UNIVERSITAT ROVIRA I VIRGILI
DEPARTAMENT D'ENGINYERIA INFORMÀTICA I MATEMÀTIQUES
MÀSTER EN INTEL·LIGÈNCIA ARTIFICIAL

MASTER THESIS

Pattern-based Automatic Annotation of Web contents

José Miguel Millan Rosa

Tarragona, 2008

Knowledge is Power,
Sir Francis Bacon (1561-1626)

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Since the creation of the World Wide Web (WWW), presented by Tim Berners-Lee in 1989, its structure and architecture have been in constant growth and development. Nowadays the Web is involved in what we familiarly know as the Social Web, where all its users are able to add and modify its contents. This has brought lots of new information to the Web and its size has grown up to 4×10^9 static pages [*Baeza-Yates,* 2004] (*the surface web*) plus the so-called *deep web*, which consists in the dynamically created web pages. Although this increase of information could seem a very interesting feature, the lack of structure brought some problems: it complicates its accessing, as it cannot be interpreted semantically by IT applications [*Fenster et al.,* 2002], both manually and in an automatic way. So, in order to solve these inconveniences a new global initiative has been proposed: the **Semantic Web** [*Berners-Lee et al.,* 2001].

The Semantic Web is an evolving extension of the World Wide Web in which the semantics of information and services on the web is defined, making it possible for the web to understand and satisfy the requests of people and machines to use the web content. It derives from W3C director Tim Berners-Lee's vision of the Web as a universal medium for data, information, and knowledge exchange.

Tim Berners-Lee originally expressed the vision of the Semantic Web as follows:

> *I have a dream for the Web [in which computers] become capable of analyzing all the data on the Web – the content, links, and transactions between people and computers. A 'Semantic Web', which should make this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy and our daily lives will be handled by machines talking to machines. The 'intelligent agents' people have touted for ages will finally materialize.*
>
> *Tim Berners-Lee*

Humans are capable of using the Web to carry out tasks such as finding the Finnish word for "cat", reserving a library book, and searching for a low price on a DVD. However, a computer cannot accomplish the same tasks without human direction because web pages are designed to be read by people, not machines. The Semantic Web is a vision of information that is understandable by computers, so that they can perform more of the tedious work involved in finding, sharing and combining information on the web.

At its core, the Semantic Web comprises a set of design principles, collaborative working groups, and a variety of enabling technologies. Some elements of the Semantic Web are expressed as prospective future possibilities that are yet to be implemented or realized. Other elements of the Semantic Web are expressed in formal specifications. Some of these include Resource Description Framework (RDF), a variety of data interchange formats (e.g. RDF/XML, N3, Turtle, N-Triples), and notations such as RDF Schema (RDFS) and the Web Ontology Language (OWL), all of which are intended to provide a formal description of concepts, terms, and relationships within a given knowledge domain.

The final objective of the Semantic Web is to be able to semantically analize and catalog the web contents. This requires a set of structures to model the knowledge, and a linkage between the knowledge and contents. In this manner the Semantic Web relies on two basic components, **ontologies** and **annotations**.

1.1 Introduction to ontologies

Regarding the first component, there are several definitions about what an ontology is:

- In [Studer *et al*., 1998], an ontology is defined as a formal, explicit specification of a shared conceptualization. *Conceptualization* refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon. *Explicit* means that the type of concepts used, and the constraints of their use, are explicitly defined. *Formal* refers to the fact that the ontology should be machine-readable. *Shared* reflects the notion that an ontology captures consensual knowledge, that is, it is not private of some individual, but accepted by a group.
- In [Neches *et al*., 1991], a definition focused on the form of an ontology is given. An ontology defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary.
- Other approaches have defined ontologies as explicit specifications of a conceptualization [Gruber, 1993] or as shared understanding of some domain of interest [Uschold and Gruninger, 1996].

Even though different knowledge representation formalisms exist for the definition of ontologies, they share the following minimal set of components:

- *Classes*: represent concepts. Classes in the ontology are usually organised in taxonomies through which inheritance mechanisms can be applied.
- *Relations*: represent a type of association between concepts of the domain. Ontologies usually contain binary relations. The first argument is known as the domain of the relation, and the second argument is the range. Binary relations are sometimes used to express concept attributes. Attributes are usually distinguished from relations because their range is a data type, such as string, numeric, *etc*., while the range of a relation is a concept.
- *Instances*: are used to represent elements or individuals in an ontology.

The set of activities that concern the ontology development process, the ontology life cycle, the principles, methods and methodologies for building ontologies, and the tool suites and languages that support them, is called *Ontological engineering* [Gómez-Pérez and Fernández-López, 2004]. With regard to methodologies, several proposals have been reported for developing ontologies manually (more details in [Gómez-Pérez and Fernández-López, 2004]).

1.2 Introduction to Semantic Annotations

The annotation of web resources -**Figure 1**- is another fundamental requirement for the Semantic Web. In [*Wikipedia:Annotation*], a Semantic Annotation is extra information asserted with a particular point in a document or other piece of information.

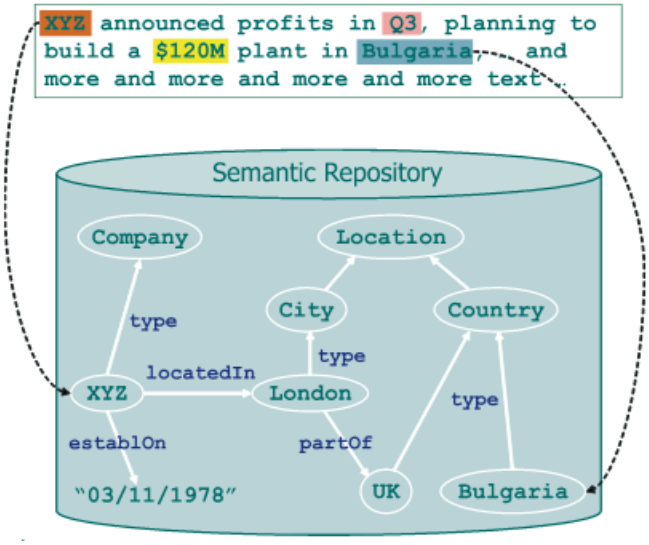


Figure 1. What is annotation?

A complete definition of Semantic Annotation is the one given in [KIM] where it is defined as information about what entities (or, more generally, semantic features) appear in a text and where they do. Formally, semantic annotations represent a specific sort of metadata, which provides references to entities in the form of URIs or other types of unique identifiers.

There are a number of alternative approaches regarding the organization, structuring, and preservation of annotations. For instance, all the markup languages (HTML, SGML, XML, etc.) can be considered schemata for embedded or in-line annotation. On the contrary, open hypermedia systems use standoff annotation models where annotations are kept detached, i.e. non-embedded in the content.

Appropriate semantic tagging of web content based on domain ontologies can certainly bring many benefits in developing intelligent Information Retrieval and Extraction techniques. However, considering the amount of information available in the Web, manual annotation processes (such as the framework proposed in [*McDowell et al.,* 2003]) are unfeasible.

In the last years, several attempts have been made to address this issue, by automating some aspects of the annotation process. The most basic ones use manually constructed rules to extract known patterns for the annotations [*Baumgartner et al.,* 2001]. There are supervised systems that use extraction rules obtained from a set of pre-tagged data [*Califf and Mooney,* 2004]. In both situations the applicability is certainly limited by the knowledge bottleneck introduced by the interaction of a domain expert and the difficulty of compiling a large and representative training set [*Cimiano et al.,* 2005]. Due to these reasons, the completely unsupervised annotation paradigm plays a fundamental role in the success of the Semantic Web. Fully unsupervised systems (such as [*Etzioni et al.,* 2004]) or bootstrap based approaches [*Ciravegna et al.,* 2004] are promising trends in the annotation research area.

As it will be seen state of the art section of this document, there is an evident lack of automatic solutions, and this absence is bigger in case of both unsupervised and automatic solutions. On the other hand, the large amount of documents to be annotated and the unfeasibility of annotate them manually, makes the need of automatic and unsupervised solutions crucial.

In this manner, this work will be focused in the annotation of web documents in an unsupervised and automatic way. For our purposes we will use the Web as a knowledge source, exploiting the large of information available for nearly every domain of knowledge, as it will be described in Chapter 3.

1.3 Objectives

To achieve our goal, we have defined a set of objectives in order to design a solution to annotate documents in an automatic and unsupervised way. Concretely:

- In order to design a novel solution, we will review and classify of the current state of the art in textual annotation. Moreover, as these solutions rely on a set of standards and specifications, they will have a set of common characteristics. For this reason, we will extract their main features to define properly the requirements that these a semantic annotation approach should, ideally, accomplish.
- As the goal of our algorithm is to analyze textual documents, we will analyse the existing tools to structure the free text in such way that information extraction processes can be applied over it.
- Due to the size of the Web, we think it is a perfect knowledge repository for assisting knowledge-related learning approaches. As we need a knowledge source to know *what-is-what,* we will study the best ways to use the amount and the heterogeneity of web information to aid the learning. For this purposes, we will deeply study its characteristics and the tools available to exploit the Web.
- All the solutions based on Web technologies, or designed for the Web, should rely on accepted and usable standards. At the moment, there isn't any study or analysis

of the current technologies to annotate textual documents for the Web. For this reason, we will analyse the most commonly used techniques to annotate such documents, in order to have a deeper vision of their advantages and to select which of them fit better for our purposes.

- During the annotation of textual documents, several tasks should be performed:
  - The first of them is the detection of the relevant text portions that should be annotated, the Named Entities. So, we will develop an algorithm to extract the Named Entities of web documents trying to increase the performance of the currently available algorithms.
  - The second task that should be tackled is the tagging of named entities with the most appropriate semantic ontological class. So, we will develop an algorithm to analyse each Named Entity and discover which class of a given ontology is the most appropriate one. This process will be designed to minimize the amount of web accesses to improve the algorithm performance.

  The results of these objectives have been presented in [*Millan et al*., 2008].
- Finally, we need to demonstrate the viability of our solution, so we should deeply evaluate it. Due to the lack of automatic evaluation procedures for this kind of algorithms, we will propose a manual methodology to evaluate the quality of the results. Using this evaluation we will quantify the quality of the results of the different stages of the algorithm over different domains. Finally, in order to have a clear idea of the algorithm's reliability, we will compare the evaluation results against other similar approaches.

1.4 Document Structure

This document is organized as follows:

- In Chapter 2 we present the requirements that an annotation solution for the Web should accomplish, e.g.- *the usage of standard formats, the ontology support, etc*. We also present the state of the art of the different existing annotation techniques classifying them in function of their architecture (e.g.- *Frameworks, Complete Solutions, Plug-ins, etc*) and in function of their automatism grade (e.g.- *manual annotation tools, semi-automatic tools or complete automatic solutions*).
- In Chapter 3, the different learning techniques and technologies that are used to automatically annotate a document are presented. The main techniques are focused on Natural Language processing techniques, the usage of the Web as a knowledge repository and the usage of electronic dictionaries to reduce the overload produced of utilizing the Web as a knowledge repository. We also describe how the inherent patterns in the texts can be used to extract class candidates for instances, and some statistical techniques to evaluate which of these candidates are the most appropriate ones. Finally, we describe the current annotation technologies.
- Chapter 4 presents our novel annotation algorithm, describing its different steps. Firstly we describe the instance detection procedure, in a second step we detail how the class candidates for each instance are extracted from the Web, and finally in a third step, we classify this instances in a predefined ontology using the class candidates extracted.

- We describe in Chapter 5 an evaluation both of the instances extraction procedure and the classification procedure. This Chapter also introduces a comparison of the algorithm performance using different configurations of learning parameters.
- Finally, in the last Chapter, the conclusions and several lines of future work are presented. In this, firstly we analyse and compare the results obtained in the evaluation, and lastly we focus on the different points that can be refined to improve the algorithm.

Chapter 2

Requirements and state of the art in Semantic Annotation

The annotation of documents is covered by a large set of solutions. For instance, the nature of these annotation is composed by automatic or manual solutions and standard-based or ad-hoc developments. This diversity of solutions produces a high heterogeneity of technologies and applications, so it is important to define which requirements these applications should accomplish in order to be feasible used in the Semantic Web.

On the other hand, it is prior to know exactly which kinds of solutions for annotate documents exists nowadays, so the realisation a State of the Art of the current applications is indispensable.

So, in this chapter, a set of requirements for the Semantic Web annotation applications is described. Next a State of the Art is presented as a means to introduce the reader to the real state of the Semantic Web annotation procedures.

2.1 Requirements

In order to analyze the quality of any annotation system, a set of requirements for these systems should be defined. Based upon the requirements set out by *[Handshuch et al.,* 2003] and the extended requirements described on [*Uren et al.,* 2006], a new set is described, proposing new ones in order to be able to do a better analysis of the annotation systems. All these requirements are based on the supposed functionalities that annotation tools shall give to the user.

2.1.1 Standard formats

This requirement is fully necessary for a future integration and cooperation of the systems, so it will be possible in a near future to integrate different annotation systems. As the annotation processes are mainly orientated to the Web, the standard formats associated to it are the most appropriate, so the different standards proposed by W3C, like RDF or OWL, seem to be the ones who fit better with this functionality,

as they are open, free and have a definition strict enough to be used without ambiguities.

2.1.2 Support of heterogeneous formats in documents

Systems designed to semantically annotate Web contents should have capabilities to support all the formats that the Web has, and this does not include only HTML and XML (as they are the most common formats in the Web). This includes formats like Flash, Java Applets, Windows Media Videos, Quicktime Movies, etc. Currently the most of the existing systems have support for XML and HTML, but it is possible to find systems like WICKOffice and OntoOffice which have support for word processor files, or systems like SMORE and Mangrove that provide facilities for handling emails. On the other hand, the Rich News application of KIM provided an example of how an area where automation expertise exists (text based IE) can be used to support the automatic annotation of more difficult media (audio visual). This last approach is a great strategy to prove fertile ground for the development of environments that take a more integrated approach to handling heterogeneous formats.

2.1.3 Ontology support

The annotation tools that have been appearing have rapidly adopted standards like OWL. However, all the support has been reduced to doing processes not more complex than searching and navigating. It means that the existing annotation tools do not give to the user functionalities to create or maintain the ontologies. It seems to be caused by the fact that the knowledge workers will use existing ontologies instead of creating or editing new ones. At the moment, the most advanced existing tools in this requirement, like The Open Ontology Forge, give to the user limited possibilities, like the creation of new classes from a root class, but more intense work should be done in this requirement.

2.1.4 Ontology domain dependence

This requirement relies on the fact that the systems are dependant on the domain of the ontology used, this means that they are not capable to identify entities from an undefined domain, giving to the system a limitation. This requirement can be solved by the use of standard formats, and the integration of the different systems, giving to the final user, a “super-annotation tool” but, by now, and while the existing systems do not have support for standard formats, it seems to be a hard requirement to defy.

2.1.5 Document and annotation consistency

This is one of the requirements with more work to be done, as keeping the annotations synchronized with changes in the documents is a great challenge and the current annotation standards are not adequate. As it will be mainly described in requirement 6 (see section 2.6), there are strong arguments in favor of having annotations separated from documents, but the problem is that current standards for annotation (like XPointer) do not manage so well the changes into the documents, and the annotations usually get broken. So, ruled out the possibility of having the documents and the annotations joined and the use of the current annotation standards, it is prior to develop efficient and consistent annotation technologies.

2.1.6 Annotation storage options

The Semantic Web proposes a decoupled model for the annotation storage, where the annotations and the documents are stored separately. In another model commonly named "word processor", the annotations are stored as an integral part of the documents and can be viewed or not by the readers. So, meanwhile the Semantic Web model fits perfectly with the Web environment as usually the annotators are separated from documents, the second model cannot be forgotten, as it is more familiar for the Knowledge Management annotators.

2.1.7 User centered/collaborative design

As the annotation can be done manually, the annotation process can be potential system impairment if the annotators have a great demand of time. So, collaborative and easy-to-use systems interfaces should be provided to them, in order to increase their productivity, and at the same time, to give them tools to share the information that they have and generate. On the other hand, these systems should also provide access control policies to share all this information, as in some fields of use, the information should be anonymized or restricted to some users.

2.1.8 Automation

This is one of the most important requirements for an annotation system, as the amount of documents present in the Web is so large, and their manual annotation is so hard and time consuming, that some automation methodologies based in knowledge extraction technologies should be provided to the user to facilitate this job.

2.1.9 Communication

Finally, another important requirement for an annotation system should be its capabilities to communicate with other systems, as it should be possible that the knowledge stored in other systems can be used to annotate a document by the first one. To achieve that, communication and annotation standards provided by the W3C, as XML, SOAP, RDF, etc. can be used. Nevertheless, this requirement depends on most of the requirements exposed before and, usually, the current systems are more centered in achieve better annotation techniques than in use other systems, although, in fact, by the use of annotation standards this communication can be indirectly done.

2.2 State of the Art

In this subsection, the current approaches for annotation will be presented. They have been classified into three categories: frameworks, annotation tools (manual and automatic), integrated annotation environments and on-demand annotation tools.

2.2.1 Annotation Frameworks

There are many annotation frameworks currently in the Web. We have chosen the three most widely known: Annotea, CREAM and NOMOS.

The Annotea [*Kahan et al.,* 2001] [*Koivunen,* 2005] system is a W3C project which specifies the infrastructure needed for annotation in the Web, with emphasis on the collaboration aspects. As a work of the W3C, it emphasizes in the use of the Web standards, so this framework works with HTML/XML-based documents and uses RDF to annotate the documents. As it is a project clearly oriented for the Web, it places the annotations separated from the documents and, to manage them, it uses XPointer technologies. This technology is another recommendation of the W3C, which works fine with detail changes in the documents, but it has problems with large-scale revisions. Annotea concentrates in a semi-formal style of annotation, in which the annotations are free text statements about the documents.

The CREAM Framework [*Handschuh et al.,* 2003] is in fact very similar to the Annotea system, as it uses W3C standards to annotate the documents (RDF or OWL) and to locate the annotations (XPointers). However, it has two important differences from Annotea. The first one is that the authors have tried to specify a more formal format for the annotations. The second one is that this framework is conceived for annotate the deep Web, this means that it has functionalities to annotate relational databases and metadata, giving, in this way, essential functionalities to provide Semantic Web services.

The NOMOS Framework [*Niekrasz and Gruenstein,* 2006] inherits the main functionalities of the Annotea system (use of standards to annotate), but it adds some user-based functionalities, as a GUI to annotate and a large set of utilities to develop the corpus and annotate it, and capabilities to annotate multimedia formats taking into account temporal restrictions of these kind of documents.

2.2.2 Annotation Tools

Having analyzed the current frameworks to annotate contents it is possible to introduce some tools which perform the annotations into the documents. These are a first generation of annotation tools, and some of the requirements presented above are provided, but, in the other hand, these tools need further work to provide a fully featured annotation environment. As the amount of tools is so large, they have been classified into two groups. The most significant of these groups will be briefly introduced.

2.2.3 Manual annotation tools

Those tools which need some kind of human support to do their annotations compose the group of manual annotation tools. Into this group, three systems should be emphasized.

In first term, the W3C Web browser and editor, Amaya [*Quint and Vatton,* 1997], can mark-up the visited Web documents in XML or HTML. It is specially significant for this tool, the fact that the user can annotate the documents in the same tool in which he can navigate through the Web. It is also remarkable the fact that some plug-ins (Annozilla and Teknowledge) for more popular browsers (Mozilla Firefox and Internet Explorer) have been developed to support this annotations.

In second place, it is important the task done in the Vannotea tool [*Schroeter et al.,* 2003], which allow to the user to annotate multimedia contents like images, video and audio from different sources like MPEG-2 (video), JPEG2000 (image) or Direct3D (mesh). It also has a particularly interesting feature: the possibility to manage distributed annotations, allowing to annotate the same document in a collaborative "fashion".

Finally, the Open Ontology Forge (OOF) [*Collier et al.,* 2004] is a tool similar to another one called "SMORE" [*SMORE*], but it adds to its functionalities (images, emails, HTML and text annotation) an integrated environment to handle documents, ontologies and annotations.

2.2.4 Automatic annotation tools

The automatic tools are those that have some kind of automation in order to suggest an annotation to the user. Some of them go one step further and try to do the annotations completely automatic.

According to the type of approach they propose, they can be classified into three groups: the first ones, which use techniques like rules or wrappers to identify patterns into the text; the ones which use learning systems which try to annotate based on previously known annotations; and the ones which use unsupervised techniques to annotate the documents, being these last ones the most interesting ones, but also being the ones with worst results.

Regarding concrete system, the first remarkable tool is Melita [*Ciravegna et al.,* 2002], a user driven automated tool, which make available to the user two main

strategies. The first one is an information extraction system that learns how to annotate the documents based on a manual annotation previous step. The second one is the provision of facilities for rule writing to allow to advanced users to define their own rules.

The second tool is KnowItAll [*Etzioni et al.,* 2005]. It uses the redundancy of the Web to do a bootstrapping process and extract information, so, after this information is extracted, the confirmation of the correctness of this information is requested to the user in order to perform this process another time system. The most interesting feature of this tool is the way in which the system verifies the trustiness of the candidate extractions. It uses the pointwise mutual information (PMI) measure, which is a roughly the ratio between the number of search engine hits obtained by querying with the discriminator phrase by the number of hits obtained by querying with the extracted fact. Also, KnowItAll has a set of extensions to improve the overall performance.

