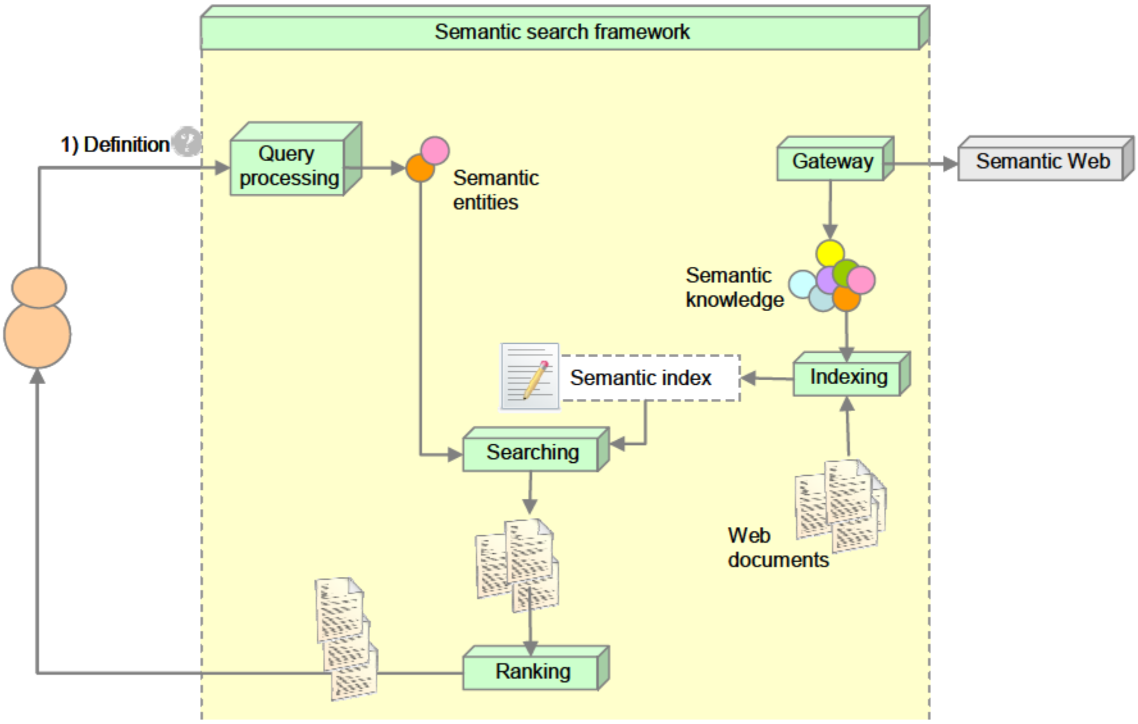


**Fig. 14. Semantic search framework from (Fernández et al., 2011)**

One of the modifications from the original model is a better interface for the definition of the query; in the first work, queries had to be defined with expert semantic languages. They also tackle the problem of covering multiple domains in the annotation process by adding a semantic gateway that provides access to large amounts of semantic metadata. This process has to be done previous to the search process, and is used at indexing time to improve the domain coverage. Instead of assign weights to the keywords of every document, as in the original vector space model, this work assigns weights to the annotations, reflecting the discriminative power of instances with respect to the documents, using an adaptation of the *tf x idf* algorithm:

$$weight(x,d) = \frac{freq(x,d)}{\max\limits_{\forall y \in Y(d)} \{freq(y,d)\}} \times \log \frac{N}{df(x)}$$

**Equation 6. Weight of an instance *x* in a document *d***

In the equation, *freq(x,d)* is the number of occurrences in *d* of the keywords attached to the instance *x*; *Y(d)* is the set of all instances in *d*, *df(x)* is the number of documents annotated with *x* and *N* is the set of all documents in the search space.

The query execution returns a set of SPARQL tuples that satisfy the query. The semantic entities are extracted from those tuples; then, the model accesses the semantic index to collect all the documents in the repository that are annotated with these semantic entities. Once the list of documents is formed, the search engine computes a ranking value for every document. To do that, the engine calculates the semantic similarity between every document vector and the query vector, as follows:

$$sim(d,q) = \frac{d \times q}{|d||q|}$$

**Equation 7. Adaptation of the classic vector-space model**

This work, though, assumes that a knowledge base has been built, instances have been created (manually or automatically) and that these instances have been associated to the documents; they do not focus on these tasks, crucial for the success of the framework. Besides, this and most of the semantic approaches design and implement new semantic information retrieval systems. They do not fully exploit the information indexed, functionalities and features (like static algorithms) provided by current, large-scale web search engines.

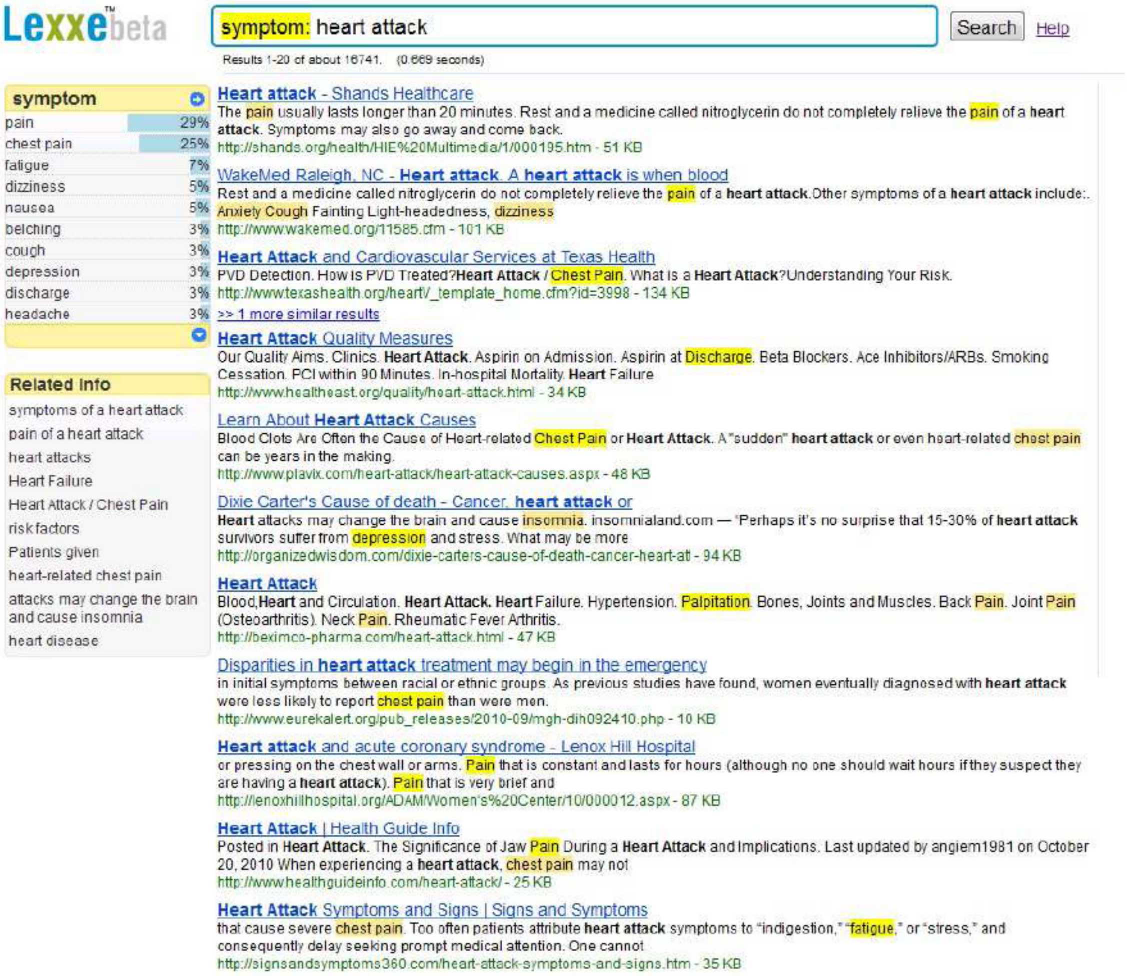
An exception in this sense is iGlue[17], created in 2007 in order to be settled on top of any web browser in such a way that, when viewing a web page and clicking on any page word (a noun, a person, a place, etc.), it would deliver to the user further information about that entity. This application was composed of an experimental database with semantic entities related to images, videos or web pages. The idea was ambitious, but the project is not alive any more.

Lexxe[18] is a web search engine which supports normal query searches. However, they have included a new search technology called *semantic key* for specific information search (specific queries - answers). Semantic keys enable users to query with a special keyword or concept (the semantic key) in order to find instances under that concept.

For example, if users want to find out what colours are associated with Toyota Camry cars, it is common that they type the words “colour toyota camry” in a search engine’s query slot. Current search engines search and return documents with a combination of the words “colour” and “toyota camry” in them. Search engines do not *know* “red” is a colour. Users cannot fully take advantage of the search results and get the information straight away, due to the missing link between “colour” and “red”, “black, “blue”, etc. Lexxe search engine calls semantic key to words like “colour”, which could point to “red”, “black” and “blue”. A query example is “colour: ocean” Not only does it return all the results with at least one colour word close to the target search term, but also it highlights them. Fig. 15 shows an example with the query “symptom: heart attack”. Besides the list of search results and the possible answers highlighted, Lexxe also runs some statistics (on the upper left corner in the figure).

[17] Interview in the Guardian online to Peter Vasko, the chief executive of the company behind iGlue: http://www.guardian.co.uk/technology/pda/2010/aug/27/iglue-semantic-web

[18] Lexxe home page: http://www.lexxe.com/



symptom: heart attack

Results 1-20 of about 16741. (0.669 seconds)

| symptom | |
|---|---|
| pain | 29% |
| chest pain | 25% |
| fatigue | 7% |
| dizziness | 5% |
| nausea | 5% |
| belching | 3% |
| cough | 3% |
| depression | 3% |
| discharge | 3% |
| headache | 3% |

Related Info

symptoms of a heart attack
pain of a heart attack
heart attacks
Heart Failure
Heart Attack / Chest Pain
risk factors
Patients given
heart-related chest pain
attacks may change the brain and cause insomnia
heart disease

**Heart attack** - Shands Healthcare
The pain usually lasts longer than 20 minutes. Rest and a medicine called nitroglycerin do not completely relieve the pain of a **heart attack**. Symptoms may also go away and come back.
http://shands.org/health/HIE%20Multimedia/1/000195.htm - 51 KB

WakeMed Raleigh, NC - **Heart attack**. A **heart attack** is when blood
Rest and a medicine called nitroglycerin do not completely relieve the pain of a **heart attack**.Other symptoms of a **heart attack** include:. Anxiety Cough Fainting Light-headedness, dizziness
http://www.wakemed.org/11585.cfm - 101 KB

**Heart Attack** and Cardiovascular Services at Texas Health
PVD Detection. How is PVD Treated?**Heart Attack** / Chest Pain. What is a **Heart Attack**?Understanding Your Risk.
http://www.texashealth.org/heart/_template_home.cfm?id=3998 - 134 KB
>> 1 more similar results

**Heart Attack** Quality Measures
Our Quality Aims. Clinics. **Heart Attack**. Aspirin on Admission. Aspirin at Discharge. Beta Blockers. Ace Inhibitors/ARBs. Smoking Cessation. PCI within 90 Minutes. In-hospital Mortality. **Heart** Failure
http://www.healtheast.org/quality/heart-attack.html - 34 KB

Learn About **Heart Attack** Causes
Blood Clots Are Often the Cause of Heart-related Chest Pain or **Heart Attack**. A "sudden" **heart attack** or even heart-related chest pain can be years in the making.
http://www.plavix.com/heart-attack/heart-attack-causes.aspx - 48 KB

Dixie Carter's Cause of death - Cancer, **heart attack** or
**Heart** attacks may change the brain and cause insomnia. insomnialand.com — "Perhaps it's no surprise that 15-30% of **heart attack** survivors suffer from depression and stress. What may be more
http://organizedwisdom.com/dixie-carters-cause-of-death-cancer-heart-at - 94 KB

**Heart Attack**
Blood,**Heart** and Circulation. **Heart Attack. Heart** Failure. Hypertension. Palpitation. Bones, Joints and Muscles. Back Pain. Joint Pain (Osteoarthritis). Neck Pain. Rheumatic Fever Arthritis.
http://beximco-pharma.com/heart-attack.html - 47 KB

Disparities in **heart attack** treatment may begin in the emergency
in initial symptoms between racial or ethnic groups. As previous studies have found, women eventually diagnosed with **heart attack** were less likely to report chest pain than were men.
http://www.eurekalert.org/pub_releases/2010-09/mgh-dih092410.php - 10 KB

**Heart attack** and acute coronary syndrome - Lenox Hill Hospital
or pressing on the chest wall or arms. Pain that is constant and lasts for hours (although no one should wait hours if they suspect they are having a **heart attack**). Pain that is very brief and
http://lenoxhillhospital.org/ADAM/Women's%20Center/10/000012.aspx - 87 KB

**Heart Attack** | Health Guide Info
Posted in **Heart Attack**. The Significance of Jaw Pain During a **Heart Attack** and Implications. Last updated by angiem1981 on October 20, 2010 When experiencing a **heart attack**, chest pain may not
http://www.healthguideinfo.com/heart-attack/ - 25 KB

**Heart Attack** Symptoms and Signs | Signs and Symptoms
that cause severe chest pain. Too often patients attribute **heart attack** symptoms to "indigestion," "fatigue," or "stress," and consequently delay seeking prompt medical attention. One cannot
http://signsandsymptoms360.com/heart-attack-symptoms-and-signs.htm - 35 KB

**Fig. 15. Results for a particular query in Lexxe search engine, screenshot**

One basic problem of Lexxe is the relatively small number of semantic keys users can operate with. In the beta version launched on 2011, there were only 500 semantic keys approximately. Besides, there are no enhancements for general informational searches, which are the target of this dissertation.

In general, even though the usage of formal annotation vocabularies produces a more expressive semantic enrichment in a searching process than merely using tags, semantic mechanisms like ontologies still lack of mass support (Rico Almodóvar, 2012), leaving its use and management to the expert community. Non-toy domain ontologies are still very limited for many areas of interest, and complex ontologies require specialized knowledge of experts.

# 3 Semantic Similarity Measures

Semantic similarity indicates how much two words are related in meaning. This chapter details the principal existing measures used to calculate semantic similarity between words. First, it resumes traditional semantic similarity methods with text corpora and well-formed hierarchies, such as WordNet. Second, it gives a review of methods which use Wikipedia as their knowledge source. Most of these methods are intended for estimating semantic relatedness in general, which is not the goal of this thesis, but they are worth mentioning. A brief comparison of their experimental results is given at the end of the chapter.

## 3.1 Non Wikipedia-based semantic similarity measures

*Semantic similarity* indicates how much two words are related in meaning – that is, the degree of synonymy between the two words - and it is different from *semantic relatedness*, which evaluates how much two words are associated in general (Resnik, 1995). For example, the pair “cough” and “common cold”, and the pair “common cold” and “influenza”, are both semantically related. However, “common cold” and “influenza” are also semantically similar, because they both are *from the same type,* illnesses – whereas “cough” is *a symptom* of “common cold”.

Traditional approaches to calculate semantic similarity can be grouped depending on the representation of their knowledge source: statistical approaches based on co-occurrence of words in big corpora; path-based methods using lexical structures; and multi-source methods which combine statistical approaches with path-based methods. This section analyses all of them, excluding Wikipedia-based methods, which are explained later on.

### 3.1.1. Co-occurrence-based measures

These metrics use statistical approaches or vector-based methods in text corpora, focusing on the co-occurrence of words. They are usually applied to situations where there is not a well-formed lexical structure - taxonomies or thesauri - to process.

The first important group is formed by *gloss-based* measures, which use word-sense glosses of machine-readable dictionaries to compute similarity and relatedness in general. One example is Lesk’s algorithm (Lesk, 1986), which uses dictionary-gloss overlapping to disambiguate the words in a phrase. Taking the disambiguation of the word “bank” in the sentence “I sat on the bank of the lake” as an example, possible definitions of “bank” are:

> def(bank)$_1$= “financial institution that accepts deposits and channels the money into lending activities”;
> def(bank)$_2$= “sloping land especially beside a body of water”.

And the definition of “lake” is:

> def(lake) = “a body of water surrounded by land”.

There is no overlap between *def(bank)$_1$* and *def(lake)*, but there exist overlap between *def(bank)$_2$* and *def(lake)*, with the words “body” and “water”. The problem with this method is that dictionary entries are short, and may not provide sufficient information about the relation of two words.

Another group of techniques uses *vector-based* methods, which also focus on the co-occurrence of words in dictionaries (Wilks et al., 1990) or large corpora (Church & Hanks, 1990). In these measures, the authors define a vocabulary from the words in the corpora or the dictionary glosses. Using this vocabulary, a co-occurrence matrix is built. This matrix indicates how often each word co-occurs with each other in the vocabulary. Thus, each word is represented by a vector, where each dimension shows how often the word occurs with another word in the vocabulary. Finally, to measure the similarity of two words, these techniques compute the similarity (i.e., cosine similarity) between their respective vectors.

A variant of measures in this group use the World Wide Web as the knowledge corpus. Using indexed documents from web search engines to compute semantic similarity has a clear advantage: almost any possible word or sense can have been indexed, and a potential measure does not have to depend on limited sources which sometimes do not have particular concepts.

One simple technique in this variant consists on obtaining the *hits* (page counts) of two words (separately and together) from a search engine and applying similarity coefficients or overlapping metrics from statistics.

(Cilibrasi & Vitanyi, 2007) calculated a distance metric based on hits and an overlapping metric, which was called *Normalized Google Distance* (NGD):

$$\mathrm{NGD}(c_1,c_2)=\frac{\max\{\log|c_1|,\log|c_2|\}-\log|c_1\cap c_2|}{\log N-\min\{\log|c_1|,\log|c_2|\}}$$

**Equation 8. Distance metric by Cilibrasi & Vitanyi**

*N* is the number of estimated indexed pages in Google web engine, and $c_1 \cap c_2$ represents the set of pages where the term "[$c_1$] AND [$c_2$]" appears. (Trillo, Gracia, Espinoza, & Mena, 2007) transformed the NGD into an exponential, monotonically increasing similarity measure:

$$\mathrm{sim}_{\mathrm{trillo}}(c_1,c_2)=e^{-2\mathrm{NGD}(c_1,c_2)}$$

**Equation 9. Similarity measure by Trillo et al.**

However, page counts ignore the position of a word in a document; even though two words may appear in a same document, one may be far apart from the other, and may not be related at all. Besides, polysemous words can also be a problem for the final results: searching for "apple" can yield pages about the fruit or about the company.

In (Bollegala, Matsuo, & Ishizuka, 2007), authors propose a model with a SVM, combining four different coefficients based on hits - Jaccard, Dice, Overlap and PMI - and one NLP technique based on the extraction of syntactic patterns from text snippets. This last approach makes this measure more computationally expensive than the previous approaches.

In general, co-occurrence measures are used to compute general semantic relatedness; they are not focused on measuring semantic similarity in particular. Besides, the election of an appropriate corpus is crucial to obtain acceptable results, especially important when working with specific domains.

### 3.1.2. Path-based measures

These measures are based on graphs of lexical taxonomies and usually focus on the paths between concepts of the hierarchy to calculate their similarity.

One of the taxonomies most frequently used in the literature is WordNet[19] (Miller, 1995), due mainly to its extensive scope and its free availability. Wordnet[20] is an English lexical database where nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms, *synsets*, each expressing a distinct concept. The most frequently encoded association among synsets is the hyponymy (also known as *is-a-type-of* or simply *is-a* relation), and represents the semantic relation of belonging to a generic concept (e.g., a *girl* is a *female person*). See Fig. 16 for a WordNet extract.

A simple approach considers the *minimal path length* between two concepts, by counting the edges (or nodes) that separate them. This idea of edge or node counting goes back to Quillian's model of semantic memory (Quillian, 1967), where concepts were represented by nodes and relationships by links. (Rada, Mili, Bicknell, & Blettner, 1989) demonstrated that counting the edges or nodes of the shortest path between two concepts in a net can be used as a measure of *conceptual distance* if just the hyponym relations are considered: the bigger the similarity between two concepts, the smaller their conceptual distance. If a word is polysemous (multiple senses represented in the net), multiple paths might exist, and the shortest path of all of them is considered. Other works such as (Rada et al., 1989) or (Lee, Kim, & Lee, 1993) used this metric as the basis for ranking documents by their similarity to a query.

As conceptual distance is a decreasing function of similarity, distance metric is usually transformed into a similarity measure by subtracting the shortest path between two concepts (henceforth, *shortest* $(c_1, c_2)$) to the longest possible path in a hierarchy (twice the maximum depth of the net, $D$):

$$\mathrm{sim}_{\mathrm{rada}}(c_1,c_2) = 2 \times D - \mathrm{shortest}(c_1,c_2)$$

**Equation 10. Similarity measure by Rada et al.**

(Leacock & Chodorow, 1994) also transform the conceptual distance into a similarity measure, but through a logarithm. Besides, they normalize the shortest path, dividing its length by the length of the longest path in the taxonomy:

$$\mathrm{sim}_{\mathrm{lc}}(c_1,c_2) = -\log\left(\mathrm{shortest}(c_1,c_2)/(2 \times D)\right)$$

**Equation 11. Similarity measure by Leacock & Chodorow**

[19] WordNet home page: http://wordnet.princeton.edu/

[20] Wordnet extracts displayed in these thesis correspond to version 3.1.

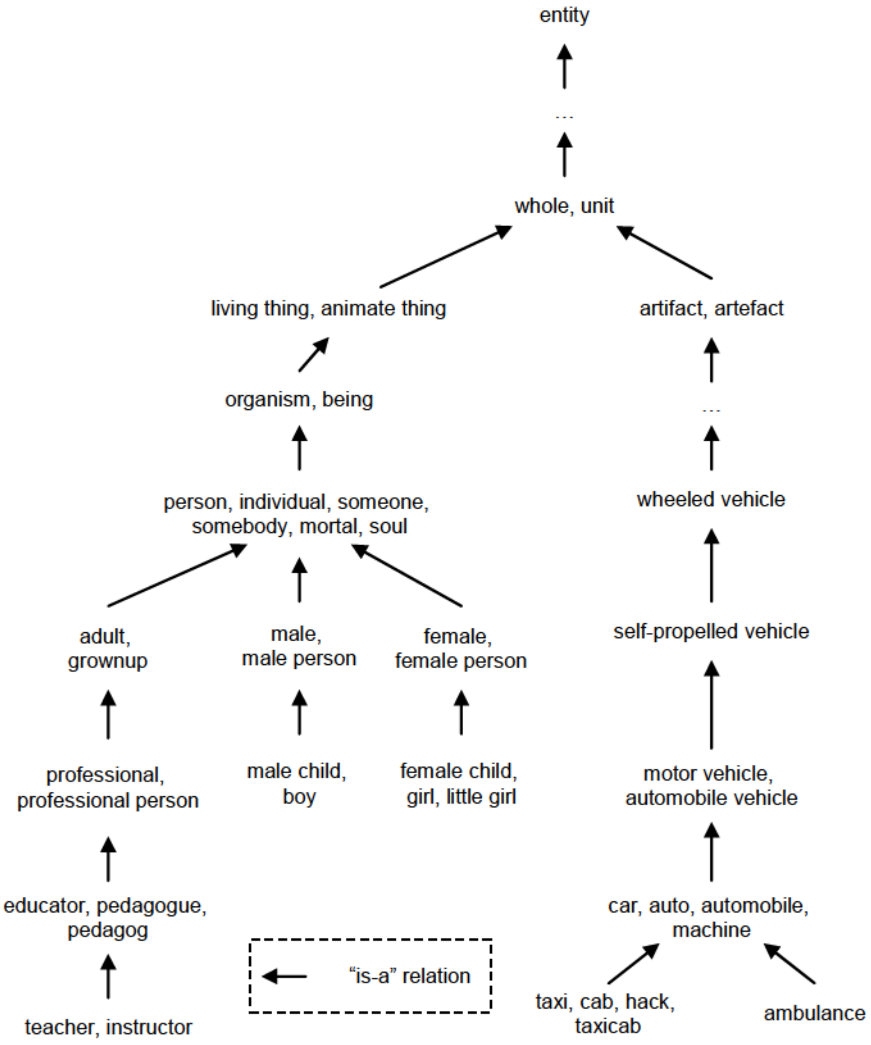

entity
…
whole, unit
living thing, animate thing
artifact, artefact
organism, being
…
person, individual, someone, somebody, mortal, soul
wheeled vehicle
adult, grownup
male, male person
female, female person
self-propelled vehicle
professional, professional person
male child, boy
female child, girl, little girl
motor vehicle, automobile vehicle
educator, pedagogue, pedagog
car, auto, automobile, machine
"is-a" relation
teacher, instructor
taxi, cab, hack, taxicab
ambulance

**Fig. 16. Extract of the WordNet 3.1 taxonomy**

The basic problem with the approaches based on shortest path is that they rely on the assumption that all relations in the hierarchy represent a uniform distance, and this is not usually true. Going back to Fig. 16, *car←taxi* seems to have a closer similarity than *whole←artifact*, but both relations are represented by the same distance. This problem is clearer when using broad-coverage sources. To avoid this, shortest-path technique is usually combined with some other taxonomic features:

- *Local density*: The density of a node in a hyponym relation is the number of its incoming links. It is considered that the greater the density, the closer the distance between the nodes involved in the association.
- *Depth of a node*: The depth of a node is the path of that node to the root of the taxonomy. Semantic distance is lower as we go down the hierarchy, because the differentiation among concepts is based on fine-grained details. Therefore, nodes in the upper levels of a hierarchy have less semantic similarity.
- *Relation type*: When not only semantic similarity is required, other hierarchical relations are used: meronymy-holonymy (also known as *part-of*, *substance-of*, etc.), associative (cause-effect), etc.

(Sussna, 1993) applies these 3 features to compute semantic relatedness. Particularly, he states that links are not semantically uniform, so a different weight is assigned to each of them.

(Z. Wu & Palmer, 1994) avoid using just the length of the shortest path. For that, they take into account both the distance of two concepts in the hierarchy and the depth of the first common node upwards that subsumes these two concepts (see Fig. 17). This node is called *least common subsumer* (henceforth, *lcs*):

$$\mathrm{sim}_{\mathrm{wp}}(c_1,c_2)=\frac{2\times\mathrm{depth}(lcs)}{\mathrm{shortest}(c_1,c_2)+2\times\mathrm{depth}(lcs)}$$

**Equation 12. Similarity measure by Wu & Palmer**

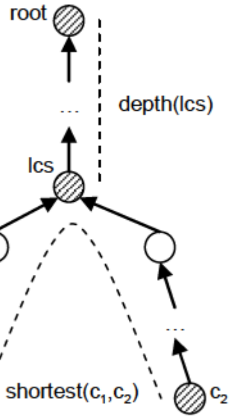


**Fig. 17. Illustrative example of factors used by Wu & Palmer**

In (Blázquez-del-Toro, Fisteus, Centeno, & Sánchez-Fernández, 2008), semantic similarity between two concepts is obtained considering the local density of the nodes in the shortest path that links those concepts, considering that the greater the density of the nodes in the path, the higher the similarity between the concepts. Initially, their measure was intended to be applied when ontologies are the knowledge source involved. However, they only use the hypernym-hyponym relations and, therefore, their measure can be applied to hierarchical structures in general. In fact, their experiments are finally made with a simplified version of WordNet, transformed into an ontology. Their measure can be reduced to the following form:

$$\mathrm{sim}_{\mathrm{blazquez}}(c_1,c_2)=\max_{lcs\in LCSs(c_1,c_2)}\left\{\frac{\mathrm{sim}_{to\_lcs}(c_1,lcs)\times\mathrm{sim}_{to\_lcs}(c_2,lcs)}{\mathrm{sim}_{to\_lcs}(c_1,lcs)+\mathrm{sim}_{to\_lcs}(c_2,lcs)-\mathrm{sim}_{to\_lcs}(c_1,lcs)\times\mathrm{sim}_{to\_lcs}(c_2,lcs)}\right\}$$

**Equation 13. Similarity measure by Blázquez-del-Toro et al.**

Multiple inheritance can appear in the taxonomy, so they choose the *lcs* of all the possible *lcs*'s of the two concepts (*LCSs($c_1$,$c_2$)*) that yields the best value (*max*). To measure the similarity between a concept *c* and an *lcs*, they apply the following formula:

$$\mathrm{sim}_{to\_lcs}(c,lcs)=\frac{k\times\mathrm{depth}(lcs)}{k\times\mathrm{depth}(lcs)+\log(E_{lcs}/E_c)}$$

**Equation 14. Similarity measure between a concept and its *lcs*, by Blázquez-del-Toro et al.**

The main assumption here is that, the more specific a concept *c*, the less the difference between it and its parent in the hierarchy. This feature is the *information ratio* between *lcs* and *c*, $E_{lcs}$ / $E_c$. To calculate this ratio, consider that a node has the 100% of the information of the subhierarchy of which is root of, (in Fig. 18, $E_{lcs}$ is 100), whereas each of its children will have an equitable fraction of that mass of information, *E / number of children* (as density of *lcs* is 4 in Fig. 18, each of its children has a mass of information of 25%).

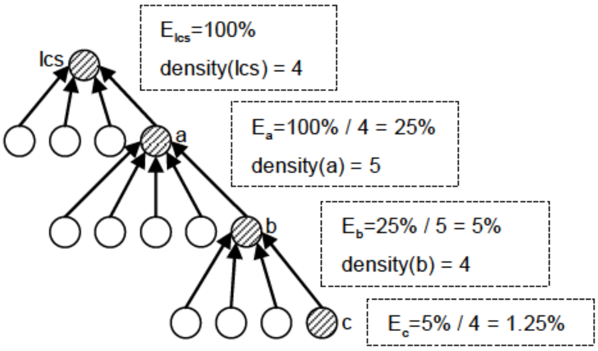


**Fig. 18. Illustrative example of information ratio**

Then, considering *parents* ($c_{lcs}$) as the set of hypernyms of *c* in the path to that *lcs*, including the *lcs*:

$$E_{lcs}/E_c = \prod_{p \in parents(c_{lcs})} density(p)$$

**Equation 15. Information ratio by Blázquez-del-Toro et al.**

Going back to the example in Fig. 18:

$$E_{lcs}/E_c = \frac{100}{1.25} = \frac{100}{100 \big/ 4 \times 5 \times 4} = 4 \times 5 \times 4 = 80$$

The taxonomy selected to compute these metrics has an important impact in the results. If these path-based measures are used in the hierarchy of verbs in WordNet, instead of the hierarchy of nouns, the results obtained are worse because the verb hierarchy is shallower and not so well formed (Pedersen, Banerjee, & Patwardhan, 2005). Besides, an implicit problem of the structure of taxonomies like WordNet is that a comparison can only be made between concepts representing the same part of speech - nouns with nouns, verbs with verbs, etc.-

### 3.1.3. MULTI-SOURCE MEASURES

These methods use different path-based techniques from taxonomies and combine them with statistical information obtained from corpora.

