

UNIVERSITAT ROVIRA I VIRGILI
DEPARTAMENT D'ENGINYERIA INFORMÀTICA I MATEMÀTIQUES
MÀSTER EN ENGINYERIA INFORMÀTICA I DE LA SEGURETAT

MASTER THESIS

Axiom discovery from the Web

Luis Miguel Del Vasto Terrientes

Directors: Dr. Antonio Moreno Ribas (URV)
Dr. David Sánchez Ruenes (URV)

Tarragona, 2010

Once we accept our limits, we go beyond them

Albert Einstein, 1879 - 1955

Contents

1. INTRODUCTION	1
1.1 INTRODUCTION TO ONTOLOGIES	3
1.2 OWL.....	5
1.2.1 <i>Class expression axioms</i>	6
1.2.2 <i>Object property axioms</i>	7
1.2.3 <i>Data property axioms</i>	9
1.2.4 <i>Assertions</i>	9
1.3 OBJECTIVES AND CONTRIBUTIONS	10
1.4 DOCUMENT STRUCTURE.....	12
2. STATE OF THE ART ON AUTOMATIC DISCOVERY OF AXIOMS.....	13
2.1 MANUAL AXIOMATIZATION TOOLS	13
2.1.1 <i>Protégé</i>	13
2.1.2 <i>OntoEdit</i>	15
2.1.3 <i>KAON</i>	16
2.2 AUTOMATIC/SEMI-AUTOMATIC AXIOMATIZATION TOOLS	17
2.2.1 <i>LExO</i>	17
2.2.2 <i>Disjointness axiom learning (LExO based)</i>	19
2.2.3 <i>RELExO</i>	19
2.2.4 <i>HASTI</i>	20
2.2.5 <i>AuCONTRAIRE</i>	22
2.3 CONCLUSIONS.....	24
3. LEARNING TECHNIQUES	27
3.1 AXIOMS COVERED IN THIS WORK	28
3.2 THE WEB AS CORPUS	28
3.3 WEB SEARCH ENGINES	30
3.3.1 <i>Search engine features</i>	30
3.3.2 <i>Search engine retrieval comparison</i>	31
3.4 LEXICO-SYNTACTIC PATTERNS.....	34
3.4.1 <i>Lexico-syntactic Patterns for Taxonomic relations</i>	34
3.4.2 <i>Lexico-syntactic Patterns for Non-Taxonomic relations</i>	36
3.5 NATURAL LANGUAGE ANALYSIS	37
3.5.1 <i>Stemming analysis</i>	38
3.5.2 <i>Detecting sentences, tokens and part-of-speech</i>	39
3.6 WEB-SCALE STATISTICS	41
3.7 CONCLUSIONS.....	43
4. OBJECT PROPERTY AXIOMS LEARNING METHODOLOGY	45
4.1 ALGORITHM DESCRIPTION	45

4.2 PATTERN CONSTRUCTION & WEB SEARCH QUERIES	47
4.2.1 <i>Patterns</i>	47
4.2.1.1 RELATION (SUBJECT, ?)	47
4.2.1.2 RELATION (SUBJECT, OBJECT)	49
4.2.1.3 ? (SUBJECT, OBJECT)	50
4.2.2 <i>Web Search Engine Retrieval</i>	51
4.3 NATURAL LANGUAGE ANALYSIS AND CANDIDATE EXTRACTION	52
4.3.1 <i>Natural Language Processing Analysis</i>	52
4.3.2 <i>Extraction Filters</i>	54
4.4 WEB SCALE STATISTICS MEASURES	56
4.5 CONCLUSION	59
5. EMPIRICAL EVALUATION	61
5.1 DISCOUNTED CUMULATIVE GAIN.....	61
5.2 RESULTS EVALUATION.....	64
5.2.1 <i>Symmetric / Assymmetric Object Property Evaluation</i>	65
5.2.2 <i>Reflexive / Irreflexive Object Property Evaluation</i>	65
5.2.3 <i>Functional Object Property Evaluation</i>	66
5.2.4 <i>Inverse Object Property Evaluation</i>	68
5.2.5 <i>Inverse Functional Object Property Evaluation</i>	70
5.2.6 <i>Transitive Object Property Evaluation</i>	72
5.3 CONCLUSIONS.....	74
6. CONCLUSIONS AND FUTURE WORK	77
6.1 SUMMARY & CONCLUSIONS.....	77
6.2 FUTURE WORK	78
7. REFERENCES	81
8. ANNEX I. ARTICLE	85

Acknowledgments

I have been supported by a grant of Fundación Carolina. This work has been supported by the Universitat Rovira i Virgili (2009AIRE-04), the Spanish Ministry of Science and Innovation (DAMASK project, Data mining algorithms with semantic knowledge, TIN2009-11005) and the Spanish Government (PlanE, Spanish Economy and Employment Stimulation Plan).

List of Figures

Figure 1. Classic Expert System Architecture	2
Figure 2. Concepts relation example	4
Figure 3. Property example (individual to individual)	5
Figure 4. Property example (individual to data value)	5
Figure 5. Snapshot of Protégé	14
Figure 6. Snapshot of OntoEdit	15
Figure 7. Snapshot of KAON tool	16
Figure 8. Ontology evolution process with LExO	18
Figure 10. RELExO	20
Figure 10. HASTI Architecture	22
Figure 11. AuCONTRAIRE Architecture	23
Figure 12. Sentence Analysis	40
Figure 13. Default fields obtained from Web search APIs	51

List of Tables

Table 1. Class expression axioms notation in OWL 2	6
Table 2. Object property axioms notation in OWL 2	7
Table 3. Data property axioms notation in OWL 2	9
Table 4. Assertion axioms notation in OWL 2	9
Table 5. Automatic and Semi-Automatic tools for axiom learning	24
Table 6. Object property axioms considered in this work and their OWL notation	28
Table 7. Query features analysis of Web Search Engines	31
Table 8. Number of results for specific queries	32
Table 9. Summary of the main characteristics of each Web Search Engine	33
Table 10. Examples of Hearts linguistic patterns	35
Table 11. part-of patterns presented by Berland and Charniak	36
Table 12. Natural language categories and entities referred to ontology concepts	37
Table 13. Results with Porter stemming algorithm	39
Table 14. Queries constructed for each axiom studied according to the example Borders (Spain, France)	45
Table 15. Conditions to validate the patterns defined	53
Table 16. Ranks defined by co-occurrence measures	54
Table 17. Test of Symmetry for several Object Properties	62
Table 18. Test of Reflexivity for several Object Properties	63
Table 19. Test of Functionality for several Object Properties	65
Table 20. Functional candidates extracted for “Microsoft competes with Google”	65
Table 21. Functional candidates found for “Brazil borders Bolivia”	66
Table 22. Functional candidates found for “Michael Jackson was born in Gary”	67
Table 23. Functional candidates found for “Raloxifene reduces breast cancer”	67
Table 24. Functional candidates found for “Chichen Itza is known as El Castillo”	67
Table 25. Discovery of Inverse relations for several Object Properties	68
Table 26. Inverse candidates found for “Global warming is caused by pollution”	68
Table 27. Inverse candidates found for “Charles Darwin proposed evolution theory”	69
Table 28. Test of the Inverse Functionality of some Object Properties	70
Table 29. Inverse Functional candidates found for “Pollution causes global warming”	71
Table 30. Inverse Functional candidates found for “Asthma is linked to stress”	71
Table 31. Inverse Functional candidates found for “Neisseria Meningitides causes meningococcal disease”	72