SmartWeb [*Buitelaar and Ramaka,* 2005] is the third tool; its most important feature is that resolves the issue of not having pre-existing mark-up to learn from, by using class and subclass names from the ontology to construct examples. The context of these examples is then learnt. In this way, instances can be identified which have similar contexts, but which may use different terminology to the ontology.

As the fourth tool we have C-Pankow [*Cimiano et al.,* 2005]. It is the second version of PANKOW (Pattern-based Annotation through Knowledge On the Web). PANKOW uses a range of well studied syntactic patterns (Hearst Patterns) to mark-up candidate phrases in Web pages without having to manually produce an initial set of marked-up Web pages. The context driven version, C-Pankow, improves the first version reducing the number of queries to the search engines and having better annotation results.

Eventually, it is interesting to remark h-TechSight [*Maynard et al.,* 2005]. It is the result from a European project in which the University Rovira i Virgili has worked to. It is a traditional approach to information extraction in which the GATE rule-based IE system is used to feed a semantic portal. It is interesting because gives automatic monitoring of the "dynamics" of concepts and instances that can be feed back to the end users.

2.2.5 Integrated Annotation Environments

As one of the requirements to the annotation systems is the need of a single entry point for the annotations, in order to give to the knowledge workers an homogeneous access way, completely integrated with their working environments, some systems conceived to be integrated with the tools used to generate documents and make annotations simultaneously. Those are presented here.

In first place, WiCKOffice [24] uses automatic assistance for form filling using data extracted from knowledge bases and demonstrates a successful way of write annotations in a knowledge aware environment.

Another remarkable tool is AktiveDoc [*Lanfranchi et al.,* 2005]. It enables annotation of documents at three levels: ontology based content annotation, free text statements and on-demand document enrichment. It gives support during both editing

and reading, and a semi-automatic annotation of content is provided via Adaptative Information Extraction from text. It is also designed for knowledge reuse, so it is able to monitor editing actions and to provide automatic suggestions about relevant content.

2.2.6 On-demand annotation

The systems described in this section are not strictly annotation tools, but they produce annotation-like services on demand for users browsing un-annotated resources.

The first system is Magpie [*Dzbor et al.,* 2004], which does "real-time" annotation of web resources by highlighting text strings related to an ontology of the user's choice; appropriate web services can be linked to highlighted strings. While the annotation is automatic, Magpie currently has the problem that subject specific parts of the lexicons of text strings for each ontology have to be produced manually.

The second system is Thresher [*Hogue and Karger,* 2005]. It is similar to Magpie in that it uses wrappers to generate RDF on the fly as users browse deep into web resources. It gives facilities to write these wrappers for the non-technical users and it is part of a Semantic Web browser called Haystack, which permits to personalize the employed ontologies.

2.2.7 Conclusions

As the amount of documents to annotate in the web is extremely large, a way to automate this annotation is needed; this is why the automation is an important part of all the systems presented above. In this section a brief revision of the techniques used to automatically annotate documents will be presented. In **Table 1** it is also possible to find a relation of the automation techniques used in the annotation tools mentioned before.

The most common way to automatically annotate the documents is using wrappers, which exploit the structure of Web pages to identify nuggets of information for mark-up. Wrappers and rules are most useful when there are very regular patterns in the documents, such as standard tables of data. In the other hand, they need skill on the part of the user to define the correct patterns.

Another automatic technique is the use of learning, in where the systems recognize the objects to be annotated by learning from a collection of previous annotated documents. This has the inconvenient that needs a large amount of previously annotated documents.

A third technique is the use of unsupervised learning techniques, like the use of the distribution of certain patterns on the Web to determine the formal annotation of entities in Web pages by a principle of 'annotation by maximal (syntactic) evidence'. These techniques are used, for example, in C-Pankow, which will be described in section 3.4.

However, the users of automatic annotation systems need to be aware of their limitations. This means that there are missing annotations (recall) and incorrect

annotations (precision), and they trade off against each other. Nevertheless, they present a valuable aid to the task of annotating large amounts of documents.

TABLE 1. SYSTEMS DESCRIBED AND AUTOMATION TECHNOLOGIES USED BY THEM.

System Methodologies	*Methodologies*
Annotea	No automation
CREAM	No automation
NOMOS	No automation
Amaya	No automation
Vannotea	No automation
OOF	No automation
SMORE	No automation
Melita	Wrappers and rules, Learning from previous knowledge
KnowItAll	Learning from previous knowledge
SmartWeb	Learning from previous knowledge
C-Pankow	Unsupervised learning techniques
h-TechSight	Wrappers and rules
WiCKOffice	No automation
AktiveDoc	Learning from previous knowledge

It is evident that there is a dearth of automatic solutions (more than the half of the presented does not present any automation at all) and, practically, there is not any unsupervised solution (only two from a list of 16 approaches). In this manner, our work has been focused on an automatic and unsupervised algorithm, due to the facts mentioned.

Considering the requirements presented at the beginning of this section, in first place, our proposal will be focused on textual documents and it will use "*into-document*" annotations. It should also be completely automatic; it should use knowledge from standardized ontologies, as these features are prior in order to be suitable for the size of the Web. On the other hand, it should be domain independent due to the heterogeneity of the Web, and we would like to give it the enough communication features in order to make it usable as a Web Service.

Based upon the nature of the Web, these requirements are based on the fact that it is impossible to annotate manually the all existing documents, and at the same time, it is very hard to do it properly with the new ones. As stated, one can see the evident scarcity of automatic solutions, in this manner we intend to contribute in this field with an automatic and unsupervised work.

Nevertheless, the existing works are mainly based on learning techniques or predefined rules, which requires a previous training or configuration work. In this manner, we like the path taken by C-Pankow, as it is completely unsupervised, based on the patterns existing on the English language and, at the same time has an automatic behaviour using the Web as the largest knowledge repository available. Eve though, it has some open issues. The first one is the detection of Named Entities, which can be improved using techniques similar to the described in [*Pasca,* 2005]. In second place, it uses a concrete set of patterns that are very present in the English language, but not all of them fit well with the annotation concept, as usually defining Named Entities fit better with the purposes of the Semantic Annotation than listing

examples. Finally, in the work of C-Pankow is not described a complete procedure to link Named Entities with knowledge, for this reason, we will also focus on describing a complete procedure to do so, starting from the relation of a Named Entity with its corresponding concept and finishing on the final textual annotation.

Chapter 3

Learning Techniques and Working Environment

Every automatic annotation solution is based upon several learning techniques. These techniques go from content analysis algorithms to information extraction procedures. For these reason it is prior to well know the different techniques that have been needed during the development of this algorithm. On the other hand, as this approach is based on the Web, and its own characteristics make it different from other sources used in classical learning techniques, it is prior to well know those characteristics in order to correctly define the set of techniques which will be used to discover, learn and apply the inherent knowledge contained in the sources. Then, this chapter is structured as follows:

- In §3.1, it will be argued that the Web can be a valid knowledge learning repository thanks to the huge amount of information available for every possible domain and its high redundancy. In this sense, the amount and heterogeneity of information is so high that it can be assumed that the Web approximates the real distribution of information [*Cilibrasi and Vitanyi,* 2004].
- The redundancy of information in such a wide environment can represent a measure of relevance and trustiness of the information. As will be introduced in §3.2, this redundancy may allow lightweight analytic approaches to obtain good quality results maintaining scalability and efficiency in this enormous and noisy environment [*Pasca,* 2005].
- Web search engines do a great job in indexing and retrieving web resources if the queries are specific enough. In consequence, if appropriate queries are performed, they can be eventually used for retrieving domain related web resources. Moreover, they can provide web-scale statistics about information distribution in a scalable and efficient way. In general, as will be justified in §3.3, they can be used as an aid in the knowledge acquisition process.
- The enormous size of the Web and the unsupervised nature of this approach make suitable the application of statistical analyses in order to infer information's relevance for a particular domain. As will be discussed in §3.4, a statistical analysis applied over knowledge acquisition tasks is a good deal if enough information is available to obtain relevant measures. The case of the Web is especially adequate as it represents the hugest repository of information available.

- The usage of lightweight analysis over the free text, as shown in §3.2, permits the application of higher-level techniques, which allows us to obtain relationships between different concepts. Some of these relationships could be hyponymy, synonymy, etc. In §3.5 will be detailed the different structures which can be used to extract those relationships between concepts, that allow the discovery of the inherent information of the text.
- In spite of the Web's powerfulness, its usage generates dependency on external tools and hampers the global performance of the solutions that rely on it. For this reason it is prior the introduction of alternative techniques to reduce these issues. In §3.6 is described one electronic dictionary, which solves the problems of the Web's dependency but it reduces the size of the knowledge source. For this reason, it cannot substitute the Web as a knowledge source, but it is useful to considerably reduce its usage.

3.1 The Web as a knowledge repository

As stated in [*Sánchez,* 2008], many classical knowledge acquisition techniques present performance limitations due to the typically reduced corpus used [*Brill,* 2003]. This idea is supported by current social studies as [*Surowiecky,* 2004], in which it is argued that collective knowledge is much more powerful than individual knowledge. The Web is the biggest repository of information available [*Brill,* 2003] with near 20,000 million web resources indexed by Google. This fact can represent a great deal when using it as a corpus for knowledge acquisition.

Apart from the huge amount of information available, another feature that characterizes the Web is its high redundancy. Several authors have mentioned this fact and it is especially important because the amount of repetition of information can represent a measure of its relevance [*Ciravegna et al.,* 2003; *Etzioni et al.,* 2004; *Brill*, 2003]. This can be a good approach to tackle the problem of untrustworthiness of the resources: we cannot trust the information contained in an individual website, but we can give more confidence to a fact that is enounced by a considerable amount of possibly independent sources. This fact is also related to the consensus that the extracted knowledge should present: implicit consensus can be achieved as concepts are selected among the terms that are frequently employed in documents produced by the virtual community of users [*Navigli and Velardi,* 2004].

Thanks to those characteristics, the Web has demonstrated its validity as a corpus for research [*Resnik and Smith,* 2003] with successful results in many areas: question answering [*Brill et al.,* 2003], question classification [*Solorio et al.,* 2004], machine translation [*Greffenstette,* 1999], anaphora resolution [*Bunescu,* 2003], Prepositional Phrase treatment [*Calvo and Gelbukh,* 2003], ontology enrichment [*Agirre et al.,* 2000], and contents annotation [*Cimiano et al.,* 2005].

3.2 Natural Language analysis

In general, the use of complex text processing tools as a step towards accessing the knowledge within a huge repository, like the Web, is nonviable [*Pasca,* 2005]. On the other hand, lightweight analyses can miss important information. Nevertheless, due to the information redundancy in the web, if that information is relevant, sooner or later it will be contained in another resource, even expressed in another formal way. Thus, one can take profit of the amount of resources available and its high redundancy to perform analyses over a large amount of resources, achieving good scalability and competent results. This is one of the basic theses that, at the end of this document, we want to proof.

This annotation methodology will be based on this premise. In general, we will perform an evaluation of a reduced corpus of resources obtained from the Web to retrieve candidates for an Entity. Then, their relevance will be checked against a large amount of resources (the whole Web). Note that to check that relevance (through a statistical analysis), it will not be necessary to analyse the whole corpus of web sites that cover a certain fact. This extracted knowledge will be used to connect a concrete content with the class to which it is related to.

A certain degree of natural language processing of the web content is needed to interpret the text and extract relations. In order to perform an efficient analysis, the amount of processed information from each web site will be reduced to the minimum. Concretely, only the nearest context of the analysed concept at each moment will be evaluated. Those pieces of relevant information are known as "*text nuggets*" (in our case, and as it will be seen later, these pieces correspond to the *snippets* retrieved by the Web Search Engines) and their analysis allows obtaining relevant results without an exhaustive processing of the whole text [*Pasca,* 2005].

Concerning the analysis of text itself, this work only considers English written resources and exploits some peculiarities of that language to extract knowledge. Therefore, a set of tools and algorithms for analysing English natural language is used for that purpose. Concretely:

- *Stemming algorithm*: allows obtaining the morphological root of a word for the English language. It is fundamental to avoid the redundancy of extracting the different equivalent morphological forms in which a word can be presented. Some examples of these algorithm could be found in [*Rijsbergen et al*., 1980] and [*Lancaster*, 2004].
- *Text processing tools for detecting sentences, tokens and parts of speech*: These tools use a piece of the text and chunk it in order to find its minimal parts. In our approach the longest context considered for a particular concept will be the whole text; however, it is also possible to consider separated sentences or paragraphs. Once the text is chunked, the different minimal pieces obtained are tagged with a *Part-Of-Speech* (POS) tagger.
- *Syntactic analyser* or *Part-Of-Speech tagging*: it will be used to perform basic morphological and syntactical analyses of particular pieces of text that can contain valuable information. This will allow us to interpret and extract potentially interesting concepts and relationships. Even though their precision is not perfect and, in consequence, some useful information may be omitted, this is not an important problem thanks to the high redundancy of information in the Web. In

§3.2.1 the most common techniques used to tag the parts of the text and to syntactically analyse text are described.

3.2.1 POS taggers

Part-of-speech tags are closely related to the notion of word class used in syntax. The assumption in linguistics is that every distinct word type will be listed in a lexicon (or dictionary), with information about its pronunciation, syntactic properties and meaning. A key component of the word's properties will be its class. When one carries out a syntactic analysis of an example like *"fruit flies like a banana"*, he will look up each word in the lexicon, determine its word class, and then group it into a hierarchy of phrases, as illustrated in the following parse tree:

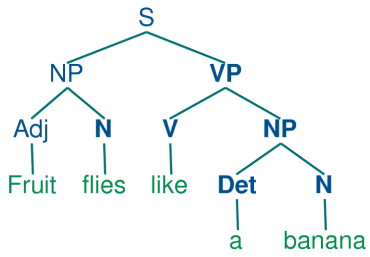


Figure 2. Example of syntactic tree

Once the different parts of the texts are discovered, they can be tagged with its corresponding POS tag, as shown in **Table 2**. It is remarkable that one can use the desired set of tags. For instance, in this work we have used the Brown Tag-set, as it is one of the more common sets.

TABLE 2. WORD CLASS LABELS AND ITS CORRESPONDING TAGS

Word Class Label	Brown Tag	Word Class
Det	AT	Article
N	NN	Noun
V	VB	Verb
Adj	JJ	Adjective
P	IN	Preposition
Card	CD	Cardinal Number
--	.	Sentence-ending punctuation

Even though this tagging task is easy for humans who are minimally familiarized with the text analysis, they are not trivial tasks for computers. In general, linguists use three criteria: *morphological* (or formal); *syntactic* (or distributional); *semantic* (or notional). A morphological criterion is one that looks at the internal structure of a word. For example, *-ness* is a suffix that combines with an adjective to produce a noun. Examples are *happy* → *happiness, ill* → *illness*. So if we encounter a word that ends in *-ness*, this is very likely to be a noun.

A *syntactic* criterion refers to the contexts in which a word can occur. For example, assume that one has already determined the category of nouns. Then he might say that a syntactic criterion for an adjective in English is that it can occur immediately before a noun, or immediately following the words be or very. According to these tests, near should be categorized as an adjective:

a. The near window
b. The end is (very) near.

A familiar example of a *semantic* criterion is that a noun is "the name of a person, place or thing". Within modern linguistics, semantic criteria for word classes are treated with suspicion, mainly because they are hard to formalize. Nevertheless, semantic criteria underpin many of our intuitions about word classes, and enable us to make a good guess about the categorization of words in languages that we are unfamiliar with. For example, if we all we know about the Dutch *verjaardag* is that it means the same as the English word birthday, then we can guess that *verjaardag* is a noun in Dutch. However, some care is needed: although we might translate *zij is vandaag jarig* as *it's her birthday today*, the word *jarig* is in fact an adjective in Dutch, and has no exact equivalent in English!

All languages acquire new lexical items. A list of words recently added to the Oxford Dictionary of English includes *cyberslacker, fatoush, blamestorm, SARS, cantopop, bupkis, noughties, muggle, and robata*. Notice that all these new words are nouns, and this is reflected in calling nouns an open class. By contrast, prepositions are regarded as a closed class. That is, there is a limited set of words belonging to the class (e.g., above, along, at, below, beside, between, during, for, from, in, near, on, outside, over, past, through, towards, under, up, with), and membership of the set only changes very gradually over time.

Then, in base of the problematics of Part-of-Speech tagging, some tagging algorithms have been designed.

The simplest algorithms are the **Regular Expression taggers**. They assign tags to tokens on the basis of matching patterns. For instance, one might guess that any word ending in *"ing"* is the gerund of a verb, and any word ending with *"'s"* is a possessive noun. One can express these as a list of regular expressions:

TABLE 3. TAGGING REGULAR EXPRESSIONS

Regular Expression	*Tag*	*Description*	*Example*
[A-Z].*$	NNP	Proper Noun	Madrid
.*ing$	VBG	Gerund verb tense	distinguishing
.*ould$	MD	Modal verb	would
.*'s	NN$	Singular common noun genitive	season's
.*s$	NNS	Plural common noun	stadiums

A more sophisticated approach would be the **Unigram taggers**, which are based on a simple statistical algorithm: for each token, assign the tag that is most likely for that particular token. For example, it will assign the tag JJ to any occurrence of the word frequent, since frequent is used as an adjective (e.g. a frequent word) more often than it is used as a verb (e.g. I frequent this cafe). A unigram tagger bases its behaviour on a previous training in which obtains the most common tags for each word.

Affix taggers are like unigram taggers, except they are trained on word prefixes or suffixes of a specified length (Using prefix and suffix in the string sense, not the morphological sense). An example would be a tagger that considers suffixes of length 3 (e.g. *-ize, -ion*), for words having at least 5 characters.

When we perform a language processing task based on unigrams, we are using one item of context. In the case of tagging, we only consider the current token, in isolation from any larger context. Given such a model, the best we can do is tag each word with its a priori most likely tag. This means we would tag a word such as *wind* with the same tag, regardless of whether it appears in the context the *wind* or *to wind*. An **n-gram tagger** is a generalization of a unigram tagger whose context is the current word together with the part-of-speech tags of the n-1 preceding tokens, as shown in **Figure 3**. The tag to be chosen, t_n, is circled, and the context is shaded in grey. In the example of an n-gram tagger shown in **Figure 3**, we have n=3; that is, we consider the tags of the two preceding words in addition to the current word. An n-gram tagger picks the tag that is most likely in the given context.

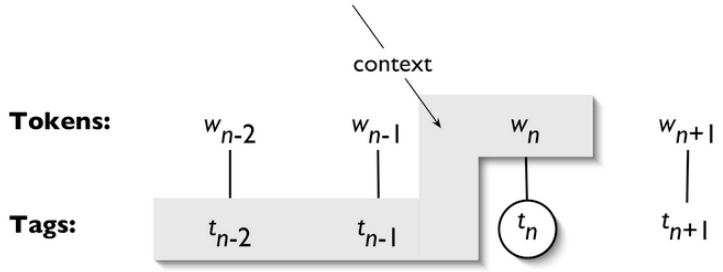


Figure 3. Tagger context

As n gets larger, the specificity of the contexts increases, as does the chance that the data we wish to tag contains contexts that were not present in the training data. This would involve in a sparse data problem. Thus, there is a trade-off between the accuracy and the coverage of the results (and this is related to the precision/recall trade-off in information retrieval).

3.3 Web search engines

The base of a knowledge acquisition methodology is the extraction of concepts and relationships from a corpus of documents that covers a certain domain. Ideally, that corpus should contain the most relevant and reliable documents for the specific domain. However, this premise requires that a certain pre-processing should be made by an expert to compile the initial set of resources.

As this work intends to develop a domain independent and unsupervised methodology, the corpus of documents has to be obtained in other manners. Concretely, a reliable way of obtaining web resources is to use a search engine to retrieve lists of web sites matching with a specific query [*Sánchez,* 2008]. In addition,

and as will be shown further in §3.4, robust web-scale statistics can be obtained directly and efficiently from queries performed into a web search engine. As a result, one may realize about the important role that a web search engine can play in the knowledge acquisition process from the Web.

In this section, we describe in detail the behaviour and possibilities that currently available Web search engines offer. The objective is to analyse the ways in which a search engine can be exploited to perform knowledge learning tasks and which is the concrete search engine that fits better with our purposes.

Concretely, in §3.3.1, an overview of the main types of search engines is presented (*keyword-based* and *taxonomic* approaches). Next, in §3.3.2, the type of search engine that fits well with the purposes of this approach is justified (*keyword-based* engines), and a discussion of the different aspects and features that can be exploited to aid the knowledge acquisition process. Lately, in §3.3.3, a comparison of keyword-based search engines is introduced, considering several parameters and functionalities that are important in our knowledge acquisition approach. Finally, it is exposed which of the current web search engine is more reliable and useful to annotate web contents.