The *information-content* approach is the most used in this group; it is based in Information Theory and was proposed by (Resnik, 1995). He defines the semantic similarity of two concepts as the maximum of the information content of their *lcs*:

$$sim_{resnik}(c_1,c_2) = \max_{lcs \in LCSs(c_1,c_2)} \{ic(lcs)\}$$

**Equation 16. Similarity measure by Resnik**

Where the information content (*ic*) of a concept *c* refers to the probability of occurrence of the concept *c* in a large text corpus:

$$\mathrm{ic}(c) = -\log(\mathrm{p}(c))$$

**Equation 17. Information content measure**

If the probability of finding a term in a set of documents is 100% (that is, the term is on every document), it has no information content; a concept with high information content is more specific. To set an example, *fork* has more information content than *thing*. Some implementations use the function *1 / p(c)* instead of *- log (p(c))*. The frequencies of concepts in the taxonomy are estimated using a large collection of text (Resnik's and similar works used the Brown Corpus of American English). Each term that occurred in the corpus was accounted as an occurrence of the concept (taxonomic class) containing it; that is, the frequency of a concept *c* is calculated counting each time a term *t* appears in the corpus *(count(t))*, where *terms(c)* is the set of terms subsumed by concept *c*:

$$freq(c) = \sum_{t \in terms(c)} count(t)$$

**Equation 18. Frequency of a concept by Resnik**

The probability is computed simply as relative frequency, where *T* is the total number of terms observed, excluding those not included in any WordNet synset:

$$p(c) = \frac{freq(c)}{T}$$

**Equation 19. Probability of occurrence of a concept**

An illustrative example can be found in Fig. 19, from (J. J. Jiang & Conrath, 1997). It depicts the fragment of the WordNet (version 1.5) noun hierarchy, and numbers in parentheses are the corresponding information content values of a particular node. The similarity between *car* and *bicycle* is the *ic* of the concept *vehicle*, 8.30, which has the maximum value among all the concepts that subsume both *car* and *bicycle*. In contrast, the similarity between *car* and *fork* is 3.53. These results conform to human perception that cars and forks are less similar than cars and bicycles.

The information content feature is considered *coarse-grained*, because it does not differentiate the similarity between any pair of concepts in a taxonomy as long as their *lcs* is the same. Given the extract on Fig. 16, semantic similarity between *boy* and *instructor* would be the same as *boy* and *girl*, as both pairs share the same *lcs*.

(J. J. Jiang & Conrath, 1997) propose a modification, where the similarity between two concepts is twice the shared information content subtracted from the sum of the individual information contents of each concept:

$$\mathrm{sim}_{\mathrm{jc}}(c_1, c_2) = \mathrm{ic}(c_1) + \mathrm{ic}(c_2) - 2 \times \mathrm{ic}\, lcs(c_1, c_2))$$

**Equation 20. Similarity measure by Jiang & Conrath**

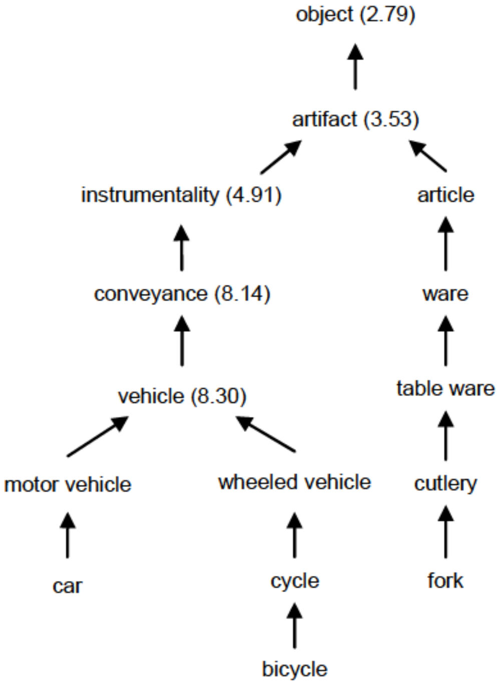


**Fig. 19. Illustrative example of information content**

(Lin, 1998) also proposes a normalization, but via ratio:

$$\mathrm{sim}_{\mathrm{lin}}(c_1,c_2)=\frac{2\times \mathrm{ic}(lcs(c_1,c_2))}{\mathrm{ic}(c_1)+\mathrm{ic}(c_2)}$$

**Equation 21. Similarity measure by Lin**

If multiple inheritance is considered, the selected *lcs* will be the one that maximizes the value.

These measures take into consideration simple terms, not word senses; therefore, strange results can arise. For example, *tobacco* and *horse* are not similar at all, but if we take the word *horse* as the colloquial term to refer to *heroin*, they are quite similar. As the information content measure always selects the maximum value between all possible concepts, in this example, it will yield the value of *ic* between *tobacco* and *heroin* instead. To partially avoid this problem, in (R. Richardson & Smeaton, 1995) the frequency of a concept is divided by the number of possible senses that the word *t* may have, *senses(t)*:

$$freq(c)=\sum_{t\in terms(c)}\frac{count(t)}{|senses(t)|}$$

**Equation 22. Frequency of a concept by Richardson & Smeaton**

An information content-based measure still uses a hierarchical structure, but is less sensitive to it; however, results with this approach, as with general corpora-based measures, also depend on the particular corpus used.

(Y. Li, Bandar, & McLean, 2003) tried different strategies, using the length of the shortest path between two words, the information content and the depth of their *lcs*. They assumed that semantic similarity does not only depend on different factors, but the correct combination of them. For that, they tried different linear and non-linear measures. At the end, the formula that yielded best results was the following:

$$\mathrm{sim}_{\mathrm{li}}(c_1,c_2)=e^{-\alpha\times\mathrm{shortest}(c_1,c_2)}\times\frac{e^{\beta\times\mathrm{depth}(lcs(c_1,c_2))}-e^{-\beta\times\mathrm{depth}(lcs(c_1,c_2))}}{e^{\beta\times\mathrm{depth}(lcs(c_1,c_2))}+e^{-\beta\times\mathrm{depth}(lcs(c_1,c_2))}}$$

**Equation 23. Similarity measure by Li et al.**

Every factor is transformed into non-linear functions. In the case of the shortest path function, *shortest($c_1$,$c_2$)*, they use an exponential (non-linear) and monotonically decreasing function. In the case of *depth* factor, they use a monotonically increasing function. They also played with the information content feature; however, outcomes showed that it did not influence the final results.

## 3.2 WIKIPEDIA-BASED SEMANTIC SIMILARITY METHODS

Approaches seen so far tackle with problems related to the source they are applied to.

Measures using taxonomies or dictionaries cannot be used in scenarios that require a great coverage of the real world; for example, words like some proper nouns (“Angela Merkel”) or specific terminology (“hyperpolarization”) are not defined in WordNet. New words or modifications in existing traditional corpora are managed slowly in time; besides, most of these sources are built just in English language.

Measures using web search engines take advantage of the huge amount of updated information stored over the World Wide Web; however, they cannot take benefit from path-based features of structured sources.

Finally, all these measures take into account words, not word senses. (Resnik, 1995) foresaw that, in measuring semantic similarity between words, “*it is really the relationship among word senses that matters and a similarity measure should be able to take this into account*”.

Wikipedia, though, provides a vast knowledge for computing semantic similarity between word senses. It is built upon a more defined structure than that from results obtained through web search engines and has more information than WordNet or specific taxonomies.

In Wikipedia, the information of every concept of the real world is represented in single pages or articles. It offers concepts from a great variety of domains - science, geography, etc. -, and all of them are updated constantly by a large community. Concepts belonging to different parts of speech are located under the same structure, whereas in many vocabularies, like WordNet, they are separated (nouns with nouns, verbs with verbs), which makes it difficult to analyse their similarity.

There are works which use traditional semantic similarity measures adapted to Wikipedia; most of these works, however, focus on measuring relationships rather than similarities. (Strube & Ponzetto, 2006) took benefit of Wikipedia to calculate the relatedness between a pair of concepts. Their work, known as *WikiRelate!*, applies a combination of several existing techniques, but adapted to the structure of categories of Wikipedia. In particular, they use 1) two path-based measures (Leacock & Chodorow, 1994), (Z. Wu & Palmer, 1994); 2) a co-occurrence measure (Lesk, 1986), for what they use the content of the articles of the two concepts; and 3) a modified version of the information content measure of (Resnik, 1995).

The two path-based measures require the *lcs* of the two concepts. Given the concepts $c_1$ and $c_2$, they extract the lists of categories $cats(c_1)$ and $cats(c_2)$ they belong to. Given those category lists, for each category pair $<cat_i, cat_j>$, $cat_i \in cats(c_1)$, $cat_j \in cats(c_2)$, they perform a depth-limited search of maximum depth of 4 for a *lcs*. Finally, given the set of paths found, they select the path that maximizes the information content.

To apply the information content measure, they do not use a specific corpus, but the *intrinsic* information content of a node in the structure of categories:

$$\mathrm{ic}(cat) = 1 - \frac{\log(\mathrm{hypo}(cat)+1)}{\log(C)}$$

**Equation 24. Information content of a category in *WikiRelate!* approach**

In Equation 24, *hypo(cat)* is the number of hyponyms of category *cat*, and *C* is the total number of nodes in the taxonomy. In this case, *cat* is one of the *lcs's* between two concepts.

(Gabrilovich & Markovitch, 2007) calculated the semantic relatedness between two arbitrary texts by an approach called ESA (*Explicit Semantic Analysis*). This approach is a co-occurrence technique which represents every Wikipedia concept as a word vector, where each dimension represents a word which occurs within the document, and is given a certain weight. This weight is calculated by the *tf x idf* technique. Then, an inverted index is constructed, where each word is assigned the list of Wikipedia concepts they appear in. With this, given a text fragment, they first iterate over the text words, retrieves the corresponding entries from the inverted index, and merges them into a weighted vector of Wikipedia concepts that represent the given text. Finally, the semantic relatedness between two texts is obtained by applying the cosine metric to the vectors of that pair of text fragments. (Wee & Hassan, 2008) also used this technique to calculate similarity between words and, further, similarity between texts.

There are other works that calculate relationships based on the (hyper)links within Wikipedia articles. A link is a connection manually-defined between two disambiguated concepts. One of these works is the WLM (*Wikipedia Link-based Measure*) (Milne & Witten, 2008). Here, the measure is a combination (the average) of two measures. The first one is defined by the angle between the vectors of the links found within the articles of the two concepts. It is similar to the *tf x idf* technique but, instead of working with term counts weighted by the probability of each term occurring, authors work with the link counts, weighted by the probability of each link. Thus, if $c_1$ and $c_2$ are the source and target concepts respectively, then the weight *w* of the link $c_{1\rightarrow}c_2$ is:

$$w(c_1 \to c_2) = \log\left(\frac{|W|}{|C_2|}\right)$$

**Equation 25. First measure in WLM**

In Equation 25, *W* is the set of all concepts in Wikipedia and $C_2$ is the number of concepts that link to $c_2$. Thus, links are considered less significant for judging the similarity between articles if many other articles also link to the same target $c_2$. These link weights are used to generate vectors to describe each of the two concepts of interest and, finally, the cosine similarity is used.

The second measure of WLM is a metric similar to NGD but, instead of working with search results, authors work with Wikipedia links:

$$\text{dist}_{\text{wlm}}(c_1, c_2) = \frac{\log(\max\{|C_1|, |C_2|\}) - \log(|C_1 \cap C_2|)}{\log(|W|) - \log(\min\{|C_1|, |C_2|\})}$$

**Equation 26. Second metric in WLM**

$C_1$ and $C_2$ are the sets of all articles that link to $c_1$ and $c_2$ respectively. $C_1 \cap C_2$ represents a co-occurrence-based factor, counting the Wikipedia pages that link to both concepts.

Another work using links to calculate the semantic relatedness between pairs of Wikipedia concepts is that of (X. Zhang, Asano, & Yoshikawa, 2011). They distinguish between *explicit* and *implicit* relationships. An *explicit* relationship is given by a hyperlink between two concepts. An *implicit* relationship is given by a page containing the two concepts.

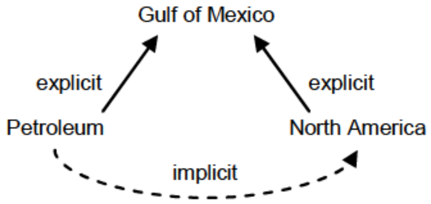


**Fig. 20. Example of relationships in Zhang et al.'s work**

They compute the strength of a relationship on a network from concept $c_1$ to concept $c_2$ using the value of the flow whose source is $c_1$ and destination is $c_2$. Every edge has a weight and the value of a flow sent along an edge is multiplied by the weight of the edge. The weight of every edge is assigned through a function based on three factors obtained from the category structure of Wikipedia: distance and co-citation, already seen in previous works, and connectivity. The distance is the length of the shortest path between two concepts. Co-citation is the reverse of co-occurrence, and measures the number of concepts linked by both the two concepts (stronger relationship when the number is larger). The connectivity from $c_1$ to $c_2$ on a network is the minimum number of vertices such that no path exists from $c_1$ to $c_2$ if the vertices are removed (more connectivity, more relationship).

## 3.3 REPORTED RESULTS

In the study by (Rubenstein & Goodenough, 1965), human volunteers gave a similarity score to 65 pair of terms. (Miller & Charles, 1991) replicated the experiment, providing human evaluation for 30 of those initial 65 pairs. These datasets are considered as the ground truth, and a similarity measure just has to look how well its ratings correlate with those human ratings.

Table 3 shows the Pearson correlation coefficient reported by the most relevant measures. In order to evaluate and compare results, most of the reported works took into account a subset of 28 pairs (henceforth, test set) from the 30 pairs of (Miller & Charles, 1991) - except Jiang & Conrath's, who took the 30 pairs as their test set -, and they correlated their results with the human ratings obtained with either Miller & Charles or Rubenstein & Goodenough's experiments. Wikipedia-based methods are not directly comparable, because authors used different sets for the evaluation, applied in the literature to measure semantic relatedness instead of semantic similarity. These sets are the PASCAL Recognizing Textual Entailment Corpus[21] for Wee & Hassan's work and WordSim353 Text Collection (Finkelstein et al., 2002) for the rest.

**Table 3. Correlation coefficients for a test set**

| Semantic similarity measure | Reported correlation |
|---|---|
| Co-occurrence based | |
| Cilibrasy & Vitanyi (2007) | 0.79 |
| Bollegala et al. (2007) | 0.79 |
| Path based | |
| Rada et al. (1989) | 0.66 |
| Wu & Palmer (1994) | 0.79 |
| Leacock & Chodorow (1994) | 0.83 |
| Blázquez-del-Toro et al. (2008) | 0.81 |
| Multi-source based | |
| Resnik (1995) | 0.74 |
| Jiang & Conrath (1997) | 0.84 |
| Lin (1998) | 0.75 |
| Li et al. (2003) | 0.89 |
| Wikipedia based | |
| WikiRelate! (2006) | 0.56 |
| Gabrilovich & Markovitch (2007) | 0.75 |
| Wee & Hassan (2008) | 0.60 |
| Milne & Witten (2008) | 0.64 |
| Zhang et al. (2011) | 0.56 |

The measures with higher coefficients (higher than 0.8) are two multi-source methods (Li et al. and Jiang & Conrath) and path-based approaches (Leacock & Chodorow, and Blázquez-del-Toro et al.). Just one out of the four (Jiang & Conrath's model) uses the information content as a feature of their final formula. Cilibrasi & Vitanyi, and Bollegala et al.'s method have a broader coverage than the rest of measures, because they have the World Wide Web as source, but their results are not better than some of the simple path-based models like the Wu and Palmer's measure.

[21] PASCAL Recognizing Textual Entailment: http://pascallin.ecs.soton.ac.uk/Challenges/RTE2

Wikipedia-based methods are not very promising either. Best result by WikiRelate! is obtained applying the shortest-path metric. Milne and Witten's measure is based on the links that relate articles, so it requires low computational effort, but its final result is far from Gabrilovich & Markovitch's work. Anyway, values in this group do not improve results obtained by traditional similarity models applied to WordNet or models based in web search engines.

# PART II. Itaca Layer

# 4 Problem Analysis and Solution

Once the general concepts, approaches and techniques involved in the context of this dissertation have been presented, I resume the problems seen in the state of the art and explain the general view of the solution offered. A global vision of Itaca will be presented, taking into account the goals listed on chapter 1, depicting the overall architecture of the Itaca approach. I also enumerate the hypotheses to be proved for the consecution of the goals and how these hypotheses will be evaluated.

## 4.1 Problems

Basic static and dynamic algorithms (sections 2.1 and 2.2) have worked well during these years. Current search engines based on these algorithms have a great number of users due to their easiness of use and their proved effectiveness. Queries are defined in free natural text (no special language is required) and results are returned in an order based on their quality and their relevance to the query. The quality is commonly based on the hyperlink structure of web documents, the number of visits, etc., and the relevance to a query is based on textual similarity.

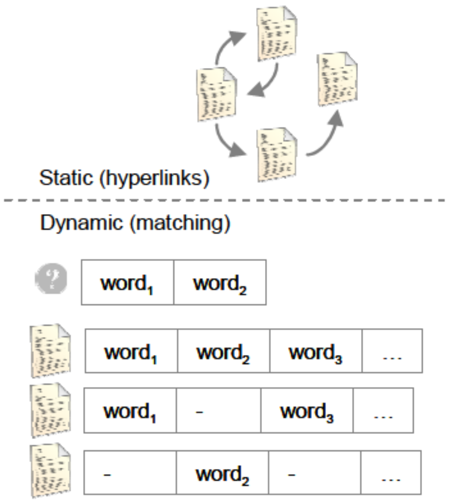


**Fig. 21. Graphical review of search and ranking algorithms (I)**

This dissertation does not have to avoid these algorithms. In fact, the solution has to consider the advances in web search obtained throughout these last two decades. However, there are still limitations that reveal an important gap in web mining:

- Query terms are merely sequence of textual words, so problems associated to natural language descriptions can appear, such as ambiguity or lack of synonym relations.
- Multimedia web resources (images, videos) do not incorporate any linkage information, so link-based approaches cannot be applied to search these types of items.
- The quality of a page in link-based techniques is implicitly stated by the web designer, instead of the reader.
- The loneliness of users in the searching process is notable; collaborative techniques used in Web 2.0 applications could be incorporated in the process to enhance the effectiveness perceived by the users.

Techniques were created in order to make final users to participate in a collaborative way in the searching process. More specifically, click-through data and user profiles appeared to log users' behaviour with respect to queries and results, to incorporate more information to further searching processes. However, the information these logs provide may not be accurate. Basically, as seen in section 2.3.1 and 2.3.2 users are more likely to click on a link high in a ranking list of documents, independently of how relevant it is or its relative importance to their interests.

To address the aforementioned limitations, this dissertation considers the necessity of a *semantic enrichment of query terms and web resources*. This additional information can enhance and facilitate the information search process. The semantic enrichment implies the creation of annotations that specify the concepts involved both in queries and web documents, and directly specify the relevance of a web document with respect to a query.

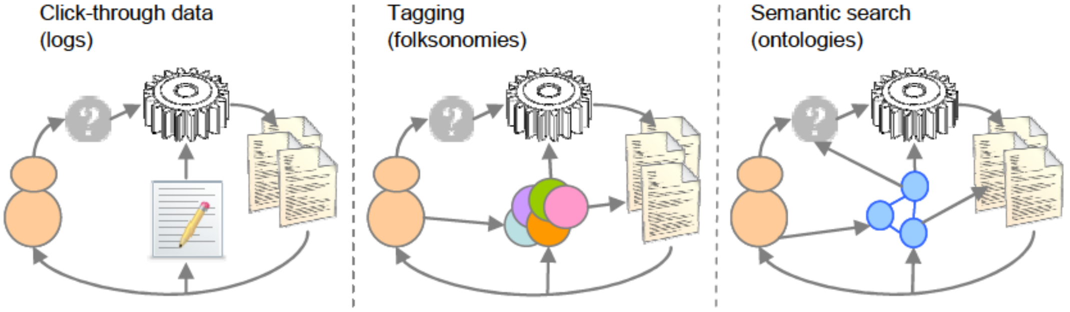


**Fig. 22. Graphical review of search and ranking algorithms (II)**

In this sense, tagging has received considerable interest as a mean for adding semantic metadata. Tagging of content in social web applications enables their organization and facilitates searching and formation of social networks for recommendation. Besides, no specific skills are needed to tag resources. The frequent use of these systems, as explained in 2.3.3, shows clearly that folksonomy-based approaches are able to overcome the knowledge acquisition bottleneck. However, some drawbacks in these systems avoid the semantic enrichment this thesis is looking for:

- Again, tags are described with natural language, so ambiguity or lack of synonym relations problems still last.
- The lack of consensus in the social community produces an inefficient tagging system.
- Another consequence of the previous item is that the internal structure of folksonomies suffers from bad organization.

Semantic web technologies avoid these issues. Semantic search has been proposed as an alternative to traditional syntax-based search in academia and industry (see section 2.4.4). However, it is not clear how to exploit their benefits without the necessity to train users in the domain of semantics and ontologies. Besides, these approaches tend to design and build new systems from scratch (crawler, indexer, etc.); thus, they do not exploit the information indexed and functionalities already presented in traditional web search engines.

## 4.2 Specific goals and solutions

This dissertation proposes a *user-support approach based on the collaborative sharing of semantic knowledge through Wikipedia to improve current web search engines*. In general, current mechanisms do not fulfil the problems to be solved. The solutions proposed in this thesis are due to tackle the set of goals established at the beginning of this dissertation.

> ***Goal 1:*** *The design and implementation of a data flow that allows collaborative 1) semantic annotations of resources without expertise knowledge about ontologies or other semantic techniques; and 2) filtering by explicit relevance feedback.*

The solution of this thesis has to minimize the problems of logs-based and social tagging models by making explicit the semantics for queries and web documents with concepts extracted from social annotations. The concepts to be used have to keep the following basic characteristics:

- Concepts used to annotate resources must be easily identified by a unique global identifier.
- Concepts must cover a great variety of domains.
- Concepts must be arranged in a structure which must be kept updated as constantly as possible, and by a large community.
- Concepts must be defined in different languages.

Folksonomies are more widely accepted for non-expert users, who have more freedom to create and use them. However, in terms of knowledge representation, the set of these keywords cannot even be considered as vocabularies, the simplest possible form of an ontology on the continuous scale of Smith & Welty (Smith & Welty, 2001).

In order to fulfil the goal, semantic annotations in this thesis are attached to the queries and documents by means of Wikipedia pages. As shown in previous chapters, Wikipedia vocabulary is an adequate source for annotations with advantages over unstructured folksonomies and other well-formed vocabularies like WordNet. The correct sense of an ambiguous word can be selected based on the context where it occurs, and this process is called word sense disambiguation. Most of the times, the number of query terms in web searches makes difficult to have a wide context to assign the appropriate meaning to those terms involved. An explicit disambiguation is then achieved with Wikipedia annotations.

There is an increasing interest in using Wikipedia as a linguistic source, and some works already saw their benefits for potential use in information retrieval and search (Damme, Hepp, & Siorpaes, 2007; Fernández García, Blázquez del Toro, José María, Sánchez Fernández, & Luque Centeno, 2006; Hepp, Siorpaes, & Bachlechner, 2007). Every conceptual entity in Wikipedia is represented in a particular web page or *article* with a unique identifier (URI). Its great coverage is another key factor of this vocabulary, which contains concepts of a huge variety of domains, like science, geography, history, etc., including proper nouns or very specific terminology in a variety of languages. Furthermore, as reflected in works like (Heflin & Hendler, 2000), the evolving nature of the information on the Web requires a continuous maintenance in the vocabulary used to annotate resources. In that sense, the open and simple editorial process of Wikipedia - compared to the formal development of ontologies or other vocabularies - makes it suitable for collaborative maintenance and rapid adaptation to information changes.

The annotation process has to be guided once users have entered the query, and before the list of results is shown. This annotation process has to be easy for the user, and natural language has to be employed, avoiding expert languages to express queries, concepts or annotations.

Finally, collaborative filtering is a key factor in the solution proposed here. The information of the significance of a web resource with respect to a query is given by users and must be stored, in order to make rankings of web documents based also in these opinions. The collaboration among multiple users is a way to improve the performance of information retrieval in web search. The filtering of documents by a user in one search can serve for the ranking of documents to another user in another search.

> ***Goal 2:*** *The design and implementation of a ranking algorithm that, along with traditional static and dynamic features existing in current web search algorithms, uses semantic annotations and social feedback information to provide more relevant results.*

This thesis proposes an unsupervised approach, because supervised machine-learning algorithms (like those based on neural networks or SVMs) usually require a large volume of training data. Semantic annotations indicate the relevance of a document given a query in two basic ways:

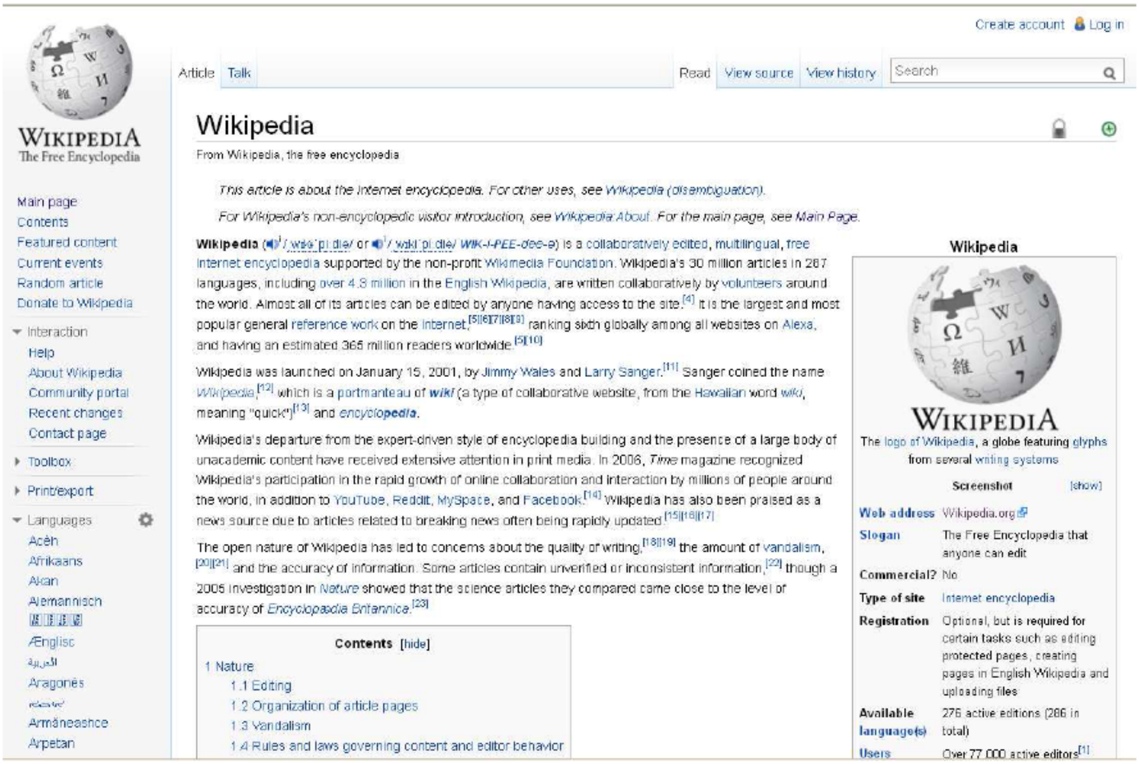
1. They provide another feature for the intrinsic static ranking of the underlying web search engine. A normalized count of the number of annotations a web page gets can indicate the relevance of that page.
2. Both the semantics associated to a query and the semantics associated to pages serve to indicate the relevance of documents in terms of similarity with queries, enhancing the dynamic ranking of the underlying web search engine.

Besides the fact that the solution proposed is unsupervised - no data is needed to train the model -, it does not require any specific context besides the concepts extracted from the keywords of a particular query.

The solution presented here comprises an algorithm that must incorporate dynamic and static characteristics from current search engines; that is, it must be easily coupled on top of traditional algorithms existing in web search engines in order to take advantage of their characteristics. Remember that there were algorithms in chapter 2 that were developed from scratch, without incorporating existing approaches which already obtained acceptable results.

> ***Goal 3:*** *The design and implementation of a semantic and domain-independent similarity algorithm that, given two semantic concepts, automatically determines a score that indicates their similarity at semantic level, in order to provide query expansion.*

The algorithm proposed as Goal 2 will have to know the *semantic similarity* between the concepts related to a query and the concepts related to a particular document, in order to determine if that document is relevant for that query. For that, an algorithm has to be developed in order to calculate the similarity between two concepts in Wikipedia.



Wikipedia

From Wikipedia, the free encyclopedia

*This article is about the Internet encyclopedia. For other uses, see Wikipedia (disambiguation).*

*For Wikipedia's non-encyclopedic visitor introduction, see Wikipedia:About. For the main page, see Main Page.*

**Wikipedia** (/ˌwɪkɨˈpiːdiə/ or /ˌwɪkiˈpiːdiə/ *WIK-i-PEE-dee-ə*) is a collaboratively edited, multilingual, free internet encyclopedia supported by the non-profit Wikimedia Foundation. Wikipedia's 30 million articles in 287 languages, including over 4.3 million in the English Wikipedia, are written collaboratively by volunteers around the world. Almost all of its articles can be edited by anyone having access to the site.[4] It is the largest and most popular general reference work on the Internet,[5][6][7][8][9] ranking sixth globally among all websites on Alexa, and having an estimated 365 million readers worldwide.[5][10]

Wikipedia was launched on January 15, 2001, by Jimmy Wales and Larry Sanger.[11] Sanger coined the name *Wikipedia*,[12] which is a portmanteau of ***wiki*** (a type of collaborative website, from the Hawaiian word *wiki*, meaning "quick")[13] and *encyclo**pedia***.

Wikipedia's departure from the expert-driven style of encyclopedia building and the presence of a large body of unacademic content have received extensive attention in print media. In 2006, *Time* magazine recognized Wikipedia's participation in the rapid growth of online collaboration and interaction by millions of people around the world, in addition to YouTube, Reddit, MySpace, and Facebook.[14] Wikipedia has also been praised as a news source due to articles related to breaking news often being rapidly updated.[15][16][17]

The open nature of Wikipedia has led to concerns about the quality of writing,[18][19] the amount of vandalism,[20][21] and the accuracy of information. Some articles contain unverified or inconsistent information,[22] though a 2005 investigation in *Nature* showed that the science articles they compared came close to the level of accuracy of *Encyclopædia Britannica*.[23]

**Wikipedia**

The logo of Wikipedia, a globe featuring glyphs from several writing systems

| Screenshot | [show] |
|---|---|
| **Web address** | Wikipedia.org |
| **Slogan** | The Free Encyclopedia that anyone can edit |
| **Commercial?** | No |
| **Type of site** | Internet encyclopedia |
| **Registration** | Optional, but is required for certain tasks such as editing protected pages, creating pages in English Wikipedia and uploading files |
| **Available language(s)** | 276 active editions (286 in total) |
| **Users** | Over 77,000 active editors[1] |

**Fig. 23. Wikipedia page about Wikipedia concept itself**

Each article in Wikipedia has a more or less fixed structure (see Fig. 23 for an example). It contains a title of the concept described, the first paragraph usually provides a brief definition of that term, and the remaining text further elaborates its content. It also offers a hierarchy of categories, and each concept can belong to one or more of these categories. Besides, there is information about polysemous concepts, through the so-called *disambiguation pages*.

However, Wikipedia is a work-in-progress project and, as such, it may contain errors, like duplicated entries or hyperlinks to Wikipedia concepts that have not been created yet, so finding an effective algorithm is not a trivial task. There are three basic Wikipedia factors that can be used in order to elaborate a semantic metric:

- The title of concepts and/or the first paragraph, after some clean-up, can be good candidates to be relevant terms in calculating similarity, with co-occurrence and matching techniques.
- Hyperlinks in Wikipedia pages are considered in existing works for comparing pairs of concepts.
- The structure of categories can be considered as a taxonomy and, as such, an algorithm based on lexical structures can be implemented.

As seen in section 3.2 from the state of the art, there are already algorithms that try to obtain a value from the comparison of two Wikipedia concepts. However, these algorithms are far from obtaining as good results as those where other sources are employed, like WordNet.