Tables 32a and 32b. Two steps in the test of Transitivity for some Object Properties	72
Table 33. Transitive candidates found for “Global warming is caused by pollution” in the first step	73
Table 34. Transitive candidates found for “Berlin is located in Germany” in the first step	73
Table 35. Transitive candidates found for “Microsoft competes with Apple” in the first step	74

Section 1

Introduction

Since John McCarthy coined the term Artificial Intelligence (AI) in 1956 as “the science and engineering of making intelligent machines”, much research in this discipline has in fact been devoted to building systems incorporating knowledge about a certain domain in order to reason on the basis of this knowledge and solve problems which were not encountered before. Regarding this research, Knowledge based systems (KBS) have been applied to many problems which require some sort of intelligence [2]. Knowledge-based Systems are artificial intelligent tools working in a narrow domain to provide intelligent decisions with justification.

Many KBS such as case-based reasoning systems, expert systems and networks attempt to provide a solution to a problem and many methods have been applied to each system in order to infer solutions. This solution is essentially the same as that concluded by a person knowledgeable about the domain of the problem when confronted with the same problem. All those systems have something in common, the creation of a symbolic knowledge base about a concrete domain. The need of a knowledge base opened a field known as *Knowledge Representation*, which is concerned in how to use a symbol system to represent a given domain. Since knowledge is used to achieve intelligent behaviour, the fundamental goal of knowledge representation is to represent knowledge in a manner as to facilitate inference process from knowledge. The symbolic representation is suitable to separate the procedural aspects related to a system in particular and what the knowledge represents, in such a way that any application can process this knowledge and query the information contained on it in order to infer conclusions.

The area of KBS development has matured over the past three decades. It started with first-generation expert systems with a single flat knowledge base and a general reasoning engine, typically built in a rapid-prototyping fashion. This has now been replaced by methodological approaches that have many similarities with general software engineering practice [3]. In **Figure 1**, the classical architecture of an expert system is presented, in which the *Knowledge Base* is the core element of the system.

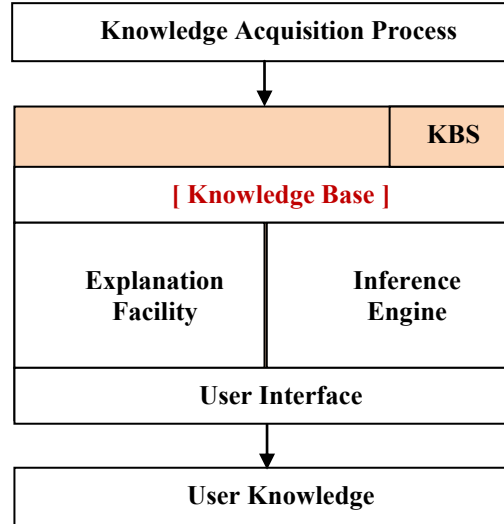


Figure 1. Classic Expert System Architecture

There are many representation techniques such as frames, scripts, rules, semantic networks and ontologies which have originated from theories of human information processing. From this set of representation techniques, the ontology is the most widely-used in Artificial Intelligence. An ontology provides a shared vocabulary, which can be used to model a domain — that is, the type of objects and/or concepts that exist, and their properties and relations.

Ontologies play a key role in the Semantic Web and other areas like agent communication and intelligent information integration, giving it a relevant importance in Computer Science. It not only captures knowledge, but it also provides meaning explicit by definition and context. However, the construction of ontologies is a bottleneck to overcome, due to the fact that manual development of ontologies can be a harsh, time consuming task. The enrichment of ontologies requires not only ontology engineers but also knowledge of domain experts, the knowledge acquisition process can be tedious and error prone. The process of incorporating knowledge to an ontology is mostly manually, making the task very harsh, which often leads to commonly create flat, lightweight ontologies that are very limited in expressivity and only populated with concepts organized hierarchically [4],[5]. This limitation neglects for instance the general concept behind ontologies, a robust and expressive knowledge of the representation of the world.

1.1 Introduction to ontologies

The term ontology is the philosophical study of the nature of being, existence, or reality in general as well as the basic categories of being and their relations.

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

- In [2], an ontology, in modern computer science, is a resource that represents the conceptual model underlying a certain domain, describing it in a declarative fashion and thus cleanly separating it from procedural aspects. An ontology is a structure:

$$O := (C, \leq_c, R, \sigma_R, \leq_R, A, \sigma_A, T)$$

consisting of:

- four disjoint sets C , R , A and T whose elements are called concept identifiers, relation identifiers, attribute identifiers and data types, respectively-
 - A semi-upper lattice $\leq_c$ on C with top element root c , called concept hierarchy or taxonomy.
 - A function $\sigma_R: R \Rightarrow C^+$ called relation signature,
 - A partial order $\leq_R$ on R , called relation hierarchy, where $r1 \leq_R r2$ implies $|\sigma_R(r1)| = |\sigma_R(r2)|$ and $\pi_i(\sigma_R(r1)) \leq_c \pi_i(\sigma_R(r2))$, for each $1 \leq i \leq |\sigma_R(r1)|$.
 - A function $\sigma_A: A \Rightarrow C \times T$, called attribute signature.
 - A set T of datatypes such as strings, integers, etc.
- In [13], 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 [14], 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 [15] or as shared understanding of some domain of interest [16].