3.3.1 Web search engines classification

There are two main types of search engines [*Yeol and Hoffman*, 2003]:

- *Keyword-based search engines* (*e.g.* Google, Altavista, MSN Search, Yahoo): by far the most successful way for accessing available web resources. They apply simple but effective automatic keyword-based algorithms in order to retrieve web sites that match with a specific query. Moreover, they try to rank the list of returned web sites according to their relevance using several heuristics (*e.g.* Pagerank®). They offer quite complete and up-to-date lists of web sites, but their accuracy depends extremely on the adequacy and concreteness of the user's query. Moreover, it is difficult to construct the most appropriate query due to the translation between the semantic concept searched (topic) to the logic keyword-based notation used. In other words, their performance is limited due to their lack of semantic analysis. So, in many situations, they return a huge amount of resources, which have to be manually evaluated. The consequence is that, usually, only the first resources are evaluated by the user [*Jans*, 2000].
- *Taxonomic approaches*: their goal is to solve the information-overload problem, caused by a usually long list of retrieved documents in a keyword-based approach, by providing a set of document clusters (or categories) and organising them in a hierarchical structure. Clusters are determined by a term taxonomy that is provided by human experts or dynamically defined in function of the retrieved documents. There are two important approaches:
  - Web catalogues or directories, such as Google, consist of a huge human-classified catalogue of documents, which can be browsed by following a pre-defined hierarchical structure (as shown in **Figure 4**). The assignment of documents to the appropriate category is accurate only in the context that the human classifier has assumed. However, the manual updating is not appropriate to match the World-Wide Web's dynamic nature.

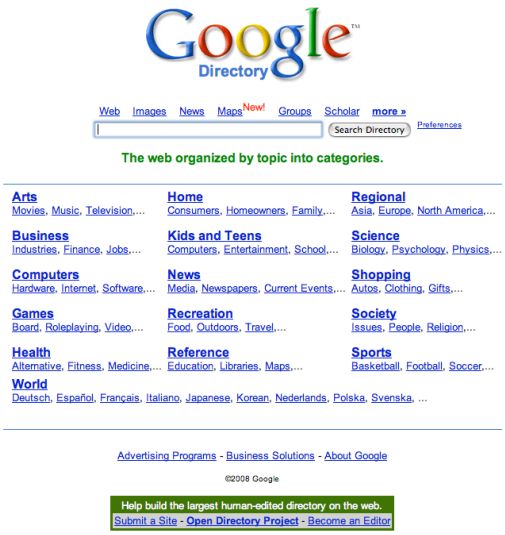


Figure 4. Google's directory start page

- Another approach consists on automatically creating a structured view of a ranked list: the idea is to group similar web resources into sets by applying clustering techniques. Some search engines are summarized in **Table 4** and several examples of the results presented by some of those systems are presented in **Figure 5** and **Figure 6**. Their goals are: 1) to create a hierarchical view automatically for each query, 2) to assign only relevant documents for a query into each category at runtime, and 3) to provide a user interface which allows iterative and hierarchical refinement of the search process. However, they offer a limited and reduced amount of web resources in comparison to term-based search engines; on the other hand, the obtained categories present poor semantics and lack of a good structure. This hampers the comprehension of the domain structure and the browsing of the available resources. Moreover, if the domain is concrete (*e.g.* a query with two keywords), in most cases, no classification will be obtained. In other cases, they only cover a certain domain of knowledge (*e.g.* scientific or technical domains) and depend on manual construction of the presented categories (even with an automatic classification of web resources). In this sense, we can offer a potential contribution in the area of structuring web resources into a meaningful representation using our automatically acquired knowledge for the domain. As will be discussed later, this can be considered as an improvement over current systems.

TABLE 4. OVERVIEW OF SEVERAL CLUSTER-BASED SEARCH ENGINES.

Cluster search engine	*URL*	*Description*
Scatter/Gather System [Cutting *et al*., 1992]	http://www.sims.berkeley.edu/~hearst/sg-overview.html	- Designed for browsing - Based on two novel clustering algorithms · *Buckshot* – fast for online clustering · *Fractionation* – accurate for offline initial clustering of the entire set
Carrot2 [Stefanowski and Weiss, 2003]	http://demo.carrot2.org/demo-stable/main	- Component framework - Allows substituting components
WiseNut	http://www.wisenut.com	- Query refinements - Online; Commercial
Vivisimo/ Clusty	http://www.vivisimo.com http://www.clusty.com	- Online; Commercial - Hierarchical - Conceptual
NorthernLight	http://www.northernlight.com	- Business research content only - Online; Commercial
Grouper [Zamir and Etzioni, 1999]	http://www.cs.washington.edu/research/projects/WebWare1/www/metacrawler	- Online - Operates on query result snippets - Clusters together documents with large common subphrases - Suffix Tree Clustering (STC) - STC induces labelling
Mapuccino [Maarek *et al*., 2000]	N/A	- Relatively efficient - Similarity-based on vector-space model
SHOC [Zhang and Dong, 2004]	N/A	- Grouper-like - Key phrase discovery

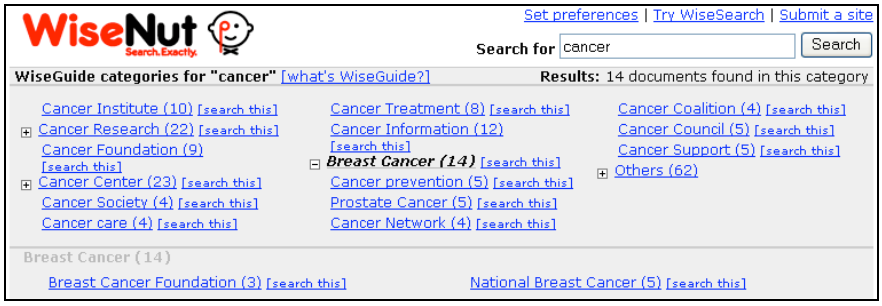


Figure 5. Clusters of web resources proposed by WiseNut for the *Cancer* domain.

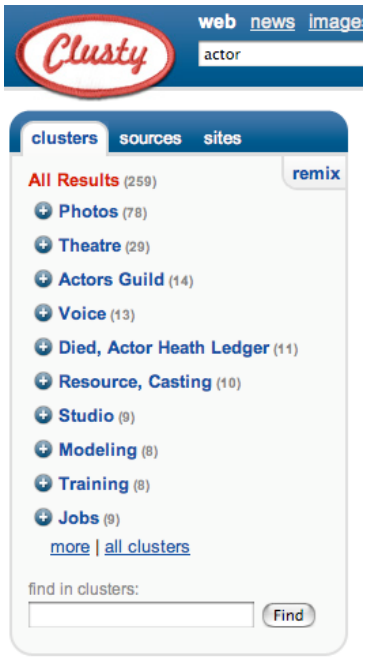


Figure 6. Clusters of web resources proposed by Clusty for the *actor* domain.

3.3.2 Web search engines as learning tools

Taking into consideration our corpus requirements (obtain a representative and up-to-date set of web resources from which to acquire knowledge) and the independency from the searched domain, we have opted for using keyword-based web search engines as the tool for obtaining the necessary corpus of web documents. They are very useful when the query is representative and concrete enough. The ranked list of web resources is quite updated and accurate thanks to the continually evolving scores obtained by the ranking methodology (*e.g.* Pagerank for Google). Moreover, the lack of any semantic analysis makes them suitable for any kind of possible domain of knowledge regardless of its generality. They will be considered as our particular experts for corpus selection with the advantage that they are experts in all types of domains. Even though the offered ranking of web sites is an added value, this proposal does not depend directly on the scoring algorithm. In other words, even without any sorting of web resources our procedure is able to use the web search engines properly, as the order of the results obtained does not influence on their usage.

In more detail, there are several aspects of web search engines that may result in a valuable aid in the knowledge acquisition process:

- The key point to obtain the maximum profit of keyword-based search engines is to construct the queries that will result in an adequate set of web resources at a certain moment of the analysis. As will be described in §4.4, these queries will be

created dynamically in function of the content to be annotated and the knowledge to extract. Query issues are closely related to the problems presented by the Web against traditional information retrieval systems. In spite the fact that most Web queries are only two words long, and that is insufficient to identify the context [*de Lima*, 1999]; our proposal Typical IR queries involve long queries [*Hearst*, 1996] that can contextualize enough to obtain a suitable and -sometimes- reduced set of results.

- In addition to the list of web sites for a certain query, search engines will be also used to obtain previews of the information contained in the Web. Those are presented in the form of the context in which the queried keyword(s) is(are) presented (see **Figure 7**). These previews, typically called *snippets*, even offering a narrow context, are informative enough to extract related knowledge without accessing the web's content.

Figure 7. Search snippet with keywords emphasized

- The last and the most important use of web search engines is to obtain global statistics about information distribution in the whole Web. These statistics about the presence of a certain query term in the Web can be computed efficiently from the estimated amount of returned results (see **Figure 8**) as described in §3.4. This is a very important point, as the discovery of the true relative frequencies of words and phrases in society is a major problem in applied linguistic research. In this sense, the number of resources of the Web is so vast, and the number of web authors generating web pages is so enormous (and can be assumed to be a truly representative very large sample from humankind) that the probabilities of web search engine terms, conceived as the frequencies of page counts returned by the search engine divided by the number of indexed pages, approximate the actual relative frequencies of those search terms as actually used in society [*Cilibrasi and Vitanyi*, 2004]. Based on this premise, some authors [*Economist*, 2005] have mentioned that the *relative page counts* of a web search engine can approximate the true societal words and phrases usage. This measure is very interesting if the adequate queries are formulated (introduced in §3.4) as it can give us an idea of the generality of a discovered concept or relation. Those measures, even estimated, can save us from analysing a large quantity of resources in order to obtain representative statistics, improving the scalability and the performance of the learning process with independence of the generality of the searched domain. The use of web search engines for obtaining valuable statistics for information retrieval and knowledge acquisition has been applied previously by several authors [*Cilibrasi and Vitanyi*, 2006; *Cimiano and Staab*, 2004; *Etzioni et al.*, 2004; *Turney*, 2001] obtaining good quality results in relation to classical statistical approaches.

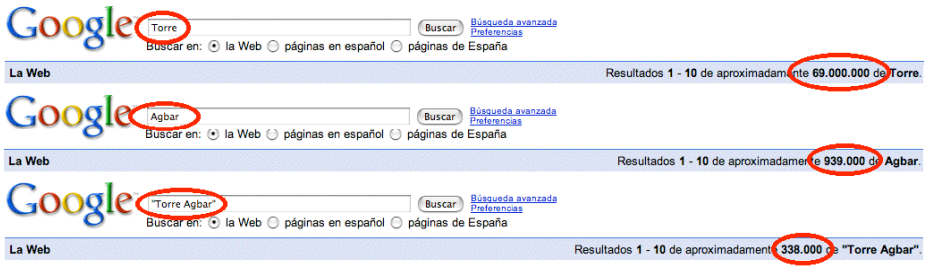


Figure 8. Statistics about query terms presence in the Web returned by Google.

Even presenting all those advantages, the best keyword-based search engines available (like Google or Altavista) have some limitations that can influence negatively in web-based information retrieval tasks [*Etzioni et al.*, 2004]:

- Assuming that with very general results (*e.g.* millions of available web resources), most users will only evaluate the first ones, which are considered the most relevant, only the first 1000 web sites are presented. So, even with a very general query we will only be able to access the first 1000 web resources. This is an assumption derived again from the redundancy of information hypothesis and the premise that web search engines are able to rank the webs according to their importance: it will be possible to find the desired information without having to analyse the whole set of web resources. However, this restriction[1] does not represent a limitation for our approach, as on the one hand we use the most common knowledge to verify the correctness of the knowledge extracted from the documents, and on the other hand, in further steps, we are focused on obtain class candidates without regarding the number of candidates, so 1000 web resources are enough to obtain this information.
- Another possible drawback can be the overhead introduced in the learning process by the response time of those web search engines for a specific query (in addition to the online accessing to the individual resources themselves). However, comparing this delay with the runtime required to obtain the same robust statistics from the analysis of a wide corpus, the benefits are clear.

3.3.3 Keyword-based search engine comparison

From the discussion presented in the previous section, it is clear the importance of the search engine for our knowledge acquisition methodology. This is why we have

[1] To overcome this restriction, a simple algorithm like Recursive Query Expansion (RQE) [*Etzioni et al.*, 2004] can coax a search engine to return most if not all of its results. In essence, the algorithm constructs recursively different queries from an initial one by adding new key terms from a repository of common words. This forces the searcher to return a different set of results but without altering the initial meaning of the word. The result is a wider set of final results with a much higher amount of web sites.

studied the available alternatives in order to select the most adequate search engine for our purposes.

Publicly available widely used keyword-based search engines have been considered. This will ensure that the search engine will be available and the quality of service maintained during the development. Concretely, Google, Yahoo and MSNSearch have been considered. Other widely used searchers such as Altavista and AlltheWeb use the database provided by Yahoo, offering very similar results.

In [*Sánchez*, 2008], each analysed search engine has been evaluated from different points of view:

- *Access*: some search engines (such as Google) offer only access for programmers through calls to a specific API. Others only allow querying the web interface and parsing the results page. The first option is preferred as it is independent of the graphical representation of the results.
- *Limitations*: most search engines include access limitations in order to avoid hacker attacks and maintain the quality of service. Those are referred to a certain amount of queries performed per day or consecutively from a particular IP address.
- *Response time*: this is referred to the amount of time in which the results for a particular query are presented. Some search engines (such as Google) offer low priority access to API-based queries or introduce courtesy waits between consecutive queries.
- *Coverage*: the amount of web resources that a particular search engine is able to index for a particular query. In our case, the web coverage for general terms is not as important as the number of results presented for very concrete queries. This is because we do not intend to analyse millions of web resources for a very general query (that will correspond to the firsts steps of the learning process); on the contrary we desire that a very concrete query (*e.g.* with less than 100 results) returns the biggest amount of resources. In this last case, the higher degree of contextualization of the learning process will allow to obtain valuable domain information. In relation to the computation of web scale statistics, the absolute measure returned is not that important, as our main statistical employed measures are *relative*.

The results of the analysis performed for each search engine are summarised in **Table 5**, **Table 6** and **Table 7**. The first two are referred to the coverage of each one, presenting some results obtained for different example queries of typical domains considered during the development.

TABLE 5. NUMBER OF ESTIMATED RESULTS OBTAINED BY SEVERAL KEY-BASED WEB SEARCH ENGINES FOR GENERAL CONCEPTS.

Concept	***Google***	***Yahoo***	***MSN Search***
Keanu Reeves	8.100.000	16.600.000	3.340.000
Barcino	461.000	554.000	158.000
The Matrix	24.000.000	73.100.000	18.500.000
Torre Agbar	338.000	327.000	73.300

In **Table 5**, general queries are performed, obtaining an enormous amount of potential results. Yahoo is offering the largest amount of web resources in all cases, Google is in the middle, and MSNSearch returns an amount that is almost one order of magnitude lower. However, it should be considered that MSNSearch does not count redundant web sites as the other search engines do by default.

TABLE 6. NUMBER OF ESTIMATED RESULTS OBTAINED BY SEVERAL KEYWORD-BASED WEB SEARCH ENGINES FOR SPECIFIC QUERIES.

Query	***Google***	***Yahoo***	***MSNSearch***
"cities like Barcelona"	10.100	24.300	8.330
"Torre Agbar is a Monument"	1	3	0
"films such as 300"	653	1.300	736
"producers including Joel Silver"	22	20	10

In **Table 6**, very specific queries are performed in order to test the effective coverage for very narrow domains. In this case it is quite evident that Google offers the highest numbers, followed by Yahoo and, to considerable distance, MSNSearch.

TABLE 7. SUMMARY OF THE MAIN CHARACTERISTICS OF EACH WEB SEARCH ENGINE.

Search engine	***Access***	***Limitations***	***Coverage***	***Response time***
Google	API Web access	1000 queries per day and account. Several accounts per IP allowed	**Highest**	*Slowest*
Yahoo	API Web access	5000 queries per day, account and IP	Medium	Medium
MSNSearch	Web access	**No limits**	*Lowest*	**Fastest**

Taking those facts into consideration, **Table 7** shows summary of the analysed features of each search engine. Google has the best Web coverage but its very limited access and extremely slow response times through the search API, introducing courtesy waits of several seconds for consecutive queries really hampers its usefulness. On the other hand, MSNSearch offers a really good performance through the web interface with no limitations (even performing thousands of consecutive queries) at the cost of a reduced coverage especially for the most concrete queries. Yahoo stays at an intermediate point with slightly lower response and better coverage time than MSNSearch, but introducing access limitations.

The results of this empirical study are quite similar to those presented in [*Dujmovic and Bai*, 2006], in which the three search engines are exhaustively compared in relation to their *functionality*, *usability*, *IR performance* and *IR quality*.

The conclusion is that there does not exist a perfect search engine for our purposes. However, Google potentially offers the best recall for concrete domains with limited resources at the cost of a very limited access (not enough for small sized documents). MSNSearch behaves in a complementary way, making it adequate for wide domains. This is because, due to the high redundancy of the Web, once a significant amount of web resources has been retrieved, the extracted knowledge using different search

engines tends to be the same. With respect to the web scale statistics, although the absolute values for a specific query may be quite different (as observed in **Table 5** and **Table 6**), due to the particular estimation algorithm employed by each web searcher, the final score computed from those values tends to be very similar as they are *relative* measures.

3.4 Web-scale statistics

In general, the use of statistical measures (*e.g.* co-occurrence measures) in knowledge related tasks for inferring the degree of relationship between concepts is a very common technique when processing unstructured text [*Grefenstette*, 1992; *Lin*, 1998]. However, statistical techniques typically suffer from the *sparse data problem* (*i.e.* the fact that data available on words of interest may not be indicative of their meaning). So, they perform poorly when the words are relatively rare, due to the scarcity of data. This problem can be addressed by using lexical databases [*Lee et al.*, 1993; *Richardson et al.*, 1994] or with a combination of statistics and lexical information, in hybrid approaches [*Jiang and Conrath*, 1997; *Resnik*, 1998]. In this sense, some authors [*Brill*, 2003] have demonstrated the convenience of using a wide corpus in order to improve the quality of classical statistical methods. Concretely, in [*Turney*, 2001] methods to address the sparse data problem are proposed by using the hugest data source: the Web.

However, the analysis of such an enormous repository for extracting candidate concepts and/or statistics is, in most cases, impracticable. Here is where the use of lightweight techniques that can scale well with high amounts of information, in combination with the statistical information obtained directly from the Web, can represent a good deal. In fact, on the one hand, some authors [*Pasca*, 2004] have enounced the need of using simple processing analysis when dealing with such a huge and noise repository like the Web; on the other hand, other authors [*Cilibrasi and Vitanyi*, 2006; *Cimiano and Staab,* 2004; *Etzioni et al.*, 2005] have demonstrated the convenience of using web search engines to obtain good quality and relevant statistics.

In order to statistically assess the relation between words, one can consider the standard collocation function between terms (**1**):

$$c_k(a,b) = \frac{p(ab)^k}{p(a)p(b)} \qquad (\mathbf{1})$$

,being $p(a)$ the probability that the word a occurring within the text and $p(ab)$ the probability of co-occurrence of words a and b. From this formula, one can define the *Symmetric Conditional Probability* (SCP) [*Ferreira da Silva and Lopes,* 1999] as c_2 and the *Pointwise Mutual Information* (PMI) [*Church et al*, 1991] in the form $log_2 c_1$. This last one has been adapted to approximate term probabilities with web search hit counts by [*Turney,* 1992].

In that work, several heuristics for exploiting the statistics provided by web search engines are presented. Those measures, known as "web scale statistics" have been further discussed in [*Etzioni et al.*, 2004]. They use a form of *pointwise mutual*

information (PMI) [*Church et al*, 1991] between words and phrases that is estimated from Web search engine hit counts for specifically formulated queries.

The conclusion is that the degree of relationship between a pair of concepts can be measured through a combination of queries made to a Web search engine (involving those concepts and, optionally, their context). Queries are constructed using the logical query language (AND, OR, NOT…) provided by the search engine. As an example, a typical score measure of co-occurrence between an initial word (*problem*) and a related candidate concept (*choice*) presented in [*Turney*, 2001] is (**2**).

$$Score(choice, problem) = \frac{hits(problem\ AND\ choice)}{hits(choice)} \quad (\mathbf{2})$$

This score is derived from probability theory. Here, p(*problem* AND *choice*) is the probability that *problem* and *choice* co-occur. If *problem* and *choice* are statistically independent, then the probability that they co-occur is given by the product p(*problem*)p(*choice*). If they are not independent, and they have a tendency to co-occur, then p(*problem* AND *choice*) will be greater than p(*problem*)p(*choice*). Therefore the ratio between p(*problem* AND *choice*) and p(*problem*)p(*choice*) is a measure of the degree of statistical dependence between *problem* and *choice*. Since we are looking for the maximum score among a set of *choices* –or *candidates*-, we can drop p(*problem*) because it has the same value for all choices, for a given problem word, obtaining the final expression.

Those measures have been extensively used to evaluate the relevance of a set of candidates [*Cimiano and Staab*, 2004]. However, the problem of obtaining those candidates remains open. In consequence, a certain degree of knowledge (*e.g.* synsets from WordNet –see §3.6- [*Turney* 2001]) or a previous analysis is still necessary in order to at least discover a representative set of candidates. On the other hand, the usage of these measures can produce a big network overload, so it is also important to find solutions to reduce this usage without losing information quality. In §3.6, we also propose a solution for this issue.

We found this work in the analysis of a corpus obtained from the search engine, to firstly use this corpus to verify the correctness of the extracted candidates to be annotated, and in further steps to verify/quantify the relationship between the class candidates extracted with the ones contained in the used ontology.