Taking this into account, this thesis proposes a set of steps in a general procedure that can be used to adapt semantic similarity metrics to use Wikipedia information. In particular, it proposes the usage of the Wikipedia categorization structure as an alternative to traditional lexical structures like WordNet. Henceforth, our analysis focuses on path-based and multi-source metrics, as these perform better than corpus-based metrics. The adapted measure with better results will be applied to the ranking algorithm of Itaca.

This dissertation will focus on semantic similarity - meaning associations - between concepts -, instead of semantic relatedness - general associations -.

## 4.3 Architecture overview

In order to attain the goals stated, this thesis proposes the development of a web layer, *Itaca*, which works on top of web search engines to improve the information provided to the final users. Itaca layer, mainly composed of designed algorithms and gathered data, must be easily settled on top of the architecture of current search engines (see Fig. 25).

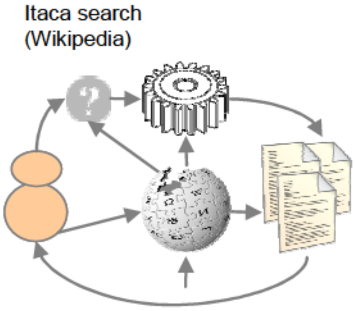


**Fig. 24. Graphical review of Itaca approach**

This layer extends the capabilities of traditional search engines and is based on the following principles:

- Collaborative tagging and filtering by means of semantic annotations and explicit feedback respectively.
- Disambiguation of query-word senses with Wikipedia to improve traditional searching models - which rely on keyword-based approaches to compare queries to documents -.
- Low response time in the results obtained, as online searches must be feasible.
- The design must cope with a huge amount of users and documents.

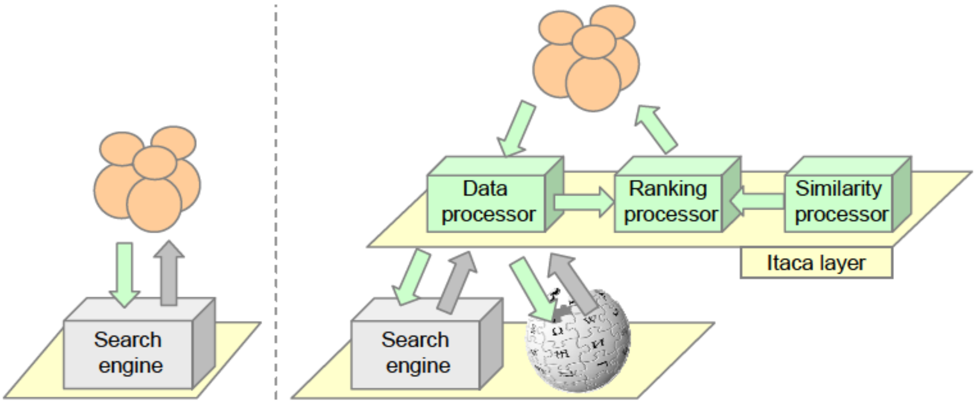


**Fig. 25. General overview without (left) and with (right) Itaca layer**

In Fig. 25, Itaca layer is mainly composed of three components:

- *Data processor*: Query input, query disambiguation, semantic annotations and other relevant feedback will take place in this part of the layer. The data gathered will serve to the other two components. This component is explained in chapter 5, and covers Goal 1.
- *Ranking processor*: With the semantic annotations and explicit feedback of users, the ranking algorithm will work in this component. It is explained in chapter 6 and covers Goal 2.
- *Similarity processor*: The ranking algorithm of the previous component will need a measure to determine the degree of similarity between queries and web documents at a semantic level. This similarity will be calculated by measuring the similarity between the concepts used to disambiguate queries and the concepts of documents potentially relevant to those queries. Then, an algorithm is needed to calculate the similarity between pairs of concepts, and it is developed in this component, which is explained in chapter 7 and covers Goal 3.

Two basic issues in any search engine are *quality* and *scalability* of results. In the present dissertation, these problems are resumed in *effectiveness* and *efficiency*; these aspects of the search results are crucial in order to satisfy final users:

- *Quality*: Effectiveness will be measured in terms of *relevance* of results.
- *Scalability*: Efficiency will be measured in terms of *response time*.

These two features will be measured at the end of the development of the present thesis to evaluate the overall solution.

## 4.4 HYPOTHESES

The hypotheses to be validated in this dissertation are the following.

> ***Hypothesis 1***. *It is feasible to improve current web search engines by means of the implementation of an independent layer on top of them with collaborative data gathering.*

This thesis considers it is feasible to implement a layer on top of current search engines to take advantage of both 1) traditional ranking algorithms and 2) new techniques based on collaborative data. This would allow incorporating an additional model instead of working on a new search engine from scratch. To prove this hypothesis, the final implementation of Itaca layer will be conducted. This will confirm the feasibility of the architecture proposed in this thesis.

> ***Hypothesis 2***. *Collaborative usage of semantic annotations in a search process, along with an appropriate ranking algorithm, produces 1) more relevant results than traditional web search engines; and 2) with a low response time.*

To prove Hypothesis 2, two types of evaluation will be considered, regarding two aspects respectively:

1. A first set of experiments will compare the relevance rate obtained with Itaca ranking algorithm and the relevance rate obtained with other well-known current search engines.
2. A second set of experiments will compare the response time obtained with Itaca ranking algorithm with different number of annotations, in order to see the variation (increment or decrement) in the time needed to obtain the final results.

***Hypothesis 3****. Wikipedia is a valid source to calculate semantic similarity. Its application in a semantic similarity method can yield as good results as existing techniques with WordNet and other knowledge sources.*

This thesis considers that the selection of the appropriate features Wikipedia offer - more specifically, its structure of categories - and their subsequent processing can allow the adaptation of path-based and multi-source metrics in the state of the art to obtain the semantic similarity of two entities, yielding the same or even better results than the original models with other knowledge sources. To prove this hypothesis, experiments will compare the correlation coefficient obtained with the adaption of the metrics implemented as Goal 3, and the correlation coefficient obtained with both existing techniques applied to Wikipedia and existing path-based and multi-source similarity methods applied to other knowledge sources like WordNet.

# 5 DATA PROCESSOR

This chapter is the first one devoted to describe the inner components of Itaca layer; more specifically, this chapter details the *Data processor* component. Query input, query disambiguation, semantic annotations and other relevant feedback will take place in this component of the layer. The data gathered will serve to the other components.

## 5.1 Overview

The part of the searching process used to collect user feedback is managed in the *Data processor* component of the Itaca layer. Fig. 26 shows a general view of the searching process flow, focusing on the steps in which data gathering (*Data processor* component) is divided: query definition (step 1), query disambiguation (step 2) and resources annotation (step 3).

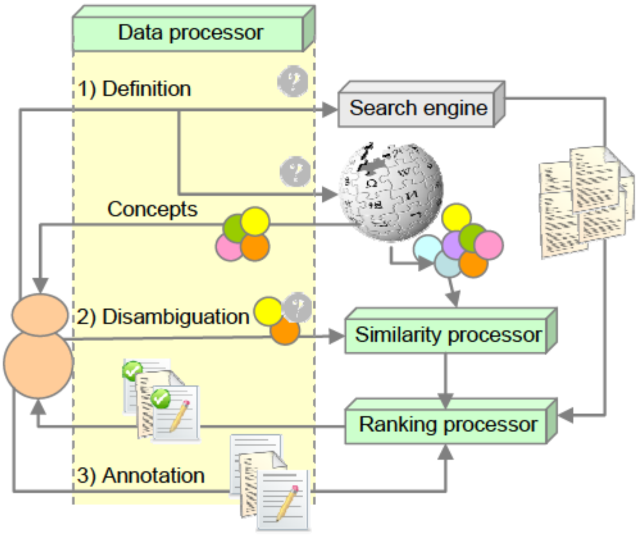


**Fig. 26. Searching process flow: *Data processor* component**

## 5.2 Query definition

Being *W* the set of possible words, any term *t* to be searched in a web search engine is composed of words and can be defined as:

$$t = \{w \in W\}$$

**Equation 27. Term definition**

A query can be formulated as:

$$q = \{t \in T\}$$

**Equation 28. Query definition**

Note that $q \subset T$, where *T* is the set of all possible terms. This simple model allows the definition of a query as a set of textual words in natural language, avoiding syntax based on complex semantics (ontologies, resource description languages, etc.).

An example of a query with two terms, each of them composed of one word, is $q_1$, where its goal is to search documents about president George Bush and the capital of Italy:

$$q_1 = \{\{\text{"Bush"}\}, \{\text{"Rome"}\}\}$$

The internal model stored is depicted in figure Fig. 27, where the query is composed of two terms, each one with a single word.

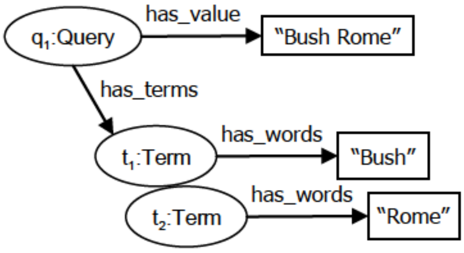


**Fig. 27. Example of query definition, information model**

> *For now on, graphical examples of information models will follow the same notation as Fig. 27. That is, objects will be displayed in oval circles, with the specific name of the object first, followed by a colon and the type of the object (the class). Textual or numerical attributes will be displayed in rectangles and, finally, relation among objects or objects and textual fields will be expressed by arrows.*

The flow of this step is depicted on the following sequence diagram:

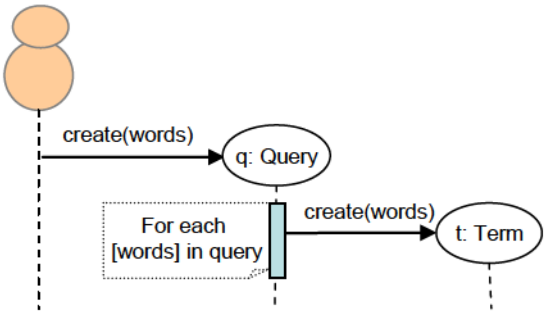


**Fig. 28. Query definition, sequence diagram**

> *In graphical examples of sequential diagrams, objects will be also displayed in oval circles, with the specific name of the object first, followed by a colon and the class of the object (the class). In traditional UML notation, these objects are represented with rectangles. The form has been changed to be consistent with the notation applied in the information model in this thesis.*

Once the query is defined and executed, the semantic enrichment of query terms and web resources have to be resolved. This is achieved by both the query disambiguation and the resources annotation processes respectively.

## 5.3 Query disambiguation

Query $q$ can be disambiguated by means of the disambiguation of its related terms:

$$d(q) = \{(t,c) \in q \times C\}$$

**Equation 29. Disambiguated query definition**

Being *C* the set of Wikipedia concepts, *t* represents a term of the original query *q* and *c* represents the particular Wikipedia concept that term has been disambiguated to. So, the query is identified with the most suitable sense of each of its terms.

> *This is the process of semantic annotation of queries. This process addresses the problems due to the natural language used in queries. Users can ommit this step, though, because it is not mandatory for continuing with the searching process.*

Consider the following concepts, retrieved through Wikipedia when searching for the term words “Bush” and “Rome”:

$$c_1 = \text{"en.wikipedia.org/wiki/George\_W\_Bush"}$$
$$c_2 = \text{"en.wikipedia.org/wiki/George\_H\_W\_Bush"}$$
$$c_3 = \text{"en.wikipedia.org/wiki/Rome"}$$
$$c_4 = \text{"en.wikipedia.org/wiki/Rome\_Georgia"}$$

Then, a possible example of the disambiguation of $q_1$ is:

$$d(q_1) = \{(\{\text{"Bush"}\}, c_1), (\{\text{"Rome"}\}, c_3)\}$$

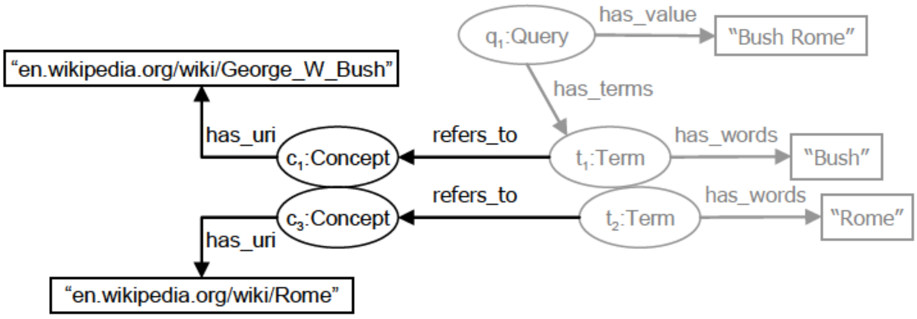


**Fig. 29. Example of query disambiguation, information model**

The flow of this process is depicted in the following figure:

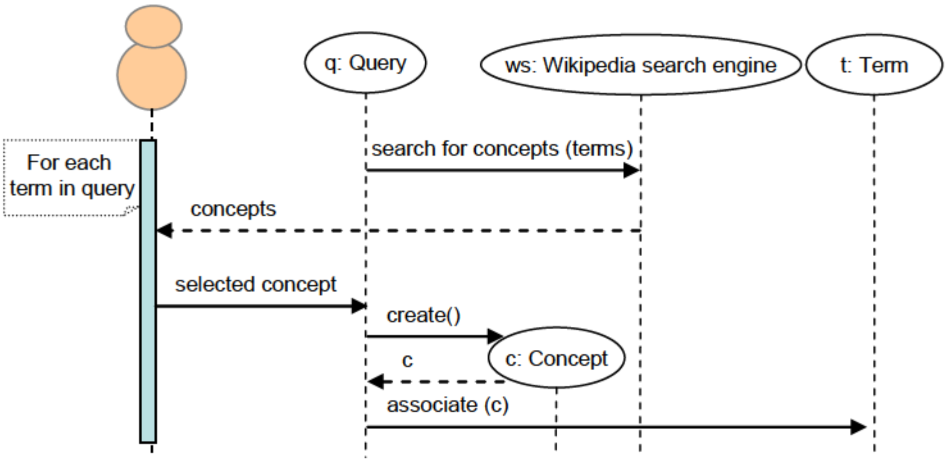


**Fig. 30. Query disambiguation, sequence diagram**

$C_{d(q)}$ is the set of concepts involved in *d(q)*, so that:

$$C_{d(q)} = \bigcup_{\forall(t,c)\in d(q)} \{c\}$$

**Equation 30. Set of concepts of a disambiguated query**

These concepts are used for the ranking algorithm to find relevant resources previously annotated with these or similar concepts. Going back to the example:

$$C_{d(q_1)} = \{c_1, c_3\}$$

## 5.4 RESOURCES ANNOTATION

When web resources (pages returned by the ranking algorithm with the help of the underlying web search engine) are presented to users, they can consider these results relevant or not to the original query.

> *This process is the collaborative semantic filtering of resources, and takes place if the user has disambiguated the query. Queries and web documents can be semantically annotated, indicating the relevance of the documents with respect to the concepts involved in the query. This is done by users in their searches, so annotations from one user are the input for the ranking in another user's search.*

Given a disambiguated query *d(q)*, users can associate a particular resource *r* with a set of annotations, $AN(r)_{d(q)}$, where:

$$AN(r)_{d(q)} = \{(t, c, score) \in q \times C \times \{-1, 0, 1\}\}$$

**Equation 31. Resource annotation definition**

A web resource *r* can be considered semantically relevant or not to a concept of the formulated query. For that, a score of -1 indicates the resource has nothing to do with the concept, 1 indicates the opposite, and 0 indicates user does not know or does not care about it.

Consider query $q_1$ again, "Bush Rome", and its disambiguation, *d(q₁)*, which searches for events in Rome about George W. Bush. Then, $r_1$ can be a possible result of a traditional search engine when query $q_1$ is executed:

$$r_1 = \text{"http://www.youtube.com/watch?v = AzJoRGTKuOE"}$$

Where $r_1$ represents a video of George W. Bush's limousine getting stuck in Rome. User can then indicate this resource is completely related to "Bush" and "Rome", adding a couple of annotations to the set:

$$AN(r_1)_{d(q_2)} = \{(\{\text{"Bush"}\}, c_1, 1), (\{\text{"Rome"}\}, c_3, 1)\}$$

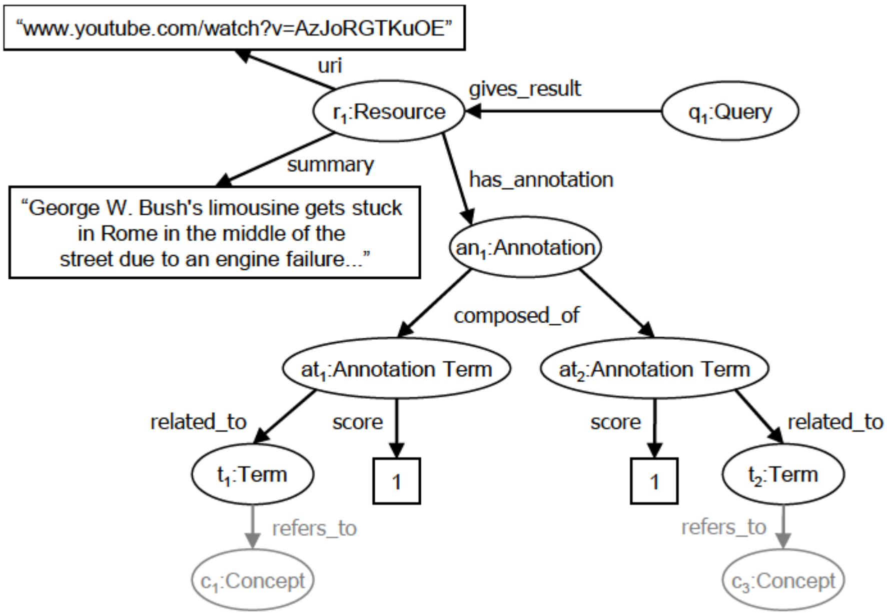


**Fig. 31. Example of resources' annotation, information model**

The process flow for this step is depicted on Fig. 32.

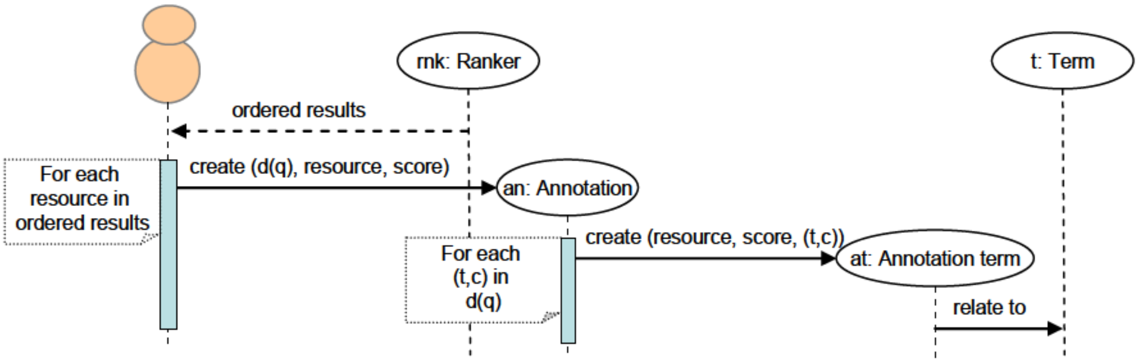

rnk: Ranker
t: Term
ordered results
For each resource in ordered results
create (d(q), resource, score)
an: Annotation
For each (t,c) in d(q)
create (resource, score, (t,c))
at: Annotation term
relate to

**Fig. 32. Resources annotation, sequence diagram**

# 6 RANKING PROCESSOR

This chapter details the *Ranking processor* component. With the semantic annotations and explicit feedback from users, gathered at the previously explained *Data Processor* component, the ranking algorithm is computed in this component.

## 6.1 OVERVIEW

Fig. 33 shows the entire searching process flow, focusing on the *Ranking processor* component and the results obtained through it, which are detailed in next sections. Chapter 7 offers further details about the *Similarity processor* component.

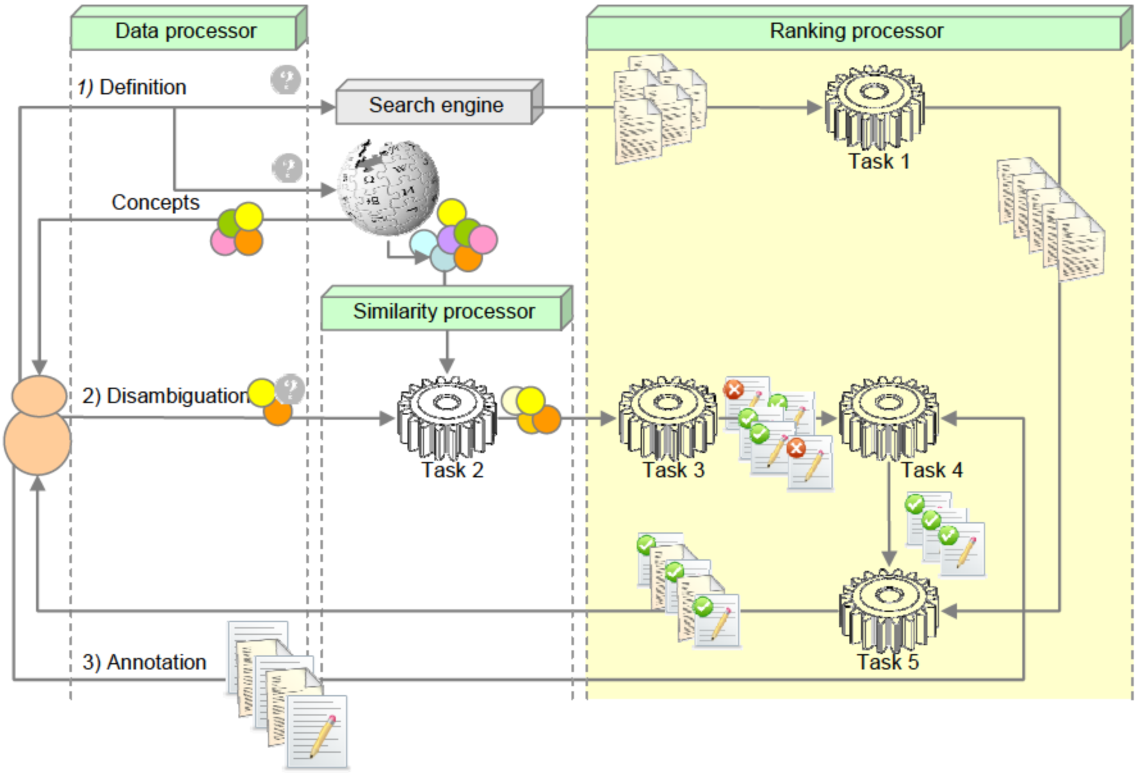


**Fig. 33. Searching process flow: *Ranking processor* component**

The *Ranking processor* component is in charge of the ranking algorithm, which combines two sources of information: a set of documents obtained by a traditional web search engine and a set of documents with semantically disambiguated annotations provided by users in the collaborative semantic filtering procedure explained in the previous chapter. This dissertation assumes the ranking value of a resource is a function that combines both the value obtained from a traditional web search engine and the value that Itaca layer estimates with users' feedback.

The ranking is computed in a process composed of five tasks (see Fig. 33). In task 1, after the formulation of the query in the traditional web search engine, a value (*web value*) is computed for every resource retrieved. Task 2 finds concepts with high semantic similarity to those involved in the query, whereas task 3 finds the set of resources annotated with any of these concepts. Task 4 calculates a second value (*annotation value*) for each resource obtained in task 3. Finally, task 5 combines the resources and the values obtained in task 1 and 4 to produce the final ranking.

The searching process does not end here, because the results returned can again be semantically annotated in the collaborative filtering of resources' annotation.

## 6.2 Task 1: Web values

After executing a query *q* in a web search engine, a set of web resources, $R_q = \{r\}$, are obtained. The number of resources obtained is limited to *s*, a configurable parameter, so that $|R_q| \leq s$. The web value (*web_val*), of a resource $r \in R_q$, ranging from 0 to 1, is calculated using the ranking of the resource in the results of the web search engine:

$$\text{web\_val}(r) = \begin{cases} 2^{-index(r)/x}, & \text{if } r \in R_q \\ 0 & \text{, if } r \notin R_q \end{cases}$$

**Equation 32. Web value function for a resource**

*web_val(r)* is a monotonically decreasing function of the position (*index*) of *r* inside $R_q$. As *index* decreases to 0 (an index of 0 represents the first position in the ranking), *r* is most relevant and *web_val(r)* increases to 1. *x* is a configurable parameter that represents the position in $R_q$ where resources become less relevant; results after position *x* are considered to have less impact in the final ranking, following a nonlinear function.

Taking as example the query “Sun”, the first 20 resources returned and its web value, considering *x* = 15, are the following:

**Table 4. First 20 results of query "Sun" and their web values**

| Position | Rq URI | Web value |
|---|---|---|
| 0 | The Sun \| The Best for News, Sport, Showbiz, Celebrities | 1,00 |
| 1 | Oracle and Sun Microsystems \| Strategic Acquisitions | 0,95 |
| 2 | Oracle España \| Hardware and Software, Engineered to… | 0,91 |
| 3 | SUN - Wikipedia, la enciclopedia libre | 0,87 |
| 4 | Sun - Wikipedia, the free encyclopedia | 0,83 |
| 5 | Sun Microsystems - Wikipedia, la enciclopedia libre | 0,79 |
| 6 | Sun Microsystems - Wikipedia, the free encyclopedia | 0,75 |
| 7 | The Sun - Wikipedia, la enciclopedia libre | 0,72 |
| 8 | Descarga gratuita de software de Java | 0,69 |
| 9 | java.com: Java y Tú | 0,65 |
| 10 | Sun Channel | 0,62 |
| 11 | Sun-Hwa Kwon - ES – Lostpedia | 0,60 |
| 12 | Sun — simple weather app – Pattern | 0,57 |
| 13 | Guardian Sun. Cristal inteligente. | 0,54 |
| 14 | Sun - Universidad de Navarra | 0,52 |
| 15 | Sun Record Company \| Where Rock & Roll Was Born | 0,50 |
| 16 | Welcome. The Official Site for Sun Studio. The Birthplace… | 0,47 |
| 17 | Techno Sun - Energía solar fotovoltaica - Paneles solares... | 0,45 |
| 18 | SUN RECORDS ··· Tu Tienda de Metal ··· | 0,43 |
| 19 | Sun Ringle | 0,41 |

The formula is inspired by studies (Baeza-Yates, Hurtado, Mendoza, & Dupret, 2005) that show the frequency of web results selected by users and the position of these results in the selected web search engine follow a similar shape (see Fig. 34).

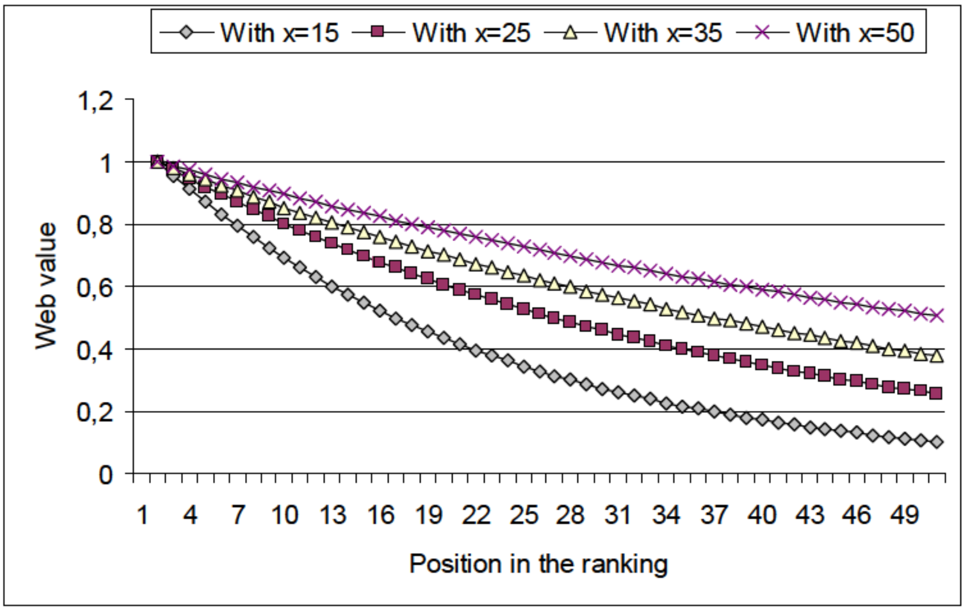


**Fig. 34. Web values for different *x* values of the first fifty ordered results in any search**

## 6.3 TASK 2: SIMILAR CONCEPTS

After querying, a set of web resources are recovered from the search engine in Task 1. However, semantically similar queries can have been executed previously, obtaining other web resources that may not appear in the current session query (usually, because the terms used to formulate the current and previous queries are different). The goal of disambiguating the queries is to obtain relevant resources that are not offered by the web search engine, through a process called *query expansion*. First, an prior to the querying, a disambiguated query *d(q)* is associated to one or more Wikipedia concepts (see section 5.3). Then, semantically similar concepts can be easily obtained, in order to recover web resources associated to them.

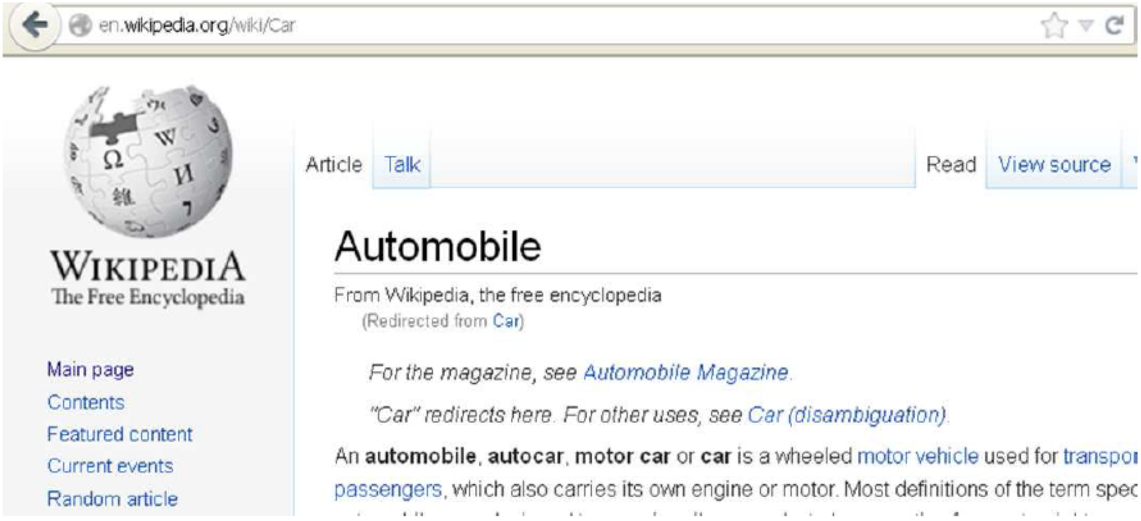
To estimate the similarity of two concepts, a function has been designed. It is based on a traditional similarity measure and the categorization schema of categories in Wikipedia, $sim_{li\_max\_avg}$. Chapter 7 explains the inner details of the procedure used to elaborate this measure. The function interval is [0, 1], where 0 means no similarity at all. There is no need to compute similarity when the result is known for sure. This is the case of Wikipedia pages that represent the same concept but in different languages, or in the case of redirection pages (see Fig. 35); in both cases, the similarity is set to 1.