Contemporary ontologies share many structural similarities, regardless of the language in which they are expressed. Common components of the ontology are:

- *Classes*: represent concepts. Classes in the ontology are usually organized in taxonomies through which inheritance mechanisms can be applied. For example, from a Vertebrate domain, we can state a taxonomy as seen in **Figure 2**. They are related through a *is-a* relationship (e.g. mammal **is a** vertebrate)

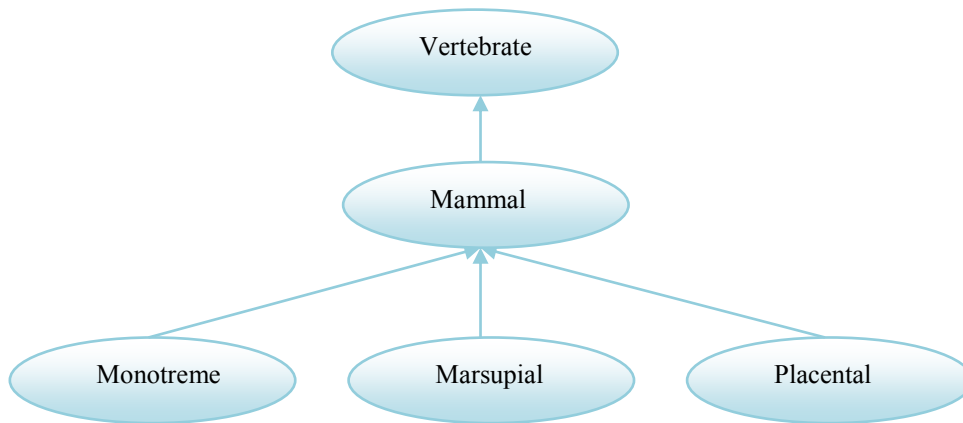


Figure 2. Concepts relation example

- *Properties*: represent a type of association between individuals of the domain or from individuals to data values. Ontologies usually contain binary relations. These relations are denoted as xRy or $R(x,y)$, in which x and y represent individuals and R the relation/property. There are three types of properties which are:
 - *Object Property*: relate individuals. In contrast to hierarchical relations, object properties represent a non taxonomic relation, which is typically expressed by a verb that relates a pair of concepts [30]. In this work we will study axioms related to this type of property. As seen in **Figure 3**, from an expression of type xRy , Barcelona represents x , is capital of represents R and Catalonia represents y .
 - *Datatype Property*: relate individuals to data values, as seen in **Figure 4**. Data value may represent an integer, string or any other data type.

- *Annotation Property*: attach metadata to classes, individuals or properties.



Figure 3. Property example (individual to individual)



Figure 4. Property example (individual to data value)

- *Instances*: are used to represent elements or individuals in an ontology.
- *Axioms*: is a sentence in first order logic that is assumed to be true without proof. It specifies constraints on the ontology and can be used in ontology consistency checking and for inferring new knowledge from the ontology through some inference mechanism.

1.2 OWL

Research in the ontology area has increased considerably during the last years, bringing the development of modern ontological languages such as Ontology Web Language (OWL 2), which provides a powerful formalism for knowledge representation and reasoning. Since its recommendation from the World Wide Web Consortium to use OWL 2, the efforts were lead into one direction. OWL 2 is a family of knowledge representation languages for authoring ontologies endorsed by the World Wide Web Consortium. They are characterized by formal semantics and RDF/XML-based serializations for the Semantic Web. OWL 2 provides a powerful formalism for knowledge representation and reasoning and contains three expressive sub languages:

- **OWL Lite**: Is the least expressive ontology language. Supports basic classification hierarchy and simple constraints features.
- **OWL DL**: Supports maximum expressiveness without losing computational completeness. It is based in description logics (DL). Description logics are a family of formal knowledge representation

languages. DLs combine a rigorous semantic based on first-order predicate logic with an intuitive way of structuring and encoding expressive conceptual knowledge [7]. OWL DL constitutes a standardized knowledge representation language well established in the Semantic Web domain.

- **OWL Full:** Supports maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. OWL Full allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary.

As seen, OWL Lite defines basic constraints in contrast to DL and Full version, which provide much more expressivity. In OWL 2, there are different types of axioms, presented in the following subsections:

1.2.1 Class expression axioms

Allow relationships to be established between class expressions.

Table 1. Class expression axioms notation in OWL 2

Axioms	OWL notation
Subclass	<code>rdfs:subClassOf</code>
Equivalent classes	<code>owl:equivalentClass</code>
Disjoint classes	<code>owl:disjointWith</code>
Pairwise disjoint classes	<code>owl:AllDisjointClasses</code>
Disjoint union	<code>owl:disjointUnionOf</code>

Subclass: States that the class C1 is a subclass of the class C2. For example, given *subClassOf(CompactCar, Car)*, states that a CompactCar class is a subclass of Car.

Equivalent classes: States that the class C1 has exactly the same members of the class C2. For example, given *equivalentClass(Car, Automobile)* we can deduce that the classes Car and Automobile contain the same members and are therefore equivalents.

Disjoint classes: States that the class C1 has exactly no member in common with the class C2. For example, given *disjointClass(Male, Female)* we can deduce that the class Male and Female are disjoint.

Pairwise disjoint classes: For ontologies in which unique classes hold, the use of *owl:AllDisjointClasses* declares that all the classes in the ontology are disjoint.

Disjoint union: States that a class *C1* is a disjoint union of the class expression *C_i*, where $1 \leq i \leq n$. For example, from *disjointUnion(Person Male Female)*, we can deduce that a person is either a male or a female, but not both.

1.2.2 Object property axioms

Characterize and establish relationships between object property expressions.

Table 2. Object property axioms notation in OWL 2

Axioms	OWL notation
Subproperty	<i>rdfs:subPropertyOf</i>
Property chain inclusion	<i>owl:propertyChainAxiom</i>
Equivalent properties	<i>owl:equivalentProperty</i>
Disjoint properties	<i>owl:propertyDisjointWith</i>
Pairwise disjoint properties	<i>owl:AllDisjointProperties</i>
Inverse properties	<i>owl:InverseOf</i>
Functional property	<i>owl:FunctionalProperty</i>
Inverse Functional property	<i>owl:InverseFunctionalProperty</i>
Reflexive property	<i>owl:ReflexiveProperty</i>
Irreflexive property	<i>owl:IrreflexiveProperty</i>
Symmetric property	<i>owl:SymmetricProperty</i>
Asymmetric property	<i>owl:AsymmetricProperty</i>
Transitive property	<i>owl:TransitiveProperty</i>

Subproperty: Object properties can be defined in a hierarchy in the same way as classes. Subproperty axiom would be a Subclass equivalent. For example, given *subPropertyOf(hasMother hasParent)* implies that having a mother implies having a parent, in the same way as *subPropertyOf(hasFather hasParent)* does.

Property chain inclusion: Allows a property to be defined as a composition of several properties.

Equivalent properties: States that the object property *OP1* is semantically equivalent to the property *OP2*. For example, given *equivalentProperty(hasBrother, hasMaleSibling)* we can deduce that both object properties are equivalent.