3.5 Instance-Concepts relationships

In spite of the above-mentioned techniques to extract the needed information from the texts, or the techniques used to quantify the relationships between concepts, we needed a solution those relationships, concretely *is-a* relationships. In this manner, there exist many approaches for performing this task. However, as we intend to define an unsupervised, domain independent approach, appropriate techniques should be employed. As stated in [Cimiano *et al*., 2004], three different learning paradigms can be exploited. First, some approaches rely on the document-based notion of term subsumption [Sanderson and Croft, 1999]. Secondly, some researchers claim that words or terms are semantically similar to the extent to which they share similar

syntactic contexts [Bisson *et al*., 2000; Caraballo, 1999]. Finally, several researches have attempted to find taxonomic relations expressed in texts by matching certain patterns associated to the language in which documents are presented [Ahmad *et al.,* 2003; Charniak and Berland, 1999].

Pattern-based approaches are heuristic methods using regular expressions that have been successfully applied in information extraction. The text is scanned for instances of distinguished lexical-syntactic patterns that indicate a relation of interest. This is especially useful for detecting specialisations of concepts that can represent *is-a* (taxonomic) relations [Hearst, 1992] or individual facts [Etzioni *et al*., 2005]. The most important precedent is [Hearst, 1992], in which a set of basic domain independent patterns for hyponymy discovery and a methodology for obtaining new patterns are described (see some examples in **Table 8**).

TABLE 8. EXAMPLES HEARST LINGUISTIC PATTERNS (NP=NOUN PHRASE).

Pattern	***Example***	***Relation***
NP {,} including {NP ,}* {or | and} NP	… countries including Spain, or France.	hyponym("Spain", "countries"), hyponym("France", "countries")
such NP as {NP ,}* {(or | and)} NP	… such actors as Keanu Reeves, Laurence Fishburne, and Bruce Lee.	hyponym("K. Reeves", "actors"), hyponym("L. Fishburne", "actors"), hyponym("Bruce Lee", "actors")
NP {,} such as {NP ,}* {or | and} NP	… films such as The Matrix, and 300.	hyponym("The Matrix", "films"), hyponym("300", "films")
NP {,} especially {NP ,}* {or| and} NP	… cities, especially Tarragona, and Reus.	hyponym("Tarragona", "cities"), hyponym("Reus", "cities")

These patterns summarize the most common ways of expressing relationships between concepts in English. Consequently, many authors [Pasca, 2004; Cimiano *et al*., 2005] have refined or used them as the base for their taxonomy learning methodologies.

However, the quality of pattern-based extractions can be compromised by the problems of *decontextualisations* and *ellipsis*. For instance, *decontextualisations* can easily be found in sentences like "*There are several newspapers sited in big cities such as El Pais and El Mundo*"; without a more exhaustive linguistic analysis we might erroneously extract *"El Pais"* and *"El Mundo"* as instances of *"city"*. For the second case, due to language conventions, we can find a sentence like "*teams such as Barcelona and Madrid*"; in this case, the ellipsis of the words "*Futbol Club*" and *"Club de Futbol Real"* respectively could result in the incorrect conclusion that "*Barcelona*" and "*Madrid*" (and not "*breast cancer*" and "*lung cancer*") are subtypes of "*teams*" instead of *"Futbol Club Barcelona"* and *"Club de Futbol Real Madrid"*.

As a final note, pattern-based approaches present a relatively high precision but typically suffer from low recall due to the fact that the patterns are rare in corpora [Cimiano *et al*., 2004]. Fortunately, as stated in §3.4, this data sparseness problem can be tackled by exploiting the Web [Buitelaar *et al*., 2003; Velardi *et al.,* 2005].

Unsupervised pattern-based learning is one of the bases of our approach. In chapter 4 we present the patterns we have used, and we also demonstrate that pattern's regular expressions can be used to construct web search engine queries to retrieve documents and compute statistics making them powerful.

3.6 Electronic dictionaries as a knowledge repository. WordNet

Even though the usage of web-based techniques obtains high quality results, it has a high temporal cost [*Cimiano et al.*, 2005] and it creates dependency of external tools. For this reason, an alternative solution is required to reduce its usage, losing the less quality as possible. The electronic dictionaries are a good solution, as they usually give us a large set of concepts with their definitions, they usually give basic classifications of concepts, and they even give relationships between the different concepts they have. Although they usually could not be used to substitute the web-based techniques, these dictionaries could be used to filter the results obtained with these measures.

WordNet is a general-purpose semantic electronic repository for the English language. In this section, an overview of its characteristics, structure and potential usefulness for our purposes is described.

The idea is to use the electronic dictionaries to reduce the usage of web-based statistics. In our proposal we use these tools to quantify the correctness of the entities to be annotated and to filter the candidates to be evaluated. In this manner we are able to reduce drastically the usage of web search engine queries.

WordNet[2] is the most commonly used online lexical and semantic repository for the English language. Many authors have contributed to it [*Daudé et al*., 2003; *Farreres et al*., 2004; *Meaning*, 2005] or used it to perform many knowledge acquisition tasks. In more detail, it offers a lexicon, a thesaurus and semantic linkage between the major part of English terms. It seeks to classify words into many categories and to interrelate the meanings of those words. It is organised in synonym sets (synsets): a set of words that are interchangeable in some context, because they share a commonly agreed upon meaning with little or no variation. Each word in English may have many different senses in which it may be interpreted: each of these distinct senses points to a different synset. Every word in WordNet has a pointer to at least one synset. Each synset, in turn, must point to at least one word. Thus, we have a many-to-many mapping between English words and synsets at the lowest level of WordNet. It is useful to think of synsets as nodes in a graph. At the next level we have lexical and semantic pointers. A semantic pointer is simply a directed edge in the graph whose nodes are synsets. The pointer has one end we call a *source* and the other end we call a *destination.*

Some interesting semantic pointers are:

- *hyponym*: X is a hyponym of Y if X is a (kind of) Y.
- *part meronym*: X is a part meronym of Y if X is a part of Y.
- *member meronym:* X is a member meronym of Y if X is a member of Y.
- *attribute*: A noun synset for which adjectives express values. The noun *weight* is an attribute, for which the adjectives *light* and *heavy* express values.
- *similar to*: A synset is similar to another one if the two synsets have meanings that are substantially similar to each other.

Finally, each synset contains a description of its meaning, expressed in natural language as a gloss. Example sentences of typical usage of that synset are also given.

[2] http://wordnet.princeton.edu/

All this information summarizes the meaning of a specific concept and models the knowledge available for a particular domain. Using this information it is possible to compute the similarity and relatedness between concepts. There have been some initiatives for computing these measures, such as the software *WordNet::Similarity* [*Pedersen et al*., 2004]. It offers an implementation of some standard measures that have been widely used by several authors to perform different WordNet-based disambiguation tasks [*Budanitsky and Hirst*, 2001; *William*, 2002].

More concretely, *similarity measures* use information found in an *is-a* hierarchy of concepts and quantify how much a concept A is like another concept B. WordNet is particularly well suited for similarity measures, since it organizes nouns into *is-a* hierarchies and, therefore, it can be adequate to evaluate taxonomic relationships. However, as described, concepts can be related in many ways beyond being similar to each other (*i.e.* through the mentioned semantic pointers). This information, in conjunction to gloss descriptions, can be brought to bear when creating *measures of relatedness*. As a result, those last measures are more general than similarity ones.

TABLE 9. CLASSIFICATION OF MEASURES OF SEMANTIC SIMILARITY AND RELATEDNESS AND THEIR RELATIVE ADVANTAGES AND DISADVANTAGES AS STATED IN [PEDERSEN *ET AL*., 2006].

Type	***Name***	***Principle***	***Pros***	***Cons***
Path Finding	Path Length	Count of edges between concepts	- Simplicity	- Requires a consistent hierarchy - No multiple inheritance - WordNet nouns only - *IS-A* relations only
	[*Wu and Palmer*, 1994]	Path length to subsumer, scaled by subsumers path to root	- Simplicity	- WordNet nouns only - *IS-A* relations only
	[*Leacock Chodorow*, 1998]	Finds the shortest path between concepts	- Simplicity	- WordNet nouns only - *IS-A* relations only
	[*Hirst and St-Onge*, 1998]	Based in WordNet synsets	- Measures relatedness of all parts of speech - More than *IS-A*	- WordNet specific
Info. Content	[*Resnik*, 1998]	Information Content (IC) of the least common subsumer (LCS)	- Uses empirical information from corpora	- Does not use the IC of individual concepts, only that of the LCS - WordNet nouns only - *IS-A* relations only
	[*Jiang and Conrath*, 1997]	Extensions of Resnik; scale LCS by IC of concepts	- Takes into account the IC of individual concepts	- WordNet nouns only - *IS-A* relations only
Context Vector Measures	[*Patwardhan and Pedersen*, 2006]	Creates context vectors that represent meaning of concepts from co-occurrence statistics	- Relatedness POS - No structure required - Uses Knowledge implicit in a corpus	- Definitions can be short, inconsistent - Computationally intensive

The available measures (compared in **Table 9**) can be grouped in three types:

- *Path finding*: as a similarity measure, it finds the path length between two concepts in the *is-a* hierarchy of WordNet. The path length is then scaled by the depth of the hierarchy in which they reside to obtain the relatedness of the two concepts.

- *Information content*: it indicates the specificity of a concept. Information content is derived from corpora, and it is used to augment the concepts in the WordNet *is-a* hierarchy. The measure of relatedness between two concepts is the information content of the most specific concept that both concepts have in common (*i.e.* their lowest common subsumer in the *is-a* hierarchy).
- *Context vector*: it does not depend on the interlinkage between words that, in some situations, has a poor coverage in the WordNet. In more detail, this measure incorporates information from WordNet glosses as a unique representation for the underlying concept, creating a co-occurrence matrix from a corpus made up of the WordNet glosses. Each content word used in a WordNet gloss has an associated context vector. Each gloss is represented by a gloss vector that is the average of all the context vectors of the words found in the gloss. Relatedness between concepts is measured by finding the cosine between a pair of gloss vectors.

From all of these measures, context vector ones offer the best performance in general situations [*Patwardhan and Pedersen*, 2006]. Moreover, they do not depend on the degree of semantic interlinkage between the considered concepts (that is mainly limited to taxonomic relationships). However, its temporal cost is high, and it works better when dealing with general –non-taxonomic- relationships.

On the other hand, the path similarity measures have better temporal costs and its behaviour fits better for our purposes, as we use these measures to quantify is-a similarity between concepts –because our work is focused on taxonomic relationships-. This set of comparisons can be large, and the differences between the results obtained using different measures are not enough relevant to use more complex measures.

Even though, all measures have limitations because they assume that all the semantic content of a particular term is modelled by semantic links and/or glosses in WordNet and, in consequence, in many situations, truly related terms obtain a low score due to the relative WordNet's poor coverage for specific domains [*Turney*, 2001]. Nevertheless, these measures are some of the very few fully automatic general purpose ways of evaluating knowledge acquisition results.

3.7 Documents Annotation: The Web approach

Although the annotation procedures require of several learning techniques and knowledge repositories, we should not forgot their purpose, the annotation of documents. In this section, we make a study of the most common solutions to annotate documents for the Web and their impact over the documents. Our analysis has been focused on both decoupled solutions (Like *RDF* or *Microformats*) as on coupled solutions (Like *XPointer*).

3.7.1 Resource Description Framework

In the case of decoupled solutions, the most common technology is *RDF*[3], it is a family of The World Wide Web Consortium (The W3C) specifications originally designed as a metadata data model, but which has come to be used as a general method of modeling information through a variety of syntax formats.

The RDF metadata model is based upon the idea of making statements about Web resources in the form of subject-predicate-object expressions, called *triples* in RDF terminology. The subject denotes the resource, and the predicate denotes traits or aspects of the resource and expresses a relationship between the subject and the object. For example, one way to represent the notion "Barcelona is a big city" in RDF is as the triple: a subject denoting "Barcelona", a predicate denoting "is a", and an object denoting "big city". RDF is an abstract model with several serialization formats (i.e., file formats), and so the particular way in which a resource or triple is encoded varies from format to format.

This mechanism for describing resources is a major component in what is proposed by the W3C's Semantic Web activity: an evolutionary stage of the World Wide Web in which automated software can store, exchange, and use machine-readable information distributed throughout the Web, in turn enabling users to deal with the information with greater efficiency and certainty. RDF's simple data model and ability to model disparate, abstract concepts has also led to its increasing use in knowledge management applications unrelated to Semantic Web activity.

The subject of an RDF statement is a resource, possibly as named by a Uniform Resource Identifier (URI). Some resources are unnamed and are called blank nodes or anonymous resources. They are not directly identifiable. The predicate is a resource as well, representing a relationship. The object is a resource or a Unicode string literal.

In Semantic Web applications, and in relatively popular applications of RDF like RSS and FOAF (Friend of a Friend), resources tend to be represented by URIs that intentionally denote actual, accessible data on the World Wide Web. But RDF, in general, is not limited to the description of Internet-based resources. In fact, the URI that names a resource does not have to be dereferenceable at all. For example, a URI that begins with "http:" and is used as the subject of an RDF statement does not necessarily have to represent a resource that is accessible via HTTP, nor does it need to represent a tangible, network-accessible resource — such a URI could represent absolutely anything (as a fanciful example, the URI could even represent the abstract notion of world peace).

Therefore, it is necessary for producers and consumers of RDF statements to be in agreement on the semantics of resource identifiers. Such agreement is not inherent to RDF itself, although there are some controlled vocabularies in common use, such as Dublin Core Metadata, which is partially mapped to a URI space for use in RDF.

A collection of RDF statements intrinsically represents a labeled, directed pseudo-graph. As such, an RDF-based data model is more naturally suited to certain kinds of knowledge representation than the relational model and other ontological models traditionally used in computing today. However, in practice, RDF data is often

[3] http://www.w3.org/RDF; http://en.wikipedia.org/wiki/Resource_Description_Framework

persisted in relational database or native representations also called Triple stores, or Quad stores if context (i.e. the named graph) is also persisted for each RDF triple. As RDFS and OWL demonstrate, additional ontology languages can be built upon RDF.

As an example of usage, given that "http://en.wikipedia.org/wiki/Tony_Benn" identifies a particular resource (regardless of whether that URI could be traversed as a hyperlink, or whether the resource is *actually* the Wikipedia article about Tony Benn), to say that the title of this resource is "Tony Benn" and its publisher is "Wikipedia" would be two assertions that could be expressed as valid RDF statements. In the N-Triples form of RDF, these statements might look like the following:

```
<http://en.wikipedia.org/wiki/Tony_Benn>
<http://purl.org/dc/elements/1.1/title> "Tony Benn" .

<http://en.wikipedia.org/wiki/Tony_Benn>
<http://purl.org/dc/elements/1.1/publisher> "Wikipedia" .
```

And these statements might be expressed in RDF/XML as:

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
       <rdf:Description
       rdf:about="http://en.wikipedia.org/wiki/Tony_Benn">
                 <dc:title>Tony Benn</dc:title>
                 <dc:publisher>Wikipedia</dc:publisher>
        </rdf:Description>
</rdf:RDF>
```

To an English-speaking person, the same information could be represented simply as:

The title of this resource, which is published by Wikipedia, is 'Tony Benn'

However, RDF puts the information in a formal way that a machine can understand. The purpose of RDF is to provide an encoding and interpretation mechanism so that resources can be described in a way that particular software can understand it; in other words, so that software can access and use information that it otherwise couldn't use.

Both versions of the statements above are wordy because one requirement for an RDF resource (as a subject or a predicate) is that it be unique. The subject resource must be unique in an attempt to pinpoint the exact resource being described. The predicate needs to be unique in order to reduce the chance that the idea of Title or Publisher will be ambiguous to software working with the description. If the software recognizes *http://purl.org/dc/elements/1.1/title* (a specific definition for the concept of a title established by the Dublin Core Metadata Initiative), it will also know that this title is different from a land title or an honorary title or just the letters t-i-t-l-e put together.

The following example shows how such simple claims can be elaborated on, by combining multiple RDF vocabularies. Here, we note that the primary topic of the Wikipedia page is a "Person" whose name is "Tony Benn":

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
        <rdf:Description
        rdf:about="http://en.wikipedia.org/wiki/Tony_Benn">
                <dc:title>Tony Benn</dc:title>
                <dc:publisher>Wikipedia</dc:publisher>
                <foaf:primaryTopic>
                     <foaf:Person>
                          <foaf:name>Tony Benn</foaf:name>
                     </foaf:Person>
                </foaf:primaryTopic>
        </rdf:Description>
</rdf:RDF>
```

3.7.2 Microformats

In the case of coupled solutions, the most common technology are the Microformats. A **microformat** (sometimes abbreviated **µF** or **uF**) is a web-baseddata formatting approach that seeks to re-use existing content as metadata, using only XHTML and HTML classes and other attributes. This approach is intended to allow information intended for end-users (such as contact information, geographic coordinates, calendar events, objects classes, and the like) to also be automatically processed by software.

Although the content of web pages is technically already capable of "automated processing," and has been since the inception of the web, there are certain limitations. This is because the traditional markup tags used to display information on the web do not describe what the information means. Microformats are intended to bridge this gap by attaching semantics, and thereby obviate other, more complicated methods of automated processing, such as natural language processing or screen scraping. The use, adoption and processing of Microformats enables data items to be indexed, searched for, saved or cross-referenced, so that information can be reused or combined.

Current Microformats allow the encoding and extraction of events, contact information, social relationships and so on. Another possible application is its usage as a very simple *is-a* relationship. The 3rd version of the Firefox browser, as well as version 8 of Internet Explorer, is expected to include native support for microformats.

Microformats emerged as part of a grassroots movement to make recognizable data items (such as events, contact details or geographical locations) capable of automated processing by software, as well as directly readable by end-users. Link-based microformats emerged first. These include vote links that express opinions of the linked page, which can be tallied into instant polls by search engines.

As the microformats community grew, CommerceNet, a nonprofit organization that promotes electronic commerce on the Internet, helped sponsor and promote the

technology and support the microformats community in various ways. CommerceNet also helped co-found the microformats community site http://www.microformats.org. CommerceNet nor Microformats.org is a standards body. The microformats community is an open wiki, mailing list, and Internet relay chat channel.

Most of the existing microformats were created at the Microformats.org wiki and associated mailing list, by a process of gathering examples of web publishing behaviour, then codifying it. Some other microformats (such as rel=nofollow and unAPI) have been proposed, or developed, elsewhere.

XHTML and HTML standards allow for semantics to be embedded and encoded within the attributes of markup tags. Microformats take advantage of these standards by indicating the presence of metadata using the following attributes:

```
class
rel
rev (in one case, otherwise deprecated in microformats)
```

For example, in the text "The birds roosted at 52.48, -1.89" is a pair of numbers which may be understood, from their context, to be a set of geographic coordinates. By wrapping them in spans (or other HTML elements) with specific class names (in this case `geo`, `latitude` and `longitude`, all part of the geo microformat specification):

```
The birds roosted at
   <span class="geo">
     <span class="latitude">52.48</span>,
     <span class="longitude">-1.89</span>
   </span>
```

With this extra information the machines can be told exactly what each value represents and can then perform a variety of tasks such as **indexing it**, looking it up on a map and exporting it to a GPS device.

3.7.3 XPointer

In the case of decoupled solutions, the most used technology is XPointer[4], it is a family of The World Wide Web Consortium (The W3C) specifications originally designed as a system for addressing components of XML based internet media; however, it can also be extended to other markup languages like HTML, making it valid as an annotation technology for the web.

At the present time, XPointer is divided among four specifications: a "framework" which forms the basis for identifying XML fragments, a positional element addressing scheme, a scheme for namespaces, and a scheme for XPath-based addressing.

The XPointer language is designed to address structural aspects of XML, including text content and other information objects created as a result of parsing the document. Thus, it could be used to point to a section of a document highlighted by a user through a mouse drag action.

[4] http://www.w3.org/TR/xptr-framework/

The `element()` scheme introduces positional addressing of child elements. This is similar to a simple XPath address, but subsequent steps can only be numbers representing the position of a descendant relative to its branch on the tree.

For instance, given the following fragment:

```
<foobar id="foo">
  <bar/>
  <baz>
    <bom a="1"/>
  </baz>
  <bom a="2"/>
</foobar>
```

results as the following examples:

```
 xpointer(id("foo")) => foobar
 xpointer(/foobar/1) => bar
 xpointer(//bom) => bom (a=1), bom (a=2)
 xpointer(/1/2/1) => bom (a=1) (/1 descend into first element
(foobar), descend into second child element (baz), select first
child element (bom))
```

3.7.4 Technologies Evaluation

In the following table (**Table 10**), we have resumed the most important features of the three technologies evaluated. We have mainly focused both on their semantic possibilities as on their overload. However it is also interesting to have an idea of their usage implications.

TABLE 10. SUMMARY OF THE MAIN CHARACTERISTICS OF EACH ANNOTATION TECHNOLOGY

Technology	*Decoupling*	*Overload*	*Current Usage*	*Semantic Possibilities*
RDF	**No**	High	OWL/RDF libraries, plugins	**High**
Microformats	**No**	**Low**	**Supported by most Web clients**	Medium
XPointer	Yes	Medium	XML Framework	Low

From this comparison, it seems logical to use the RDF format or the Microformats, as they have better semantic coverage, and on the other hand, they are not decoupled. Perhaps this last feature is not desirable, the maintenance of the coherence between the document and the annotations is still under development for the XPointer technology, and so it is better to have the annotations in the same document.

On the other hand, RDF has better semantic coverage, even though the Microformats have the enough capabilities to be used as a good annotation solution and their usage implies a lower overhead.

For the reasons mentioned above, we have decided to use the Microformats to annotate the documents in our work. Moreover, their usage is not very extended in any of the current solutions, it is supported by most of the Web Clients (making the results "*understandable*"), and they fit perfectly for the annotation tasks, so it would be an extra effort of this work.