Given a disambiguated query *d(q)*, and the set of concepts $C_{d(q)} \subset C$ used to disambiguate its terms - being *C* the set of Wikipedia concepts -, query expansion begins. Its goal is to find resources related not only with concepts in $C_{d(q)}$, but also related with concepts semantically similar to those in $C_{d(q)}$, the set $C'_{d(q)}$, where:

$$C'd(q) = \cup_{\forall c \in C_{d(q)}} \{d \in C \mid \mathrm{sim}_{\mathrm{li\_max\_avg}}(c,d) \geq \mu\}$$

**Equation 33 Semantically similar concepts**

$\mu \in [0, 1]$ is the threshold to consider a concept *c* semantically similar to another concept *d*.



en.wikipedia.org/wiki/Car

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article

Article Talk Read View source

Automobile

From Wikipedia, the free encyclopedia
(Redirected from Car)

*For the magazine, see Automobile Magazine.*

*"Car" redirects here. For other uses, see Car (disambiguation).*

An **automobile**, **autocar**, **motor car** or **car** is a wheeled motor vehicle used for transpo passengers, which also carries its own engine or motor. Most definitions of the term spec

**Fig. 35. Example of a redirection page, from *Car* to *Automobile***

The process flow of task 1 and task 2 is depicted on Fig. 36, where these tasks are put in context with the sequential steps followed in the *Data processor* component seen in the previous chapter.

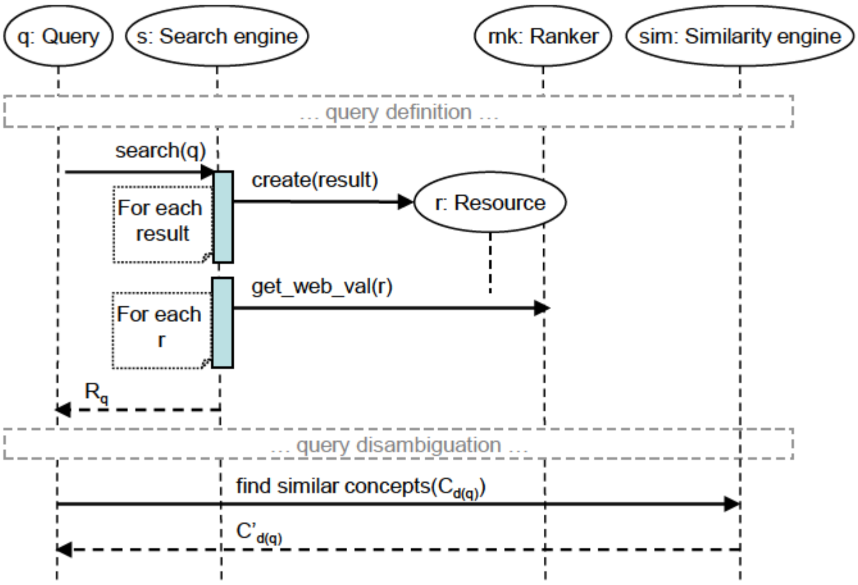


**Fig. 36. Task 1 and task 2, sequence diagram**

## 6.4 Task 3: Relevant resources from user annotations

In this task, the algorithm searches the set of resources that were annotated with any of the concepts, either in $C_{d(q)}$ or $C'_{d(q)}$ (see section 5.4 for details about annotation of resources). The annotation set can be enormous and its computation cost may be high, as happened in some works exposed in Chapter 2, like *FolkRank* or *SocialPageRank*. A subset of resources could be selected (e.g., those with the most recent annotations), but this would reduce the whole working space and final results offered to users could be inaccurate.

In order to solve this problem, and following an item-based approach which, as seen previously, gives better results than user-based approaches, the algorithm in Itaca layer makes use of *accumulators* to obtain a summary of which resources were annotated with which concepts. The set of accumulators is defined as:

$$AC = \{(r, c, rel, ind, unrel) \in S \times C \times N \times N \times N\}$$

**Equation 34 Accumulators set**

*S* is the set of web pages indexed by the web search engine considered in Itaca layer and *N* is the set of natural numbers. Value *rel* indicates the number of times *r* has been set as related to *c*; *unrel* indicates the opposite, and *ind* indicates the number of times a user did not know or did not care about its relatedness. Every time a user makes an annotation about a particular resource *r* that involves a particular concept *c*, the corresponding accumulator is updated.

For example, consider $r_1$ and $c_1$ again:

$$r_1 = \text{"http : //www.youtube.com/watch?v = AzJoRGTKuOE"}$$
$$c_1 = \text{"es.wikipedia.org/wiki/George\_W\_Bush"}$$

One possible accumulator can be:

$$ac_1 = (r_1, c_1, 6900, 3000, 100)$$

In this case, the web resource $r_1$ has been annotated with concept $c_1$ 10000 times. In 3000 annotations, users did not know/care about the relatedness; 100 annotations state that $r_1$ had nothing to do with $c_1$ and was annotated as unrelated; finally, 6900 annotations indicate that $r_1$ was indeed relevant for $c_1$ (see Fig. 37).

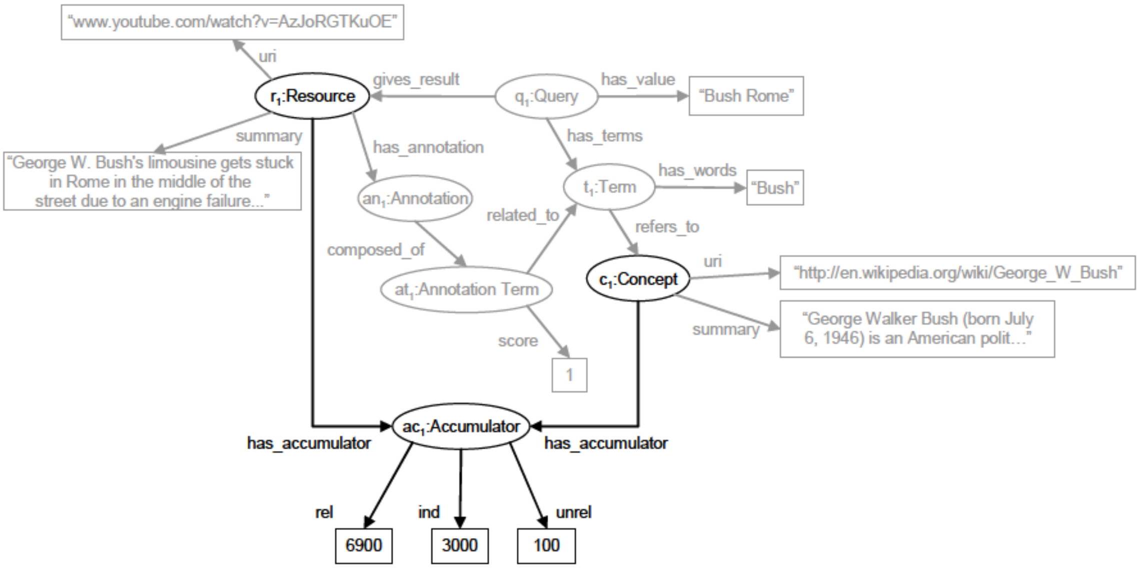


**Fig. 37. Example (I) of accumulators, information model**

In a particular query session, the algorithm uses the subset $AC_{d(q)} \in AC$, where:

$$AC_{d(q)} = \bigcup_{\forall(r,c,rel,ind,unrel)\in AC} \{(r,c,rel,ind,unrel) | c \in (C_{d(q)} \cup C'_{d(q)})\}$$

**Equation 35 Accumulators set of a given disambiguated query**

This way, the algorithm obtains the resources annotated with any of the concepts implied, by means of the annotation's accumulators.

Consider the following query and its disambiguated form in this second example:

$$q_2 = \{\{\text{"Rodrigo"}, \text{"Rato"}\}, \{\text{"teatro"}\}\}$$
$$c_5 = \text{"es.wikipedia.org/wiki/Rodrigo\_Rato", an Spanish politician}$$
$$c_6 = \text{"es.wikipedia.org/wiki/Teatro", the art of performing}$$
$$d(q_2) = \{(\{\text{"Rodrigo"}, \text{"Rato"}\}, c_5), (\{\text{"teatro"}\}, c_6)\}$$

Now, consider the description of the following web resources:

$$r_2 = \text{"Telefonica appoints Rodrigo Rato to advisory boards"}$$
$$r_3 = \text{"Fraud trial for Rodrigo Rato over Bankia collapse"}$$
$$r_4 = \text{"Rodrigo Rato rehearses "Don Mendo's Revenge" with a group of Spanish actors in Washington"}$$

The accumulators related to the disambiguated query can be (see Fig. 38 ):

$$AC_{d(q_2)} = \{(r_2, c_5, 1000, 50, 10), (r_3, c_5, 700, 40, 5), (r_4, c_5, 650, 40, 2), (r_4, c_6, 650, 50, 10)\}$$

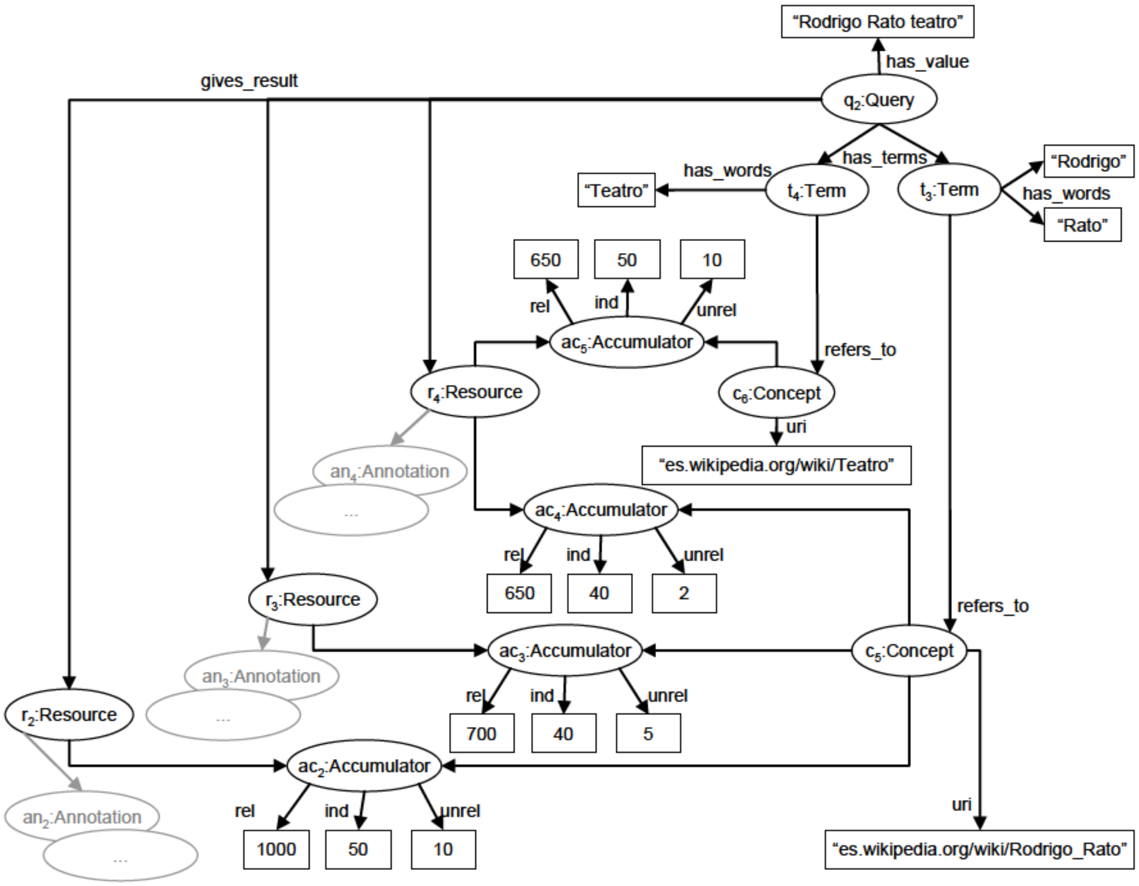


**Fig. 38. Example (II) of accumulators, information model**

Every single annotation will be stored (see the faded objects of type *Annotation* in Fig. 38). However, as each annotation is created, the particular accumulator associated is updated. At the end, the ranking algorithm will operate with these accumulators instead of the single annotations. This fact avoids four original problems of other works:

- The quality of a web page is stated by the reader, instead of the web designer.
- The algorithm is given the whole context surrounded the resource and its associated concept; that is, it counts with every annotation stored - by means of their accumulators - instead of with a small particular set (the set of the last annotations, the set of the most annotated resources, etc.).
- The quality of a web page can be explored no matter its structure (a text page with hyperlinks, an image file with no hyperlinks, etc.).
- As accumulators offer the summary of the context, the response time to calculate the ranking algorithm can be presumably low.

The set containing the different web resources associated to a set of accumulators from a disambiguated query *d(q)* is $R_{d(q)}$ (independent of how relevant, irrelevant or indifferent they are with respect to the query):

$$R_{d(q)} = \bigcup_{\forall (r,c,rel,ind,unrel) \in AC_{d(q)}} \{r\}$$

**Equation 36 Resources set of a given disambiguated query**

In the example, the set of resources implied in *d(q₂)* are:

$$R_{d(q_2)} = \{r_2, r_3, r_4\}$$

The simple process flow is depicted on Fig. 39.

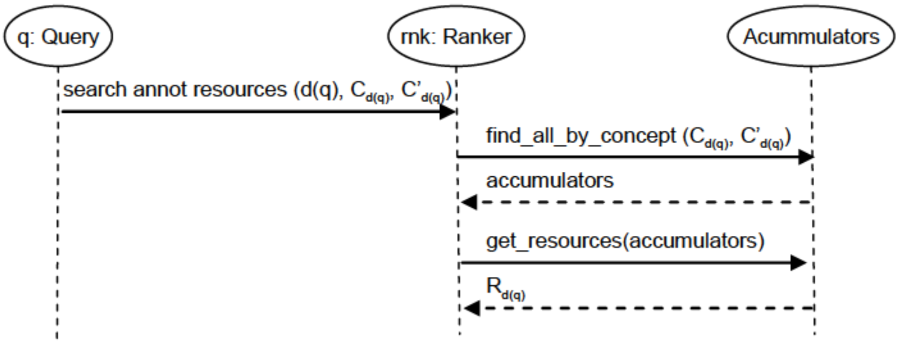

q: Query
rnk: Ranker
Acummulators
search annot resources (d(q), C_d(q), C'_d(q))
find_all_by_concept (C_d(q), C'_d(q))
accumulators
get_resources(accumulators)
R_d(q)

**Fig. 39. Task 3, sequence diagram**

## 6.5 Task 4: Annotation values

Task 1, given a query *q*, yields a particular value for every resource *r* returned by the web search engine. This value, *web_val(r)*, was obtained from the position of *r* in the returned list of results, $R_q$.

Now, Task 4 calculates another value, *annotation value*, which represents the relevance of a resource in $R_{d(q)}$ given its associated annotations. Notice that if a resource is not in the set of $R_{d(q)}$, that means that no annotation is associated yet to this resource and its annotation value will be 0. The function to get this value, *annot_val(r)*, ranges from -1 to 1 and is defined as:

$$\text{annot\_val}(r) = \begin{cases} \dfrac{\text{annotations\_score}(r)}{\text{annotations\_number}(r)} \times \text{sim}_{\text{total}}(C_{d(q)}, C^{r}_{d(q)}), \text{ if } r \in R_{d(q)} \\ 0 \qquad\qquad , \text{ if } r \notin R_{d(q)} \end{cases}$$

**Equation 37 Annotation value function for a resource**

Three functions are involved: *annotations_score*, *sim$_{total}$* and *annotations_number*.

First, the function *annotations_score(r)* represents the total score of a resource *r* given its annotations, calculated as the the sum of the total score of its associated accumulators. Just the annotations related to the concepts involved in $AC_{d(q)}$ are considered:

$$\text{annotations\_score}(r) = \sum_{\forall(res,c,rel,ind,unrel)\in AC_{d(q)}|res=r} (rel \times 1 + unrel \times (-1))$$

**Equation 38 Total score of a resource given its annotations**

This sum is calculated as the weighted sum of *rel* and *unrel* for every accumulator, where *rel* value is multiplied by 1 and *unrel* value is multiplied by -1.

The annotations' score obtained is weighted with the similarity of the concepts involved in those annotations. $C^{r}_{d(q)}$ is the set of concepts related to a particular resource *r* through its accumulators.

$$C^{r}_{d(q)} = \bigcup_{\forall(res,c,rel,ind,unrel)\in AC_{d(q)}|res=r} \{c\}$$

**Equation 39 Concepts set of a particular resource through its accumulators**

Considering the example of *d(q$_2$)* seen so far, there exist the following set of concepts:

$$C^{r_2}_{d(q_2)} = \{c_5\}$$
$$C^{r_3}_{d(q_2)} = \{c_5\}$$
$$C^{r_4}_{d(q_2)} = \{c_5, c_6\}$$

Second, *sim$_{total}$* is the total similarity between the concepts of a disambiguated query, $C_{d(q)}$, and the concepts belonging to the set of annotations of the resource, $C^{r}_{d(q)}$. This function, given two set of concepts $C_1$ and $C_2$, operates as follows:

$$\text{sim}_{total}(C_1, C_2) = \frac{\sum_{\forall c_1 \in C_1} \forall c_2 \in C_2 \max\{\text{sim}_{\text{li\_max\_avg}}(c_1, c_2)\}}{|C_1|}$$

**Equation 40 Semantic similarity of two set of concepts**

In (Haase & Siebes, 2004), authors applied the same equation in an answering peer-to-peer system to calculate the similarity among ACM categories, in which the set of categories of the query are compared in similarity with the set of expertise categories of the peers involved in the community. The only change in Itaca is that the function to calculate the similarity between two single concepts $c_1$ and $c_2$ is a particular function called $sim_{li\_max\_avg}$, instead of the measure used by the authors for hierarchical structured semantic networks. This similarity function will be explained in chapter 7.

Suppose the similarities among $c_5$ and $c_6$ are the following:

$$\text{sim}_{\text{li\_max\_avg}}(c_5, c_5) = 1$$
$$\text{sim}_{\text{li\_max\_avg}}(c_5, c_6) = 0.02$$
$$\text{sim}_{\text{li\_max\_avg}}(c_6, c_6) = 1$$

Then, the similarity between the set of concepts in $d(q_2)$ and the set of concepts of resource $r_3$ and $r_4$ is obtained as follows:

$$\text{sim}_{total}(C_{d(q_2)}, C^{r_3}_{d(q_2)}) = \frac{\max\{\text{sim}_{\text{li\_max\_avg}}(c_5, c_5)\} + \max\{\text{sim}_{\text{li\_max\_avg}}(c_6, c_5)\}}{2} = \frac{1+0.02}{2} = 0.51$$

$$\text{sim}_{total}(C_{d(q_2)}, C^{r_4}_{d(q_2)}) = \frac{\max\{\text{sim}_{\text{li\_max\_avg}}(c_5, c_5), \text{sim}_{\text{li\_max\_avg}}(c_5, c_6)\}}{2} + \\ + \frac{\max\{\text{sim}_{\text{li\_max\_avg}}(c_6, c_5), \text{sim}_{\text{li\_max\_avg}}(c_6, c_6)\}}{2} = \frac{1+1}{2} = 1$$

Logically, comparing the two concepts implied in the disambiguated query $d(q_2)$ with the concepts used to annotate both $r_3$ and $r_4$ yields different results. Similarity for $r_4$ is maximal, because the set of concepts for $r_4$ are the same as the concepts in the disambiguated query. However, just one of the two concepts in $d(q_2)$ is present in the annotations for $r_3$, so similarity between the sets is just of nearly a 50%.

Third, the total number of annotations of a resource is calculated with another simple function, *annotations_number*, in order to normalize the annotations score obtained previously:

$$\text{annotations\_number}(r) = \sum_{\forall (res,c,rel,ind,unrel) \in AC_{d(q)} | res=r} (rel + ind + unrel)$$

**Equation 41 Total number of annotations of a resource**

Focusing again on two of the resources of the previous example and their related accumulators:

$$r_3 = \text{"Fraud trial for Rodrigo Rato over Bankia collapse"}$$
$$r_4 = \text{"Rodrigo Rato rehearses "Don Mendo's Revenge" with a group of Spanish actors in Washington"}$$

$$\{(r_3, c_5, 700, 40, 5), (r_4, c_5, 650, 40, 2), (r_4, c_6, 650, 50, 10)\}$$

Values for *annotations_score* and *annotations_numbers* are the following:

$$\text{annotations\_score}(r_3) = (700 \times 1 + 5 \times (-1)) = 695$$
$$\text{annotations\_}\mathit{number}(r_3) = (700 + 40 + 5) = 745$$
$$\text{annotations\_score}(r_4) = (650 \times 1 + 2 \times (-1)) + (650 \times 1 + 10 \times (-1)) = 1288$$
$$\text{annotations\_}\mathit{number}(r_4) = (650 + 40 + 2) + (650 + 50 + 10) = 1402$$

Therefore, their final annotation values are:

$$\text{annot\_val}(r_3) = \frac{695 \times 0.51}{745} = 0.47$$
$$\text{annot\_val}(r_4) = \frac{1288 \times 1}{1402} = 0.91$$

## 6.6 TASK 5: RANKING RESULTS

For every resource $r \in R_q \cup R_{d(q)}$, this task combines the value obtained from the traditional web search engine, *web_val(r)* (Task 1), and the value obtained from users annotations, *annot_val(r)* (Task 4). The result of this combination is a new value, *final_val(r)*, whose function ranges from -1 to 1 and it is defined as follows:

$$\text{final\_val}(r) = \alpha \times \mathit{annot\_val}(r) + (1 - \alpha) \times \mathit{web\_val}(r)$$

**Equation 42 Final ranking value of a resource**

Final web resources will be sorted by this final value (from the highest to the lowest value). Constant $\alpha$ can be adjusted depending on user necessities or the annotations status. For example, if there are just a few annotations stored, $\alpha$ can be set to a minimum value in order to avoid sparse data.

Notice that, if there is no annotation in Itaca layer, *annot_val* function for every resource will be 0. The algorithm in this case will return results in the same order they were returned by the web search engine; that is, the order established by the web values of every resource, *web_val*. There were existing works that implemented new search engines from scratch. With the ranking algorithm proposed in this thesis, no matter if no annotation has been already done for a particular resource or concept, because the results will be still ordered by the ranking algorithm of the underlying engine.

The process flow of task 4 and task 5 can be resumed in the following sequence diagram:

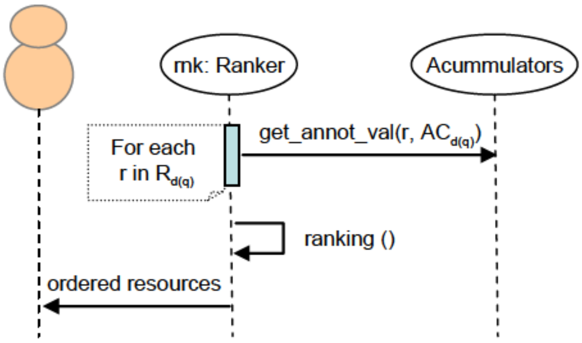


**Fig. 40. Task 4 and task 5, sequence diagram**

# 7 SIMILARITY PROCESSOR

The ranking algorithm of *Ranking processor* needs a measure to determine the degree of similarity between queries and web documents at a semantic level. This similarity is calculated by measuring the similarity between the concepts used to disambiguate queries and the concepts of documents potentially relevant to those queries. Then, an algorithm is needed to calculate the similarity between pairs of Wikipedia concepts, and it is implemented in this component.

## 7.1 OVERVIEW

Table 3 in section 3.3 showed the Pearson correlation coefficient reported for the most relevant semantic similarity measures for the test set.

The measures with values higher than 0.8 are methods based on hierarchical sources like WordNet (path-based and multi-source measures). Web-based and Wikipedia-based methods have a broader coverage than the rest of measures, because they have the World Wide Web or Wikipedia as information sources, but their results are not better than some of the simple path-based models. Coefficients higher than 0.8 are obtained from path-based and multi-source methods and most of them have been developed and evaluated for WordNet taxonomy. However, these methods using taxonomies or dictionaries suffer from several drawbacks that make their use in Itaca difficult to apply:

- They cannot be used in a web search engine, which requires a great coverage of the real world; for example, words such as some proper nouns ("Angela Merkel") or specific terminology ("hyperpolarization") are not defined in WordNet.
- Creating or modifying words in existing traditional corpora is managed slowly in time.
- Most of these sources are built just in English, and metrics that perform well cannot be used in other languages.
- These approaches measures similarity between words, and not word senses.

Wikipedia, though, solves these drawbacks by providing a vast knowledge for computing semantic similarity between word senses. Since 2006, there are multitudes of works which confirm Wikipedia as a faithful and complete source in a wide variety of applications in areas of Computational Linguistics and Artificial Intelligence, such as disambiguation of words (C. Li, Sun, & Datta, 2011), text annotation (N. Fernández, Fisteus, Fuentes, Sánchez, & Luque, 2011; Makris, Plegas, & Theodoridis, 2013) or text classification (P. Jiang et al., 2013). Its main facilities are:

- It offers concepts from a great variety of domains, like science, geography, etc.
- Its information is constantly updated by a large community.
- Its contents have been translated to numerous languages.
- Concepts belonging to different parts of speech are located under the same structure, whereas in many vocabularies, like WordNet, they are separated (nouns with nouns, verbs with verbs), which makes it difficult to analyse their similarity.
- Wikipedia concepts represent particular word senses, and not mere terms, an issue important to calculate semantic similarity, foreseen in (Resnik, 1995).

As expressed in (Strube & Ponzetto, 2006), the strength of Wikipedia lies in its size; however, despite its advantages, the size itself is also a disadvantage; the search space in the Wikipedia category graph is very large in terms of depth, branching factor and multiple inheritance relations, which create problems related to finding efficient mining methods. Besides, the category relations cannot be interpreted only as hyponym associations (*is-a-type-of* relations) of well-formed taxonomies. These characteristics have to be considered prior to the formalization of a semantic similarity metric.

This dissertation *exploits Wikipedia as a valid semantic source to compute semantic similarity between two identified concepts. In particular, as metrics based on lexical structures yielded better results, this dissertation proposes the structure of categories in Wikipedia to be applied to those existing metrics. The Wikipedia structure features will be adapted with diverse techniques to the most important path-based and multi-source measures*.

## 7.2 CHARACTERISTICS OF WIKIPEDIA STRUCTURE

The categorization schema in Wikipedia has the form of a directed cyclic graph; it is not a hierarchical acyclic structure as WordNet. Due to this aspect, potential problems can arise, mainly: selection of a *root node*, *cycles*, and *multiple inheritance*.

Fig. 41 shows an extract of the top level of the structure of categories in Wikipedia[22]. The root node is *Cat: Contents*. This category groups every page type in Wikipedia in a variety of forms. Among its children, *Cat: Articles* divides Wikipedia pages by content. Other subcategories under *Cat: Contents* distribute articles by administrative characteristics like their state.

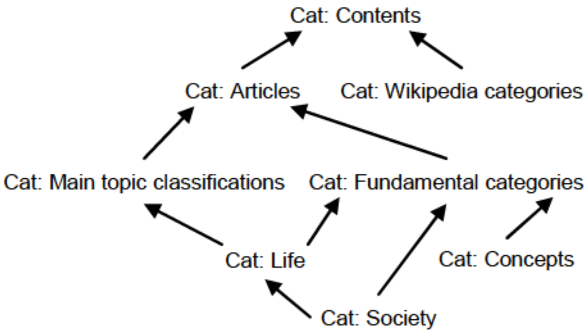

Cat: Contents
Cat: Articles
Cat: Wikipedia categories
Cat: Main topic classifications
Cat: Fundamental categories
Cat: Life
Cat: Concepts
Cat: Society

**Fig. 41. Top levels of Wikipedia categories structure**

In order to facilitate further processing, a single root node has to be considered. Below *Cat: Articles*, *Cat: Fundamental categories* distributes the articles in a more logic and progressive way than the rest of subcategories, which make merely a division by main broad topics. Therefore, this thesis considers *Cat: Fundamental categories* as the actual root node for the categorization scheme.

An example of existing cycles can be seen in Fig. 42, where categories *Cat: Coastal geography*, *Cat: Coasts* and *Cat: Coastal and oceanic landforms* form a cycle. These cycles have to be considered when processing the structure, in order to avoid loops.

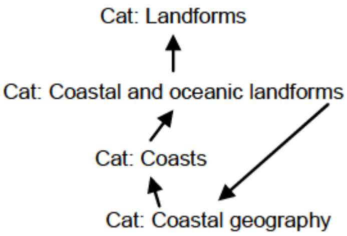


**Fig. 42. Extract of a cyclic subgraph**

Multiple inheritance among categories and concepts coexists in Wikipedia. The first form of multiple inheritance involves categories. Fig. 43 shows an extract where *Cat: Fruit* has 3 different parents.

[22] Categories are identified with the prefix *Cat:* in this thesis.

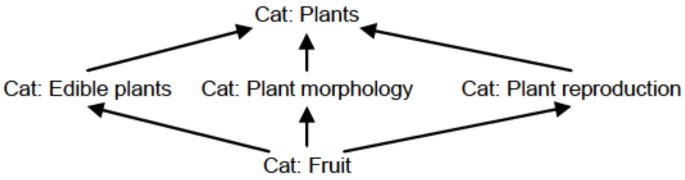
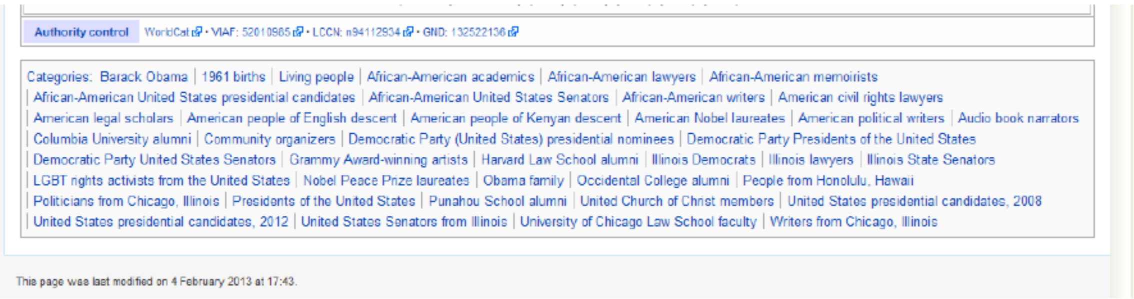


**Fig. 43. Extract with multiple inheritance in categories**

The second form of multiple inheritance involves both categories and concepts, and makes the scheme of categorization of concepts resemble a tagging system more than a taxonomy; i.e., a *folksonomy*, a lightweight conceptual structure created by users. The example of Fig. 44, which shows the categories for the concept *Barack Obama*, illustrates this form of multiple inheritance.


Authority control WorldCat • VIAF: 52010985 • LCCN: n94112934 • GND: 132522136

Categories: Barack Obama | 1961 births | Living people | African-American academics | African-American lawyers | African-American memoirists
| African-American United States presidential candidates | African-American United States Senators | African-American writers | American civil rights lawyers
| American legal scholars | American people of English descent | American people of Kenyan descent | American Nobel laureates | American political writers | Audio book narrators
| Columbia University alumni | Community organizers | Democratic Party (United States) presidential nominees | Democratic Party Presidents of the United States
| Democratic Party United States Senators | Grammy Award-winning artists | Harvard Law School alumni | Illinois Democrats | Illinois lawyers | Illinois State Senators
| LGBT rights activists from the United States | Nobel Peace Prize laureates | Obama family | Occidental College alumni | People from Honolulu, Hawaii
| Politicians from Chicago, Illinois | Presidents of the United States | Punahou School alumni | United Church of Christ members | United States presidential candidates, 2008
| United States presidential candidates, 2012 | United States Senators from Illinois | University of Chicago Law School faculty | Writers from Chicago, Illinois

This page was last modified on 4 February 2013 at 17:43.