Disjoint properties: States that the object property OP1 is semantically disjoint to the object property OP2 in the way that an individual x cannot be connected to another individual y by OP1 and OP2. For example, given *propertyDisjointWith(hasFather, hasMother)* implies that fatherhood is disjoint with motherhood.

Inverse property: If some object property OP1 links individual x to individual y then its inverse will link individual y to individual x . For example, given the relation *hasParent(Matt, John)* an inverse property can be *hasChild(John, Matt)* established by *inverseOf(hasParent, hasChild)*.

Functional property: If an object property is functional, there can only be one individual y related to an individual x via the object property OP1. For example, given the relation *hasFather(Matt, John)*, if we define *functionalProperty(hasFather Matt John)*, then *Matt* can only be related to *John* by OP1 *hasFather* (Matt can only have one father).

Inverse Functional property: The same concept is applied as in Functional property, but to the Inverse property. As stated in Inverse property, if *hasChild(John, Matt)* is the inverse of *hasParent(Matt, John)*, considering the inverse property if we state that *InverseFunctionalProperty(hasChild John Matt)*, then *John* can only be related to *Matt* by OP1 *hasChild* (by then *John* only has one child, *Matt*).

Reflexive / Irreflexive property: An object property OP1 is said to be reflexive when the property must relate individual x to itself. For example, if we state *ReflexiveProperty(knows)* and if there exists a relation *knows(Peter, Mary)* then, *Peter* knows itself.

Symmetric / Asymmetric property: If an object property OP1 relates individual x and individual y , if y can be related to x by OP1 as well then the object property is symmetric, if not is asymmetric. For example, if we state *SymmetricProperty(friend)* and if exists a relation *friend(Peter, Mary)* then, *Mary* is friend of *Peter*.

Transitive property: If an object property is transitive, and the object property OP1 relates individual x to individual y , and the same object property instance OP1 related individual y to individual z then we can infer that individual x can be related through OP1 to individual z .

1.2.3 Data property axioms

Establish relations between data properties.

Table 3. Data property axioms notation in OWL 2

Axioms	OWL notation
Subproperty	rdfs:subPropertyOf
Equivalent properties	owl:equivalentProperty
Disjoint properties	owl:propertyDisjointWith
Pairwise disjoint properties	owl:AllDisjointProperties
Functional property	owl:FunctionalProperty

Data property axioms apply some of the object property axioms, as seen in **Table 2**. Taking this into account, all the axioms expressed in data property reflects the same definition found in object properties axioms but applied to data properties.

1.2.4 Assertions

Axioms about individuals that are often called *facts*.

Table 4. Assertion axioms notation in OWL 2

Axioms	OWL notation
Individual equality	owl:sameAs
Individual inequality	owl:differentFrom
Pairwise individual inequality	owl:AllDifferent
Class assertion	rdf:type
Positive object property assertion	a1 PN a2
Positive data property assertion	a R v
Negative object property assertion	owl:NegativePropertyAssertion
Negative data property assertion	owl:NegativePropertyAssertion

Individual equality / inequality: States than an individual x and the individual y have (or not) the same “identity”. For example, for two individuals Gary *Jefferson* and G. *Jefferson* we can state by applying *sameAs* (*Gary Jefferson*, *G.Jefferson*) that both are the same individual.

Pairwise individual inequality: For ontologies in which the unique-names assumption holds, the use of *owl:differentFrom* is likely to lead to a large number of statements, as all individuals have to be declared pairwise disjoint.

Class assertion: Allows one to state that an individual is an instance of a particular class.

Positive / Negative object property assertion: A positive object property assertion *ObjectPropertyAssertion*(OP1 x y) states that the individual x is connected by the object property expression OP1 to the individual y . On the other hand, a negative object property assertion states that the individuals x and y are not connected.

Positive / Negative data property assertion: A positive data property assertion *DataPropertyAssertion*(DP1 lt) states that the individual x is connected by the data property expression DP1 to the literal lt . On the other hand, a negative data property assertion states that the individual x and the literal lt are not connected.

1.3 Objectives and Contributions

During the last decade, many efforts to automate the ontology acquisition process have been carried out. Ontology Learning refers to extracting ontological elements from input and building an ontology from them. Several efforts have been made in order to automate the ontology acquisition and construction process, but most of them are related to taxonomic knowledge [5] and, more rarely, on non-taxonomic knowledge [6]. Other complex elements regarding ontologies such as **object properties axioms**, often named **relation axioms**, involving logical connectives, role restrictions and other semantic features that can provide expressivity to ontologies remain largely unexplored [4].

Our work presents a novel automatic axiom learning algorithm which, starting from a set of non-taxonomic relations (denoted xRy or $R(x,y)$, in which x and y represent concepts or individuals and R the relation/property) defined in an input ontology, explores and verifies which axioms those relations fulfill. Specifically, the axioms studied in our approach are *symmetry*, *asymmetry*, *reflexivity*, *irreflexivity*, *functionality*, *transitivity*, *inverse* and *inverse functionality*. Our approach relies on linguistic techniques and statistical analyses to unsupervisedly and automatically extract and evaluate semantic evidences found in Web resources [1], which support the definition of ontological axioms.

The main objectives of the present work are the following:

- Make a comprehensive study of the main techniques used to learn automatic axiom definitions.
- Study the OWL DL (Ontology Web Language – Description Logics) formalism supported by the World Wide Web Consortium (W3C), and analyse, for each kind of object property axiom, the feasibility of developing techniques based on natural language analysis of Web pages to discover automatically evidences to define the relations characteristics, more concretely object properties axioms from a given relation (e.g. symmetry, transitivity, relation inverse, functionality, etc).
- Due to the quantity of valuable information contained in the Web, we will study the best ways to use the amount and the heterogeneity of Web information to aid the learning. For these purposes, we will to study the several Web search engines available, its characteristics, accessibility and other properties.
- Develop and implement a subset of text-based patterns to exploit the Web and extract semantic evidences to support or not an object property axiom.
- Finally, we need to demonstrate the viability of our solution, so we will evaluate the decisions taken by our algorithm. Due to the lack of automatic evaluation procedures for this kind of algorithms, we will propose a manual methodology by using a human expert to evaluate the quality of the results. Using this evaluation we will quantify the quality of the results of the different decisions taken by the expert against the decision taken by the algorithm.