3.8 Conclusions

As seen, the aim to develop an automatic and unsupervised solution needs a big amount of techniques and technologies, furthermore if it has to be based on an unstructured repository like the Web. As stated in several works, the Web is a large and valid knowledge repository, even though it does not a have a good structure, it solves this problem by having large quantities of information in every field. Thus, it is an interesting knowledge entity if the precise tools are used over it. However, we have currently the enough tools to do so.

On the one hand, big efforts have been done to develop reliable and high-performance **Web search engines** (like Google or MSN Live!). These tools give us an easy way to access the raw knowledge, and they are very powerful if they are used properly. However, they retrieve us raw knowledge, then we need extra tools to completely understand and structure that knowledge. In this manner, the **Natural Language Processing tools** are a very good complement. Several works have been developed in this field in the last years and, as stated in §3.2, their results are good enough. Although their performance is not perfect, they have enough quality to be used in order to analyse any kind of textual contents, and in a high percentage we can obtain good results.

Although we can have the knowledge structured using the mentioned tools, if we want to quantify the quality of the new discovered knowledge we will need some **statistical measures**. In §3.4 we have seen that using the tools mentioned above over the Web, we could also extract statistical values of the knowledge we have. In this manner, we can quantify how related are two concepts, and be sure of the validity of the results obtained, as they are extracted from a large and heterogeneous repository.

As we wanted an unsupervised system, we have seen that we need a technique to relate concepts and instances in this manner. Works like [Hearst, 1992] have shown that is possible to extract this knowledge without supervision, only using the inherent **patterns** present in a language (in our case the English language), it only needs a big text repository, and tools for accessing it. Once again, the Web fits perfectly as it has large quantities of text, and the perfect tools to access it.

However, the usage of the Web is not the solution to every problem, though it solves the need of a big repository, it has performance problems associated to its usage. In this manner, we needed a solution to minimize the usage of the Web, without losing final performance. For this reason, we thought the usage of extra knowledge repositories, which could be accessed even without time or resources

penalisation than without losing the domain independence. In this manner, we thought in use **Electronic Dictionaries** to solve those problems. Effectively, solutions like **WordNet**, with its tools and knowledge "*web*" is a good solution if we want to reduce the usage of the Web, even though it cannot replace it.

Finally, any annotation system can be developed without deeply knowing what annotation means. Then, a deep analysis of the associated techniques to annotate the documents we want to annotate was prior. In this manner, we have seen that to annotate textual documents in the Web, basically we have three solutions. Each of them has its inconveniences and its advantages, but all of they can be used for this purpose, always taking into account which of their features is more interesting for the purpose they would be used.

Chapter 4

Methodology

Up to this moment, we have described in detail all the used learning methodologies and technologies, so now we are able to describe the annotation algorithm we have developed. Our algorithm is based on previous works like [*Cimiano et al*., 2005], but we have introduced some refinement, to improve both the Named Entities extraction as the Named Entities classification. In order to improve the coverage of the class candidates' extraction procedure, we have also introduced different text patterns to obtain these class candidates.

In this chapter we describe the three steps of our algorithm:

- As stated in §1.2, any automatic and unsupervised Annotation procedure should be divided in several parts: *Detection of Instances to annotate* and *Semantic tagging of those Instances to the most appropriate class of an Ontology*. In §4.1 we will describe the context of those pieces and how the techniques explained in the previous chapter fit on it.
- In §4.2, a general description of the algorithm will be done.
- The §4.3 shows the Named Entities extraction procedure. This procedure is based in four consecutive steps. The three first steps consist in extract the free text of web content, tag it and extract the Noun Phrases present there. The final step consists in a refinement of the Noun Phrases extracted, in order to discard the ones, which cannot be catalogued as Named Entities.
- In §4.4 the used set of text patterns to extract class candidates is presented. As a further matter, it is also described how this set is applied using a web search engine and how the class candidates are obtained.
- Finally, in §4.5 the procedure to relate a Named Entity with the corresponding class of the ontology (if there is one corresponding class) is shown. We describe how electronic dictionaries can be used to reduce the usage of web-based statistics measures, and how to use these measures to relate Named Entities with ontology classes.
-

4.1 Learning annotations in text

In order to tackle the annotation of textual content, two tasks should be performed. First, one has to retrieve the entities to annotate. Next, those should be tagged with the most appropriate conceptual label (class) contained in a domain ontology.

4.1.1 Detection of entities to annotate

As annotated entities are considered as ontological instances, the first step should be able to discover those real-word entities within the text. This task is typically addressed by the detection of Named Entities [*Cimiano et al.*, 2005]. Without relying on specific set of extraction rules learned from pre-tagged examples, one can extract Named Entities in an unsupervised way by detecting Proper Nouns. Those are presented in languages such as English as Noun Phrases, which can be distinguished from normal words by the presence of alphanumeric terms and capitalized letters. Those simple heuristics have been the base for developing reliable Named Entity detection techniques [*Etzioni et al*., 2004][*Pasca*, 2004]. In addition, other approaches [*Lamparter et al.*, 2004] use a thesaurus to perform the detection: if the word is not found in the dictionary, it is assumed to be a Named Entity.

We have based our approach in the combination of those two unsupervised techniques. During the extraction stage, several analyses are performed over the text to retrieve Named Entity candidates. Next, those are individually checked against a thesaurus. We use a general-purpose electronic repository for this task: WordNet. As seen in §3.6 WordNet offers a lexicon, a thesaurus and semantic linkage between the major parts of the English words. It is possible to distinguish a common word from a Named Entity using WordNet, as Named Entities usually are not present in WordNet or, in case they are, they have a semantic pointer with an *"instance of"* relationship. However, being a manually composed tool, WordNet has coverage limitations in some situations. In order to obtain more robust candidates and overpass WordNet's limitations, we use additional web resources as a learning corpus to check the suitability of the Named Entity candidates.

4.1.2 Ontology-based annotation

Once the relevant entities have been extracted from the text, they should be associated to their formal semantics (i.e. an ontological concept from which they are *instances*). This is a difficult process as, on the one hand, Named Entities are unstructured and unbounded by nature; on the other hand, the semantics which can be exploited to detect these relationships is hidden in the text from which the extraction has been performed.

Some authors [*Fleischman and Hovy*, 2002] [*Evans*, 2003] simplify this problem by using a predefined and reduced set of categories (e.g. *organization, person, location*, etc). This reduces the degree of generality of the algorithm as; in general, domain dependant relationships are omitted. Other authors try to learn instance-concept pairs from the text [*Cimiano,* 2004]. From an unsupervised point of view, this

can be done by applying distributional hypothesis (i.e. words sharing linguistic contexts are similar) [*Alfonseca*, 2002]. The degree of similarity can be also computed from the statistical analysis of term co-occurrence. In [*Evans*, 2003], web based statistics are used to cluster the most similar entities, deriving the class topology from the cluster tree.

Statistical analysis is an effective technique to develop learning techniques from large corpus (see §3.4). However, the problem of such unsupervised approach is the *data sparseness*: the fact that the available text may not be indicative of a word's meaning. So, they perform poorly when the words are relatively rare, due to the scarcity of available data. Some authors [*Brill*, 2003] have demonstrated the convenience of using a wide corpus in order to improve the quality of classical statistical methods. In this sense, the widest repository available is the Web (see §3.1). Some authors have stated that the amount and heterogeneity of information in the Web is so high that it can be assumed to approximate the real distribution of information

The use of the Web as the source from which to compute statistics has been very successful, as stated in §3.4. In fact, not only robust statistics can be extracted from the available data, but also they can be retrieved easily from the web hit count of standard web search engines. In [*Turney*, 2001], several scores are proposed to approximate collocation measures between terms (Point-wise Mutual Information – PMI-) from the web hit count retrieved from specific search queries. From the annotation point of view, those scores have been applied to detect the most suitable class to a given Named Entity [*Cimiano et al.,* 2004].

In the present work, web-based statistical measures can be directly applied to infer the Named Entity – ontological class relationship. However, considering the amount of candidates and the potentially high number of ontological classes, the number of required queries could be overwhelming. As stated in §3.6 this approach would cause scalability problems.

Without relying exclusively in web based statistical measures, it is also possible to compute the degree of taxonomic relationships between words using WordNet (see §3.6). WordNet includes a set of semantic pointers, which interrelate terms with predefined relationships (e.g. *hyponomy, meronymy, synonymy*, etc). Counting and weighting the number of semantic links between terms, it is possible to compute similarity measures. From those measures, *path-based* scores are particularly useful in this case as they measure the path length between two concepts in the WordNet's *is-a* hierarchy.

Offline WordNet queries for similarity computation are extremely efficient when compared to on-line web-based ones. However, WordNet measures are hampered by the limited coverage offered for class instances. In general, it is unlikely to find Named Entities in WordNet and, in consequence, it is not possible to compute the similarity measures required to annotate. In order to solve this issue, we have introduced an intermediate step in which Named Entities are associated unsupervisedly to concept classes automatically acquired from the Web. Those are then compared via WordNet to the classes contained in the input ontology. As we only need to discover the most similar ontological class form a reduced set, it is possible to directly apply web scale statistics to finally select/validate and assign the most appropriate annotation label.

In order to retrieve the candidate classes for a Named entity it is possible to use linguistic patterns. Those patterns express language regularities, which can be exploited to detect predefined relationships (e.g. *is-a, part-of, causation*, etc.). In this sense, the work of Hearst (see §3.5) is particularly important. She described a set of text patterns and a method to acquire the hyponymy lexical relation from unrestricted text. Nonetheless, this technique has also been used to discover instance/concept relations (as seen in [*Hearst,* 1992]). From the annotation point of view, this approach has been applied by some authors [*Cimiano et al.,* 2005] to retrieve a set of feasible concept candidates from which to directly annotate a named entity.

4.2 Algorithm description

As introduced previously, we have divided the annotation procedure in three basic steps. The first one is the detection of the Named Entities (also named Instances) in the document. The second one is an intermediate stage, which consists on the detection of the classes to which they may belong by linguistic pattern analysis over Web documents. Finally, in the third step, the class candidates are matched with the ones in the given input ontology by means of WordNet similarity measures and web-scale statistics.

MAIN ANNOTATION PROCEDURE'S PSEUDO CODE

```
Annotation (Document d, Ontology o)
{
   tagged_document = tag_document(d)
   ne = extract_named_entities(tagged_document)

   for entity in ne
   {
      for pattern in TEXT_PATTERNS
      {
        abstracts =
        download_abstracts(build_pattern(entity, pattern))

        entity.class_candidates +=
        extract_class_candidates (abstracts)
      }
      entity.class = Search_class_candidate (entity, o)
   }
   return generate_annotated_document(d, ne)
}
```

4.3 Named Entities Extraction

As stated in §4.1, we have addressed the discovery and selection of Named Entities using several techniques. Firstly, the Noun Phrases are detected using a combination

of text taggers[5]. After that, the resulting set of Noun Phrases is refined using capitalized words filtering, statistical analysis (making queries to Web search engines) and checks against WordNet.

The designed process begins when the document's HTML markup is cleaned to prepare the text to be annotated. Over this text, a four-step procedure is applied to detect Named Entities candidates. The first step consists on the detection of Noun Phrases that may contain Named Entities. This procedure is based in the composition of three taggers. The first tagging procedure uses general regular expressions (see **Table 11**) to prioritize the mark of capitalized words as Proper Nouns. Then, the text is passed over two *n-gram taggers*, which are trained with the Brown Corpus[6], in order to refine the tagging of the rest of the words. First a Unigram tagger, which tags the words assigning the tag that, is most likely for that particular word. After that, a Bigram tagger (which assigns tags depending on the preceding word) is passed over the resultant text. Once this combination of taggers is trained, it presented a tagging precision of 93.4% over the Brown Corpus.

TABLE 11. REGULAR EXPRESSIONS USED TO ANALYSE THE TEXT.

Regular Expression	***Tag***	***Description***	***Example***
[A-Z].*$	NNP	Proper Noun	Madrid
.*ing$	VBG	Gerund verb tense	distinguishing
.*ed$	VBD	Regular verb in past tense	distinguished
.*es$	VBZ	Verb in 3rd singular person, present tense	distinguishes
.*ould$	MD	Modal verb	would
.*'s	NN$	Singular common noun genitive	season's
.*s$	NNS	Plural common noun	stadiums
.*al$	JJ	Adjective	global
^-?[0-9]+(.[0-9]+)?$ | [0-9]*((\.|,)[0-9]*)*$	CD	Cardinal Number	125,000
.*	NN	Singular common noun	word

After the text has been tagged, a grammar, based on the tags shown in Figure 9, is used in order to detect the Noun Phrases, which may contain the *full name* of a Named Entity. This grammar describes the structure of a Noun Phrase which is usually composed by a central particle with one or more Proper Nouns, <NNP>+, followed and/or leaded by zero or more Nouns (both in singular as in plural), <NN|NNS>* (e.g. "*Paris*"). Usually this central particle is leaded by some optional determinants or/and adjectives forming a Noun Phrase (e.g. "*the city*", "*of lights*"). Eventually, a completely well formed Noun Phrase is composed by one or more single Noun Phrases (e.g. "*Paris the city of lights*").

[5] To tag the text the NLTK taggers are used (http://nltk.sourceforge.net, last accessed on March 28, 2008). The system uses the Brown tag-set http://www.comp.leeds.ac.uk/amalgam/tagsets/brown.html (last accessed on March 28, 2008).

[6] See http://icame.uib.no/brown/bcm.html (Last accessed on March 28, 2008)

```
NOUNP: {<NN|NNS>*<NNP>+<NN|NNS>*}
UNINP: {<DT|DTI|DTS|DTX|PP\$>?<JJ|JJ-TL|JJR|JJT|JJS>?<NOUNP>}
NP:    {<UNINP>?<UNINP><UNINP>?}
```

Figure 9. Noun Phrases detection grammar

Considering the possibilities of WordNet, as introduced in §4.1, in the third step it is used to distinguish common words from the proper nouns. In case that all the NNPs in the Noun Phrase are found in WordNet the candidate is immediately discarded, as it is a commonly used word. However, if one or more of them are not found, or they are found as WordNet *Instances*, they are considered as valid candidates and they are evaluated in the fourth step.

The fourth step is based on the statistical distribution of the candidates over the Web. It is used to detect and discard misspelled terms and to confirm, with the information distribution from the Web, that the candidate is certainly a Named Entity. So, the remaining candidates are evaluated against their spelling in the Web. Each candidate is queried in a publicly available web search engine. The abstracts obtained are joined in one piece of text, and the text composing the candidate is searched into this snippet set. The probability to find the text written as it is in the original form (i.e. in a Named Entity-like form) is evaluated using the following formula (**3**):

$$Score(Named\ Entity) = \frac{\#\ Case\ Sensitive\ Matchings}{\#\ Case\ Sensitive\ Matchings + \#\ Case\ Unsensitive\ Matchings} \quad (\mathbf{3})$$

It compares the number of matches written equally (same uppercase letters position and same letters) with the total matches (same letters) found. If it is higher than a certain threshold, the candidate is considered as a Named Entity. A minimum number of total matchings is also required to filter misspelled entities (quite common in the case of proper names) that cannot be detected using the previous analyses.

4.4 Class candidates extraction

When the set of Named Entities for a web resource has been found, it is imperative to determinate the domain class to which they should be annotated. As introduced in §4.1, we have introduced an intermediate step to extract class candidates. Pattern-based constructions are used as queries to a web search engine and the retrieved web resources are used as the corpus from which to extract class candidates. We use WordNet to relate those class candidates with the input ontology classes (as it is likely that both are contained on it).

Using the approach proposed by [*Hearst*, 1992], we use a set of taxonomical patterns to discover Named Entities' classes (see **Table 12**, where CONCEPT is the last Noun in the last Noun Phrase before the mark -in the first three patterns- or in the first Noun Phrase after the text mark -in the last two-).

TABLE 12. LIST OF HEARST PATTERNS USED TO RETRIEVE CLASS CANDIDATES

Pattern Name	*Pattern structure*	*Example*
HEARST 1	CONCEPT such as (INSTANCE)+ ((and | or) INSTANCE)?	Cities such as Barcelona or Madrid
HEARST 2	CONCEPT (,?) especially (INSTANCE)+ ((and | or) INSTANCE)?	Countries especially Spain and France
HEARST 3	CONCEPT (,?) including (INSTANCE)+ ((and | or) INSTANCE)?	Capitals including London and Paris
HEARST 4	INSTANCE (,?)+ and other CONCEPT	Eiffel Tower and other monuments
HEARST 5	INSTANCE (,?)+ or other CONCEPT	Coliseum or other historical places

In addition, two new patterns have been added to this list as, after some experimental results, we found that they provide good contextualization. They are formally described in **Table 13**.

TABLE 13. ADDITIONAL TEXT PATTERNS

Pattern Name	*Pattern structure*	*Example*
PATTERN 1	INSTANCE (,?)+ is a | are a CONCEPT	Paris is a beautiful city
PATTERN 2	INSTANCE (,?)+ like other CONCEPT	Taj Mahal like other mausoleums

All the patterns are used in conjunction with the Named Entity candidates extracted in the previous step as queries to the web search engine, replacing the INSTANCE part by each Named Entity. After that, the obtained snippet set of web resources is analyzed to extract the CONCEPT part from them in an efficient way. Each discovered CONCEPT is added to the corresponding Named Entity class candidates list.

4.5 Ontology-based annotation

In this third stage we associate the most appropriate class from the input domain ontology with each of the Named Entities we found in the first step, taking into consideration the class candidates found in the second stage. WordNet's path-length similarity measure is used as the relatedness score between the ontology classes and the class candidates of each Named Entity.

ONTOLOGY CLASS SELECTION ALGORITHM

```
Search_class_candidate (Named_Entity entity, Ontology o)
{
      similarities = []
      entity.class =""
      add_direct_match(entity.class_candidates, o)

      if entity.onto_candidates.size() == 0
      {
        similarities =
        compute_WN_similarity (entity.class_candidates, o)

        for value in similarities
        {
           if value.similarity > SIM_THRESHOLD
           {
            entity.onto_candidates.append(value)
           }
        }
      }
      entity.class =
        class_of_Max_PMIIR(entity, entity.onto_candidates)

      return entity.class
}
```

For each Named Entity, the algorithm takes all the class candidates found in the previous stage. It directly compares each possible pair of class candidates and ontological classes. When a candidate is composed by several words, it syntactically tags the class candidates and compares the main NN | NNP contained in the noun phrase. In case that one of the ontology classes is the same –word- than the class candidate, it is added to the list of possible annotating classes for further validation.

Otherwise, if a direct matching is not found, it is possible that the ontology contains one of the appropriate semantic concepts but expressed with different words. It is here where the WordNet-based similarity measure is used to assess which of the ontology classes is more similar to one of the class candidates. A threshold is set in order to demand a minimum degree of similarity. If there is not any class similar enough, we suppose that the input ontology does not have any concept related with the concrete Named Entity. As a result of the described filtering process, the most similar classes from the ontology are obtained. From our experiments, the class candidates total number is reduced from several dozens to 4-8.

A final selection step is done over the subset of classes (which have been previously selected) to choose the annotation label. In this stage, in order to properly assess/validate the most suitable ontological class for the Named Entity, we use web-based collocation measures. Concretely the previously mentioned PMI score [*Turney*, 2001] is computed (**4**). We calculate the value from the web hit count provided by a web search engine when querying the Named Entity and each of the filtered ontological classes. We choose the one with the highest value, which is selected as the final annotation label.

$$Score(Class\,Candidate) = \frac{hits(Named\ Entity\ AND\ OntologyClass)}{hits(OntologyClass)} \quad (4)$$

4.6 Annotation

Once we have obtained the different Named Entities from the document, and discovered to which class in the ontology they belong, we tag the document following an annotation standard. Several standards such as *XMLPointer*[7], *RDF*[8] and *HTML MicroFormats*[9] have been evaluated but, as our priority was to use a standard useful nowadays, we decided to use *HTML MicroFormats*. They are an extension of the basic HTML, which let enrich it with semantic information. Even though it has semantic limitations, it gives the possibility to describe the class to which one concrete web content belongs. So, each Named Entity found in a document is annotated as follows:

```
<a href="Ontology Class URL" class=" Ontology Class">Named Entity</a>
```

Figure 10. HTML MicroFormats usage

Using this notation we are able to include the semantic information needed with a low increase in the size of the document, making it usable by existing tools. For example, for the Named Entity "*Shelbyville*", which is a city, the resulting annotation would be:

```
<a href="http://Ontologies/0.3/space.owl#[City]" class="http:// Ontologies/0.3/space.owl#[City]">
Shelbyville </a>
```

Figure 11. Resulting annotation

4.7 Computational complexity

The runtime complexity of our algorithm for one document is $O(|Q|)$ where $|Q|$ is the number of queries done to the web search engine.

$|Q|$ can be split in $|N|+|N|\cdot|P|+|N|\cdot|OC|$ where $|OC|$ is the maximum number of classes in the ontology. In our case, $|P| = 7$ as we use seven patterns, so, for one document the cost is $O(|N|+7\cdot|N|+|OC|\cdot|N|)$. From this, we can conclude that the algorithm has a cost depending on the product of the number of Named Entities found and the Ontology Classes $O(|OC|\cdot|N|)$.