**Fig. 44. Screenshot of the categories established for concept *Barack Obama***

Also notice that there are categories which do not represent hyponym-hypernym relations, such as *Cat: Living people*, but they indicate characteristics of the concept, like *Cat: 1961 births*. Due to this multiple inheritance, factors applied in well-formed taxonomies, such as a unique *lcs* between nodes, cannot be directly obtained in the structure of Wikipedia.

Because Wikipedia is crowd-sourced self-organized human knowledge, it undergoes constant change and development. Its branching factor and depth steadily increase over time, and does not follow the strict rules of well-formed taxonomies, making more difficult to find efficient mining methods. In this chapter, I develop techniques that, being applied to the features in the Wikipedia categorization structure, can be integrated in existing metrics.

## 7.3 Information elements and nomenclature

This dissertation will just consider a portion of the overall Wikipedia to achieve its goals. Just the pages included in two of the Wikipedia namespaces[23] are considered: articles (namespace 0) and categories (namespace 14). Other namespaces such as Users, Talks, etc. are obviated. More specifically, the considered information is the following:

- *Articles related to specific concepts*: In this type of articles, disambiguation pages, redirection pages or lists pages are obviated for the information model.
- *Articles related to categories*: The URLs of these articles start with the prefix *Category* (in English).

[23] Wikipedia namespaces: http://en.wikipedia.org/wiki/Wikipedia:Namespace

- *Relations between pairs of concepts and categories*: A concept can belong to one or more categories. *cats(c)* is the set of *parent* categories a concept *c* belongs to.
- *Relations between categories*: A category can belong to one or more categories. *cats(cat)* is the set of *parent* categories a category *cat* belongs to, forming a hierarchy.

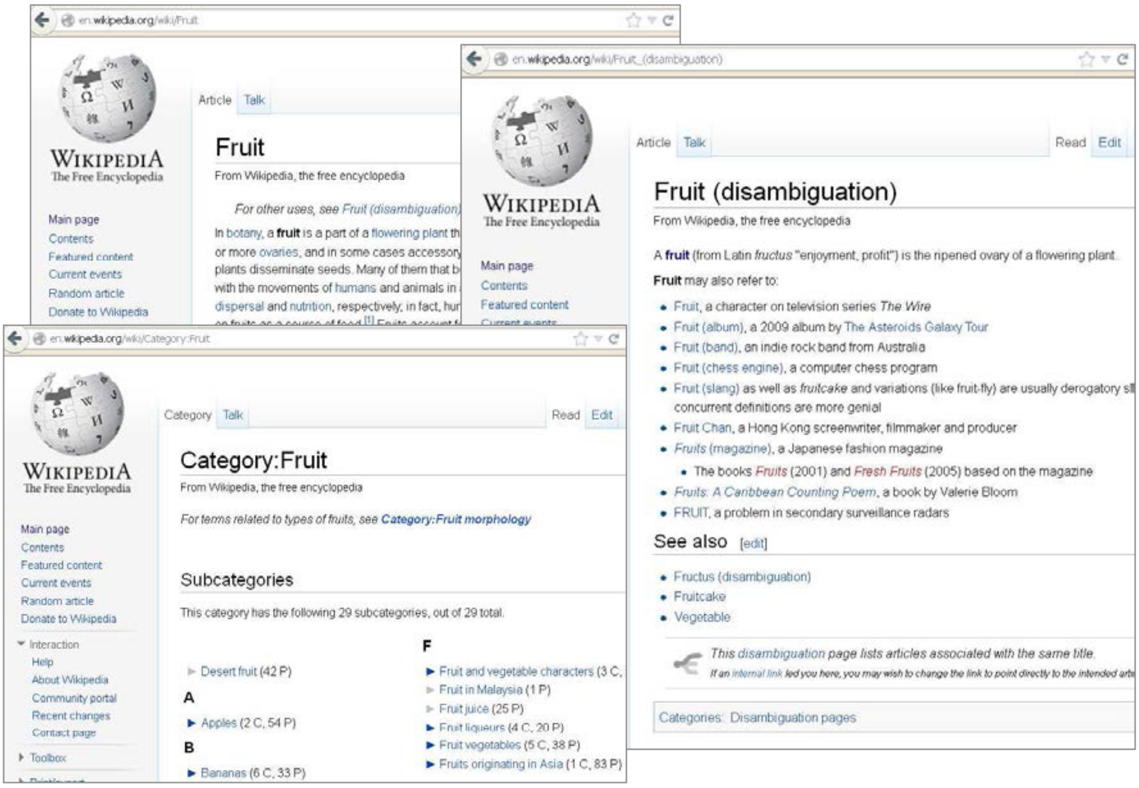
*Parent* categories are the immediately above categories in the graph structure of Wikipedia. Taking as an example the structure fragment of Fig. 43, then:

$$cats(\text{Cat : Fruit}) = \{\text{Cat : Edible plants}, \text{Cat : Plant morphology}, \text{Cat : Plant reproduction}\}$$

**Equation 43. Example of the set of categories of a category**

That is, this sample set does not include *Cat: Plants* as a parent category of *Cat: Fruit*.

An example of the types of articles is depicted on Fig. 45. Page with title *Fruit* is an article related to the specific concept of fruit. Page with title *Fruit (disambiguation)* is a disambiguation page. This kind of articles usually has their URL ending with *_disambiguation*, but not always. To identify them correctly, they must be located under the category *Disambiguated pages*, as can be seen in the figure. Finally, the page with title *Category: Fruit* is a categorization page.



Fruit

From Wikipedia, the free encyclopedia

*For other uses, see Fruit (disambiguation)*

In botany, a **fruit** is a part of a flowering plant th
or more ovaries, and in some cases accessory
plants disseminate seeds. Many of them that b
with the movements of humans and animals in
dispersal and nutrition, respectively; in fact, hur

Fruit (disambiguation)

From Wikipedia, the free encyclopedia

A **fruit** (from Latin *fructus* "enjoyment, profit") is the ripened ovary of a flowering plant.

**Fruit** may also refer to:

- Fruit, a character on television series *The Wire*
- Fruit (album), a 2009 album by The Asteroids Galaxy Tour
- Fruit (band), an indie rock band from Australia
- Fruit (chess engine), a computer chess program
- Fruit (slang) as well as *fruitcake* and variations (like fruit-fly) are usually derogatory sl concurrent definitions are more genial
- Fruit Chan, a Hong Kong screenwriter, filmmaker and producer
- *Fruits* (magazine), a Japanese fashion magazine
  - The books *Fruits* (2001) and *Fresh Fruits* (2005) based on the magazine
- *Fruits: A Caribbean Counting Poem*, a book by Valerie Bloom
- FRUIT, a problem in secondary surveillance radars

See also [edit]

- Fructus (disambiguation)
- Fruitcake
- Vegetable

*This disambiguation page lists articles associated with the same title.*
*If an internal link led you here, you may wish to change the link to point directly to the intended arti*

Categories: Disambiguation pages

Category:Fruit

From Wikipedia, the free encyclopedia

*For terms related to types of fruits, see* ***Category:Fruit morphology***

Subcategories

This category has the following 29 subcategories, out of 29 total.

- ► Desert fruit (42 P)

A

- ► Apples (2 C, 54 P)

B

- ► Bananas (6 C, 33 P)

F

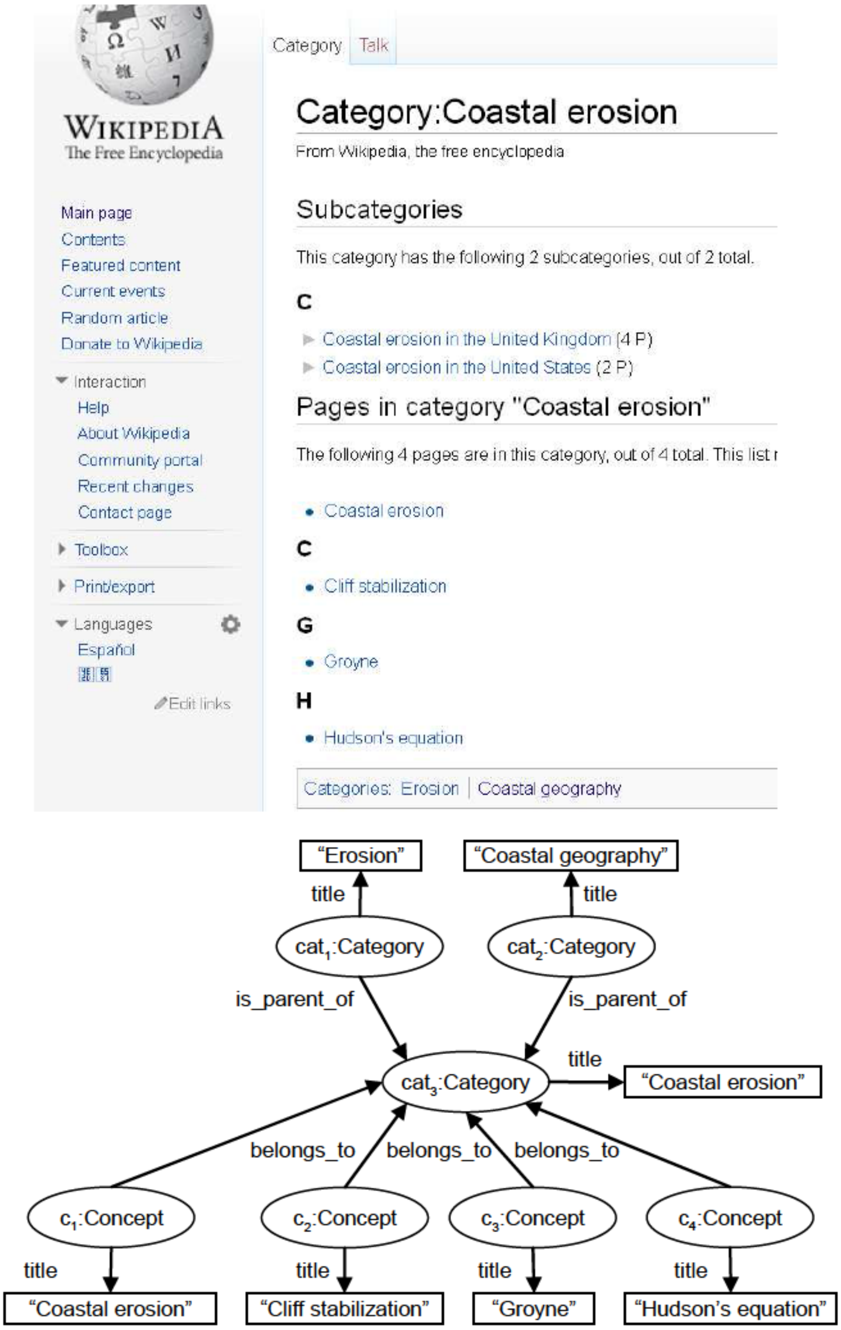
- ► Fruit and vegetable characters (3 C,
- ► Fruit in Malaysia (1 P)
- ► Fruit juice (25 P)
- ► Fruit liqueurs (4 C, 20 P)
- ► Fruit vegetables (5 C, 38 P)
- ► Fruits originating in Asia (1 C, 83 P)

**Fig. 45. Several Wikipedia screenshots to identify different types of pages**

Fig. 46 illustrates the page of a category (top) and the items implied in the information model (bottom). In the example, there are 2 subcategories and 4 concepts (section *Pages*).

For each category, the elements to store are:

1. The category itself (*$cat_3$* in the example of Fig. 46)
2. Concepts belonging to the category (*$c_1$*, *$c_2$*, *$c_3$*, *$c_4$*)
3. The relation of the concepts and the category they belong to (*belongs_to* arrows)
4. The relation between the category and their parent categories (*is_parent_of* arrows)



Category:Coastal erosion

From Wikipedia, the free encyclopedia

Subcategories

This category has the following 2 subcategories, out of 2 total.

C

- Coastal erosion in the United Kingdom (4 P)
- Coastal erosion in the United States (2 P)

Pages in category "Coastal erosion"

The following 4 pages are in this category, out of 4 total. This list r

- Coastal erosion

C

- Cliff stabilization

G

- Groyne

H

- Hudson's equation

Categories: Erosion | Coastal geography

**Fig. 46. Example of a Wikipedia category page (top) and its information model (bottom)**

After this, the pages of the two subcategories are processed and their information stored, and so on with their children, until the categorization structure is completely crawled. If this crawling algorithm of storage goes through an element (a category, concept, or a relation among them) which has been already visited, this element is ignored. The English

Wikipedia version used for developing the semantic algorithm is dated in January 2012 and contains 715,890 categories and 4,245,659 concepts. For our approach, just the titles and URLs of concepts, categories and their relations are stored; articles texts are obviated.

## 7.4 Features adaptation

Features used on path-based and multi-source approaches must be redefined to be aligned with the Wikipedia characteristics seen in the previous sections, because Wikipedia categorization scheme is not a well-formed taxonomy.

Even though semantic similarity is traditionally obtained working through a structure of concepts, this thesis will work with the structure of categories, as Wikipedia does not contain an explicit taxonomy of concepts. Next, we will explain the different features involved in our proposal; for that, we make use of Fig. 47 as an example to illustrate how the features are computed.

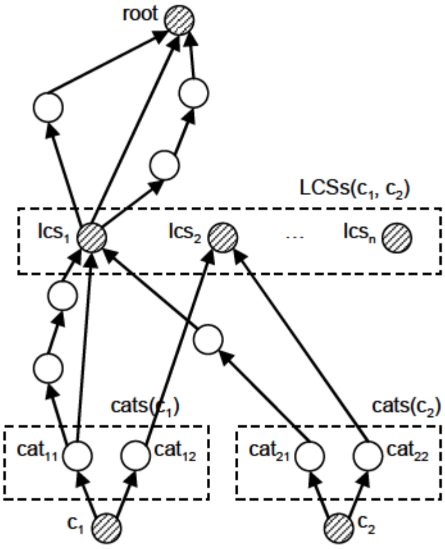


**Fig. 47. Illustrative example to explain features in Wikipedia**

The first feature to consider is the *maximum depth*, *D*, associated to a hierarchical tree, used in some of the traditional measures. It refers to the longest path from the root to the deepest node (a *leaf*) in the tree - loops are eliminated in this computing process -.

The second feature to consider is the *lcs* between two concepts. First, the lists of categories from $c_1$ and $c_2$, $cats(c_1)$ and $cats(c_2)$ respectively, are extracted. Given these lists, for each category pair $\{cat_x \in cats(c_1), cat_y \in cats(c_2)\}$, all of their *lcs's* are extracted in subsets, $LCSs(cat_x,cat_y)$. The final set for every *lcs* between $c_1$ and $c_2$, $LCSs(c_1,c_2)$, is calculated as the union of the previous subsets:

$$LCSs(c_1,c_2) = \bigcup_{\forall cat_x \in cats(c_1), cat_y \in cats(c_2)} LCSs(cat_x,cat_y)$$

**Equation 44. LCSs set**

The third feature is the shortest path between two concepts through a single *lcs* that subsumes them. Given the multiple inheritance in Wikipedia categories, there may be multiple shortest paths - with different *lcs* - between concepts. In Fig. 47 there is a shortest path between $c_1$ and $c_2$ associated to $lcs_1$, another shortest path associated to $lcs_2$, etc.

To compute this feature, a second vector is introduced, *shortest*, with the same number of elements than *LCSs*. Each dimension in *shortest* vector corresponds to an *lcs* from *LCSs*; the value of each dimension ranges from 0 to 2×*D* and is calculated as follows:

$$shortest < c_1, c_2 > (lcs) = \min_{\forall cat_x \in cats(c_1)} \{\text{shortest}(cat_x, lcs)\} + \min_{\forall cat_y \in cats(c_2)} \{\text{shortest}(cat_y, lcs)\}$$

**Equation 45: Shortest-paths vector**

In Equation 45, do not confuse *shortest* $<c_1, c_2>$ with *shortest* $(c_1, c_2)$. The former refers to the vector; the latter refers to the minimal shortest path function. Considering Fig. 47, $lcs_1$ is the unique *lcs* between categories $cat_{11}$ and $cat_{21}$. There are several paths joining these categories through $lcs_1$, but the shortest is selected; that is, the path between $cat_{11}$ and $lcs_1$ with one edge, and the path between $lcs_1$ and $cat_{21}$ with 2 edges:

| | | | |
|---|---|---|---|
| LCSs $(c_1,c_2)$ | $lcs_1$ | ... | $lcs_n$ |
| shortest $<c_1, c_2>$ | 1 + 2 = 3 | ... | ... |

**Fig. 48. Illustrative example of *shortest* vector**

The <u>fourth feature</u> to be considered is the *depth* of a node. It is commonly used in traditional measures to compute the length between the *lcs* of two concepts and the root of the hierarchy. Again, given the multiple inheritance in Wikipedia categories, there may be multiple *lcs's* between concepts and, given one of these *lcs's*, there may be multiple paths between that *lcs* and the root. In Fig. 47, $lcs_1$ has several paths leading to the root, with 2, 1 and 3 edges respectively.

Initially, the shortest path can be considered, selecting the depth that minimises the distance between a node and the root. However, a small distance to the root indicates less specialization of that node. To deal with this case, this thesis takes into account every single path to the root and applies three functions (minimum, average and maximum) to them. A new vector is introduced, *depth*, with the same number of elements as *LCSs*. Each dimension in *depth* vector corresponds to an *lcs* from *LCSs*. Being *distances (x, y)* the set of lengths of the different paths from node *x* to *y*, the value of each dimension is composed of three new values, from 0 to 2×*D*, and are calculated as follows:

$$depth < c_1, c_2 > (lcs) = (depth < c_1, c_2 > (lcs)_{min}, depth < c_1, c_2 > (lcs)_{avg}, depth < c_1, c_2 > (lcs)_{max})$$

$$\text{Where:}$$

$$depth < c_1, c_2 > (lcs)_{min} = \min\{\text{distances}(lcs, root)\},$$

$$depth < c_1, c_2 > (lcs)_{avg} = \text{avg}\{\text{distances}(lcs, root)\},$$

$$depth < c_1, c_2 > (lcs)_{max} = \max\{\text{distances}(lcs, root)\}$$

**Equation 46: *depth* vector**

The first dimension (related to $lcs_1$) of *depth* vector on the subgraph in Fig. 47 will have the following set of values:

| | | | |
|---|---|---|---|
| LCSs ($c_1$, $c_2$) | $lcs_1$ | … | $lcs_n$ |
| shortest <$c_1$, $c_2$> | 1 + 2 = 3 | … | … |
| depth <$c_1$, $c_2$> | (min = 1, avg = 2, max = 3) | … | … |

**Fig. 49. Illustrative example of *depth* vector**

## 7.5 Measures adaptation

The adaptation proposed in this thesis focuses on the most important path-based and multi-source measures; this section explains the details to adapt them to Wikipedia. There are two basic steps in the general procedure of adaptation: 1) Obtaining an intermediate vector with the measures for every *lcs* in *LCSs* vector, using *shortest* and/or *depth* vector; and 2) applying basic functions (minimum, average, maximum) to that intermediate vector.

### 7.5.1. Rada et al. (1989) adaptation

(Rada et al., 1989) use the shortest path between two concepts to calculate their semantic similarity (see Equation 10). Its adaptation is made by means of *shortest* vector. First, a new vector is obtained, *rada* <$c_1$, $c_2$>, with the result of the measure for every *lcs*, ranging from 0 to 2›*D*:

$$rada < c_1, c_2 > (lcs) = 2 \times D - shortest < c_1, c_2 > (lcs)$$

**Equation 47: Vector with Rada et al.'s adapted measure for each *lcs***

The first dimension (related to $lcs_1$) of this vector on the subgraph in Fig. 47 will have the following values:

| | | | |
|---|---|---|---|
| LCSs ($c_1$, $c_2$) | $lcs_1$ | … | $lcs_n$ |
| shortest <$c_1$, $c_2$> | 3 | … | … |
| depth <$c_1$, $c_2$> | (min = 1, avg = 2, max = 3) | (…, …, …) | (…, …, …) |
| rada <$c_1$, $c_2$> | 2 › 20 - 3 = 37 | … | … |

**Fig. 50. Illustrative example of *rada* vector**

Second, 3 different adapted measures are obtained by selecting the minimum, average and maximum values of *rada* vector.

$$\begin{aligned}
\text{sim}_{\text{rada\_min}}(c_1, c_2) &= \min\{rada < c_1, c_2 >\} \\
\text{sim}_{\text{rada\_avg}}(c_1, c_2) &= \text{avg}\{rada < c_1, c_2 >\} \\
\text{sim}_{\text{rada\_max}}(c_1, c_2) &= \max\{rada < c_1, c_2 >\}
\end{aligned}$$

**Equation 48. Adapted similarity measures, based on Rada et al.'s**

### 7.5.2. Wu & Palmer (1994) adaptation

(Z. Wu & Palmer, 1994) use the shortest path between two concepts and the depth of their *lcs* (see Equation 12). Then, its adaptation is made by means of *shortest* and *depth* vector.

In this case, every dimension in *depth* vector has 3 different values (the shortest path to the root (minimum), the longest path (maximum), and the average of all paths' lengths). So first, a new vector is obtained, *wp*, with the measure calculated for every *lcs* and every depth value, ranging from 0 to 1:

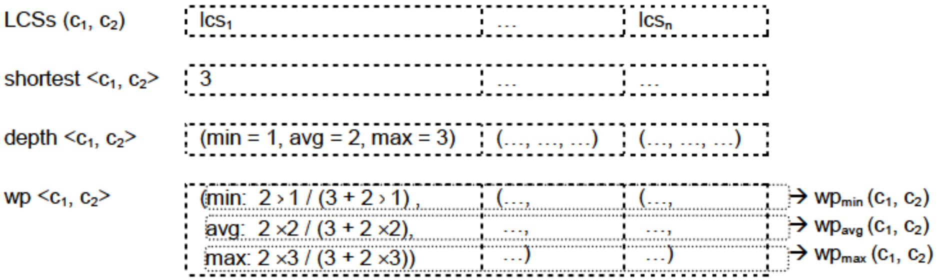
$$
\begin{aligned}
& wp < c_1,c_2 > (lcs) = \left( wp < c_1,c_2 > (lcs)_{min}, wp < c_1,c_2 > (lcs)_{avg}, wp < c_1,c_2 > (lcs)_{max} \right) \\
& \text{Where:} \\
& wp < c_1,c_2 > (lcs)_{min} = \frac{2 \times depth < c_1,c_2 > (lcs)_{min}}{shortest < c_1,c_2 > (lcs) + 2 \times depth < c_1,c_2 > (lcs)_{min}}, \\
& wp < c_1,c_2 > (lcs)_{avg} = \frac{2 \times depth < c_1,c_2 > (lcs)_{avg}}{shortest < c_1,c_2 > (lcs) + 2 \times depth < c_1,c_2 > (lcs)_{avg}}, \\
& wp < c_1,c_2 > (lcs)_{max} = \frac{2 \times depth < c_1,c_2 > (lcs)_{max}}{shortest < c_1,c_2 > (lcs) + 2 \times depth < c_1,c_2 > (lcs)_{max}}
\end{aligned}
$$

**Equation 49: Vector with Wu & Palmer's adapted measure for each *lcs***

Then, 3 subsets are obtained by grouping the results generated in the previous step depending on the depth value used (see example in Fig. 51):

$$
\begin{aligned}
wp_{min}(c_1,c_2) &= \bigcup_{lcs \in LCSs(c_1,c_2)} wp < c_1,c_2 > (lcs)_{min}; \\
wp_{avg}(c_1,c_2) &= \bigcup_{lcs \in LCSs(c_1,c_2)} wp < c_1,c_2 > (lcs)_{avg}; \\
wp_{max}(c_1,c_2) &= \bigcup_{lcs \in LCSs(c_1,c_2)} wp < c_1,c_2 > (lcs)_{max}
\end{aligned}
$$

**Equation 50. Subsets with Wu & Palmer's adapted measure with minimal, average and maximum depths**



| | | | | |
|---|---|---|---|---|
| LCSs ($c_1$, $c_2$) | lcs$_1$ | … | lcs$_n$ | |
| shortest <$c_1$, $c_2$> | 3 | … | … | |
| depth <$c_1$, $c_2$> | (min = 1, avg = 2, max = 3) | (…, …, …) | (…, …, …) | |
| wp <$c_1$, $c_2$> | (min: 2 › 1 / (3 + 2 › 1) , | (…, | (…, | → wp$_{min}$ ($c_1$, $c_2$) |
| | avg: 2 ×2 / (3 + 2 ×2), | …, | …, | → wp$_{avg}$ ($c_1$, $c_2$) |
| | max: 2 ×3 / (3 + 2 ×3)) | …) | …) | → wp$_{max}$ ($c_1$, $c_2$) |

**Fig. 51. Illustrative example of *wp* subsets**

Finally, with the help of these 3 subsets, 9 adapted measures are obtained:

$$
\begin{aligned}
& \text{sim}_{\text{wp\_min\_min}}(c_1,c_2) = \min\{wp_{min}(c_1,c_2)\}; \\
& \text{sim}_{\text{wp\_min\_avg}}(c_1,c_2) = \min\{wp_{avg}(c_1,c_2)\}; \\
& \text{sim}_{\text{wp\_min\_max}}(c_1,c_2) = \min\{wp_{max}(c_1,c_2)\}; \\
& \text{sim}_{\text{wp\_avg\_min}}(c_1,c_2) = \text{avg}\{wp_{min}(c_1,c_2)\}; \\
& \text{sim}_{\text{wp\_avg\_avg}}(c_1,c_2) = \text{avg}\{wp_{avg}(c_1,c_2)\}; \\
& \text{sim}_{\text{wp\_avg\_max}}(c_1,c_2) = \text{avg}\{wp_{max}(c_1,c_2)\}; \\
& \text{sim}_{\text{wp\_ max\_min}}(c_1,c_2) = \max\{wp_{min}(c_1,c_2)\}; \\
& \text{sim}_{\text{wp\_ max\_avg}}(c_1,c_2) = \max\{wp_{avg}(c_1,c_2)\}; \\
& \text{sim}_{\text{wp\_ max\_max}}(c_1,c_2) = \max\{wp_{max}(c_1,c_2)\}
\end{aligned}
$$

**Equation 51. Adapted similarity measures, based on Wu & Palmer's**

### 7.5.3. Leacock & Chodorow (1994) adaptation

(Leacock & Chodorow, 1994) use the shortest path between two concepts to calculate their semantic similarity, as (Rada et al., 1989); therefore, its adaptation is also made by means of *shortest* vector used to compute the *lc* vector (Equation 52), obtaining 3 different adapted measures (Equation 53):

$$lc < c_1,c_2 > (lcs) = -\log\left(shortest < c_1,c_2 > (lcs) + 1/(2 \times D)\right)$$

**Equation 52: Vector with Leacock & Chodorow's adapted measure for each *lcs***

$$\begin{aligned}
&\mathrm{sim}_{\mathrm{lc\_min}}(c_1,c_2) = \min\{lc < c_1,c_2 >\};\\
&\mathrm{sim}_{\mathrm{lc\_avg}}(c_1,c_2) = \mathrm{avg}\{lc < c_1,c_2 >\};\\
&\mathrm{sim}_{\mathrm{lc\_max}}(c_1,c_2) = \max\{lc < c_1,c_2 >\}
\end{aligned}$$

**Equation 53. Adapted similarity measures, based on Leacock & Chodorow's**

### 7.5.4. Blázquez-del-Toro et al. (2008) adaptation

The adaptation of this measure uses *depth* vector and a constant *k*. Besides, the shortest path is needed to obtain the information ratio, $E_{lcs}$ / $E_c$, and the value of $sim_{to\_lcs}$ (see Fig. 18, Equation 13 and Equation 14). Again, an intermediate vector is obtained, *bl*, with the measure for every *lcs* and the corresponding depth value, ranging from 0 to 1:

$$\begin{aligned}
&bl < c_1,c_2 > (lcs) = \left(bl < c_1,c_2 > (lcs)_{min}, bl < c_1,c_2 > (lcs)_{avg}, bl < c_1,c_2 > (lcs)_{max}\right)\\
&\text{Where:}\\
&bl < c_1,c_2 > (lcs)_{min} = \frac{\mathrm{sim}_{to\_lcs}(lcs,c_1) \times \mathrm{sim}_{to\_lcs}(lcs,c_2)}{\mathrm{sim}_{to\_lcs}(lcs,c_1) + \mathrm{sim}_{to\_lcs}(lcs,c_2) - \mathrm{sim}_{to\_lcs}(lcs,c_1) \times \mathrm{sim}_{to\_lcs}(lcs,c_2)},\\
&\text{using depth} < \mathrm{c}_1,\mathrm{c}_2 > (\mathrm{lcs})_{\mathrm{min}};\\
&bl < c_1,c_2 > (lcs)_{avg} = \frac{\mathrm{sim}_{to\_lcs}(lcs,c_1) \times \mathrm{sim}_{to\_lcs}(lcs,c_2)}{\mathrm{sim}_{to\_lcs}(lcs,c_1) + \mathrm{sim}_{to\_lcs}(lcs,c_2) - \mathrm{sim}_{to\_lcs}(lcs,c_1) \times \mathrm{sim}_{to\_lcs}(lcs,c_2)},\\
&\text{using depth} < \mathrm{c}_1,\mathrm{c}_2 > (\mathrm{lcs})_{\mathrm{avg}};\\
&bl < c_1,c_2 > (lcs)_{max} = \frac{\mathrm{sim}_{to\_lcs}(lcs,c_1) \times \mathrm{sim}_{to\_lcs}(lcs,c_2)}{\mathrm{sim}_{to\_lcs}(lcs,c_1) + \mathrm{sim}_{to\_lcs}(lcs,c_2) - \mathrm{sim}_{to\_lcs}(lcs,c_1) \times \mathrm{sim}_{to\_lcs}(lcs,c_2)},\\
&\text{using depth} < \mathrm{c}_1,\mathrm{c}_2 > (\mathrm{lcs})_{\mathrm{max}}
\end{aligned}$$

**Equation 54: Vector with Blázquez-del-Toro et al.'s adapted measure for each *lcs***

Experiments will work with different *k* values. Then, 3 subsets are obtained by grouping the results generated in the previous step depending on the depth value used, as done in Wu & Palmer's adaptation:

$$\begin{aligned}
&bl_{min}(c_1,c_2) = \bigcup_{lcs \in LCSs(c_1,c_2)} bl < c_1,c_2 > (lcs)_{min};\\
&bl_{avg}(c_1,c_2) = \bigcup_{lcs \in LCSs(c_1,c_2)} bl < c_1,c_2 > (lcs)_{avg};\\
&bl_{max}(c_1,c_2) = \bigcup_{lcs \in LCSs(c_1,c_2)} bl < c_1,c_2 > (lcs)_{max}
\end{aligned}$$