Our core contribution is an automatic axiom learning algorithm taking as starting point the extraction of non taxonomic relations of a well-defined ontology as input in order to verify what properties these non taxonomic relations fulfill, where the non taxonomic relations are of type xRy , in which x and y represent individuals and R the relation. Specifically, the object properties studied in our approach are symmetric, asymmetric, reflexive, functional, transitive and inverse property, in which the inverse property would allow us to detect if the original non-taxonomic relation fulfills the inverse functional property as well. Using the Web as corpus, linguistic techniques based on text patterns and statistical analysis from the distribution of the information contained in the Web are used to evaluate the extracted observations. This unsupervised mechanism allows an automated learning of

many object properties presented in OWL 2, avoiding the process of manually adding object properties in the ontology.

1.4 Document Structure

The document is structured as follows:

- Section 2 presents the state of the art of the different existing axiomatization techniques, classifying them in manual, semi automatic and automatic processes. We are going to define the most common approaches defined for the automatic and semi automatic procedures and what axioms they cover.
- Section 3 presents the relation axioms covered in this work and their OWL notation. We argue that the Web is a valid knowledge repository to support the use of the Web as the corpus for our work and how the Web search engines available allow us to obtain the resources from the Web. We introduce as well text patterns that will allow us to retrieve information from the Web by constructing Web search queries and how Web-scale statistics can be useful for inferring the degree of relationship between concepts.
- Section 4 describes our approach in detail by first introducing an algorithm of the process. This algorithm will be explained in detail through the content of the section, which is divided mainly in 3 process for axiom learning from the Web which are text patterns, natural language processing and Web-scale statistics, learning techniques explained in the previous Section 2.
- In Section 5, an evaluation of the results obtained from our algorithm is performed and analyse against the results from an expert. Based on these results we defined the effectivity of our algorithm.
- Finally, Section 6 presents a summary and conclusions about the research and further works.

Section 2

State of the Art on Automatic Discovery of Axioms

In this section, the current approaches to add logical axioms to ontologies will be presented. These approaches will be classified in manual axiomatization tools and automatic/semi-automatic axiomatization tools.

2.1 Manual Axiomatization Tools

This type of tools requires completely the work of ontology engineers and domain experts in order to axiomatize an ontology. The ontology engineers are in charge of the knowledge acquisition about the different relations identified for a particular domain ontology by the domain experts. These tools allow the user (ontology engineers preferably) to add, edit or delete axioms in a modeled ontology.

2.1.1 Protégé

In the first term, Protégé is one of the most used tools to create OWL ontologies, construct domain ontologies, customize data entry forms and enter data. The tool can be easily extended to access other knowledge-based embedded applications. Protégé allows the feasibility to add basic OWL components such as classes, relations and datatypes, but it also allows the addition of several axioms such as disjointness, universal and existential restrictions, functionality, reflexivity, transitivity, symmetric, cardinality, datatype properties and others. In this sense, Protégé is a very powerful tool in order to create complex and expressive ontologies manually. In **Figure 5**, we show the interface of Protégé.

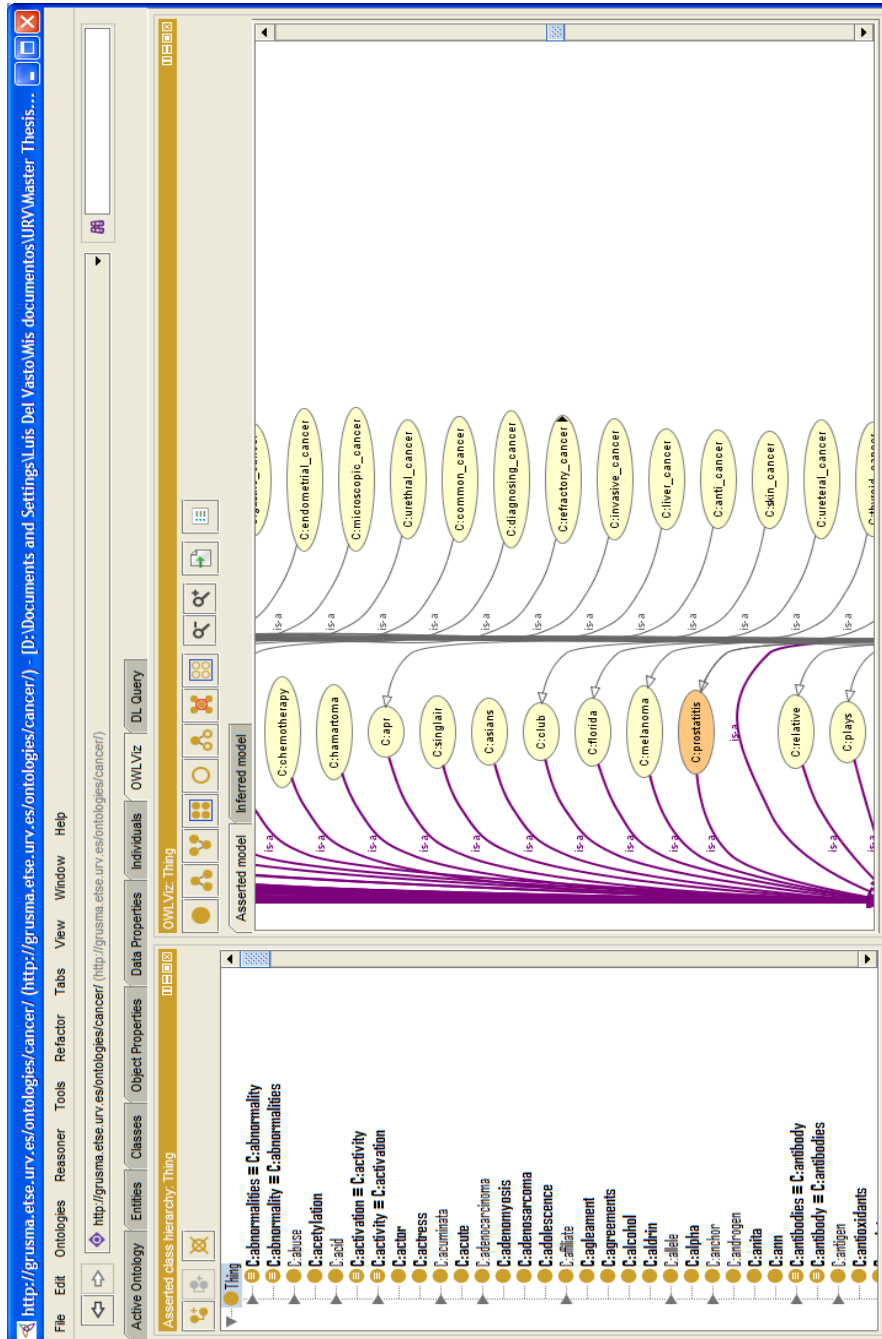


Figure 5. Snapshot of Protégé

2.1.2 OntoEdit

OntoEdit is an Ontology Engineering Environment supporting the development and maintenance of ontologies by using graphical means. OntoEdit is built on top of a powerful internal ontology model. This paradigm supports representation-language neutral modeling for concepts, relations and axioms. Several graphical views onto the structures contained in the ontology support modeling the different phases of the ontology engineering cycle - the ontology kickoff phase, refinement phase, and evaluation phase. It relies on W3C standards and offers a multitude of export interfaces to all major ontology representation languages. The ontology engineer may develop the concept hierarchy, relations and axioms in a way independent from any concrete representation language. OntoEdit is delivered with many standard plugins which includes relation axioms such as symmetry, transitive and inverse, and a disjoint concepts axiom plugin.