7 http://www.w3.org/TR/WD-xptr
8 http://www.w3.org/RDF/
9 http://microformats.org

4.8 Algorithm implementation

As this algorithm should be used on the Web, we designed it as a system based on Web Services. A Web Service is defined by the W3C as "a software system designed to support interoperable Machine to Machine interaction over a network". Then, the first decision we should carry out was to choose the proper Web Service technology over which we should develop our architecture.

There are several Web Service technologies as *SOAP*[10], *XML-RPC*[11], *JSON-RPC*[12], *raw communication, etc*. Although *raw communication* is the most efficient technology, it is quite difficult to use it, and there are some programming languages, which have different implementation of this communication.

On the other hand, the rest are easier to use and implement (most of them are supported by international organisations) and their communication is based on the *HTTP protocol* making them suitable for Web Service definition, but they have important differences at their specification level. These differences imply that each one has its advantages and inconveniences. These are mainly focused on their efficiency, their resources usage and their flexibility.

In this manner, *SOAP* is the most flexible one, as for example its basic types can be extended with descriptors WSDL, but it uses more resources than the other *HTTP-based protocols*. The other specifications are very similar (in terms of efficiency and type flexibility), having only differences at the communication language (*XML* or *JSON*), although the *JSON-RPC* is considered the alternative for the *"de facto"* standard *XML-RPC*.

Finally, *XML-RPC* is the technology we have used as it has the enough support between programming languages, and at the same time it has the enough data types for our purposes.

4.8.1 Service architecture

We have divided the algorithm in three XML-RPC accessible services -**Figure 12**-. The first one gives an API to read the ontology given, the second has an API to obtain morphological and syntactical analysis from the given text; and the last one, gives an API to annotate the web document given.

These three services can be used by a GUI (it can be even Web-based than Desktop-based), where the document to annotate could be given, and the results of the annotation can be shown. However, they can also be used alone; for example, programming languages without support for reading OWL files can use the ontology API can use. Or they can also be used as complements for other Semantic Services or Web Services.

[10] http://www.w3.org/TR/soap/
[11] http://www.xmlrpc.com
[12] http://json-rpc.org

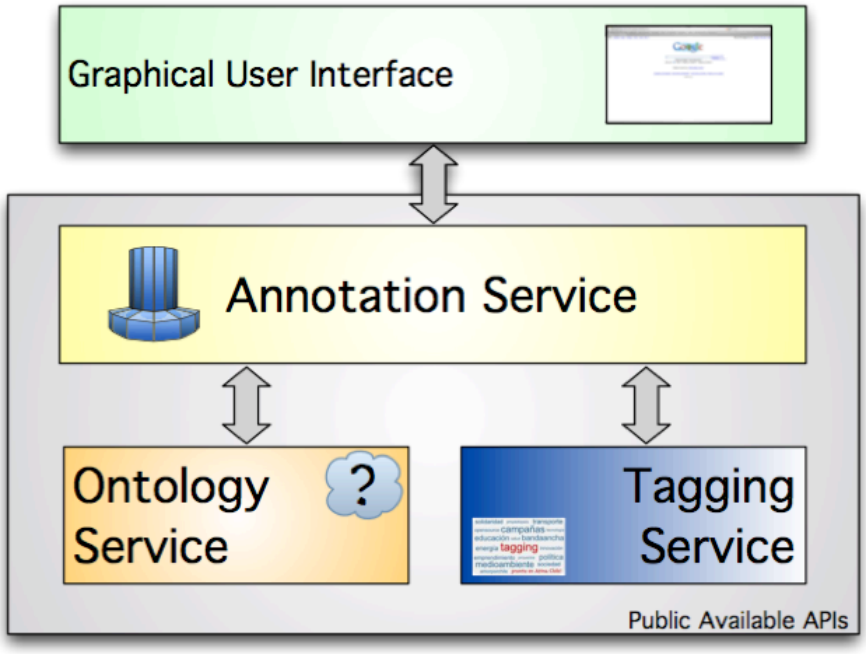


Figure 12. Annotation System Architecture

The Ontology Service offers the following functions:

- *getNumberOfClassesFromURL:* Returns the number of classes represented in the ontology contained in the passed URL.
- *getClassesFromURL:* Returns a set with the classes represented in the ontology contained in the passed URL.
- *getClassAncestorsFromURL:* Returns the class ancestors of the class passed by parameter, if it is in the URL given.
- *getClassDescendantsFromURL:* Returns the class descendants of the class passed by parameter, if it is in the URL given.

The Tagging Service offers the following functions:

- *getTextNounStatements:* Returns the Noun-Phrases which fulfil the grammar described in **Figure 9**.
- *getTextNounPhrases:* Applies the procedure described in §4.3.
- *trainTagger:* Used to train the tagger, with the Brown Corpus. It should be used before annotating contents.
- *tagText:* Tags the text passed by parameter.

Finally, the Annotation Service offers the following functions:

- *startProcess:* Applies the described annotation algorithm over the given URL by parameter.

.

Actually, implementing our algorithm in this manner we obtain some advantages. On the one hand, as it is implemented as a set of Web Services, it can be easily accessible by users and other systems, making it suitable for the Communication requirement as it can be easily executed using this APIs. On the other hand, it uses standard formats and protocols, carrying out the standard requirements. Finally, the presence of an Ontology Service makes it able to fulfil the Ontology support requirement.

Chapter 5

Evaluation

Due to the lack of solutions to carryout automatic evaluations, authors [*Alfonseca and Manandhar,* 2002] [*Cimiano et al*., 2005] [*Hahn and Schnattinger*, 1998] focus the checking of the results on the manual side. In our case, a two-step expert-based evaluation procedure has been designed. In the first step, the quality of the detected Named Entities is checked against manually extracted ones; in the second step, annotation labels assigned to those entities are manually evaluated considering the input ontology. Moreover, learning parameters which may influence in the results (e.g. selection thresholds) have been evaluated independently to empirically select the most appropriate set of values considering we focus the development on the result's reliability.

Additionally to that qualitative evaluation, we have also examined the results from the qualitative point of view. This analysis shows the type of results one may expect from the system and which kind of mistakes may appear. This information is very useful in order to have an idea about the system's applicability.

5.1 Quantitative Evaluation

Even though expert-based evaluations can lead to accurate results, they present some problems. On the one hand, manual extraction and classification of Named Entities is a laborious task, so the amount of evaluated texts and domains has to be limited. On the other hand, the annotations generated by one expert may disagree with the ones made by another expert [*Cimiano et al*., 2005].

The general evaluation has been applied over two sets of 5 Wikipedia articles. Geographical articles for well-known cities compose the first one, and cinematographic resources define the second one. Those articles have been annotated using two ontologies corresponding to both domains. In the first set we used an ontology[13] **-Figure 13-** with 188 concepts related with spatial entities; in the second

[13] http://itaka2-deim.urv.cat/ontologies/location.owl.
Original source http://212.119.9.180/Ontologies/0.3/space.owl

set, the ontology[14] -**Figure 14**- used has 59 concepts related with movie-related entities.

In order to define a relevant expert-based evaluation baseline, two domain experts have manually checked both sets of articles. In a first stage, each one is requested to extract entities, which may be suitable to be annotated. Individual results are then put together and the experts are requested to agree in those cases in which there is a lack of consensus. The final set of extracted entities for each article (an average size of 15 Named Entities for the first set, and 20 Named Entities for the second set) is compared against the automatically extracted ones. In a second stage, both experts are requested again to check the suitability of the annotations proposed by the automatic procedure, considering the classes available for each domain ontology.

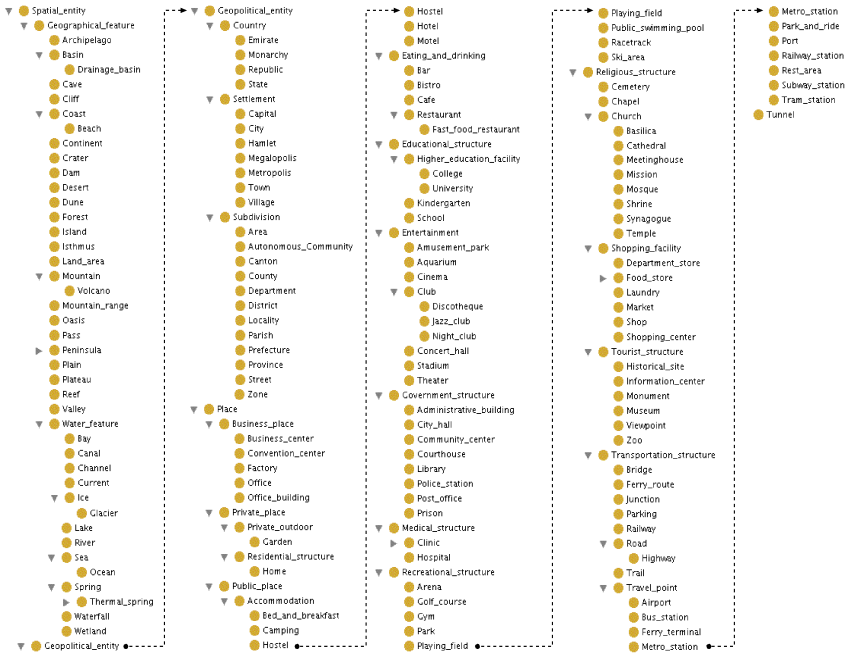


Figure 13. Location ontology diagram

[14] http://itaka2-deim.urv.cat/ontologies/film.owl.
Original source http://lsdis.cs.uga.edu/semdis/entertainment

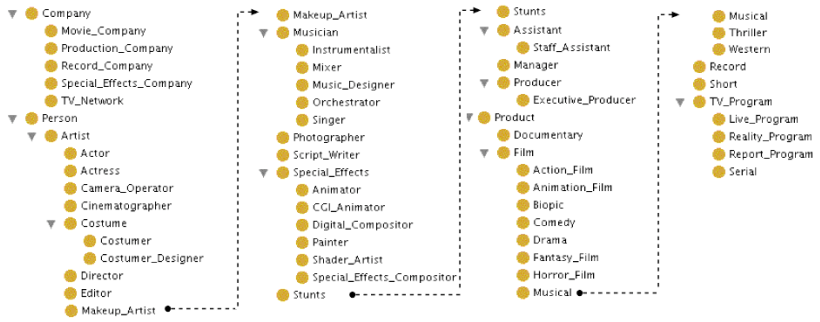


Figure 14. Film ontology diagram

Once all the results are tagged as correctly, or incorrectly retrieved/tagged, the algorithm's performance can been estimated using the following metrics [*Yang*, 1999]:

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Fallout = \frac{FP}{FP + TN} \tag{7}$$

$$Accuracy = \frac{TP + TN}{N} \tag{8}$$

$$Error = \frac{FP + FN}{N} \tag{9}$$

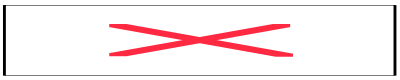
Where N is the total number of items:

$$N = TP + FP + FN + TN \tag{10}$$

And the values used are:

- *TP,* number of items correctly detected or classified (true positives)
- *FP,* number of items incorrectly detected or classified (false positives)
- *FN*, number of items incorrectly rejected (false negatives)
- *TN,* number of items correctly rejected (true negatives)

From these values we have also calculated the *F-measure*, the harmonic mean of recall and precision:



(11)

From our early experiments we set the thresholds, mentioned in §4.3 and §4.5, to the following values: the fourth step of the Named Entities detection threshold used is 0.7; and the similarity threshold when using WordNet similarity measures has been set in 0.3. However, in §5.1.3 we do a deep evaluation of these thresholds, and we extract why these values seem the most appropriate ones.

5.1.1 Evaluation of Named Entities

The results of the evaluation of this first step for both domains are summarized in **Table 14** and **Table 15**:

TABLE 14. NAMED ENTITY DETECTION EVALUATION METRICS (ARTICLE SET 1: GEOGRAPHICAL DATA)

Article	*Recall*	*Precision*	*F-Meas.*	*Fallout*	*Accuracy*	*Error*
Andorra	66.66 %	67.64%	67.15%	26.19%	70.58%	29.41%
Barcelona	37.43 %	89.72%	52.82%	11.02%	51.85%	48.14%
Tarragona	46.42 %	82.97%	59.54%	8.98%	69.36%	30.63%
Reus	15.38 %	80%	25.80%	100%	14.81%	85.18%
Palma	37.89 %	85.71%	52.55%	8.45%	60.84%	39.15%
Average	**40.76 %**	**81.21%**	**51.57%**	**30.93%**	**53.49%**	**46.50%**

TABLE 15. NAMED ENTITY DETECTION EVALUATION METRICS (ARTICLE SET 2: CINEMA DATA)

Article	*Recall*	*Precision*	*F-Meas.*	*Fallout*	*Accuracy*	*Error*
The Matrix	41.05%	77.5%	53.67%	15.51%	59.92%	40.07%
War Games	30.61%	68.18%	42.25%	15.21%	56.84%	43.15%
LOTR 3	36.15%	64.38%	46.30%	16%	62.73%	37.26%
I, Robot	30.37%	58.53%	40%	17.70%	58.85%	41.14%
300	40.20%	75%	52.34%	11.71%	65.86%	34.13%
Average	**35.68%**	**68.72%**	**46.91%**	**15.23%**	**60.84%**	**39.15%**

From the presented results, we can observe a relatively high precision (60-75%) with a lower recall (30-40%). Those are interesting results, as the final annotation precision will depend on the individual precision of each learning stage. Note also that the wrongly extracted entities may be implicitly discarded when no suitable annotations are found in the second stage. Recall is also important but can be compensated by the high redundancy of information among web resources [*Brill*, 2003]. In general, we prefer to introduce stronger constraints in the Named Entity selection procedure and present more reliable results.

We have also compared those results with other automatic approaches ([*Cimiano et al*., 2005]) and a similar corpus (also Location articles). We are able to nearly double the precision, obtaining a final F-Measure, which is around a 3% higher in average. In that approach (which represents an improvement over previous ones [*Alfonseca and Manandhar,* 2002], [*Hahn and Schnattinger*, 1998]), the Named Entity selection constraints are more relaxed, resulting in a higher number of mistakes. In our case, the use of WordNet and the introduction of the web-based assessor stage tend to present more reliable results.

We have also observed a dependency between the result's quality and the amount of English resources available in the article itself and in the Web. The fact of using several tools, which are exclusive to that language, may hamper the performance with more scarce domains (e.g. tourist destinations such as *Andorra* or *Barcelona* provide better results than *Reus*). Movie-related results are more similar as evaluated items are a more homogeneous (i.e. well known American titles in all cases).

5.1.2 Evaluation of the annotation procedure

In this second stage, the list of selected entities with the automatically proposed ontology-based annotations is presented to the human experts. They evaluate them considering the suitability of the annotation and the ontology scope. In this manner we can compute the amount of correctly and incorrectly tagged/rejected terms. Using the same metrics than in the first evaluation step, the evaluation results are presented in **Table 16** and **Table 17**.

TABLE 16. ANNOTATION EVALUATION METRICS (ARTICLE SET 1: GEOGRAPHICAL DATA)

Article	*Recall*	*Precision*	*F-Meas.*	*Fallout*	*Accuracy*	*Error*
Andorra	81.25%	76.47%	78.78%	18.18%	81.57%	18.42%
Barcelona	79.41%	70.12%	74.48%	22.77%	78.10%	21.89%
Tarragona	82.60%	73.07%	77.55%	22.58%	79.62%	20.37%
Reus	40%	50%	44.44%	28.57%	58.33%	41.66%
Palma	63.15%	80%	70.58%	11.53%	77.77%	22.22%
Average	**69.28**%	**69.93**%	**69.17**%	**20.72**%	**75.08**%	**24.91**%

TABLE 17. ANNOTATION EVALUATION METRICS (ARTICLE SET 2: CINEMA DATA)

Article	*Recall*	*Precision*	*F-Meas.*	*Fallout*	*Accuracy*	*Error*
The Matrix	76.74%	70.21%	73.33%	27.45%	74.46%	2553%
War Games	70%	58.33%	63.63%	29.41%	70.37%	29.62%
LOTR 3	70.45%	55.35%	62%	19.68%	77.77%	22.22%
I, Robot	40%	42.85%	41.37%	24.24%	64.58%	35.41%
300	50%	56.25%	52.94%	29.57%	61.6%	38.4%
Average	**61.43%**	**56.6%**	**58.65%**	**26.07%**	**69.75%**	**30.24%**

In this second stage, the results are quite consistent. For the first set, the F-Measure is above 70% (with a similar precision and recall) except for the case of the scarcest domain (Reus), for which the limited amount of English written-resources hampers the performance of the annotation selection procedure.

For the second set, results are a little lower, with an average F-Measure below 60%. We observed that there is a dependency between the annotation quality and the degree of generality and coverage of the employed ontology. For the first case, the size of the ontology (188 classes) allows a better definition of domain concepts and specializations, increasing the probability of a direct matching. In the second case (only 17 classes) entities can only be tagged with general concepts which results in less accurate results. In addition, the homogeneity of the article contents against the ontology coverage is also important. In the second test, for example, movie-related articles usually have contents related with the film's plot (including fictitious and ambiguous film characters) or off-topic information about actor's biography, which may introduce noise and hampers the annotation quality.

In this case, results are not directly comparable to C-Pankow, as they use different –more specific- ontologies to perform the annotation procedure and the expert's evaluation criteria may have an important influence. However, our results present a higher precision and recall (above 20% for the location articles) thanks to the use of WordNet similarity measures and web-based statistical analyses to assess the suitability of the selected concept candidates. All these data configure a knowledge background, which aids the annotation process and allows more accurate results. On the other hand, C-Pankow performs a lower amount of web queries that may result in a lower annotation runtime.

5.1.3 Learning parameters

In this section, we will evaluate the influence of thresholds' values used on our algorithm on each step.

Actually our algorithm uses 2 thresholds. The first one is used in the first algorithm step to set the minimal value to consider a possible instance as a Named

Entity. The second threshold is used in the third algorithm step; we use it to filter the class candidates using the WordNet similarity measures, and minimize the set of candidates to which the PMI-IR should be calculated.

In order to evaluate the quality of the thresholds used, we set the annotation content (concretely we used the Wikipedia article about "*Palma de Mallorca*", as it is middle-sized) and we modified the thresholds' values, in order to examine the quality of the results obtained. We increased the thresholds with an increase of 0.2, being 0.1 the initial value.

5.1.3.1 Named Entities detection threshold

For this first threshold, we wanted a high precision rather than a high recall. As the system should be automatic, we considered more important the correct selection of Named Entities than to have a large set of candidates, as the redundancy of the Web can aid this second point. Moreover, a more compact set of candidates result in a better learning performance as fewer web queries are required in the second stage.

In this manner, we decided that the precision of the system, in this step, should be near or high to the 80%; and, with this parameter fixed, used the threshold with higher recall, as we also considered very important. In **Table 18** we present the performance of the Named Entities detection procedure depending on the threshold used:

TABLE 18. NAMED ENTITIES DETECTION PERFORMANCE VARYING ITS THRESHOLD

Threshold	*Recall*	*Precision*	*F-Measure*	*Fallout*	*Accuracy*	*Error*
0.1	76.84%	64.6%	70.19%	58.82%	61.96%	38.03%
0.3	77.65%	64.6%	70.53%	58.82%	62.34%	37.65%
0.5	60%	66.27%	62.98%	42.64%	58.89%	41.1%
0.7	**37.89%**	**85.71%**	**52.55%**	**8.45%**	**60.84%**	**39.15%**
0.9	14.73%	100%	25.68%	0%	50.3%	49.69%

As it can be observed in **Figure 15**, the threshold which best fulfils the previous requirements in this step was **0.7**, although there are other values which fulfil our initial requirements, they are out of our threshold modification policy (see §5.3), or have a too low recall values.

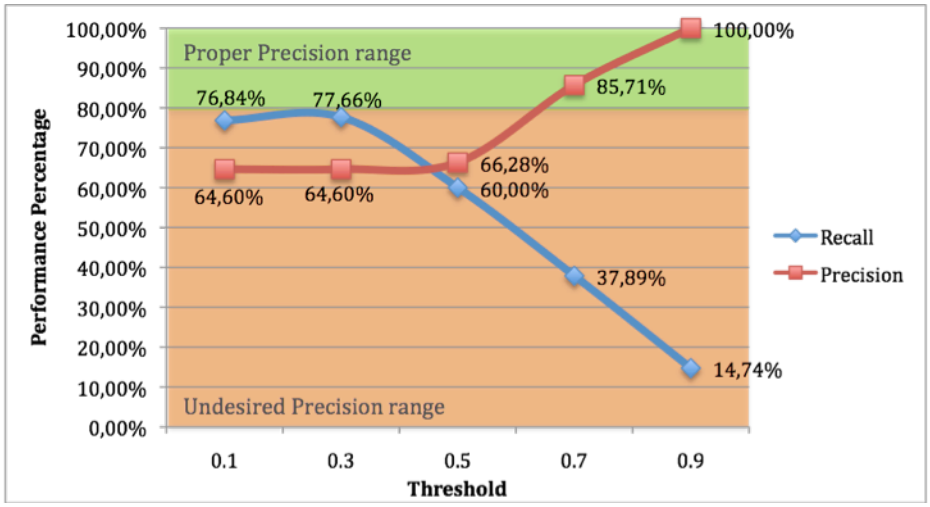


Figure 15. Recall-Precision relationship in Named Entity detection procedure

Other observations that can be extracted from this evaluation are:

- The recall is stable in a threshold range of values starting from 0.1 and finishing in 0.3 approximately. When the threshold is over these values, the recall decreases very fast.
- The higher is the obtained recall; the lower is the precision of this step. Moreover, the precision curve has the inverted behaviour if compared with the recall.
-

From these observations, we conclude that there is not any better value, as each value has its advantages and inconveniences, and it is the final user who should define what is worthwhile.