**Equation 55. Subsets with Blázquez-del-Toro et al.'s adapted measure with minimal, average and maximum depths**

With the help of these 3 subsets, 9 adapted measures are obtained:

$$
\begin{aligned}
&\mathrm{sim}_{\mathrm{bl\_min\_min}}(c_1,c_2) = \min\{bl_{min}(c_1,c_2)\};\\
&\mathrm{sim}_{\mathrm{bl\_min\_avg}}(c_1,c_2) = \min\{bl_{avg}(c_1,c_2)\};\\
&\mathrm{sim}_{\mathrm{bl\_min\_max}}(c_1,c_2) = \min\{bl_{max}(c_1,c_2)\};\\
&\mathrm{sim}_{\mathrm{bl\_avg\_min}}(c_1,c_2) = \mathrm{avg}\{bl_{min}(c_1,c_2)\};\\
&\mathrm{sim}_{\mathrm{bl\_avg\_avg}}(c_1,c_2) = \mathrm{avg}\{bl_{avg}(c_1,c_2)\};\\
&\mathrm{sim}_{\mathrm{bl\_avg\_max}}(c_1,c_2) = \mathrm{avg}\{bl_{max}(c_1,c_2)\};\\
&\mathrm{sim}_{\mathrm{bl\_max\_min}}(c_1,c_2) = \max\{bl_{min}(c_1,c_2)\};\\
&\mathrm{sim}_{\mathrm{bl\_max\_avg}}(c_1,c_2) = \max\{bl_{avg}(c_1,c_2)\};\\
&\mathrm{sim}_{\mathrm{bl\_max\_max}}(c_1,c_2) = \max\{bl_{max}(c_1,c_2)\}
\end{aligned}
$$

**Equation 56. Adapted similarity measures, based on Blázquez-del-Toro et al.'s**

### 7.5.5. Li et al. (2003) adaptation

Li et al.'s measure is based on the non-linear combination of the shortest path between $c_1$ and $c_2$ and the *depth* of their *lcs*. A new vector is obtained, *li*, with the measure for every *lcs* and every depth value, ranging from 0 to 1:

$$
\begin{aligned}
&li < c_1,c_2 > (lcs) = \left(li < c_1,c_2 > (lcs)_{min}, li < c_1,c_2 > (lcs)_{avg}, li < c_1,c_2 > (lcs)_{max}\right)\\
&\text{Where:}\\
&li < c_1,c_2 > (lcs)_{min} = e^{-\alpha \times shortest<c_1,c_2>} \times \frac{e^{\beta \times depth<c_1,c_2>(lcs)_{min}} - e^{-\beta \times depth<c_1,c_2>(lcs)_{min}}}{e^{\beta \times depth<c_1,c_2>(lcs)_{min}} + e^{-\beta \times depth<c_1,c_2>(lcs)_{min}}},\\
&li < c_1,c_2 > (lcs)_{avg} = e^{-\alpha \times shortest<c_1,c_2>} \times \frac{e^{\beta \times depth<c_1,c_2>(lcs)_{avg}} - e^{-\beta \times depth<c_1,c_2>(lcs)_{avg}}}{e^{\beta \times depth<c_1,c_2>(lcs)_{avg}} + e^{-\beta \times depth<c_1,c_2>(lcs)_{avg}}},\\
&li < c_1,c_2 > (lcs)_{max} = e^{-\alpha \times shortest<c_1,c_2>} \times \frac{e^{\beta \times depth<c_1,c_2>(lcs)_{max}} - e^{-\beta \times depth<c_1,c_2>(lcs)_{max}}}{e^{\beta \times depth<c_1,c_2>(lcs)_{max}} + e^{-\beta \times depth<c_1,c_2>(lcs)_{max}}}
\end{aligned}
$$

**Equation 57: Vector with Li et al.'s adapted measure for each *lcs***

3 subsets are obtained by grouping the results generated in the previous step depending on the depth value used:

$$
\begin{aligned}
&li_{min}(c_1,c_2) = \bigcup_{lcs \in LCSs(c_1,c_2)} li < c_1,c_2 > (lcs)_{min};\\
&li_{avg}(c_1,c_2) = \bigcup_{lcs \in LCSs(c_1,c_2)} li < c_1,c_2 > (lcs)_{avg};\\
&li_{max}(c_1,c_2) = \bigcup_{lcs \in LCSs(c_1,c_2)} li < c_1,c_2 > (lcs)_{max}
\end{aligned}
$$

**Equation 58. Subsets with Li et al.'s adapted measure with minimal, average and maximum depths**

Finally, with the help of these 3 subsets, 9 adapted measures are obtained:

$$\begin{aligned}
&\mathrm{sim}_{\mathrm{li\_min\_min}}(c_1, c_2) = \min\{li_{min}(c_1, c_2)\};\\
&\mathrm{sim}_{\mathrm{li\_min\_avg}}(c_1, c_2) = \min\{li_{avg}(c_1, c_2)\};\\
&\mathrm{sim}_{\mathrm{li\_min\_max}}(c_1, c_2) = \min\{li_{max}(c_1, c_2)\};\\
&\mathrm{sim}_{\mathrm{li\_avg\_min}}(c_1, c_2) = \mathrm{avg}\{li_{min}(c_1, c_2)\};\\
&\mathrm{sim}_{\mathrm{li\_avg\_avg}}(c_1, c_2) = \mathrm{avg}\{li_{avg}(c_1, c_2)\};\\
&\mathrm{sim}_{\mathrm{li\_avg\_max}}(c_1, c_2) = \mathrm{avg}\{li_{max}(c_1, c_2)\};\\
&\mathrm{sim}_{\mathrm{li\_max\_min}}(c_1, c_2) = \max\{li_{min}(c_1, c_2)\};\\
&\mathrm{sim}_{\mathrm{li\_max\_avg}}(c_1, c_2) = \max\{li_{avg}(c_1, c_2)\};\\
&\mathrm{sim}_{\mathrm{li\_max\_max}}(c_1, c_2) = \max\{li_{max}(c_1, c_2)\}
\end{aligned}$$

**Equation 59. Adapted similarity measures, based on Li et al.'s**

# PART III. Evaluation and Conclusions

# 8 EVALUATION

This chapter exposes the experiments carried out to support hypotheses 1, 2 and 3, stated in section 4.4. The feasibility of Itaca layer, hypothesis 1, is proven on section 8.1, showing the implementation of the layer. Section 8.2 is devoted to Hypothesis 2, where it is shown that the usage of semantic annotations in an algorithm to rank web results yields better results than current ranking algorithms. The validity of Wikipedia as a source to calculate semantic similarity, hypothesis 3, is proven in section 8.3.

## 8.1 Hypothesis 1: Web application

> *It is feasible to improve current web search engines by means of the implementation of an independent layer on top of them with collaborative data gathering.*

This thesis considers it is feasible to implement a layer on top of current web search engines to take advantage of both 1) traditional ranking algorithms and 2) new techniques based on collaborative data. This would allow incorporating an additional model instead of working on a new search engine from scratch. In order to prove this hypothesis, its development has been conducted. The web layer has been implemented as a centralized web-based site and the information is stored in a relational database server.

The site has been built on a *Rails* environment (Thomas, Heinemeier Hansson, & Breedt, 2005). Rails is a framework for the development of web applications, with basic principles which make it quite suitable:

- *Less software*: Developers need fewer lines of code to implement an application. Less code means less bugs and the resulting implementation is easier to maintain. This principle is basically obtained because of its implementation language, *Ruby* (Flanagan & Matsumoto, 2008).
- *Convention over configuration*: There are no complex configuration files, like in other frameworks; instead, some convention rules are applied.
- *DRY (Don't Repeat Yourself)*: Every element or piece of code is located in a single place, never repeated.

The storing system selected is *MySQL Server 5.0*[24] because of its simplicity. This relational database server stores the basic elements (queries, terms, Wikipedia concepts, web resources, accumulators, and relations between all of them) using separate tables. The operational flow is explained in next sections.

### 8.1.1. Query definition

Itaca offers a Google-like graphical interface to formulate a query, with a simple text field to insert the terms to search and a button to start the searching process (see Fig. 53). When clicking the *Search* button, the application stores the current query and its terms. The query in Fig. 53 will produce the following objects:

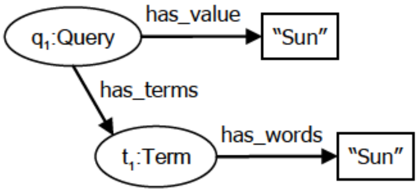
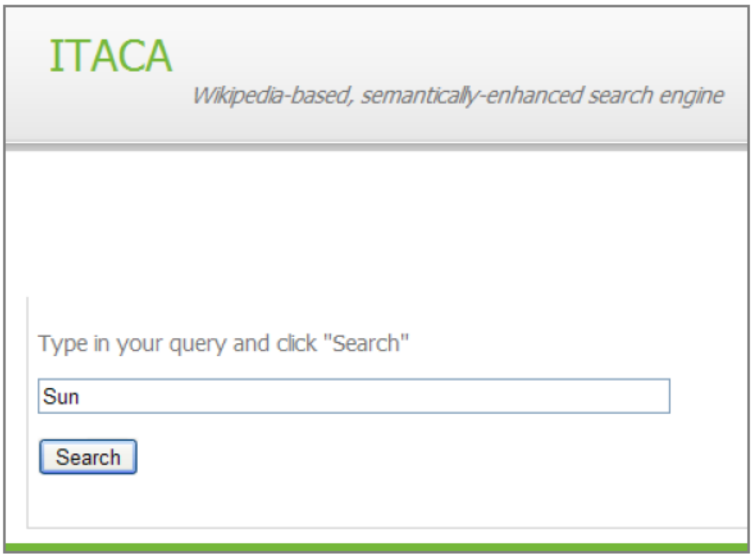


**Fig. 52. Query with one term, information model**

[24] MySQL Server 5.0 download page: http:/dev.mysql.com/downloads/mysql/



ITACA

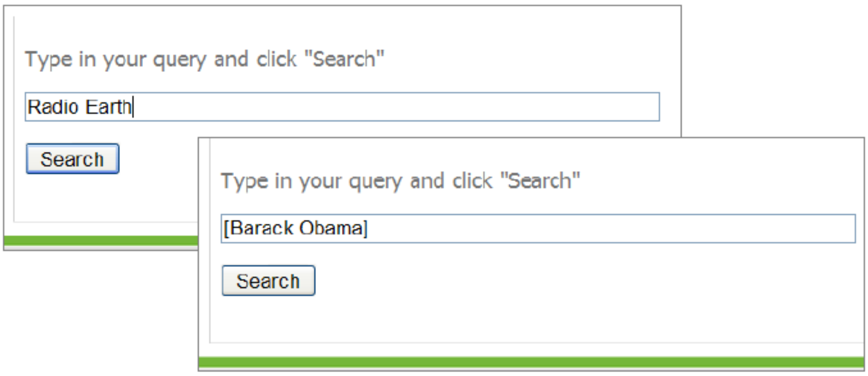
Wikipedia-based, semantically-enhanced search engine

Type in your query and click "Search"

Sun

**Fig. 53. Query with one term, screenshot**

Terms can be composed of more than one word; in that case, users have to surround those words with brackets. Fig. 54 shows two samples: a query with two different terms (left), and a query with one term composed of two words (right).



Type in your query and click "Search"

Radio Earth

Type in your query and click "Search"

[Barack Obama]

**Fig. 54. Query with two terms (left) and query with one term (right), screenshot**

The objects generated in these samples are depicted in Fig. 55.

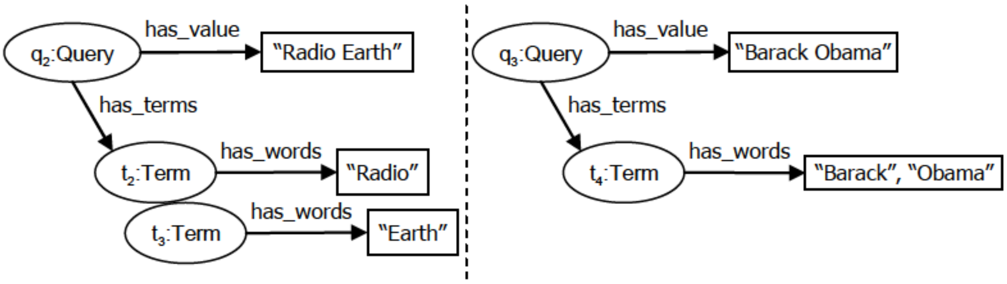
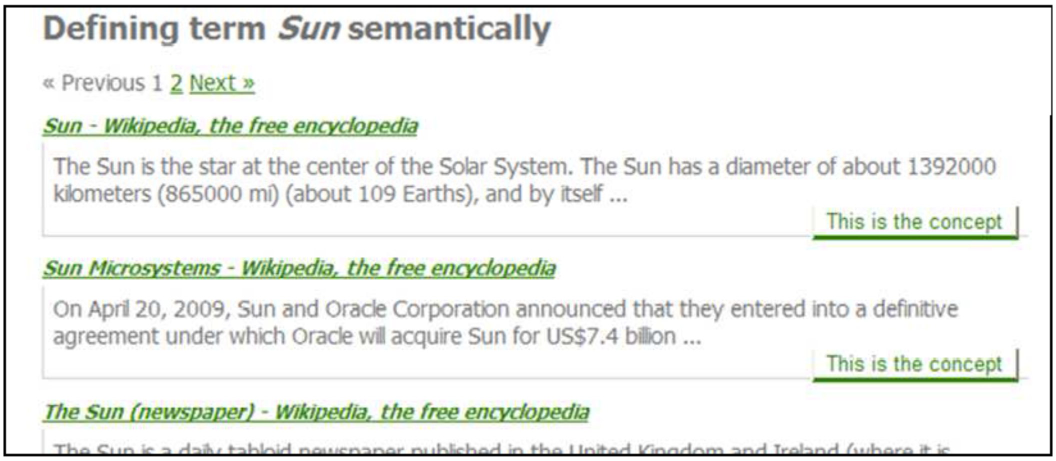


**Fig. 55. Query with two terms (left) and query with one term (right), information model**

### 8.1.2. Query disambiguation

In the second step of the searching process with Itaca layer, and before obtaining the final results of the query, users are intended to semantically disambiguate the terms of their query[25]. For each term, users are presented with a set of Wikipedia pages which may refer to that term, as Fig. 56 shows.



Defining term *Sun* semantically

« Previous 1 2 Next »

Sun - Wikipedia, the free encyclopedia

The Sun is the star at the center of the Solar System. The Sun has a diameter of about 1392000 kilometers (865000 mi) (about 109 Earths), and by itself ...

This is the concept

Sun Microsystems - Wikipedia, the free encyclopedia

On April 20, 2009, Sun and Oracle Corporation announced that they entered into a definitive agreement under which Oracle will acquire Sun for US$7.4 billion ...

This is the concept

The Sun (newspaper) - Wikipedia, the free encyclopedia

The Sun is a daily tabloid newspaper published in the United Kingdom and Ireland (where it is

Fig. 56. Query disambiguation, screenshot

The system does not take into account pages that do not represent real concepts in Wikipedia, such as user, discussion or disambiguation pages.

When users press any of the *This is the concept* buttons established for every Wikipedia concept, the query term is automatically associated with that concept, remaining the latter as one of the tags of the query. This way, instead of using traditional mechanisms of implicit feedback like query logs, terms co-occurrence, etc. (see disadvantages on 2.3), query disambiguation is made by means of well-defined concepts explicitly identified by users. Fig. 57 shows the information model for the sample of $q_{1,}$ if the first concept on the list shown in Fig. 56 is selected.

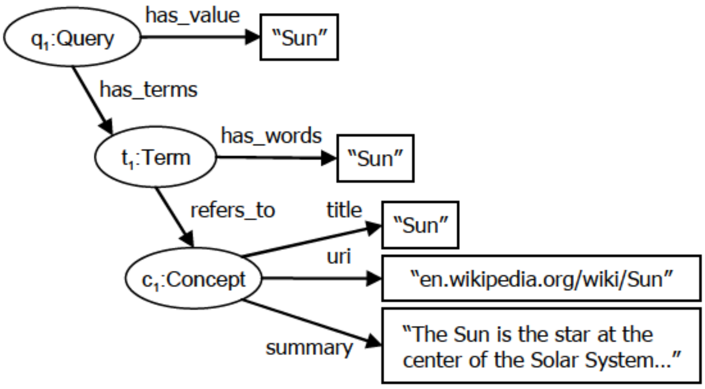


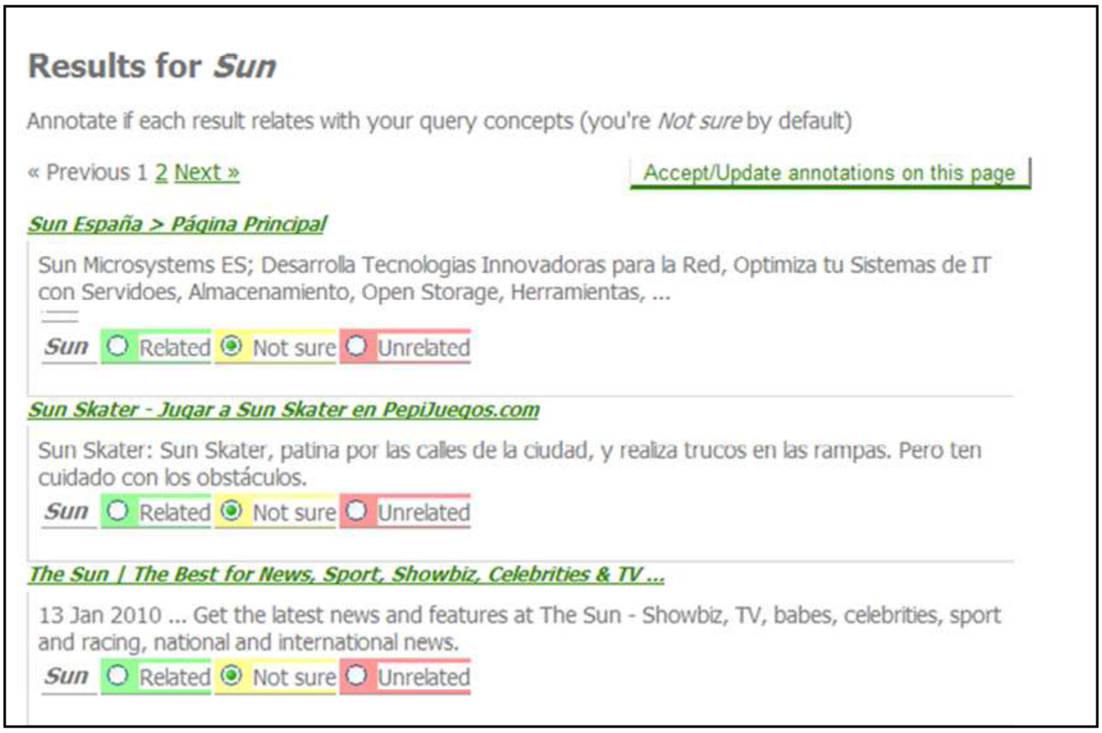
Fig. 57. Query disambiguation, information model

### 8.1.3. FINAL RESULTS AND RESOURCES ANNOTATION

After selecting the concept (or concepts, in case of more than one term) associated to the query, Itaca will return the web resources in the order the ranking algorithm establishes. Fig. 58 displays the screen of this step, consisting mainly on a list of web pages. In this

[25] Even though the current implementation of the GUI does not allow to skip this step, the theoretical model, presented in chapter 5, does allow it.

example, no annotation has been made with the concept *Sun* before; therefore, ranking presented is the same offered by the web search engine that underlies the layer.



## Results for *Sun*

Annotate if each result relates with your query concepts (you're *Not sure* by default)

« Previous 1 2 Next » Accept/Update annotations on this page

*Sun España > Página Principal*

Sun Microsystems ES; Desarrolla Tecnologias Innovadoras para la Red, Optimiza tu Sistemas de IT con Servidoes, Almacenamiento, Open Storage, Herramientas, ...

*Sun* ○ Related ● Not sure ○ Unrelated

*Sun Skater - Jugar a Sun Skater en PepiJuegos.com*

Sun Skater: Sun Skater, patina por las calles de la ciudad, y realiza trucos en las rampas. Pero ten cuidado con los obstáculos.

*Sun* ○ Related ● Not sure ○ Unrelated

*The Sun | The Best for News, Sport, Showbiz, Celebrities & TV ...*

13 Jan 2010 ... Get the latest news and features at The Sun - Showbiz, TV, babes, celebrities, sport and racing, national and international news.

*Sun* ○ Related ● Not sure ○ Unrelated

**Fig. 58. Query results, screenshot**

The difference with the graphical interface of traditional search engines is that every web resource is accompanied by a set of semaphore-like radio buttons to annotate whether the returned result is indeed related with the query. More specifically, there is a set of green, yellow and red radio buttons for every concept implied in the query. Selecting the green button means the web resource is related to the query (*Related* radio button in the figure); selecting red button means the opposite (*Unrelated* radio button). Yellow button is marked as default (*Not sure* radio button), which means that user does not know or does not care about that particular resource.

This way, and reinforcing the query disambiguation, web resources are also annotated with the appropriate concepts, and users express whether they have found the web page relevant or not with respect to the query. Fig. 59 shows the information model if user selects the first web resource (that of *Sun Microsystems* company) as unrelated with the query.

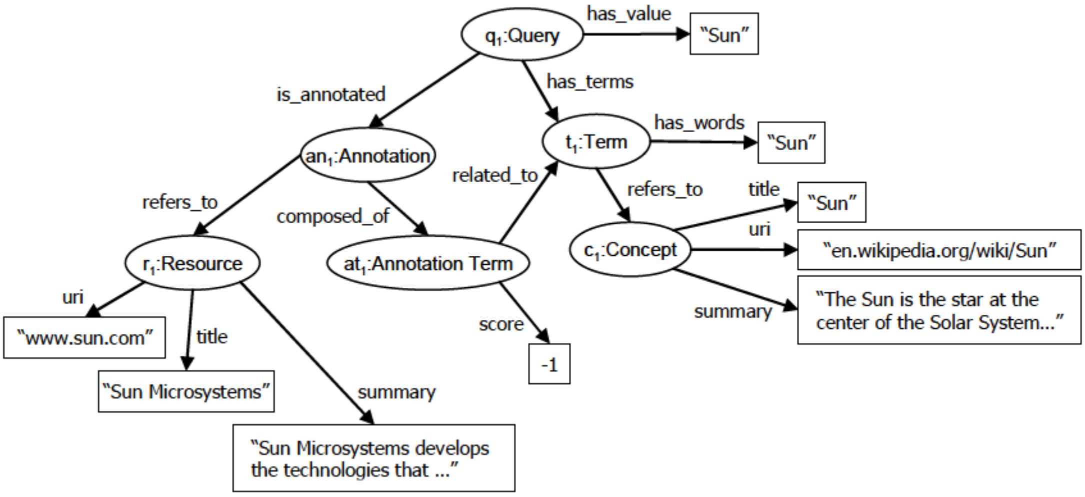
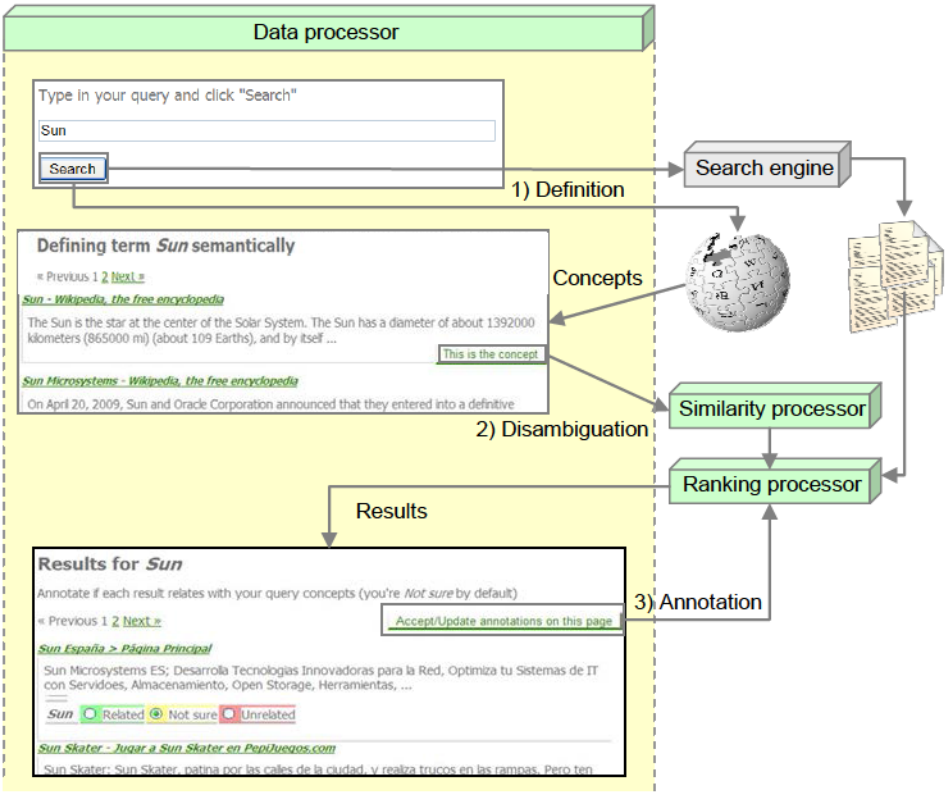


**Fig. 59. Resources annotation: information model**

With this filtering activity, a more trustworthy opinion about the relevance of the resource is obtained. Itaca layer does not consider or assume that just selecting a web resource (when user presses its hyperlink) is a fact of its relevance.

### 8.1.4. DATA PROCESSOR VS GUI

The graphical user interface and the *Data processor* component in Itaca layer are quite interconnected, because every step in the latter is represented through a different view at the former. Fig. 60 shows a resume of each step in the *Data processor* component and the view of the graphical interface that makes it possible.



Type in your query and click "Search"

Sun

Search

**Defining term *Sun* semantically**

« Previous 1 2 Next »

*Sun - Wikipedia, the free encyclopedia*

The Sun is the star at the center of the Solar System. The Sun has a diameter of about 1392000 kilometers (865000 mi) (about 109 Earths), and by itself ...

This is the concept

*Sun Microsystems - Wikipedia, the free encyclopedia*

On April 20, 2009, Sun and Oracle Corporation announced that they entered into a definitive

**Results for *Sun***

Annotate if each result relates with your query concepts (you're *Not sure* by default)

« Previous 1 2 Next »

Accept/Update annotations on this page

*Sun España > Página Principal*

Sun Microsystems ES; Desarrolla Tecnologias Innovadoras para la Red, Optimiza tu Sistemas de IT con Servidoes, Almacenamiento, Open Storage, Herramientas, ...

*Sun* ○ Related ● Not sure ○ Unrelated

*Sun Skater - Jugar a Sun Skater en PepiJuegos.com*

Sun Skater: Sun Skater, patina por las calles de la ciudad, y realiza trucos en las rampas. Pero ten

**Fig. 60. Data processor steps working throughout the GUI of Itaca layer**

## 8.2 Hypothesis 2: Ranking processor

> *Collaborative usage of semantic annotations in a search process, along with an appropriate ranking algorithm, produces 1) more relevant results than traditional web search engines; and 2) with a low response time.*

Evaluating if a ranking algorithm produces a good or better search engine is a difficult task, because it is not clear enough what a "good" search engine means. It may depend on different factors, such as the final users or the use of the application. This thesis has focused on assessing the *effectiveness* of the layer, measured with:

1. The *quality* of its search results
2. The *time* needed to process it

For the *quality* feature, the two quality parameters used in the state of the art are:

- *Precision*: The fraction of the returned results which are relevant for the query.
- *Recall*: The fraction of the relevant documents in the collection which were returned by the engine.

In this thesis, the evaluation of the recall would require the calculation of relevance ratings for the whole data collection of the web search engines involved in the evaluation, but this information is not available. Due to this, recall is not considered in the experiments. The evaluation focuses on precision, comparing the relevance rate - relevant resources - obtained with Itaca ranking algorithm and the relevance rate obtained with other well-known current search engines without the Itaca layer. A second evaluation is executed to know the influence of annotations when their number increases.

Regarding processing *time*, the evaluation compares the response time obtained with Itaca ranking algorithm with different number of annotations, in order to see the variation (increment or decrement) in the time needed to obtain the final results. Besides, this dissertation analyses the potential internal model to reach low response times when items (web resources and concepts) increase.

### 8.2.1. Data set and parameter values

Evaluating and comparing a web ranking metric is a difficult task, because of its subjectivity and the lack of standard corpus for evaluating web searching. Studies exposed in the state of the art related to resources retrieval rely on the TREC data[26], but this collection does not distinguish the concepts or meanings of a given query and it is focused on finding a set of instances for a given query, indicating only the binary relevance (0 or 1) of each page to a number of predefined queries.

Focusing on semantic search for information retrieval in the Web, studies from (Castells et al., 2007)  and (M. Fernández et al., 2011) are the most similar to this thesis. As documented in section 2.4.4, their works support semantic search capabilities (as a

[26] Test REtrieval Conference (TREC) Home Page: http://trec.nist.gov/

question answering system) in large document repositories. The semantic annotations take place with the use of ontologies instead of Wikipedia. They elaborated 20 queries to compare their searches with conventional keyword-only search. Later on, the queries were modified in order for the results to be compared with TREC systems. However, the documents to search and keywords were limited to the selected repository they used and the domains covered by the ontologies they elaborated, respectively.

Therefore, in order to evaluate the core of the ranking algorithm elaborated in this thesis, and as the intention of the present dissertation is the use of a new search paradigm in web search engines, with a huge number of documents from every possible domain, a new query collection has been defined for this purpose. This data set is composed of *informational queries*; that is, queries involving a need to find a selection of documents. This type of queries has been estimated to account for 80% of queries in the web (Jansen, Booth, & Spink, 2008); this prominent use is the reason to be chosen for the evaluation of Hypothesis 2.

20 different informational queries have been processed by 8 human users, given a total data set of 160 queries. Most of them use similar or even identical terms, in order to evaluate the response with similar queries. Most of the related works exposed in chapter 2 were evaluated with human judgements, so I have decided to use the same evaluation procedure. Regarding the size, the data set has more queries than similar collections for the evaluation of semantic retrieval models like AMBIENT (Carpineto, Mizzaro, Romano, & Snidero, 2009), MORESQUE (Navigli & Crisafulli, 2010), or those used in web search studies evaluated in (Hawking, Craswell, Bailey, & Griffihs, 2001). Table 5 shows the different queries used.

**Table 5. Informational queries for the evaluation**

| Id query | String |
|---|---|
| 1 | Árbol [de hoja caduca] |
| 2 | Pisos [alquiler con opción a compra] Leganés |
| 3 | [journal citation reports] |
| 4 | Árbol [de hoja perenne] |
| 5 | Pisos [alquiler con derecho a compra] Leganés |
| 6 | stars hotel hollywood |
| 7 | Join unix separator |
| 8 | JCR |
| 9 | Sun England |
| 10 | [Rodrigo Rato] Teatro |
| 11 | Join unix delimiter |
| 12 | ruby fixtures |
| 13 | iphone features |
| 14 | JCR 2009 |
| 15 | software fixtures |
| 16 | kiwi [new zealand] |
| 17 | Earth radius |
| 18 | ipad features |
| 19 | [software testing] fixtures |
| 20 | kiwi inhabitant [new zealand] |

Users were informed of the different goals persecuted in each query, detailed in Table 6.