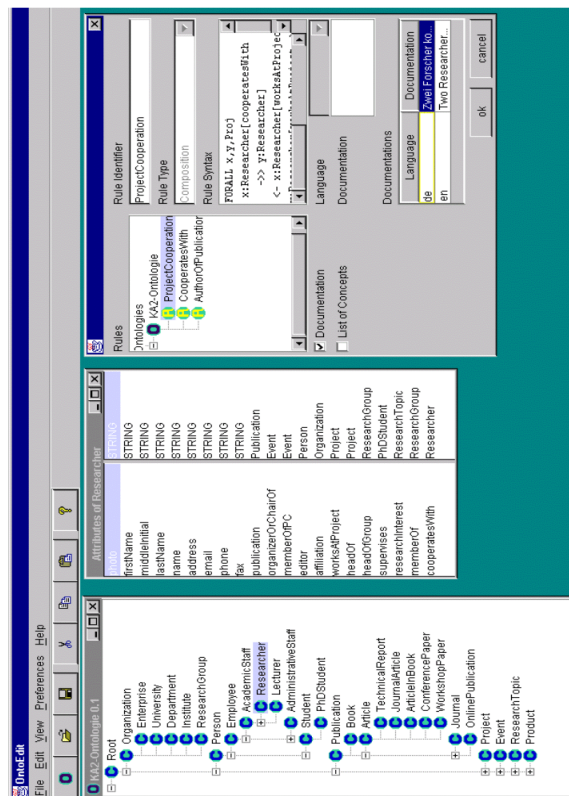


Figure 6. Snapshot of OntoEdit

2.1.3 KAON

Another manual tool is KAON, an open-source ontology management system targeted for business applications. It includes a comprehensive tool suite allowing easy ontology creation and management and provides a framework for building ontology-based applications. It supports frequently occurring patterns of logical axioms such as the specification of algebraic characteristics of relations, e.g. symmetry, inverse, transitivity, and constraints on the cardinality of relations.

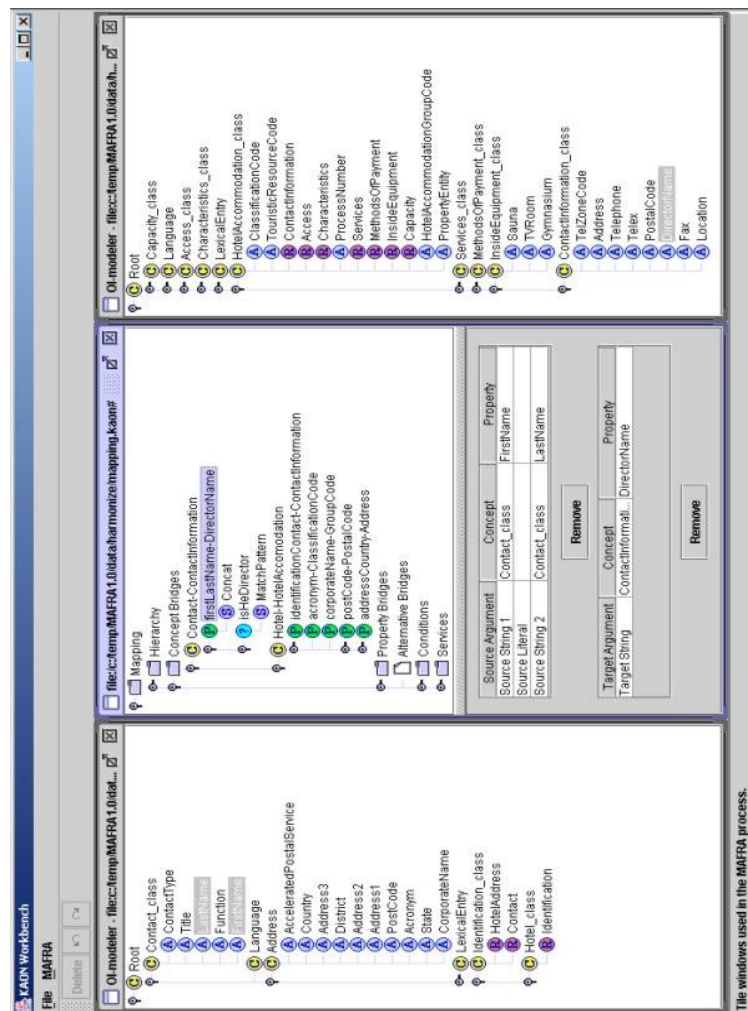


Figure 7. Snapshot of KAON tool

2.2 Automatic/Semi-Automatic Axiomatization Tools

The automatic tools are those that have some kind of automation in order to add ontology axioms under the evaluation and supervision of an expert (semi-automated) and in some cases without it (totally automated).

Several natural language processes have been applied to acquire ontology axioms from lexical resources in tools such as LExO (that stands for Learning Expressive Ontologies), RELExO (Relational Exploration) and HASTI, among others.

2.2.1 LExO

The implementation of LExO basically relies on KAON2, an ontology management infrastructure for OWL, and the Minipar dependency parser [7]. The process starts with a relatively inexpressive ontology, possibly a bare taxonomy given in RDFS. Given a natural language definition of a class to enrich, LExO starts by analyzing the syntactic structure of an input sentence. An appropriate resource (manual user input, definitions available from glossaries, encyclopedias, textual descriptions of the domain obtained from Google search, among others) could be used. The resulting dependency tree is then transformed into a set of OWL axioms (concept inclusion, transitivity, role inclusion, role assertions, concept assertions and individual equalities) by means of manually engineered transformation rules [4]. As an example, the Disjunction rule is defined as “NP₀ OR NP₁”, in which NP represents a noun phrase. If the Minipar dependency parser finds “Mazda or Toyota”, both brands concepts are set as disjoint. This approach translates hypernyms and antonyms as found in WordNet into DL Axioms. The overall scenario in order to axiomatize ontologies is a semi-automatic cyclic process [7], which contains an ontology evaluation step to make sure that the axiom candidates are correct. One of the main problems raised by this approach are the semantic inconsistencies due to natural language expressions.