5.1.3.2 Class Candidates classification threshold

For this second threshold, we wanted to maximize the final system performance. However, the final results of this step depend on the ones obtained in the previous step (for this reason, the results obtained in those step should be precise), on the ontology used and on the PMI-IR value with the results obtained after applying this threshold and filtering the candidates. In this manner, the modification of this threshold is not crucial for the final system performance, but it has enough influence to be taken into account.

Having maximized the precision at the first stage with a restrictive threshold, in this step, we aim to present the most suitable set of annotations associated to the retrieved named entities. As it will be shown in the next evaluations, this second parameters shows its highest influence over the recall value, as it defines the amount of candidates that would be filtered using WordNet. In consequence, the threshold study would be focused on maximizing the recall.

TABLE 19. CLASS CLASSIFICATION PERFORMANCE VARYING ITS THRESHOLD

Threshold	*Recall*	*Precision*	*F-Measure*	*Fallout*	*Accuracy*	*Error*
0.1	50%	78.57%	61.11%	13.04%	68.88%	31.11%
0.3	**63.15%**	**80%**	**70.58%**	**11.53%**	**77.77%**	**22.22%**
0.5	39.13%	69.23%	50%	18.18%	60%	40%
0.7	37.5%	69.23%	48.64%	19.04%	57.77%	42.22%
0.9	36%	69.23%	47.36%	20%	55.55%	44.44%

As it can be stated, the best threshold value is **0.3**. This result is given by different reasons (see **Figure 16**). The first one is the range of the values used by the WordNet similarity measures, which usually are lower than 0.4, making the higher thresholds values to this limit, very restrictive, and in this way, they do not allow the enough candidates to be evaluated. On the other hand, using a very low value reduces the recall too, mainly because in this case we permit to too much candidates be evaluated, introducing noise in the PMI-IR calculation.

Anyway, as a conclusion, we can state that the effects of this threshold are not completely decisive for the final annotation results, but it has a remarkable influence if it is not properly set.

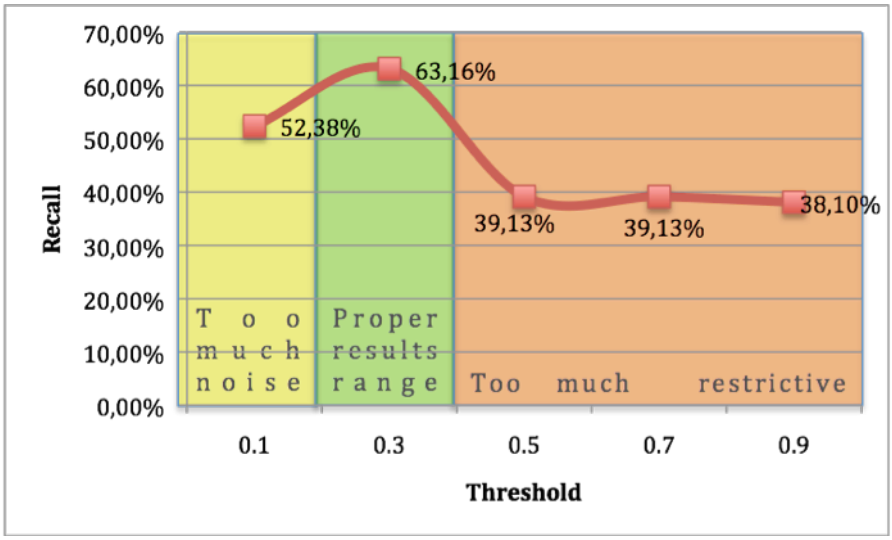


Figure 16. Threshold influence over the Recall in Class Classification

5.2 Qualitative Evaluation

In this section we will do a qualitative evaluation to examine the results' structure and the nature of the typical semantic mistakes in order to derive interesting conclusions. This qualitative evaluation, though subjective, will be useful to determine the usefulness degree of the obtained results. We have analysed the quality of the results at three levels: the Named Entities detection, the class candidates extraction and the Named Entity annotation.

5.2.1 Named Entities detection

In the following tables we present an extract of some of the candidates retrieved by the algorithm in this step:

TABLE 20. NAMED ENTITIES EXTRACTED FROM BARCELONA ARTICLE

Named Entities
Catalan IPA, Spanish IPA, rivers Llobregat, Collserola ridge, Antoni Gaudi, **Barcelona houses**, Generalitat, Barchinona, Carthaginian Hamilcar Barca, city Barcino, century BC, Hamilcar Barca, Mons Taber , Faventia, Pia Barcino, Faventia Paterna Barcino, Placa del Rei, Museu d'Historia, Barri Gotic, Visigoths, Carolingian, Marca Hispanica, Castile**, ft Torre**, Ciutadella, Gramenet, Sant Adria, Llobregat, Esplugues, Montcada i Reixac, Fabra Observatory, Tibidabo hill, …. Teatre Lliure, Eiji Oue, Museums Barcelona, Fundacio Antoni Tapies, Egiptian Museum, architect Antoni Gaudi, Musica Catalana, Sant Pau, Palau Guell, Casa Mila, Casa Vicens, Colonia Guell, Mundo Deportivo, Spanish, Catalunya Radio, Catalonia's, Championships, **FC Barcelona**, **UEFA Champions**, handball ASOBAL league, FC Barcelona-Cifec, CE Europa, UE Sant Andreu, **Estadi Olimpic Lluis Companys**, Tenis Barcelona, Montmelo, Martinair, Girona-Costa Brava, Vell area, TEU's, Old Port, Tramvia Blau, Transports Metropolitans, L9, Trambaix, Trambesos, Transports Ciutat Comtal, Montjuic hill, Vallvidrera, Torre Jaume, Torre Sant Sebastia, AVE high-speed rail system, Ferrocarrils, Estacio del Nord, Olympic Games, Taxi Institute, yellow livery, Bicing service, Ronda del Litoral, Corts Catalanes, Montpellier, Sao Paulo, Montevideo, Tel Aviv, Sarajevo, Bosnia-Herzegovina, Dubai, Arab Emirates, Serbia, Casa Batllo, Palau Nacional, MNAC, Torre Agbar, **Gaudi's**, Enciclopedia Catalana, **1971**, Informa Australia, Enciclopedia Catalana S, Wikipedia's sister projects:, d'en Grassot, Batllori, Baix Centre, Vallbona, Helsinki, Melbourne, Search Toolbox, Copyrights, Wikimedia Foundation.

TABLE 21. NAMED ENTITIES EXTRACTED FROM 300 (FILM) ARTICLE

Named Entities
Zack Snyder Produced, Gianni Nunnari, Headey, Warner Bros, Thermopylae, Zack Snyder, Leonidas, Spartans fight, Rodrigo Santoro, soldier Dilios, HD DVD, sacred Carneian festival, Xerxes messenger, **their covert support**, Persians' advance, Ephialties, Ephialtes, Leonidas refuses, heavy casualties, approaches Leonidas, his loyalty, Gorgo attempts, Hot Gates, Leonidas orders, Theron, his betrayal, Plataea, Leonidas' wife, Dilios, soldier Dominic, Stelios, Pleavin, Astinos, McHattie, Neitzel, Uber Immortal, Nunnari, character Dilios, Employing, Snyder's direction, **his performance**, Hybride Technologies, Azam Ali, Yared, Elliot Goldenthal, Internet, **Inch Nails**, **Ice Age**, Lost World Jurassic, R-rated film ever, Matrix Reloaded, **Canton said**, North America, Kirk Honeycutt, Cronan, MTV Movie Awards, Gerard Butler, Golden Icon Awards, Travolta, Leonidas' description, University, Herodotus, Simonides, Touraj Daryaee, Plutarch, Greco-Persian conflict, Film Festival, Orlando Sentinel, Sontag, Ahmadinejad, Steven Rea, Azadeh Moaveni, Norouz, Achaemenid, Iranian, Uzbekistan, Jacobins, American Revolutionary War, Universal Studios, National Lampoon's , Awesomest Maximus, D-Yikes!, Retrieved, Stax Report Script, Wloszczyna, WB, Rejects , UFC's, box-office triumph, Berlinale Update, Rotten Tomatoes, Movie Awards, Another View, Peneaud, **Iranian**, UNESCO, Explore Internet Memes, Park D-Yikes, Internet Movie Database, Superhero Hype!, English-language films, **Search Toolbox**, **Copyrights**, Wikimedia Foundation.

From these tables we have highlighted some doubtful Named Entities to comment their nature.

From the location articles, like the *Barcelona* one, the algorithm designed has been able to extract well-formed candidates but it has also extracted candidates non correctly formed or it has evaded proper candidates:

- Well formed candidates like *FC Barcelona* or *Estadi Olimpic Lluis Companys*, which in fact are formed by Proper Nouns, the Regular Expression grammar has tagged them as a set of Proper Nouns and the next taggers have not re-tagged them as they are well tagged. Then they have been passed over the next processes, which have confirmed their good formation.
- Others like *1971, ft Torre* or *Gaudi's* are non-relevant Named Entities or bad formed Named Entities, as they are numbers or they are formed by pieces of proper Named Entities with non-relevant pieces (*ft Torre* should be *Torre de Collserola*), or they are derived forms of proper Named Entities which are considered as Named Entities, like *Gaudi's*. However, even they are not completely correct, they have pieces of proper Named Entities, or they can be considered as Named Entities, like *1971* as it is a year number, but at the same time it is the name of Hotels or Clubs.
- Other cases are considered as proper Named Entities even they are not, like *Barcelona houses.* This happens due to the fact that the Named Entity has a Proper Noun into its structure, and we have given more priority to the fact of having a Proper Noun into the structure to consider a candidate as Named Entity. On the other hand, the fact of having non-English words in the structure makes the next taggers to the Regular Expression keep its initial tagging.

.
.

From the films articles, like the *300* one, we have also remarked some candidates which are interesting to comment:

- The algorithm extracts also well-formed candidates like *Ice Age* (the film), and parts of well-formed Named Entities like *Inch Nails* (it should be *Nine Inch Nails*), that even if they are not well formed they are well detected.
- As in the case of location articles, some candidates like *Canton said*, are also considered as well formed, even though they have parts that are not well tagged, because we have considered in the further steps of the algorithm that if it has a Proper Noun it is a Named Entity.
- Finally, for a set of reasons, like words which are not in the training corpus of the taggers and are classified as Proper Nouns or Nouns even they are not, and the further steps like WordNet filtering do not filter them, candidates like *Iranian*, *Search Toolbox, Copyright, their covert support* or *his performance* are considered as Named Entities.
-

From these results, we can conclude that the algorithm extracts a big amount of good candidates, but the fact of depending on a set of regular expressions or a previous learning for tagging the text, introduces some noise in the classification. On the other hand, even if the classification grammar is good, it can be refined to avoid some mistakes like the described.

The WordNet filtering step is good, but it has a limited coverage and depending on the form of the Named Entity it cannot give us good results as we pass to it the complete detected Named Entity.

Finally, the algorithm also discards good candidates like *Cuba* or *Havana*. The reasons why they are not detected could be focused on the Unigram and the Bigram taggers would have they tagged as Nouns in their training corpus (and then the grammar will discard them), as the further steps would classify them properly.

5.2.2 Class candidates extraction

In this section, we have extracted some class candidates that the system has been retrieved for some of the Named Entities detected. They are summarized in the following tables:

TABLE 22. CLASS CANDIDATES SUMMARY RETRIEVED FOR NAMED ENTITIES OR BARCELONA ARTICLE

Named Entity	***Class candidates***
Generalitat	Building, government building, information, public entities, local governments…
Placa del Rei	Captivating place, time capsule, magical square
Barri Gotic	gothic buildings, narrow street, great place, nice place to hang around places, medieval areas, old district…
Diagonal Mar	Area, stylish hotel, special hotel, superb hotel, 23-story hotel, new tourist place, places, sites, tall buildings…
Baix Centre	-

TABLE 23. CLASS CANDIDATES SUMMARY RETRIEVED FOR NAMED ENTITIES ON 300 (FILM) ARTICLE

Named Entity	*Class candidates*
Movie Awards	programs
Rotten Tomatoes	Movies, Zip Code: Film, products …
Warner Bros	film companies, company, well-known media company, big US film companies, major entertainment companies, classic films, additional companies…
Gianni Nunnari	-
Zack Snyder	director, visionary director, Jewish director, edgy director, spiffy director, directors, classic movie, his own films…

In this case, as it can be perceived the extraction of terms obtains, in general, good candidates. However, it has a high dependence on the distribution of the information in the Web, and it also depends on the capacity of the Web Search Engines to retrieve this information. In this manner, good annotable candidates, like *Gianni Nunnari* or *Baix Centre*, do not obtain any candidate, although they can be annotated in further steps, but the Web has no information matching any of the patterns constructed. On the other hand, well known Named Entities like *Warner Bros* or *Generalitat*, obtain relevant candidates, which match with them properly.

It is also remarkable that some candidates, as for example *Zack Snyder*, obtain good candidates, but other noisy candidates like *classic movie* which do not match very correctly with the meaning of the Named Entity.

As stated, the knowledge extracted on this step depends on the patterns constructed, which can be considered quite good, as they extract good candidates; but it also depends on the information contained on the Web, which size is enormous for some topics, but for other topics it has not any information. From this we can conclude that the quality of the candidates depends in a higher level on the contents of the Web than on the patterns constructed.

5.2.3 Named Entities annotation

In this section, we have extracted some class candidates that the system has extracted for some of the Named Entities detected. They are summarized in the **Table 24** and the **Table 25**.

In these tables, it is possible to detect some facts. In first place, depending on the distribution of the Web's candidate information it is possible to obtain more ontology candidates. In this manner, for some candidates it is possible to obtain a large set of candidates for them.

However, the Web's distribution of Information introduces a problem as for those Named Entities with large sets of ontology candidates, it chooses the more general candidate, making the annotation good for the experts, but improvable as there are more specific candidates, which fit better with the Named Entity. An example of this issue is *Barri Gotic,* which has candidates like *District* or *Place*, but the selected one is the more general *Area.*

TABLE 24. ANNOTATION DONE FOR SOME NAMED ENTITIES EXTRACTED ON BARCELONA ARTICLE

Named Entity	*Class candidates*	*Ontology candidates*	*Final Annotation*
Generalitat	Building, government building, information, public entities, local governments…	Office building Administrative building Information center Spatial entity Geopolitical entity Government structure	Geopolitical entity
Placa del Rei	Captivating place, time capsule, magical square	Private place Place Business place Public place	Private place
Barri Gotic	gothic buildings, narrow street, great place, nice place to hang around places, medieval areas, old district…	Office building Administrative building Street Private place Place Business place Public place District Ski area Area Land area Rest area Historical site City City hall	Area
Diagonal Mar	Area, stylish hotel, special hotel, superb hotel, 23-story hotel, new tourist place, places, sites, tall buildings…	Ski area Area Land area Rest area Hotel Private place Place Business place Public place District Historical site Office building Administrative building Bus station	Area
Baix Centre	-	-	-

TABLE 25. ANNOTATION DONE FOR SOME NAMED ENTITIES EXTRACTED ON 300 (FILM) ARTICLE

Named Entity	*Class candidates*	*Ontology candidates*	*Final Annotation*
Movie Awards	programs	TV Program	TV Program
Rotten Tomatoes	Movies, Zip Code: Film, products …	Movie Company Film Product	Movie Company
Warner Bros	film companies, company, well-known media company, big US film companies, major entertainment companies, classic films, additional companies…	TV Program Product Film Movie Company Record Company Company	Movie Company
Gianni Nunnari	-	-	-
Zack Snyder	director, visionary director, Jewish director, edgy director, spiffy director, directors, classic movie, his own films…	Director Movie Company Film	Movie Company

Finally, there are candidates, which are well annotated (i.e.- *Generalitat* as *Geopolitical Entity*, or *Rotten Tomatoes* as *Movie Company*)

Chapter 6

Conclusions and Future Work

The manual annotation of web contents is a hard task, so the design of automatic solutions is fundamental to the success of the Semantic Web. Even though the quality of the automatic solutions, like the one proposed in this work, is still far away from the one obtained with manual annotations, their performance is promising. From the evaluation, we observed that we were able to detect around a 40% of the Named Entities that a human expert can detect, with and accuracy above a 70%. Ontology-based annotations provide also reliable good results with accuracy around 70-75%.

This is possible due to the definition of a constrained retrieval process based on the combination of well-known techniques in the Named Entity detection phase, giving us well-defined Named Entities, and extracting suitable class candidates. The suitability of those candidates is influenced, on the one hand, by the quality of the Named Entities extracted, and on the other hand by the use of text patterns as a powerful unsupervised technique to retrieve this knowledge.

Analyzing our algorithm, we can also conclude that it accomplishes many of the requirements we have described. It uses standard formats (like *HTML or Microformats*), and also uses ontologies defined in standard formats too (*OWL*). It is an automatic solution, and the consistency of the annotations is maintained, as they are stored into documents without modifying their internal structure. On the other hand, we do not support several document formats (like *PDF, Word, etc*), but our approach supports many of the commonly used formats (e.g.- *HTML, XHTML, XML, SGML, etc*). Finally, from the point of view of communication, our tool is compliant, as the different services in which it is structured have a public XML-RPC API, which can be accessed by any other software.

Based on these observations and on the done effort, the main contributions of this work are:

- A list of the requirements that an annotation solution for the Web should accomplish, and an updated state of the art of the different existing annotation techniques classifying them in function of their architecture (e.g.- *Frameworks, Complete Solutions, Plug-ins, etc*) and in function of their automatism grade (e.g.- *manual annotation tools, semi-automatic tools or complete automatic solutions*).
- A review of the use of the Web as a knowledge repository. We have assessed the current Web Search engines, their performance, their accessibility and the complementary tools they give. We have also analyzed the usage of these Engines

as a tool to extract taxonomical relationships and to quantify the quality of these relationships.

- As the massive usage of the Web has a high temporal cost, we have proposed a complementary technique, the use of electronic dictionaries, to reduce the utilization of the Web, and the overload this use produce, without compromising the quality of the results obtained, and even improving them.
- The annotation of web contents requires a technological base on which rely on. We have reviewed the common technologies that can be used to annotate documents. We have analyzed their best features and their worst too. Finally we have compared them, in order to have a clear idea about where they fit better.
- We have contributed in a refined Named Entities detection procedure. The annotation of documents is mainly based on the results obtained of such kind of procedures, and based on previous works in this matter; we have introduced a high refined and automatic Named Entity detection procedure into an annotation system.
- As the computers are not able to relate different concepts, even though they can be the same, we have contributed with a Web-based class classification procedure, which relies on the high amount of information present in the Web and the heterogeneity this information has.
- The evaluation of this kind of systems is always a hard task, and many authors have proposed their evaluation methodology. However, we think these methodologies are always focused on one kind of documents (e.g.- *Location descriptions*), which is good to compare them, but do not give a clear idea of their performance. For this reason, we have prepared a methodology using different topics, this can generate different textual structures and contents, and thus gives us a better idea of the performance of our procedure.

Moreover, the designed algorithms and the results obtained have been published in [*Millan et al*., 2008].

As further work, it is a priority to study how to reduce the number of queries to web search engines, as they are the slowest part of the algorithm and introduce a dependency on external resources. During the development we have observed that some of the patterns (like the *"is a", "and other", "or other"* ones) retrieve more precise and useful candidates than others. This may lead to further research in composing a more concrete set, which results in a lower amount of web queries. Regarding this last point, it is also remarkable the use of dictionaries, like WordNet, as a good solution to reduce the total number of queries to web search engines.

It is also important to take into account semantic ambiguity problems (e.g. "*Barcelona*" could be "*a geographical place*" or "*a sports team*"). When using a unique ontology as input, the problem can be solved implicitly by the unambiguous definition of ontological classes. Nevertheless, in the context of the Semantic Web, annotators may have to deal with many ontologies, which probably will introduce different annotation possibilities. A solution to this issue would be to annotate the possible ambiguous Named Entities with each ontology, and based on the context (i.e. the rest of the annotated entities) choose the dominant annotation sense.

Finally, we think that the design of a standard evaluation, with a large set of documents to be annotated and the annotations done over these documents by experts, is another matter of future. It is not clear how to develop it, but inspiring ideas can be taken from works like the Brown Corpus used to tag the text. Nevertheless, this is a hard task, and cannot be done in a single work.

Bibliography

[*Agirre et al*., 2000] Agirre, E., Ansa, O., Hovy, E., and Martinez, D.: Enriching very large ontologies using the WWW. In Proceedings of the Workshop on Ontology Construction of the European Conference of AI (ECAI-00). Berlin, Germany, 2000.

[*Ahmad et al.*, 2003] Ahmad, K., Tariq, M., Vrusias, B. and Handy, C.: Corpus-based thesaurus construction for image retrieval in specialist domains. In Proceedings of the 25th European Conference on Advances in Information Retrieval (ECIR). 2003. 502-510.

[*Alfonseca and Manandhar,* 2002] Alfonseca, E., Manandhar, S.: Extending a lexical ontology by a combination of distributional semantics signatures. In Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management, pp. 1-7 (2002).

[*Baeza-Yates*, 2004] R. Baeza-Yates, Excavando la web, El profesional de la información 13 (1) (2004) 4–10.