**Table 6. Query goals**

| Id query | Goal |
|---|---|
| 1 | General information about deciduous trees. |
| 2 | Homes/buildings in Leganés (a place in Madrid, Spain) for leasing. |
| 3 | Information about the Journal Citation Reports publication. |
| 4 | General information about evergreen trees. |
| 5 | Same goal as query 2, but expressed with a different term that means the same. |
| 6 | Hotels in Hollywood were famous movie celebrities had been hosted. Users are not intended to look for hotels with certain ranking classification. |
| 7 | Join is a command in Unix-like operating systems that merges the lines of two sorted text files based on the presence of a common field. Fields are separated by a certain delimiter. In this query we are looking for information of this delimiter (which is the argument to settle the delimiter, etc.) |
| 8 | Information about the Journal Citation Reports publication. |
| 9 | Information about the sun (the solar star) in England (that is, the weather), and not other references, such as the newspaper. |
| 10 | Rodrigo Rato was a Spanish minister and director of the International Monetary Fund. In this query, users have to look for web resources about his role as an actor in theatre plays. |
| 11 | Same goal as query 7. |
| 12 | Fixtures are used to develop testing in software programming. Users have to find information about this element in Ruby (or Rails). |
| 13 | General features of an Iphone. |
| 14 | Information about the Journal Citation Reports publication in 2009. |
| 15 | Similar goal as query 12, but users have to look for testing fixtures in any programming language. |
| 16 | Information about the inhabitants of New Zealand. Users are not intended to look for information about the fruit or about the bird in New Zealand, also called 'kiwi'. |
| 17 | Information about the Earth radius. |
| 18 | General features of an Ipad. |
| 19 | Same goal as query 12, but focusing on fixtures for any programming language. |
| 20 | Same goal as query 16, but with a new term. |

The disambiguation of each query term was intended to be done with the Wikipedia concepts stated in Table 7. Users were not informed about the disambiguation of the terms, to analyse the possible issues in the process.

As the Wikipedia information stored for this dissertation corresponds to the English version, Spanish concepts – those starting with the prefix http://es.wikipedia... – were replaced with their counterparts from the English version.

The informational queries, the returned web resources, query disambiguation and users explicit feedback (relevance judgements of web resources) complete the dataset, obtaining a total of 6,556 annotations, 14,441 annotation terms, 42 different concepts and 2,386 web resources.

**Table 7. Queries disambiguation**

| Id query | Term | Concept |
|---|---|---|
| 1 | Árbol<br>[de hoja caduca] | http://es.wikipedia.org/wiki/Arbol<br>http://es.wikipedia.org/wiki/Caducifolio |
| 2 | Pisos<br>[alquiler con opción a compra]<br>Leganés | http://es.wikipedia.org/wiki/Casa<br>http://es.wikipedia.org/wiki/Arrendamiento_financiero<br>http://es.wikipedia.org/wiki/Legan%C3%A9s |
| 3 | [journal citation reports] | http://en.wikipedia.org/wiki/Journal_Citation_Reports |
| 4 | Árbol<br>[de hoja perenne] | http://es.wikipedia.org/wiki/Arbol<br>http://es.wikipedia.org/wiki/Perennifolio |
| 5 | Pisos<br>[alquiler con derecho a compra]<br>Leganés | http://es.wikipedia.org/wiki/Casa<br>http://es.wikipedia.org/wiki/Arrendamiento_financiero<br>http://es.wikipedia.org/wiki/Legan%C3%A9s |
| 6 | stars<br>hotel<br>hollywood | http://en.wikipedia.org/wiki/Movie_star<br>http://en.wikipedia.org/wiki/Hotel<br>http://en.wikipedia.org/wiki/Hollywood |
| 7 | Join<br>unix<br>separator | http://en.wikipedia.org/wiki/Join_(Unix)<br>http://en.wikipedia.org/wiki/Unix<br>http://en.wikipedia.org/wiki/Delimiter |
| 8 | JCR | http://en.wikipedia.org/wiki/Journal_Citation_Reports |
| 9 | Sun<br>England | http://en.wikipedia.org/wiki/Sun<br>http://en.wikipedia.org/wiki/England |
| 10 | [Rodrigo Rato]<br>Teatro | http://es.wikipedia.org/wiki/Rodrigo_Rato<br>http://es.wikipedia.org/wiki/Teatro |
| 11 | Join<br>unix<br>delimiter | http://en.wikipedia.org/wiki/Join_(Unix)<br>http://en.wikipedia.org/wiki/Unix<br>http://en.wikipedia.org/wiki/Delimiter |
| 12 | ruby<br>fixtures | http://en.wikipedia.org/wiki/Ruby_(programming_language)<br>http://en.wikipedia.org/wiki/Test_fixture |
| 13 | Iphone<br>features | http://en.wikipedia.org/wiki/Iphone<br>http://en.wikipedia.org/wiki/Feature_(software_design) |
| 14 | JCR<br>2009 | http://en.wikipedia.org/wiki/Journal_Citation_Reports<br>http://en.wikipedia.org/wiki/2009 |
| 15 | software<br>fixtures | http://en.wikipedia.org/wiki/Computer_software<br>http://en.wikipedia.org/wiki/Test_fixture |
| 16 | kiwi<br>[new zealand] | http://en.wikipedia.org/wiki/Kiwi_(people)<br>http://en.wikipedia.org/wiki/New_zealand |
| 17 | Earth<br>radius | http://en.wikipedia.org/wiki/Earth<br>http://en.wikipedia.org/wiki/Earth_radius |
| 18 | ipad<br>features | http://en.wikipedia.org/wiki/Ipad<br>http://en.wikipedia.org/wiki/Feature_(software_design) |
| 19 | [software testing]<br>fixtures | http://en.wikipedia.org/wiki/Software_testing<br>http://en.wikipedia.org/wiki/Test_fixture |
| 20 | kiwi<br>inhabitant<br>[new zealand] | http://en.wikipedia.org/wiki/Kiwi_(people)<br>http://en.wikipedia.org/wiki/Residency_(domicile)<br>http://en.wikipedia.org/wiki/New_zealand |

Finally, the parameter values used in the evaluation are specified in Table 8.

**Table 8. Parameter values for the evaluation**

| Parameter | Value | Description |
|---|---|---|
| *s* | 80 | Maximum number of web resources obtained from the web search engine (see section 6.2). Even though the precision rate has been calculated for the top 30 results (see section 8.2.2), *s* was set to 80 pages per query in order to obtain a big number of web documents for the repository. This is also the number of resources the application with Itaca layer will return in response after a search. |
| *x* | 15 | Position in the list of web resources obtained from the web search engine where resources become less relevant (see section 6.2). Users usually pay attention to the first or second page out of all the pages a search engine returns after the execution of a query. |
| $\mu$ | 0.8 | Minimum similarity to consider a concept similar to other (see section 6.3). |
| $\alpha$ | 0.6 | Weight factor given to the resources obtained from user annotations; resources obtained from the web search engine are given a weight of 0.4 (see section 6.6). |

Users involved in the evaluation were asked to execute and disambiguate the set of queries in traditional web search engines; more specifically, Google and Yahoo search engines. For every query and every web resource returned, users judged the relevance of the resource with respect to the concepts involved in that query.

### 8.2.2. Precision rate

The first evaluation process compares the precision of the results obtained through traditional search engines with the results obtained through Itaca layer and its ranking algorithm. The precision is calculated for the top 30 results returned for every query (P@30). Fig. 61 shows this precision rate for the 20 different informational queries.

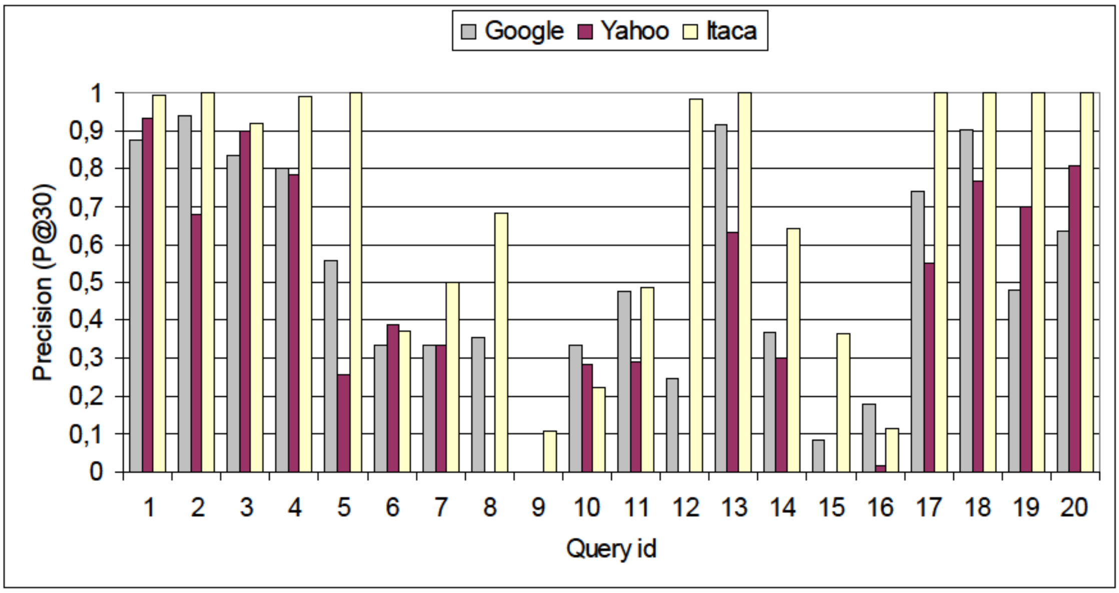


**Fig. 61. Precision rate obtained for Google, Yahoo and Itaca layer**

The graphic shows that the number of relevant results obtained is higher than in the well-known search engines in 90% of the evaluation set. More than that; in some special queries, relevance results with Itaca layer are overwhelming. This is the case of queries 9, 12 and 15, where Yahoo did not return any relevant web resource in their first 30 top results. Taking query 9 as an example - also with a precision rate of 0 in the case of

Google -, its string was "Sun England", referring to the weather in that country; however, at least the 30 first pages returned by these web search engines were related to the English newspaper.

Queries 10 and 16 present worse results applying Itaca layer in the searching process. The reason of the lack of precision in these cases was that query terms were annotated with wrong Wikipedia concepts by most of the users involved in the evaluation. For example, one of the concepts in query 10 was *Theatre*, referring to the act of playing. However, some users misunderstood the intention of the given query and annotated the term with the concept *Theatre (structure)*, referring to the building.

The second evaluation is executed to understand the extent of the annotations in Itaca layer and how they influence the final results. For that, the precision rate has been computed with different number of annotations.

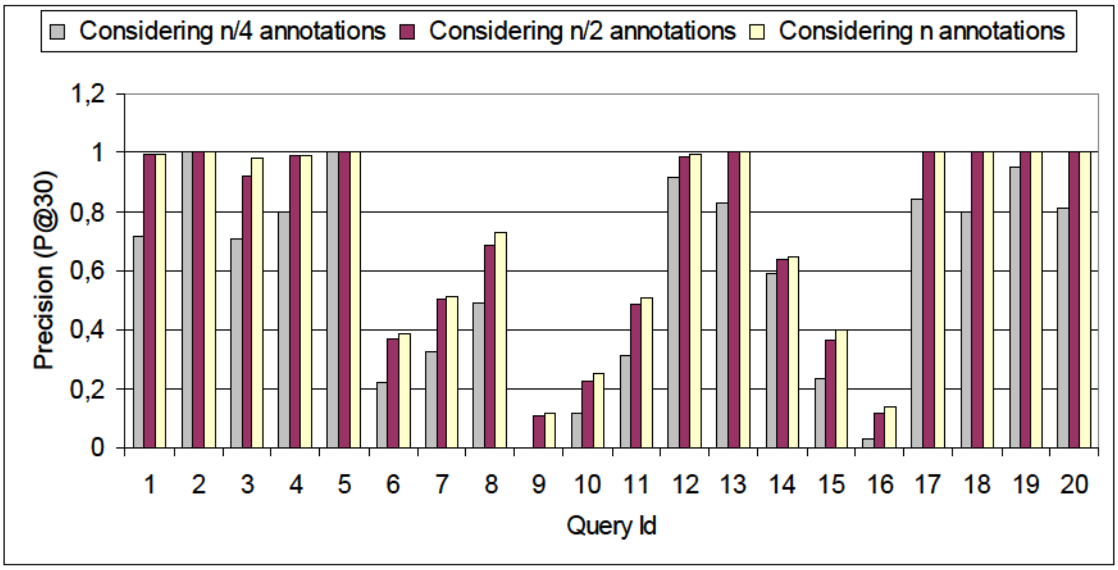


**Fig. 62. Precision rate with Itaca layer and different number of annotations**

Fig. 62 shows how the precision rate slightly increments when the ranking algorithm uses more annotations. However, increasing the number of annotations does not make precision increase at the same rate. Fig. 63 presents the same information of Fig. 62, but with a different view, where it is clearer that the precision rate tends to stabilise when the number of annotations increases.

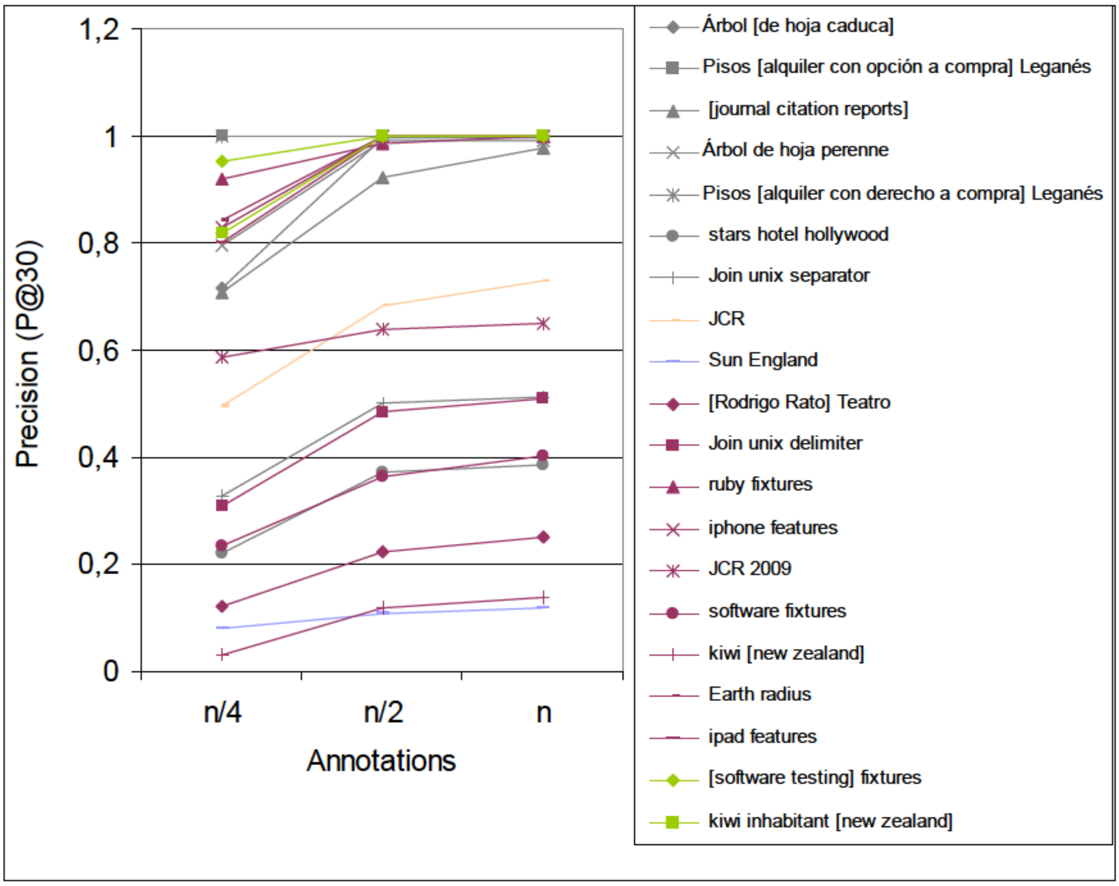


**Fig. 63. Precision rate with Itaca layer and different number of annotations, second view**

### 8.2.3. Response time

Fig. 64 shows the processing time (in seconds) needed for the ranking algorithm in Itaca layer to obtain the final results. The response time does not exceed 4.5 seconds except in one of the queries.

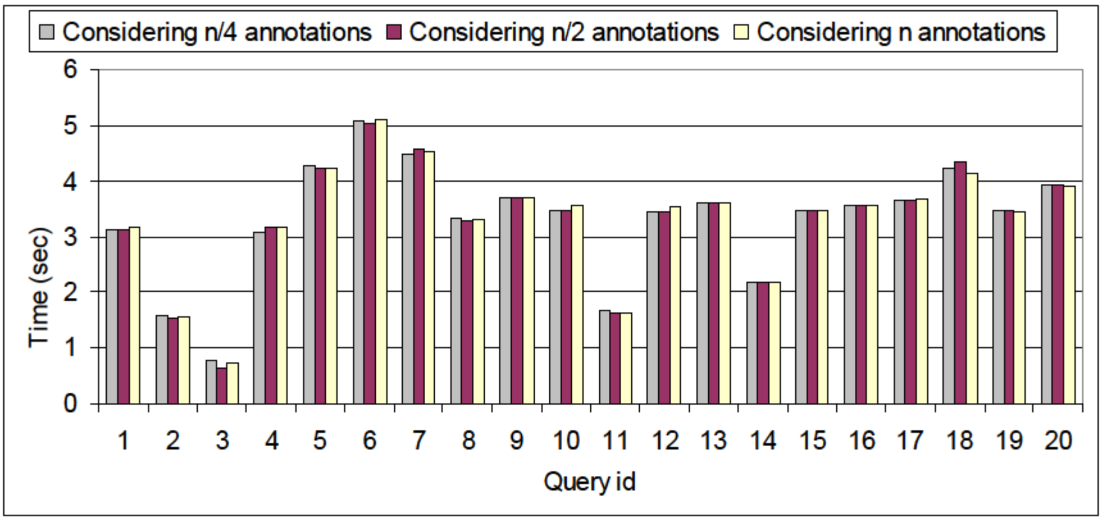


**Fig. 64. Response time obtained in Itaca layer with different number of annotations**

Due to the accumulators explained in section 6.4, the response time needed for the ranking algorithm does not increase when the number of user annotations also increases. However, response times can be altered when the number of web resources and concepts increases over time. To cope with this problem, resources and concepts are suitable to be stored in an inverted index architecture (see Fig. 65), already used for other purposes in retrieval tasks (see 2.1.1).

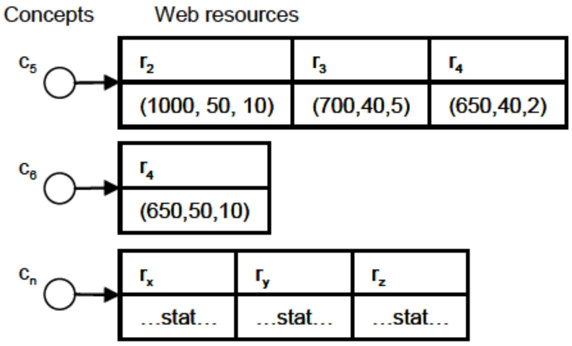


**Fig. 65. Structure of an inverted index for Itaca layer items**

The index should keep a list of the different concepts. For each concept, this index should store a list of the web resources annotated with that concept. Finally, the statistics for every concept-resource pair should also be stored, where these statistics are the triple values of the accumulators representing the concept-resource pair (number of times it has been annotated as related, unrelated or indifferent).

As large collections of resources may be involved, indexing should have to be distributed over computer clusters. In fact, web search engines use distributed indexing algorithms for index construction, and these algorithms can be exploited as well by Itaca layer. As index construction is not in the scope of this dissertation, see (Manning et al., 2008) for more information.

## 8.3 HYPOTHESIS 3 EVALUATION: SIMILARITY PROCESSOR

> *Wikipedia is a valid source to calculate semantic similarity. Its application in a semantic similarity method can yield as good results as existing techniques with WordNet and other knowledge sources.*

The selection of the structure of categories in Wikipedia and their subsequent processing to be applied in existing path-based and multi-source metrics allows calculating the semantic similarity of two concepts, yielding the same or even better results than the original techniques with other knowledge sources. To prove this hypothesis, the evaluation compares the correlation coefficient obtained using Wikipedia with the adapted measures developed as Goal 3, and the correlation coefficient obtained by the original techniques. Besides, the adapted measures are compared with other Wikipedia-based solutions.

### 8.3.1. DATA SET

The same data set used in the evaluations of previous works, those of Rubenstein and Goodenough's (Rubenstein & Goodenough, 1965), has been taken to prove the third hypothesis. More specifically, the test set is composed by the 28 pairs traditionally used for evaluation, and the training set, used to tune the adapted measures, is composed of the remaining 37 pairs out the 65.

Notice that the terms used in Rubenstein & Goodenough's work are not concepts, but merely bag of words, and there is no information about the sense of those words. As this thesis works with disambiguated Wikipedia entities, a pair of concepts - senses - has to be assigned to every word pair in both training and test sets. Table 9 and Table 10 show the concepts that identify the words in the sets of Rubenstein & Goodenough's work. The *Wikipedia concepts* column represents the URIs of the concepts without the prefix *http://en.wikipedia.org/wiki/*.

**Table 9. Correspondence between original pairs and Wikipedia concepts, training set**

| **Original pairs** | | **Wikipedia concepts** | |
|---|---|---|---|
| Asylum | Cemetery | Psychiatric_hospital | Cemetery |
| Asylum | Fruit | Psychiatric_hospital | Fruit |
| Asylum | Monk | Psychiatric_hospital | Monk |
| Autograph | Shore | Autograph | Shore |
| Autograph | Signature | Autograph | Signature |
| Automobile | Wizard | Automobile | Magician_(fantasy) |
| Automobile | Cushion | Automobile | Cushion |
| Bird | Woodland | Bird | Woodland |
| Boy | Rooster | Boy | Rooster |
| Boy | Sage | Boy | Philosophy |
| Cemetery | Mound | Cemetery | Mound |
| Cemetery | Graveyard | Cemetery | Graveyard |
| Cemetery | Woodland | Cemetery | Woodland |
| Cord | String | Rope | Rope |
| Cock | Rooster | Rooster | Rooster |
| Crane | Rooster | Crane_(bird) | Rooster |
| Cushion | Jewel | Cushion | Jewellery |
| Cushion | Pillow | Cushion | Pillow |
| Forest | Woodland | Forest | Woodland |
| Fruit | Furnace | Fruit | Furnace |
| Furnace | Implement | Furnace | Tool |
| Glass | Jewel | Glass | Jewellery |
| Glass | Tumbler | Glass | Glass |
| Graveyard | Madhouse | Graveyard | Psychiatric_hospital |
| Grin | Implement | Smile | Tool |
| Grin | Lad | Smile | Boy |
| Grin | Smile | Smile | Smile |
| Hill | Mound | Hill | Mound |
| Hill | Woodland | Hill | Woodland |
| Magician | Oracle | Magician_(fantasy) | Oracle |
| Mound | Stove | Mound | Stove |
| Mound | Shore | Mound | Shore |
| Oracle | Sage | Oracle | Philosophy |
| Sage | Wizard | Philosophy | Magician_(fantasy) |
| Serf | Slave | Serfdom | Slavery |
| Shore | Voyage | Shore | Travel |
| Shore | Woodland | Shore | Woodland |

**Table 10. Correspondence between original pairs and Wikipedia concepts, test set**

| Original pairs | | Wikipedia concepts | |
|---|---|---|---|
| Asylum | Madhouse | Psychiatric_hospital | Psychiatric_hospital |
| Automobile | Car | Automobile | Automobile |
| Bird | Cock | Bird | Rooster |
| Bird | Crane | Bird | Crane_(bird) |
| Boy | Lad | Boy | Boy |
| Brother | Lad | Sibling | Boy |
| Brother | Monk | Broter_(Catholic) | Monk |
| Car | Journey | Automobile | Travel |
| Cord | Smile | Rope | Smile |
| Coast | Forest | Coast | Forest |
| Coast | Hill | Coast | Hill |
| Coast | Shore | Coast | Shore |
| Crane | Implement | Crane_(machine) | Tool |
| Food | Fruit | Food | Fruit |
| Food | Rooster | Food | Rooster |
| Forest | Graveyard | Forest | Graveyard |
| Furnace | Stove | Furnace | Stove |
| Gem | Jewel | Jewellery | Jewellery |
| Glass | Magician | Glass | Magician_(fantasy) |
| Implement | Tool | Tool | Tool |
| Journey | Voyage | Travel | Travel |
| Lad | Wizard | Boy | Magician_(fantasy) |
| Magician | Wizard | Magician_(fantasy) | Magician_(fantasy) |
| Midday | Noon | Noon | Noon |
| Monk | Oracle | Monk | Oracle |
| Monk | Slave | Monk | Slavery |
| Noon | String | Noon | Rope |
| Rooster | Voyage | Rooster | Travel |

### 8.3.2. Evaluation

In order to homogenise the results, I have recalculated the correlation coefficients of the traditional measures explained in section 3 for test and training sets. The reason of doing this task is twofold. First, metrics results were compared with distinct human judgements for the dataset - some reported correlations were obtained after comparing the results of the metrics to the human values of Rubenstein & Goodenough's experiments and some other were obtained after the comparison with those of Miller and Charles's -. Second, the metrics which used the WordNet taxonomy as their knowledge source did not use the same version - versions used goes from WordNet 1.5 to WordNet 1.7 -.

To solve the first issue, results obtained after the replication have been compared with a single set of human judgements - the ones of Rubenstein & Goodenough's work -, avoiding the problem of correlating with different set values. For the second issue, we have used 1) the Semantic Similarity System[27] (SSST) to replicate path-based and multi-source methods; and 2) the Google web search engine to replicate the co-occurrence metrics.

[27] Semantic Similarity System Tool: http://www.intelligence.tuc.gr/similarity/index.php

The replication has been made for both the test and training sets. The usage of SSST allows working with the same WordNet version for every measure - the tool uses WordNet 2.0 -, and the replication of co-occurrence metrics through Google allows working with the same Web status. Blázquez-del-Toro et al.'s work was not replicated because their measure was not available on SSST and its implementation supposes the transformation of the structure - in this case, the Wikipedia structure - into an ontology.

Table 11 shows the replicated correlation coefficients for test and training sets. As far as the test set is concerned, some methods yield lower values than the reported ones (see Table 3). This can be due to an increment in the number of concepts in the taxonomy for new versions of WordNet, and the increment of indexed documents for the co-occurrence web based methods, but this issue is out of the scope of this thesis. Besides, a weighted average coefficient has been also calculated for the whole collection (65 pairs), to obtain an approximation without overfitting to a particular subset.

**Table 11. Replicated correlation coefficients for existing measures**

| Semantic similarity measure | Training set | Test set | Whole set |
|---|---|---|---|
| Co-occurrence based | | | |
| Cilibrasi & Vetanyi (2007) | 0.54 | 0.51 | 0.52 |
| Bollegalla (2007) | 0.67 | 0.76 | 0.70 |
| Path-based | | | |
| Rada et al. (1989) | 0.55 | 0.62 | 0.58 |
| Wu & Palmer (1994) | 0.81 | 0.75 | 0.78 |
| Leacock & Chodorow (1994) | 0.86 | 0.83 | 0.84 |
| Multi-source based | | | |
| Resnik (1995) | 0.88 | 0.77 | 0.83 |
| Jiang & Conrath (1997) | 0.85 | 0.83 | 0.84 |
| Lin (1998) | 0.89 | 0.82 | 0.85 |
| Li et al. (2003) | 0.87 | 0.82 | 0.84 |

The adapted measures exposed in section 7.5 were trained with the training set and then executed with the test set. Table 12 shows the results of these adapted measures for Wikipedia. The table still shows the coefficients for the original version of the measure for comparison purposes, extracted from Table 11. When parameters are needed, the table displays the value which maximises the results. In the same way, when adapted approach is composed of 9 different measures, they are grouped in 3 main subsets (the set which applies the minimum, average and maximum functions respectively), and only the best value is selected. For each set (column), the best value is printed in bold.

After the experiments, best results for training and test datasets are achieved with the adapted measure of Blázquez-del-Toro et al's and Li et al.'s respectively, whereas Blázquez-del-Toro et al's measure maximizes the whole data collection. From the results reported in Table 12, some conclusions can be drawn. First, when the category distance is the only feature of the measure to adapt, the minimum value among all the LCSs is that with best results (see *rada_min* and *lc_min* values). Second, when taking depth feature as one of the factors, the set of values obtained with the average of depths of the set of the LCSs between two concepts yields better correlation values. Therefore, it gives better correlation to consider the average height of every *lcs* between the categories of concepts, instead of selecting a minimum or maximum value.

**Table 12. Correlation of the path-based original measures' adaptations**

| Semantic similarity measure | Training set | Test set | Whole set |
|---|---|---|---|
| Rada et al. (1989) | *0.55* | *0.62* | *0.58* |
| rada_min | 0.72 | 0.78 | 0.74 |
| rada_avg | 0.57 | 0.54 | 0.55 |
| rada_max | 0.09 | 0.24 | 0.15 |
| Wu & Palmer (1994) | *0.81* | *0.75* | *078* |
| wp_min_ (using $wp_{max}$ set) | 0.19 | 0.18 | 0.18 |
| wp_avg_ (using $wp_{min}$ set) | 0.75 | 0.77 | 0.75 |
| wp_max_ (using $wp_{avg}$ set) | 0.78 | 0.81 | 0.79 |
| Leacock & Chodorow (1994) | *0.86* | *0.83* | *0.84* |
| lc_min | 0.74 | 0.63 | 0.69 |
| lc_avg | 0.62 | 0.49 | 0.56 |
| lc_max | 0.41 | 0.32 | 0.37 |
| Blázquez-del-Toro et al. (2008) | | | |
| bl_min_ (k = 2.5, using $bl_{max}$ set) | 0.30 | 0.21 | 0.26 |
| bl_avg_ (k = 2.0, using $bl_{min}$ or $bl_{avg}$ set) | 0.78 | 0.82 | 0.79 |
| bl_max_ (k = 0.25, using $bl_{avg}$ set) | **0.80** | 0.84 | **0.81** |
| Li et al. (2003) | *0.87* | *0.82* | *0.84* |
| li_min_ (α = 0.4; β = 1, using $li_{max}$ set) | 0.77 | 0.84 | 0.80 |
| li_avg_ (α = 0.35; β = 1, using $li_{max}$ set) | 0.77 | 0.82 | 0.79 |
| li_max_ (α = 0.35; β = 0.2, using $li_{avg}$ set) | 0.77 | **0.85** | 0.80 |

### 8.3.3. Discussion

The third hypothesis is proved looking at the results in the previous section and comparing them with the results of existing measures explained in chapter 3, namely existing path-based and multi-source methods and Wikipedia-based approaches.

First, final results show that Wikipedia is a knowledge source as faithful as well-formed taxonomies like WordNet or other dictionaries and corpora for calculating semantic similarity using an existing measure based on a lexical structure. For comparative purposes, Table 13 shows, in ascending order, the correlation coefficients of the original measures and the best adapted measure, for both the test set (28 pairs) and the whole set (65 pairs).

**Table 13. Correlation coefficients for test and whole set**

| Measure | Test set | Measure | Whole set |
|---|---|---|---|
| Shortest path | 0.62 | Cilibrasi and Vetanyi | 0.52 |
| Wu and Palmer | 0.75 | Shortest path | 0.58 |
| Bollegalla | 0.76 | Bollegalla | 0.70 |
| Resnik | 0.77 | Wu and Palmer | 0.78 |
| Blázquez-del-Toro et al. | 0.81 | Adapted (bl_max_avg) | **0.81** |
| Lin | 0.82 | Resnik | 0.83 |
| Li et al. | 0.82 | Leacock and Chodorow | 0.84 |
| Leacock and Chodorow | 0.83 | Jiang and Conrath | 0.84 |
| Jiang and Conrath | 0.83 | Li et al. | 0.84 |
| Adapted (li_max_avg) | **0.85** | Lin | 0.85 |

Best adapted results approximate and even improve traditional approaches with WordNet, as in the case of the test set. Even though the best correlation obtained for the whole set is slightly smaller than those obtained with other traditional sources, it is still over a correlation of 0.80. Besides, adapted measures take the inherent advantages of using Wikipedia, such as greater coverage, multiple domains, or the possibility of comparing concepts from different parts of speech, showing that Wikipedia is another valid source to calculate semantic similarity, obtaining better correlation than existing works.