In [4], the authors defined a vision of a cyclic semi-automatic ontology engineering process in order to create expressive OWL ontologies, illustrated in **Figure 8**.

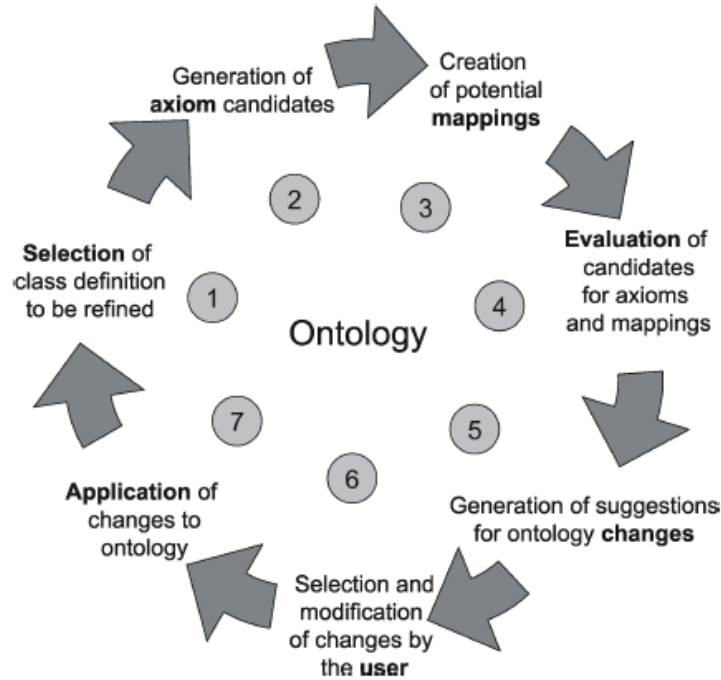


Figure 8. Ontology evolution process with LExO

- **Step 1:** The user selects the class to be refined and optionally resources for the ontology generation phase which may include manual user inputs, available glossaries and encyclopedias, textual descriptions of the domain (e.g. definitions from Google).
- **Step 2:** The definitions are parsed and transformed into OWL DL axioms that can be presented to the user. The user can intervene at this point.
- **Step 3:** The system directly proceeds to the mapping phase which aims at relating the newly generated entities and axioms to elements in the initial ontology (the ontology set on step 1).
- **Step 4:** Methods for consistent ontology evolution check for logical inconsistencies or potential modeling errors.
- **Step 5:** Based on the learned axiomatization and additional mappings the system now suggests ontology changes or extensions to the user.
- **Step 6, 7:** User evaluation of the ontology by modifying or removing some of the axioms.

2.2.2 Disjointness axiom learning (LExO based)

The feasibility of learning disjointness based on simple lexical evidence for the example presented in 2.2.1 is not enough, and a single heuristic is not suitable. Based on this, another automated attempt was performed in [10], including Taxonomic Overlap, Semantic Similarity and the OntoClean methodology.

This approach is based on several methods which include Taxonomic overlap, Subsumption and Patterns.

- The taxonomic overlap method consists of two sub methods in order to determine whether two concepts overlap or not:
 - Two classes are disjoint if their taxonomic overlap, e.g. the set of common individuals, must be empty. Individuals and subclasses which may serve as indicators for taxonomic overlap can be imported either from an ontology, or from a given corpus of text documents. In order to evaluate this, they defined a confidence $\text{conf}(p, -)$ for a pair $p = (c1, c2)$ (where $c1$ and $c2$ are concepts) not to be disjoint based on the taxonomic overlap of $c1$ and $c2$ with respect to common subclasses.
 - Del.icio.us is a server-based system with a simple-to-use interface that allows users to organize and share bookmarks on the Internet. It associates each URL with a description, a note, and a set of tags. They compute the co-occurrence of 2 concepts of two classes $c1$ and $c2$ with respect to WordNet [40].
- In Subsumption, if one class is a subclass of another, it's assumed that these two classes are not disjoint. The assumption is that a direct correspondence between the semantic similarities of two classes indicates their likeliness to be disjoint.
- The OntoClean methodology [41] is used to obtain evidence of strict disjointness between two classes. A pattern-based approach was used to analyse text taken from the Web specifically to verify that two classes have incompatible unity or identity criteria.

2.2.3 RELExO

To tackle the problem of the semantic inconsistencies due to natural language expressions presented in LExO, the relational exploration [8] method was used, which is an adaptation of attribute exploration from Formal

Concept Analysis (FCA) to description logics. This related prototype application is known as RELExO. Exploration techniques are good means to overcome the lack of completeness and precision in learned ontologies. It is just natural to apply them in order to further specify the knowledge beyond the information extractable from the corpus, which makes relational exploration a perfect complement to automatic approaches for ontology generation based on lexical resources [9]. The **Figure 9** illustrates the RELExO process, where gear wheels indicate ontology management activities including reasoning and updates, whereas the thinker icon marks user involvement. In this figure, KB represents the OWL DL ontology to be refined. The new hypothesis is tried to be asserted or denied by processing the ontology, if not is asked to the user, where I represents interpretation. This function I is extended to roles and concepts in the domain.

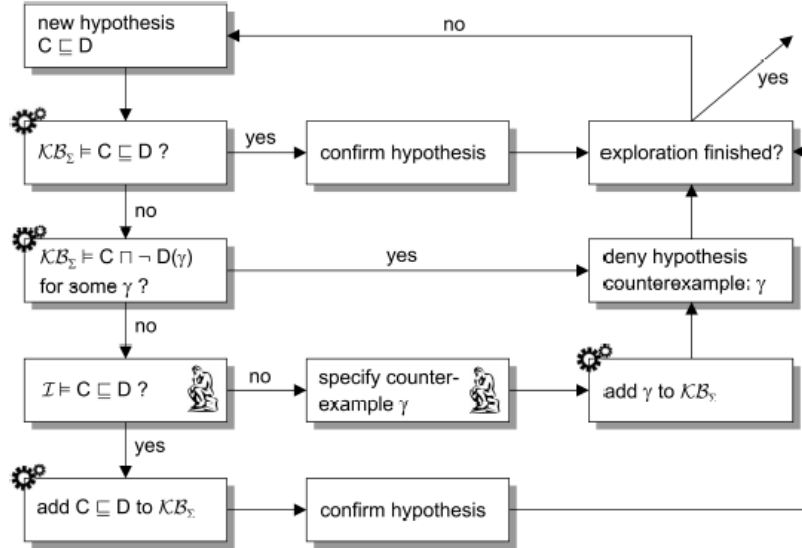


Figure 9. RELExO process

2.2.4 HASTI

HASTI [12] is a system for automatic ontology construction, based on Persian (Farsi) text understanding. It uses an initial kernel, which has the essential metaknowledge (primitive concepts and operators) to build an ontology including axioms. HASTI implements a rule base which contains linguistic rules, inference rules and heuristics to extract lexical elements from texts. HASTI uses Sentence Structures (SST) to extract taxonomic relations, non-taxonomic relations and axioms. A SST is a case frame and contains

information about thematic roles, extracted from a sentence, and their relations. HASTI extracts axioms (disjointness, transitivity and functionality) from conditional and quantified sentences. It uses Semantic Templates, which convert SSTs to ontels (ontology elements), written in Knowledge Interchange Format (KIF). In **Figure 9**, the architecture of HASTI is illustrated with all its components.