[*Baumgartner et al.,* 2001] Baumgartner, R., Flesca, R., Gottlob, G.: Visual web information extraction with Lixto. In Proceedings of the International Conference on Very Large Data Bases (2001).

[*Berners-Lee et al.,* 2001] Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web – a new form of web content that is meaningful to computers will unleash a revolution of new possibilities, Scientific American 284 (5) 34–43 (2001).

[*Bisson et al.,* 2000] Bisson, G., Nedellec, C. and Cañamero, D.: Designing Clustering Methods for Ontology Building. The Mo'K Workbench. In S. Staab, A. Maedche, C. Nedellec, P. WiemerHasting (eds.), Proceedings of the Workshop on Ontology Learning, 14th European Conference on Artificial Intelligence, ECAI'00, Berlin, Germany. August 20-25, 2000. 13-19.

[*Brill et al*., 2001] Brill, E., Lin, J., Banko, M. and Dumais, S.: Data-intensive Question Answering. In Proceedings of the Tenth Text Retrieval Conference TREC-2001. 2001. 393-400.

[*Brill*, 2003] Brill, E.: Processing Natural Language without Natural Language Processing. In Proceedings of CICLing 2003, LNCS 2588. 2003. 360–369

[*Budanitsky and Hirst*, 2001] Budanitsky, A. and Hirst, G: Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. NAACL01. 2001.

[*Buitelaar and Ramaka*, 2005] P. Buitelaar, S. Ramaka, Unsupervised ontology based semantic tagging for knowledge markup, in Proceedings of the Workshop on Learning in Web Search at International Conference on Machine Learning, 2005.

[*Bunescu*, 2003] Bunescu, R.: Associative Anaphora Resolution: A Web-Based Approach. In Proceedings of the EACL-2003 Workshop on the Computational Treatment of Anaphora, Budapest, Hungary, April, 2003. 47-52.

[*Califf and Mooney,* 2004] Califf, M.E., Mooney, R.J.: Bottom-up relational learning of pattern matching rules for information extraction. Machine Learning Research 4(2) 177-210 (2004).

[*Calvo and Gelbukh*, 2003] Calvo, H. and Gelbukh., A.: Improving Prepositional Phrase Attachment Disambiguation Using the Web as Corpus. LNCS 2905. 2003. 604–610.

[*Caraballo*, 1999] Caraballo, S.A.: Automatic construction of a hypernym-labeled noun hierarchy from text. In Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics. 1999. 120-126.

[*Carr et al*., 2004] L. Carr, T. Miles-Board, A. Woukeu, G. Wills, W. Hall, The case for explicit knowledge in documents, in: Proceedings of the ACM Symposium on Document Engineering (DocEng '04), Oct 28-30, Milwaukee, Wisconsin, USA, 2004, pp. 90- 98.

[*Charniak et al.*, 1999] Charniak, E. and Berland, M.: Finding parts in very large corpora. In Proceedings of the 37th Annual Meeting of the ACL. 1999. 57-64.

[*Church et al*, 1991] Church, K.W., Gale, W., Hanks, P. and Hindle, D.: Using Statistics in Lexical Analysis. In: Uri Zernik (ed.), Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon. New Jersey: Lawrence Erlbaum. 1991. 115-164.

[*Cilibrasi and Vitanyi*, 2004] Cilibrasi, R. and Vitanyi, P.M.B.: Automatic meaning discovery using Google. Available at: http://xxx.lanl.gov/abs/cs.CL/0412098. 2004.

[*Cilibrasi and Vitanyi*, 2006] Cilibrasi, R. and Vitanyi, P.M.B.: The Google Similarity Distance. IEEE Transaction on Knowledge and Data Engineering. 19(3). 2006. 370-383.

[*Cimiano et al*., 2004] Cimiano, P., Pick, A., Schmidt, L. and Staab, S.: Learning Taxonomic Relations from Heterogeneous Sources of Evidence. In Procedings of the ECAI 2004 Ontology Learning Workshop. 2004.

[*Cimiano et al*., 2005] P. Cimiano, G. Ladwig, S. Staab, Gimme' The Context: Context-driven Automatic Semantic Annotation with C-PANKOW, in the Proceedings of the International World Wide Web Conference (WWW2005), 2005.

[*Ciravegna et al*., 2002] F. Ciravegna, A. Dingli, D. Petrelli, Y. Wilks, User-system cooperation in document annotation based on information, In Proceedings of the 13th International Conference on Knowledge Engineering and KM (EKAW02), 2002.

[*Ciravegna et al*., 2003] Ciravegna, F., Dingli, A., Guthrie, D. and Wilks, Y: Integrating Information to Bootstrap Information Extraction from Web Sites. In Proceedings of the IJCAI Workshop on Information Integration on the Web. 2003. 9-14.

[*Collier et al*., 2004] N. Collier, A. Kawazoe, A.A. Kitamoto, T. Wattarujeekrit, T.Y. Mizuta, A. Mullen, Integrating deep and shallow semantic structures in open ontology forge, in proceedings of the Special Interest Group on Semantic Web and Ontology, JSAI (Japanese Society for Artificial Intelligence), vol. SIG-SWO-A402-05, 2004.

[*Cutting et al.*, 1992] Cutting, D., Karger, D., Pedersen, J. and Tukey, J.W.: Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. In Proceedings of the 15th Annual International ACM/SIGIR Conference, Copenhagen. 1992. 318-329.

[*Daudé et al*., 2003] Daudé J., Padró L. and Rigau G.: Validation and Tuning of WordNet Mapping Techniques. In Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP'03). Borovets, Bulgaria, 2003.

[*de Lima*, 1999] de Lima, E.F. and Pedersen, J.O.: Phrase Recognition and Expansion for Short, Precision biased Queries based on a Query Log. In Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 1999. 145-152.

[*Dujmovic and Bai*, 2006] Dujmovic, J. and Bai, H.: Evaluation and Comparison of Search Engines Using the LSP Method. ComSIS 3(2). 2006. 711-722.

[*Dzbor et al*., 2004] M. Dzbor, E. Motta, J. Domingue, Opening up magpie via semantic services, in Proceedings of the 3rd International Semantic Web Conference, November 2004, Hiroshima, Japan, 2004.

[*Economist*, 2005] Economist: Corpus collosal: How well does the world wide web represent human language? The Economist, January 20, 2005. Available at: http://www.economist.com/science/displayStory.cfm?story id=3576374. 2005.

[*Etzioni et al*., 2004] Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A., Shaked, T., Soderland, S. and Weld, D.S.: WebScale Information Extraction in KnowItAll. In Proceedings of WWW2004, New York, USA. 2004.

[*Etzioni et al*., 2005] O. Etzioni, M.J. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D.S. Weld, A. Yates, Unsupervised named-entity extraction from the Web: an eperimental study, Artif. Intelligence 165 (1), 2005, 91-134.

[*Evans*, 2003] Evans, R.: A framework for named entity recognition in the open domain. In Proceedings of the Recent Advances in Natural Language Processing, pp. 137-144 (2003)

[*Farreres et al*., 2004] Farreres, J., Gibert, K. and Rodríguez, H.: Towards Binding Spanish Senses to WordNet Senses through Taxonomy Alignment. In Proceedings of GWC 2004. Masaryk University. 2004. 259-264.

[*Fensel et al., 2002*] D. Fensel, C. Bussler, Y. Ding, V. Kartseva, M. Klein, M. Korotkiy, B. Omelayenko, R. Siebes, Semantic web application areas, in: 7th International Workshop on Application of Natural Language to Information Systems, Stockholm, Sweden, 2002

[*Ferreira da Silva and Lopes,* 1999] J. Ferreira da Silva, and G.P. Lopes. 1999. A local maxima method and a fair dispersion normalization for extracting multi-word units from corpora. In Proceedings of Sixth Meeting on Mathematics of Language. 369-381.

[*Fleischman and Hovy*, 2002] Fleischman, M., Hovy, E.: Fine grained classification of named entities. In Proceedings of the Conference on Computational Linguistics (2002).

[*Gómez-Pérez et al*., 2004] Gómez-Pérez, A., Fernández-López, M. and Corcho, O.: Ontological Engineering, 2nd printing. Springer Verlag.. ISBN: 1-85233-551-3. 2004

[*Grefenstette*, 1992] Grefenstette, G.: Finding Semantic Similarity in Raw Text: The Deese Antonyms. In: R. Goldman, P. Norvig, E. Charniak and B. Gale (eds.), Working Notes of the AAAI Fall Symposium on Probabilistic Approaches to Natural Language. AAAI Press. 1992. 61-65.

[*Greffenstette*, 1999] Grefenstette, G.: The World Wide Web as a resource for example-based Machine Translation Tasks. In Proceedings of Aslib Conference on Translating and the Computer. London. 1999.

[*Gruber*, 1993] Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing. In Guarino, N., Poli, R. (eds) International Workshop on Formal Ontology in Conceptual Analysis and Knowledge Representation. Padova, Italy. 1993. 907-928.

[*Hahn and Schnattinger*, 1998] Hahn, U. and Schnattinger, K.: Towards text knowledge engineering. In AAAI'98/IAAI'98 Proceedings of the 15th National Conference on Artificial Intelligence and the 10th Conference on Innovative Applications of Artificial Intelligence (1998).

[*Handschuh et al*., 2003] S. Handschuh, S. Staab, R. Studer, Leveraging metadata creation for the semantic web with CREAM, KI '2003 - Advances in artificial intelligence, in: Proceedings of the Annual German Conference on AI, September 2003, 2003.

[*Hearst*, 1992] Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In Proceedings of the 14th International Conference on Computational Linguistics (COLING-92). Nantes, France, 1992. 539–545.

[*Hearst*, 1996] Hearst, M.A.: Improving Full-Text Precision on Short Queries using Simple Constraints. In Proceedings of the Symposium on Document Analysis and Information Retrieval. Las Vegas, NV. 1996.

[*Hirst and St-Onge*, 1998] Hirst, G. and St-Onge D.: Lexical chains as representations of context for the detection and correction of malapropisms. In Fellbaum C, editor. WordNet: An electronic lexical database. Cambridge, MA: MIT Press. 1998. 305–321.

[*Hogue and Karger*, 2005] A. Hogue, D. Karger, Thresher: automating the unwrapping of semantic content from the world wide web, in Proceedings of the 14th International World Wide Web Conference (WWW2005), May 10-14, Chiba, Japan, 2005, pp. 86-95.

[*Jans*, 2000] Jans, T.B.: The effect of query complexity on Web searching results. Information Research, 6(1). October 2000.

[*Jiang and Conrath*, 1997] Jiang, J. and Conrath, D.: Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In Proceedings of the 10th International Conference on Research on Computational Linguistics. Taiwan. 1997.

[*Kahan et al*., 2001] J. Kahan, M.J. Koivunen, E. Prud'Hommeaux, R. Swick, Annotea: an open RDF infrastructure for shared web annotations In Proceedings of the 10th World Wide Web Conference, Hong Kong, 2001.

[*KIM*] http://www.ontotext.com/kim/semanticannotation.html

[*Koivunen*, 2005] M.J. Koivunen, Annotea and Semantic Web supported col laboration Invited talk at Workshop on User Aspects of the Semantic Web (UsersWeb) at European Web Conference, Heraklion, Greece, 2005.

[*Lamparter et al.,* 2004] Lamparter, S., Ehrig, M. and Tempich, C.: Knowledge Extraction from Classification Schemas. In Proceedings of the CoopIS/DOA/ODBASE, pp, 618-636 (2004)

[*Lancaster*, 2004] The Lancaster stemming algorithm. Retrieved March 15, 2004, from http://www.comp.lancs.ac.uk/computing/research/stemming/

[*Lanfranchi et al*., 2005] V. Lanfranchi, F. Ciravegna, D. Petrelli, Semantic Web-based document: editing and browsing in AktiveDoc , in: Proceedings of the 2nd European Semantic Web Conference, May 29,June 1, 2005, Heraklion, Greece, 2005.

[*Leacock and Chodorow*, 1998] Leacock, C. and Chodorow, M.: Combining local context and WordNet similarity for word sense identification. In C. Fellbaum, editor, WordNet: An electronic lexical database. MIT Press.1998. 265–283.

[*Lee et al*., 1993] Lee, J.H., Kim, M.H. and Lee, Y.J.: Information Retrieval Based on Conceptual Distance in ISA Hierarchies. Journal of Documentation, 49. 1993. 188-207.

[*Lin*, 1998] Lin, D.: Automatic Retrieval and Clustering of Similar Words. In Proceedings of the 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics. Montreal. 1998. 768-773.

[*Maynard et al*., 2005] D. Maynard, M. Yankova, A. Kourakis, A. Kokossis, Ontology-based information extraction for market monitoring and technology watch, Proceedings of the Workshop on User Aspects of the Semantic Web (UserSWeb) a European Semantic Web Conference, 2005.

[*Maarek et al*., 2000] Maarek, Y.S., Fagin, R., Ben-Shaul I.Z. and Pelleg, D.: Ephemeral document clustering for web applications. Technical Report RJ 10186. IBM Research. 2000.

[*Meaning*, 2005] Meaning Project: Developing Multilinguial Web Scale Technologies. IST-2001-34460. http://nipadio.lsi.upc.edu/wei4/doc/mcr/meaning.html. 2005.

[*McDowell et al.,* 2003] McDowell, L., Etzioni, O., Gribble, S., Halevy, A., Levy, H., Pentney, W., Verma, D., Vlasseva, S.: Enticing ordinary people onto the Semantic Web via instant gratification. In Proceedings of the 2nd International Semantic Web Conference (2003).

[*Millan et al.,* 2008] Millan, M., Sánchez, D., Moreno, A.: UWAA: Unsupervised Web-based Automatic Annotation. To be appeard in Proceedings of the 4th European Starting AI Researcher Symposium (STAIRS) in European Conference on AI (ECAI), Patras, 2008.

[*Navigli and Velardi,* 2004] Navigli, R. and Velardi, P.: Learning Domain Ontologies from Document Warehouses and Dedicated Web Sites. In Computational Linguistics 30(2). June, 2004. 151-179.

[*Neches et al*., 1991] Neches, R., Fickes, R.E., Finin, T. Gruber, T.R., Senator, T. and Swartout W.R.: Enabling technology for knowledge sharing. AI Magazine 12(3). 1991. 36-56.

[*Niekrasz and Gruenstein*, 2006] J. Niekrasz, A. Gruenstein, NOMOS: A Semantic Web Software Framework for Annotation of Multimodal Corpora In Proceedings of the 5th International Conference on Language Resources and Evaluation, Genoa, 2006.

[*Pasca,* 2004] Pasca, M.: Acquisition of Categorized Named Entities for Web Search. In Proceedings of the thirteenth ACM International Conference on Information and Knowledge Management, pp. 137-145 (2004)

[*Pasca*, 2005] Pasca, M.: Finding Instance Names and Alternative Glosses on the Web: WordNet Reloaded. In Proceedings of CICLing 2005. LNCS 3406. 2005. 280-292.

[*Patwardhan and Pedersen*, 2006] Patwardhan S, Pedersen T.: Using WordNet-based context vectors to estimate the semantic relatedness of concepts. In: Proceedings of the EACL 2006 workshop, making sense of sense: Bringing computational linguistics and psycholinguistics together. Trento, Italy. 2006. 1–8.

[*Pedersen, et al*., 2004] Pedersen, T., Patwardhan, S. and Michelizzi, J.: WordNet::Similarity – Measuring the Relatedness of Concepts. http://search.cpan.org/dist/WordNet-Similarity. American Association for Artificial Intelligence. 2004.

[*Pedersen et al*., 2006] Pedersen, T., Serguei, Pakhomov, S., Patwardhan, S., Chute, C.: Measures of semantic similarity and relatedness in the biomedical domain. Journal of Biomedical Informatics. 2006.

[*Quint and Vatton*, 1997] V. Quint, I. Vatton, An introduction to Amaya, W3C NOTE 20-February-1997.

[*Resnik*, 1998] Resnik, P.: Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. Journal of Artificial Intelligence Research, 11. 1998. 95-130.

[*Resnik and Smith*, 2003] Resnik, P. and Smith, N.: The web as a parallel corpus. Computational Linguistics, 29(3). 2003. 349-380.

[*Richardson et al*., 1994] Richardson, R., Smeaton, A. and Murphy, J.: Using WordNet as a Knowledge Base for Measuring Semantic Similarity between Words. In Proceedings of the AICS Conference. Trinity College, Dublin. 1994.

[*Rijsbergen et al*., 1980] C.J. van Rijsbergen, S.E. Robertson and M.F. Porter, 1980. *New models in probabilistic information retrieval.* London: British Library. (British Library Research and Development Report, no. 5587).

[*Sánchez*, 2008] D. Sánchez, 2008. *Domain Ontology Learning From the Web*. VDM Verlag Dr. Mueller Akt.ges. & Co.KG. ISBN: 3836470691

[*Sanderson and Croft*, 1999] Sanderson, M. and Croft, B.: Deriving concept hierarchies from text. In Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Berkeley, USA. 1999. 206-213.

[*Schroeter et al*., 2003] R. Schroeter, J. Hunter, D. Kosovic, Vannotea, a col laborative video indexing, annotation and discussion system for broadband networks In proceedings of the K-CAP 2003 Workshop on "Knowledge Markup and Semantic Annotation", IOS Press, Amsterdam 2003.

[*Solorio et al*., 2004] Solorio, T., Pérez, M., Montes, M., Villaseñor, L. and López, A.: A Language Independent Method for Question Classification. In Proceedings of the 20th International Conference on Computational Linguistics (COLING-04). Geneva, Switzerland. 2004. 1374-1380.

[*SMORE*] SMORE: Semantic Markup, Ontology and RDF Editor http://www.mindswap.org/ adiktal/editor.shtml.

[*Studer et al*., 1998] Studer, R., Benjamins, V.R. and Fensel., D.: Knowledge Engineering: Principles and Methods. IEEE Transactions on Knowledge and Data Engineering 25(1-2). 1998. 161-197.

[*Surowiecky*, 2004] Surowiecki, J.: The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations. Doubleday Books, 2004.

[*Turney*, 2001] Turney, P.D.: Mining the Web for synonyms: PMI-IR versus LSA on TOEFL. In Proceedings of the Twelfth European Conference on Machine Learning. Freiburg, Germany 2001. 491-499.

[*Uren et al*., 2006] V. Uren, P. Cimiano, J. Iria, S. Handschuh, M. Vargas-Vera, E.Motta, F. Ciravegna, Semantic annotation for knowledge management: Requirements and a survey of the state of the art Web Semantics: Science, Services and Agents on the WWW 4 (2006), 14-28.

[*Uschold and Grunninger*, 1996] Uschold, M. and Grunninger, M.: Ontologies. Principles, Methods and Applications. Knowledge Engineering Review 11(2), 1996. 93-155.

[*Wikipedia:Annotation*] http://en.wikipedia.org/wiki/Annotation

[*William*, 2002] William, L.: Measuring Conceptual Distance Using WordNet: The Design of a Metric for Measuring Semantic Similarity. R. Hayes, W. Lewis, E. Obryan, and T. Zamuner (Eds.), The University of Arizona Working Papers in Linguistics. Tucson: University of Arizona. 2002.

[*Wu and Palmer*, 1994] Wu Z. and Palmer M.: Verb semantics and lexical selection. In Proceedings of the 32nd annual meeting of the association for computational linguistics. Las Cruces. 1994. 133–8.

[*Yang*, 1999] Yang, Y: An evaluation of statistical approaches to text categorization. Journal of Information Retrieval, 1999, 1 (1-2), 67-88.

[*Yeol and Hoffman*, 2003] Yeol Yoo, S. and Hoffmann, A.: A New Approach for Concept-Based Web Search. In Proceedings of the Australian Conference on Artificial Intelligence. LNAI 2903. 2003. 65–76.

[*Zamir and Etzioni*, 1999] Zamir, O. and Etzioni, O.: Grouper: A dynamic clustering interface to web search results. Computer Networks 31. 1999. 1361–1374.

[*Zhang and Dong*, 2004] Zhang, D. and Dong, Y.: Semantic, Hierarchical, Online Clustering of Web Search Results. In Proceedings of the 6th Asia Pacific Web Conference (APWEB), Hangzhou, China. 2004.

Annex I. List of Articles

The results of this work have been presented in the following articles:

- **Unsupervised Web-based Automatic Annotation**

  Abstract: The success of the Semantic Web depends both on the definition of ontologies used to represent the knowledge as on the annotations performed of the web contents. As manual approaches have low scalability, there is a need of tools capable to generate all this knowledge in an automatic and reliable way. In this paper is presented a complete algorithm to annotate web contents in an automatic and unsupervised manner. It is structured in a three-stepped procedure, based on the usage of several concept similarity measures and linguistic patterns. It is able to detect the entities to annotate, the candidate classes of these entities and, finally, associate them with the classes of an ontology. Some prospective results are presented.

  Conference: The *STAIRS-08* is the fourth European Starting AI Researcher Symposium, an international meeting intended to AI researchers, from all countries, at the beginning of their career: PhD students or people holding a PhD for less than one year. STAIRS'08 will be held jointly with ECAI-08 in Patras, Greece, on July 21st to 25th. STAIRS offers doctoral students and young post-doctoral AI fellows:
  - a first experience on submitting and presenting a paper in an international forum with a broad scope and a thorough selection process
  - an opportunity to gather knowledge and exchange ideas related to their research problems and approaches together with information on European research careers and mobility