Table 14 shows the coefficients in ascending order reported in Wikipedia-based measures and the result of the adaptation of Li et al.'s measure for the test set[28], which clearly improves them. However, the results of these existing Wikipedia-based methods cannot be directly compared in this evaluation, due to the different experimental sets used, explained in section 3.3.

**Table 14. Correlation coefficients of Wikipedia-based measures**

| Measure | |
|---|---|
| Zhang et al. (2011) | 0.56 |
| Strube and Ponzetto (2006) | 0.56 |
| Wee and Hassan (2008) | 0.60 |
| Milne and Witten (2008) | 0.64 |
| Nastase and Strube (2013) | 0.70 |
| Gabrilovitch and Markovitch (2007) | 0.75 |
| Adapted measure (li_max_avg) | **0.85** |

The general process to compute the semantic similarity of two concepts with the adapted measures is simple and cost-effective, because there is no need to process big amounts of text corpora like in (Resnik, 1995). The structure - the Wikipedia categorization taxonomy - is used as it is; there is no need to modify the underlying taxonomy as in the original measure from (Blázquez-del-Toro et al., 2008) or generate a new taxonomy from the category structure such as in (Nastase & Strube, 2013). Results obtained with the peculiar structure of Wikipedia are promising, in the sense that they may be applied to other non-well-formed hierarchies, even though this is out of the scope in this thesis.

[28] Note that, as information about training set in Wikipedia-based metrics is inexistent, the table just shows the test set correlation coefficient.

# 9 CONCLUSIONS AND FUTURE WORK

This chapter offers an overview of the main important aspects related to the present dissertation, the main goals achieved and the hypothesis proved. Some future points will be listed in order to enhance the work and to encourage further research about the initial thesis proposed here.

## 9.1 Brief resume

The *main purpose* of the present dissertation is developing an infrastructure to obtain more relevant web pages from a large-scale, traditional web search engine. It makes use of semantic and social techniques but, instead of building a new information retrieval system from scratch, a semantic layer is proposed, Itaca. This layer, mainly composed of designed algorithms and gathered data, can be easily settled on top of the architecture of current search engines.

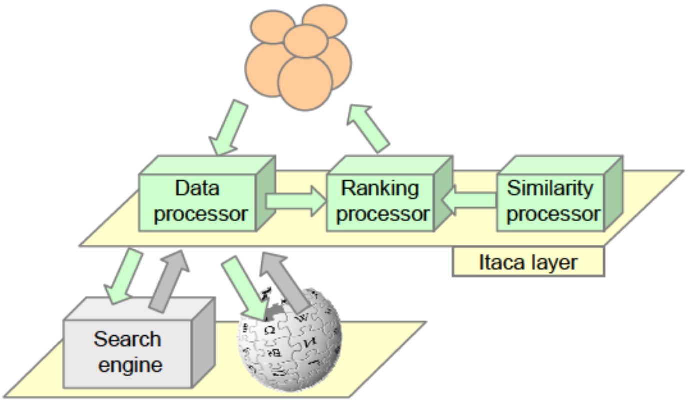


**Fig. 66. General overview with Itaca layer**

The dissertation takes into account two of the basic problems that still appear in well-known web search engines: 1) the loneliness of the searching process; and 2) the simple recovery techniques, based mainly on offering the documents that contain the exact terms used to describe a query.

For this thesis, the proposed layer relies on semantic annotations to unambiguously describe queries and web documents. These annotations are gathered by means of the collaborative usage of information generated by users while searching, obtained through explicit relevance feedback techniques.

This dissertation uses Wikipedia as the source for the semantic annotations. It is basically composed of articles, which define and describe concepts. Each of these articles is referenced by a unique identifier. Every element involved in a searching process, like queries and documents, can then be related to the particular Wikipedia article it is referring to. Wikipedia offers more advantages than WordNet or domain-specific taxonomies:

- Greater coverage over a variety of domains
- Specific concepts such as named entities and specific nouns
- Flexible and rapid updates
- Elaborated by consensus of a community
- Different parts of speech (nouns, verbs, adjectives) coexisting in the same structure
- Translated to different languages

These properties have made a suitable knowledge source for semantic annotations.

Itaca extends the functional capabilities of current web search engines, providing a new architecture and ranking algorithm without getting rid of traditional ranking models. Experiments show that this new architecture offers more precision in the final results obtained, keeping the simplicity and usability of the web search engines existing so far. Its particular design as a layer makes feasible its inclusion to current engines in a simple way.

## 9.2 Initial goals achieved

The main goals attained in the development of Itaca layer consist on:

- The implementation of a ranking algorithm that, using semantic annotations obtained from user feedback information, produces more relevance results after a search than a traditional search engine alone.
- The implementation of a similarity algorithm - in this case, the adaptation of an existing one - that, given two Wikipedia concepts, automatically determines a score that indicates their similarity at semantic level. This algorithm is fully automatic and can be used independently of the domain of the concepts.
- Both algorithms are settled in a layer, Itaca, which takes advantage of collaborative tagging and filtering to semantically annotate the resources these algorithms need. This is achieved by a guided graphical user interface which does not require any expert knowledge about taxonomies or special languages to define queries.

Every stated goal has been proved with their initial hypotheses.

## 9.3 Contributions

The main contributions of this dissertation can be resumed in the following list:

- Design of a new semantic search model set over current search engines that, using Wikipedia concepts, allows for more accurate results than traditional web searches.
- Implementation of a new ranking algorithm based on this model.
- Implementation of a semantic search engine based on this algorithm.
- Design of a procedure to adapt existing semantic similarity measures based on lexical structures to the Wikipedia categorization taxonomy.
- Creation of an evaluation benchmark for future research in semantic search and semantic similarity.

The main advantage of the new layer is that it can coexist with existing traditional search engines and enhance their results. The collaborative process of annotation makes the search task a social process where users can take benefit from each other.

An additional part of this dissertation and its results is that it indicates that a collaboratively-created structure like Wikipedia can actually be used in the fields of information retrieval or natural language processing with the same quality as well-formed taxonomies or ontologies.

## 9.4 FUTURE WORK

Finally, this section mentions potential lines and tips for further research.

Wikipedia has been used as the only knowledge source for the core search engine and for the semantic similarity algorithm. Other approaches used WordNet instead. However, results might be improved if both or more sources were available. Such idea of unifying knowledge structures have been already covered in (Suchanek, Kasneci, & Weikum, 2007), where authors present YAGO, a light-weight and extensible ontology, or in (Nastase & Strube, 2013). Even though this dissertation focuses on the simplicity, the use of combined sources might boost the precision rate of the thesis presented here.

In a general search, users would take benefit from Wikipedia categories if they could find the most relevant pages given a certain category, instead of the pages from a certain query. For this purpose, a static method should be implemented, in order to recover the most relevant documents under a category. This could be considered as a facility for browsing the Web instead of searching for informational queries.

Social network theories can also enhance the search process. By constructing topic experience profiles for each user, Itaca could infer who in the social network knows what and who the most trustworthy source of information on a topic is. For example, if a web resource about "semantic web" has been frequently selected by many semantic-web experts, it may be a high quality document on this topic. The reinforcement of the algorithm with users' expertise can be also enhanced with queries themselves; that is, a query may be of high quality if it can retrieve high quality resources.

Individual search archives could also be provided. Users could view their top searches, the most frequently visited pages, and the annotations they issued in these pages.

For automatic word sense disambiguation of query terms - useful in the first step of the searching process -, Itaca can take benefit of models such as that proposed on (Mihalcea, 2007). However, these approaches are developed to work within a wider context than a query, and the meaning of an ambiguous term is selected based on the context of the corpora where it occurs.

For semantic annotation and obtaining relevance feedback about a document in the last step of the searching process, works like Wikify (Csomai & Mihalcea, 2008) can be used for automatic keyword extraction. Specifically, given an input document, the Wikify system could identify the important concepts in the text of web pages and link them to the corresponding Wikipedia concepts. This would not mean that the web page truly refers to those concepts, but it could be an input to consider in Itaca ranking algorithm.

Finally, more hypotheses can be formulated, with their respective evaluations:

- Level of user satisfaction with respect to the graphical user interface.
- Total time saved during collaborative searching as compared with traditional personal web searching, with the equivalent set of informational queries to search.

# PART IV. Appendices and References

# Appendix A. Acronyms and Definitions

This appendix lists the most important acronyms and definitions used throughout this thesis.

## 9.4.1. A.1. Acronyms

**DRY**

Don't Repeat Yourself

**GUI**

Graphical User Interface

**HTML**

HyperText Markup Language

**HTTP**

HyperText Transfer Protocol

**IR**

Information retrieval

**LCS**

Least Common Subsumer

**NGD**

Normalized Google Distance

**NLP**

Natural Language Processing

**PMI**

Pointwise Mutual Information

**RDF**

Resource Description Framework

**SPARQL**

Simple Protocol And RDF Query Language

**SSST**

Semantic Similarity System Tool

**SVM**

Support Vector Machine/Model

**TF x IDF**

Term Frequency x Inverse Document Frequency

**TREC**

Test REtrieval Conference

**URI**

Unified Resource Identifier

**URL**

Unified Resource Locator

**VSM**

Vector Space Model

**WLM**

Wikipedia Link-based Measure

**YAGO**

Yet Another Great Ontology

### 9.4.2. A.2. Definitions

**Cosine similarity**

The cosine similarity measures the angle between two vectors *A* and *B*, which determines whether these vectors are pointing in roughly the same direction:

$$\cos(A,B) = \frac{A \times B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

**Equation 60. Cosine similarity metric**

In information retrieval, the attribute vectors are usually the term frequency vectors of documents. The cosine similarity of two documents will range from 0 to 1, since the term frequencies cannot be negative.

**Cycle**

A cycle in a graph is a path from a node to itself.

**Dice's coefficient**

It is a similarity measure between sets, and is defined as twice the size of the intersection divided by the sum of the size of each of the sets:

$$Dice's(A,B) = \frac{2|A \cap B|}{|A| + |B|}$$

**Equation 61. Dice's coefficient**

**Graph**

A graph is a representation of a set of objects, also called *nodes*, where some pairs of the objects are connected by links, also called *edges*.

**Hierarchy**

A hierarchy is an arrangement of items (objects, categories, etc.) in which the items are represented as being "above," "below," or "at the same level as" one another. A hierarchy can be modelled mathematically as a rooted *tree*.

**Hyponym / Hypernym**

In linguistics, a hyponym is a more specific term; a subordinate grouping word or phrase whose semantic field is included within that of another word, its hypernym.

**Information retrieval**

Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers). The field of IR also covers supporting users in browsing or filtering document collections or further processing - such as classifying - a set of retrieved documents.

**Jaccard coefficient**

It is a similarity measure between sets, and is defined as the size of the intersection divided by the size of the union of the sets:

$$Jaccard(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

**Equation 62. Jaccard coefficient**

**Overlap coefficient**

It is a similarity measure between sets, and is defined as the size of the intersection divided by the size of the minimum set:

$$Overlap(A,B) = \frac{|A \cap B|}{\min\{|A|,|B|\}}$$

**Equation 63. Overlap coefficient**

**Path**

A path in a graph is a sequence of edges which connect a sequence of nodes.

**PMI coefficient**

It is a measure of association between two discrete items that quantifies the discrepancy between the probability of their coincidence given their joint distribution and their individual distributions, assuming independence:

$$PMI(a,b) = \log \frac{p(a,b)}{p(a) \times p(b)}$$

**Equation 64. PMI coefficient**

**Taxonomy**

A taxonomy is a classification of a particular domain, arranged in a hierarchical structure. Typically, it is organized by hyponym-hypernym relationships, also called *generalization-specialization* relationships, or, less formally, *parent-child* relationships. In such an inheritance relationship, the hypernym has the same properties, behaviours, and constraints as the hyponym plus one or more additional properties, behaviours, or constraints. For example, *car* is a hyponym of *vehicle*. So any *car* is also a *vehicle*, but not every *vehicle* is a *car*.

**Tree**

A tree is an acyclic graph in which edges have no orientation.

# APPENDIX B. DISSEMINATION

Main contributions of this thesis (international journals) are listed here by year in descending order:

- Fuentes-Lorenzo, D., Fernández, N., Fisteus, J. A. & Sánchez, L. (2013). *Improving large-scale search engines with semantic annotations*. In *Expert Systems With Applications*, 40(6), pp. 2287-2296.
  Impact factor (2013): 1.965.
- Fernández, N., Fisteus, J. A., Sánchez, L. & Fuentes-Lorenzo, D. (2012). *WikiIdRank: An unsupervised approach for entity linking based on instance co-occurrence*. In *Innovative Computing Information and Control*, 8(11), pp. 7519-7541.
- Fernández, N., Fisteus, J. A., Fuentes, D., Sánchez, L. & Luque, V. (2011). *A Wikipedia-Based Framework For Collaborative Semantic Annotation*. In *International Journal on Artificial Intelligence Tools*, 20(5), 847-886.
  Impact factor: 0.217.

Other main works of the author during this dissertation period are:

- Fuentes-Lorenzo, D., Sánchez, L. & Cuadra, A., Cutanda, M. (2014). A RESTful and Semantic Framework for Data Integration. In *Software Practice & Experience* (to publish).
  Impact factor: 1.008.
- Fuentes-Lorenzo, D., Sánchez L., Cuadra Sánchez & Cutanda Rodríguez, M. M. (2011). *Managing Legacy Telco Data using RESTful Web Service*. In C. Pautasso & E. Wilde (Eds.), *REST: From Research to Practice* (pp. 303-317). Springer.
- Cuadra, A., Cutanda, M. M., Fuentes-Lorenzo, D. & Sánchez, L. (2011). *A Semantic Web-based Integration Framework. Seventh International Conference on Next Generation Web Services Practices (NWeSP' 11)*, 19-21 October, Salamanca, Spain.
- Fernández, N., Fuentes-Lorenzo, D., Sánchez, L. & Fisteus, J. A. (2010). *The NEWS ontology: design and applications*. In *Expert Systems With Applications*, 37(12), 8694-8704.
  Impact factor: 1.926.
- Fuentes-Lorenzo, D., Morato, J, & Gómez, J. M. (2009). *Knowledge Management in Biomedical Libraries: A Semantic Web Approach*. In *Information Systems Frontier*, 11(4), 471-480.
  Impact factor: 1.309.

Further published works can be found at http://www.it.uc3m.es/dfuentes/index.html

# REFERENCES

Baeza-Yates, R., Hurtado, C., Mendoza, M., & Dupret, G. (2005). Modeling user search behavior. Proceedings of the Third Latin American Web Congress (LA-WEB'2005), Buenos Aires, Argentina, 242-251.

Baeza-Yates, R., & Tiberi, A. (2007). Extracting semantic relations from query logs. *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* San Jose, California, USA, 76-85.

Bao, S., Xue, G., Wu, X., Yu, Y., Fei, B., & Su, Z. (2007). Optimizing web search using social annotations. *Proceedings of the 16th International Conference on World Wide Web (WWW 2007),* Banff, Alberta, Canada, 501-510.

Bates, M. E., & Anderson, D. (2002). *Free, fee-based and value-added information services* Factiva, Dow-Jones Reuters Business Interactive, LLC.

Begelman, G., Keller, P., & Smadja, F. (2006). Automated tag clustering: Improving search and exploration in the tag space. *Collaborative Web Tagging Workshop at WWW2006,* Edinburgh, Scotland.

Bernardini, A., Carpineto, C., & D'Amico, M. (2009). Full-subtopic retrieval with keyphrase-based search results clustering. *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT '09),* Milan, Italy, 206-213.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. *Scientific American, 284*(5), 34-44.

Blázquez-del-Toro, J. M., Fisteus, J. A., Centeno, V. L., & Sánchez-Fernández, L. (2008). A semantic similarity measure in the context of semantic queries. *International Journal of Computer Applications in Technology, 33*(4), 285-291.

Bollegala, D., Matsuo, Y., & Ishizuka, M. (2007). Measuring semantic similarity between words using web search engines. *Proceedings of the 16th International Conference on World Wide Web (WWW '07),* Banff, Alberta, Canada. 757-766.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems, 30*(1-7), 107-117.

Carpineto, C., Mizzaro, S., Romano, G., & Snidero, M. (2009). Mobile information retrieval with search results clustering: Prototypes and evaluations. *Journal of American Society for Information Science and Technology, 60*(5), 877-895.

Castells, P., Fernández, M., & Vallet, D. (2007). An adaptation of the vector-space model for ontology-based information retrieval. *IEEE Transactions on Knowledge and Data Engineering, 19*(2), 261-272.

Cho, J., Roy, S., & Adams, R. E. (2005). Page quality: In search of an unbiased web ranking. *SIGMOD '05: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data,* Baltimore, Maryland, 551-562.

Church, K., & Hanks, P. (1990). Word association norms, mutual information, and lexicography. *Computational Linguistics, 16*(1), 22-29.

Cilibrasi, R. L., & Vitanyi, P. M. B. (2007). The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering, 19*(3), 370-383.

Csomai, A., & Mihalcea, R. (2008). Linking documents to encyclopedic knowledge. *IEEE Intelligent Systems, 23*(5), 34-41.

Damme, C. v., Hepp, M., & Siorpaes, K. (2007). FolksOntology: An integrated approach for turning folksonomies into ontologies. *Proceedings of the ESWC 2007, Workshop "Bridging the Gap between Semantic Web and Web 2.0",* Innsbruck, Austria. 57-70.

d'Aquin, M., & Motta, E. (2011). Watson, more than a semantic web search engine. *Semant.Web, 2*(1), 55-63.

Experian Hitwise. (2011). Experian hitwise reports bing.com searches increase 5 percent in february 2011. Retrieved from http://www.hitwise.com/us/press-center/press-releases/experian-hitwise-reports-bing-search-increase/

Fernández García, N., Blázquez del Toro, José María, Sánchez Fernández, L., & Luque Centeno, V. (2006). Exploiting wikipedia in integrating semantic annotation with information retrieval. In M. Last, P. S. Szczepaniak, Z. Volkovich & A. Kandel (Eds.), *Advances in web intelligence and data mining* (pp. 61-70) Springer.

Fernández, N., Fisteus, J. A., Fuentes, D., Sánchez, L., & Luque, V. (2011). A wikipedia-based framework for collaborative semantic annotation. *International Journal on Artificial Intelligence Tools, 20*(05), 847-886.

Fernández, M., Cantador, I., López, V., Vallet, D., Castells, P., & Motta, E. (2011). Semantically enhanced information retrieval: An ontology-based approach. *Web Semantics: Science, Services and Agents on the World Wide Web, 9*(4), 434-452.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., & Ruppin, E. (2002). Placing search in context: The concept revisited. *ACM Transactions on Information Systems, 20*(1), 116-131.

Flanagan, D., & Matsumoto, Y. (2008). *The ruby programming language* O'Reilly.

Gabrilovich, E., & Markovitch, S. (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. *Proceedings of the Twentieth International Joint Conference for Artificial Intelligence,* 1606-1611.

Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM, 35*(12), 61-70.

Golder, S. A., & Huberman, B. A. (2006). Usage patterns of collaborative tagging systems. *Journal of Information Science, 32*(2), 198-208.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition, 5*(2), 199-220.

Gulli, A., & Signorini, A. (2005). The indexable web is more than 11.5 billion pages. *Special Interest Tracks and Posters of the 14th International Conference on World Wide Web*

*(WWW '05),* Chiba, Japan, 902-903.

Haase, P., & Siebes, R. (2004). Peer selection in peer-to-peer networks with semantic topologies. *Proceedings of the 13th World Wide Web Conference (WWW'04),* New York, USA. 108-125.

Hansen, M. H., & Shriver, E. (2001). Using navigation data to improve IR functions in the context of web search. *Proceedings of the Tenth International Conference on Information and Knowledge Management (CIKM'01),* Atlanta, Georgia, USA, 135-142.

Haveliwala, T. H. (2002). Topic-sensitive PageRank. *Proceedings of the 11th International Conference on World Wide Web,* Honolulu, Hawaii, USA, 517-526.

Hawking, D., Craswell, N., Bailey, P., & Griffihs, K. (2001). Measuring search engine quality. *Information Retrieval, 4*(1), 33-59.

Heflin, J., & Hendler, J. A. (2000). Dynamic ontologies on the web. *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence,* Austin, Texas, USA, 443-449.

Hepp, M., Siorpaes, K., & Bachlechner, D. (2007). Harvesting wiki consensus: Using wikipedia entries as vocabulary for knowledge management. *IEEE Internet Computing, 11*(5), 54-65.

Heymann, P., Koutrika, G., & Garcia-Molina, H. (2008). Can social bookmarking improve web search? *Proceedings of the International Conference on Web Search and Web Data Mining (WSDM '08),* Palo Alto, California, USA, 195-206.

Hotho, A., Jäschke, R., Schmitz, C., & Stumme, G. (2006). Information retrieval in folksonomies: Search and ranking. *Proceedings of the 3rd European Semantic Web Conference (ESWC 2006),* Budva, Montenegro, 411-426.

Hruschka, E. R., Campello, R., Freitas, A. A., & De Carvalho, A. (2009). A survey of evolutionary algorithms for clustering. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 39*(2), 133-155.

Jansen, B. J., Booth, D. L., & Spink, A. (2008). Determining the informational, navigational, and transactional intent of web queries. *Information Processing and Management, 44*(3), 1251-1266.

Jiang, J. J., & Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. *Proceedings of the 10th International Conference on Research in Computational Linguistics,* Taiwan, 19-33.

Jiang, P., Hou, H., Chen, L., Chen, S., Yao, C., Li, C., & Wang, M. (2013). Wiki3C: Exploiting wikipedia for context-aware concept categorization. *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining,* Rome, Italy, 345-354.

Jie, S., Chen, C., Hui, Z., Rong-Shuang, S., Yan, Z., & Kun, H. (2008). TagRank: A new rank algorithm for webpage based on social web. *Proceedings of the 2008 International Conference on Computer Science and Information Technology (ICCSIT'08),* Singapure, 254-258.

Joachims, T. (2002). Optimizing search engines using clickthrough data. *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*

*(KDD '02),* Edmonton, Alberta, Canada. 133-142.

Jung, J. J. (2005). Collaborative web browsing based on semantic extraction of user interests with bookmarks. *Journal of Universal Computer Science, 11*(2), 213-228.

Kiryakov, A., Popov, B., Terziev, I., Manov, D., & Ognyanoff, D. (2004). Semantic annotation, indexing, and retrieval. *Web Semantics: Science, Services and Agents on the World Wide Web, 2*(1), 49-79.

Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *J.ACM, 46*(5), 604-632.

Krause, B., Jäschke, R., Hotho, A., & Stumme, G. (2008). Logsonomy - social information retrieval with logdata. *Proceedings of the 19th ACM Conference on Hypertext and Hypermedia (HYPERTEXT 2008),* Pittsburgh, PA, USA, 157-166.

Lathia, N., Hailes, S., & Capra, L. (2008). The effect of correlation coefficients on communities of recommenders. *SAC '08: Proceedings of the 2008 ACM Symposium on Applied Computing,* Fortaleza, Ceara, Brazil, 2000-2005.

Leacock, C., & Chodorow, M. (1994). *Filling in a sparse training space for word sense disambiguation.* Unpublished manuscript.

Lee, J. H., Kim, M. H., & Lee, Y. J. (1993). Information retrieval based on conceptual distance in is-A hierarchies. *Journal of Documentation, 49*(2), 188-207.

Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. *Proceedings of the 5th Annual International Conference on Systems Documentation (SIGDOC '86 ),* Toronto, Ontario, Canada, 24-26.

Li, C., Sun, A., & Datta, A. (2011). A generalized method for word sense disambiguation based on wikipedia. *Proceedings of the 33rd European Conference on Advances in Information Retrieval,* Dublin, Ireland, 653-664.

Li, Y., Bandar, Z. A., & McLean, D. (2003). An approach for measuring semantic similarity between words using multiple information sources. *IEEE Transactions on Knowledge and Data Engineering, 15*(4), 871-882.

Lin, D. (1998). An information-theoretic definition of similarity. *Proceedings of the Fifteenth International Conference on Machine Learning (ICML '98),* 296-304.

Makris, C., Plegas, Y., & Theodoridis, E. (2013). Improved text annotation with wikipedia entities. *Proceedings of the 28th Annual ACM Symposium on Applied Computing,* Coimbra, Portugal, 288-295.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. New York: Cambridge University Press.

Michlmayr, E., & Cayzer, S. (2007). Learning user profiles from tagging data and leveraging them for personal(ized) information access. *Proceedings of the Workshop on Tagging and Metadata for Social Information Organization, in the International World Wide Web Conference (WWW 2007),* Banff, Canada, 1-7.

Mihalcea, R. (2007). Using wikipedia for automatic word sense disambiguation. *Human*

*Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics,* Rochester, New York.

Miller, G. A. (1995). WordNet: A lexical database for english. *Communications of the ACM, 38*(11), 39-41.

Miller, G. A., & Charles, W. G. (1991). Contextual correlates of semantic similarity. *Language and Cognitive Processes, 6*(1), 1-28.

Milne, D., & Witten, I. H. (2008). An effective, low-cost measure of semantic relatedness obtained from wikipedia links. *Proceedings of the First AAAI Workshop on Wikipedia and Artificial Intelligence (WIKIAI'08),* Chicago, Illinois, USA, 25-30.

Motta, E., & Specia, L. (2007). Integrating folksonomies with the semantic web. *Proceedings of the 4th European Semantic Web Conference (ESWC2007), Lecture Notes in Computer Science,* Innsbruck, Austria. *, 4519* 624-639.

Nastase, V., & Strube, M. (2013). Transforming wikipedia into a large scale multilingual concept network. *Artificial Intelligence, 194*, 62-85.

Navigli, R., & Crisafulli, G. (2010). Inducing word senses to improve web search result clustering. *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP '10),* Cambridge, Massachusetts, 116-126.

Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). *The PageRank citation ranking: Bringing order to the web.* (Technical Report No. SIDL-WP-1999-0120). Stanford InfoLab: Stanford InfoLab. Retrieved from http://ilpubs.stanford.edu:8090/422/

Passant, A., & Laublet, P. (2008). Meaning of A tag: A collaborative approach to bridge the gap between tagging and linked data. *Proceedings of the WWW 2008 Workshop Linked Data on the Web (LDOW2008),* Beijing, China.

Pedersen, T., Banerjee, S., & Patwardhan, S. (2005). *Maximizing semantic relatedness to perform word sense disambiguation.* ( No. UMSI 2005/25). University of Minnesota, Duluth: Supercomputing Institute.

Quillian, M. R. (1967). Word concepts: A theory and simulation of some basic semantic capabilities. *Behavioral Science, 12*(5), 410-430.

Rada, R., Mili, H., Bicknell, E., & Blettner, M. (1989). Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man, and Cybernetics, 19*(1), 17-30.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). GroupLens: An open architecture for collaborative filtering of netnews. *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work (CSCW '94),* Chapel Hill, North Carolina, United States, 175-186.

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. *Proceedings of the 14th International Joint Conference on Artificial Intelligence,* Montreal, Quebec, Canada, 1, 448-453.

Richardson, L., & Ruby, S. (2007). *RESTful web services* (1st ed.). USA: O'Reilly Media Inc.

Richardson, R., & Smeaton, A. (1995). *Using WordNet in a knowledge-based approach to* *information retrieval.* ( No. CA-0395). School of Computer Applications, Dublin City University, Ireland.

Richardson, M., Prakash, A., & Brill, E. (2006). Beyond PageRank: Machine learning for static ranking. *Proceedings of the 15th International Conference on World Wide Web (WWW '06),* Edinburgh, Scotland, 707-715.

Rico Almodóvar, R. (2012). *Aplicaciones web semánticas y datos semánticos: Una aproximación para la simplificación de su desarrollo y de su uso,* Editorial Académica Española.

Rubenstein, H., & Goodenough, J. B. (1965). Contextual correlates of synonymy. *Communications of the ACM, 8*(10), 627-633.

Salton, G. (1971). *The SMART retrieval system - experiments in automatic document processing*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

Sarwar, B., Karypis, G., Konstan, J., & Reidl, J. (2001). Item-based collaborative filtering recommendation algorithms. *Proceedings of the 10th International Conference on World Wide Web (WWW '01),* Hong Kong, Hong Kong, 285-295.

Shadbolt, N., Berners-Lee, T., & Hall, W. (2006). The semantic web revisited. *IEEE Intelligent Systems, 21*(3), 96-101.

Smith, B., & Welty, C. A. (2001). FOIS introduction: Ontology - towards a new synthesis. *Proceedings of the International Conference on Formal Ontology in Information Systems (FOIS '01),* Ogunquit, Maine, USA, 3-9.

Strube, M., & Ponzetto, S. P. (2006). WikiRelate! computing semantic relatedness using wikipedia. Paper presented at the *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI'06),* Boston, Massachusets, 1419-1424.

Suchanek, F. M., Kasneci, G., & Weikum, G. (2007). Yago: A core of semantic knowledge. *Proceedings of the 16th International Conference on World Wide Web (WWW '07),* Banff, Alberta, Canada, 697-706.

Sussna, M. (1993). Word sense disambiguation for free-text indexing using a massive semantic network. *Proceedings of the Second International Conference on Information and Knowledge Management (CIKM-93),* Washington, D.C., United States, 67-74.

Telang, P. (2013). Semantic web why it will be more and more important for ecommerce website development? Retrieved from http://www.transpacific-software.com/blog/semantic-web-why-it-will-be-more-and-more-important-for-ecommerce-website-development/

Thomas, D., Heinemeier Hansson, D., & Breedt, L. (2005). *Agile web development with rails : A pragmatic guide*. Raleigh, N.C: The Pragmatic Bookshelf.

Trillo, R., Gracia, J., Espinoza, M., & Mena, E. (2007). Discovering the semantics of user keywords. *Journal of Universal Computer Science, 13*(12), 1908-1935.

Van Rijsbergen, C. J. (1979). *Information retrieval* (2nd ed.). Newton, MA, USA: Butterworth-Heinemann.

Wee, L. C., & Hassan, S. (2008). Exploiting wikipedia for directional inferential text similarity.

*Proceedings of the Fifth International Conference on Information Technology: New Generations (ITNG '08),* Las Vegas, Nevada, USA, 686-691.

Wilks, Y., Fass, D., Guo, C., McDonald, J. E., Plate, T., & Slator, B. M. (1990). Providing machine tractable dictionary tools. *Machine Translation, 5*(2), 99-154.

Williams, S. (2013). Better search through query expansion using controlled vocabularies and apache solr. *Code4lib Journal, 20,* October 2013. Retrieved from http://journal.code4lib.org/articles/7787

Worldwidewebsize.com. (2012). The size of the world wide web. Retrieved from http://www.worldwidewebsize.com/

Wu, Z., & Palmer, M. (1994). Verbs semantics and lexical selection. *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics (ACL '94 ),* Las Cruces, New Mexico, 133-138.

Wu, X., Zhang, L., & Yu, Y. (2006). Exploring social annotations for the semantic web. *Proceedings of the 15th International Conference on World Wide Web (WWW '06),* Edinburgh, Scotland, 417-426.

Xue, G., Xing, D., Yang, Q., & Yu, Y. (2008). Deep classification in large-scale text hierarchies. *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* Singapore, Singapore, 619-626.

Zhang, D., & Dong, Y. (2002). A novel web usage mining approach for search engines. *Computer Networks, 39*(3), 303-310.

Zhang, X., Asano, Y., & Yoshikawa, M. (2011). A generalized flow based method for analysis of implicit relationships on wikipedia. *Knowledge and Data Engineering, IEEE Transactions On, 6*(1), 1-14.