Natural Language Processor: Once the Persian texts are introduced, the input has to be processed and understood in order to extract its implicit knowledge. A part of this process is done in the NLP component, which is a simple Persian text processing subsystem. It analyses sentences morpho-syntactically and writes its intermediate and final results into working memory. It builds sentence structures to be used by the knowledge extractor and also sends lexical knowledge about words to the lexicon manager.

The Working Memory Manager: The working memory manager manages the operations on the working memory (WM). It is the repository of intermediate structures each component of Hasti writes its intermediate results or reads where the other components. As a matter of fact, it is a blackboard, acting as a bridge between the functional components of Hasti.

The Knowledge Extractor: The knowledge extractor is responsible for extracting knowledge from sentence structures. It uses semantic templates and does inferences to extract knowledge. Then the extracted knowledge about concepts and their relations will be converted to ontology elements by the ontel creator and will be written to an intermediate structure in the working memory to be placed in the ontology by the ontology manager.

The Knowledge base: The knowledge base consists of the lexicon, the ontology and the rule base. The Lexicon contains knowledge about words and phrases as the building blocks of language. The Ontology contains knowledge about concepts as the building blocks of human conceptualization (the world model) and the rule base contains the rules and methods to extract lexical and ontological elements from texts.

Each component in the knowledge base (except the rule base) is managed by its own manager. The rule base has no manager. It is fixed and is written once by the system programmer and will not be changed during the system's performance.

The Lexicon Manager: The duty of the lexicon manager is to manage the lexical knowledge sent to the lexicon. It adds new words and new senses of a word, updates existing words and retrieves word's knowledge to and from the lexicon. It is also responsible for keeping the lexicon consistent.

The Ontology Manager: The ontology manager is responsible for updating the ontology according to the ontels created by the knowledge extractor. This includes directing the ontel to a proper place in the hierarchy, eliminating the redundancies, keeping the consistency, updating concepts and relations in the

ontology, clustering and reorganization of the ontology to obtain new concept(s) and relations regarding to common / different features, attribute and values or relations.

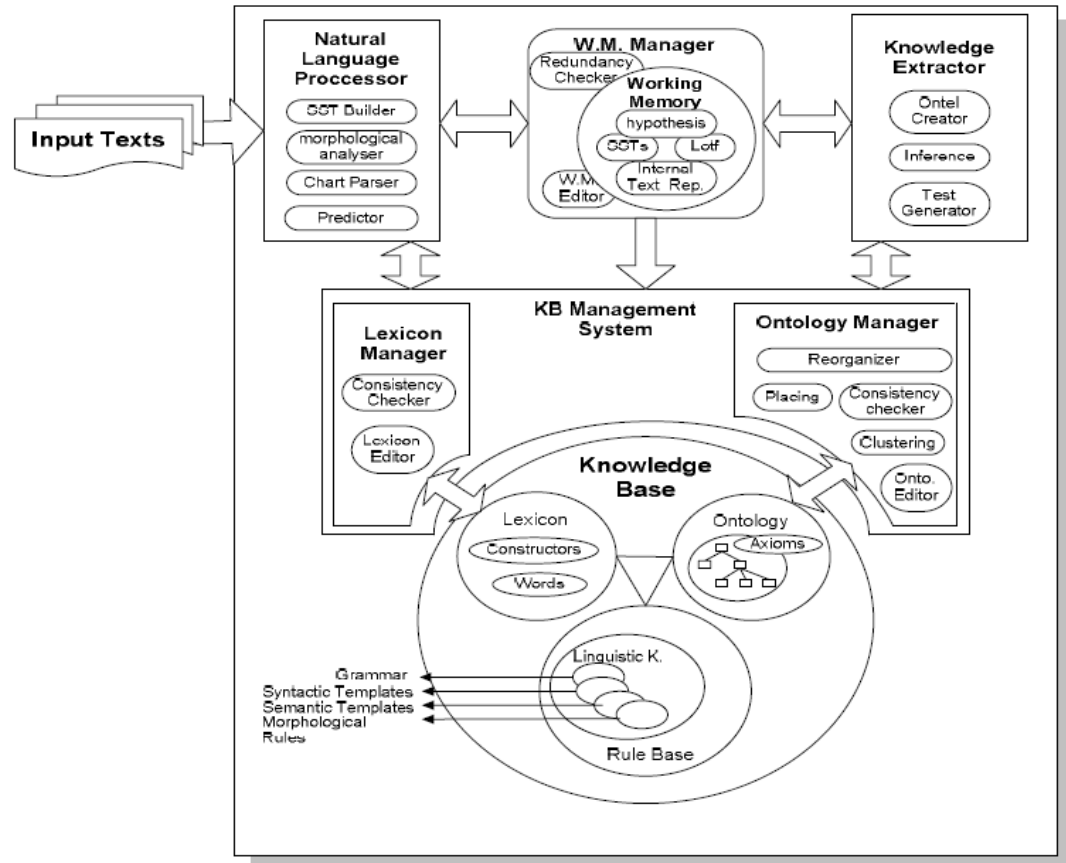


Figure 10. HASTI architecture

2.2.5 AuCONTRAIRE

Following the lexico-syntactic approach, AuCONTRAIRE is a Contradiction Detection (CD) system which relies on the TextRunner system, an Open Information Extraction system that uses the Web as corpus to obtain relations between entities [11]. In order to detect functional relations, it checks if a domain and its relation map to a unique variable using probabilistic assessment based on aggregated textual evidence. The fact that a CD system cannot assume that it knows all the functional relations makes determining

when a possible candidate for a relation is or not valid an important task to handle natural language phenomena such as synonyms, meronyms and ambiguity.

In **Figure 10**, the main components of AuCONTRAIRE are illustrated. The Extractor creates a set of extracted assertions ξ from a large corpus of Web pages or other documents, obtaining extractions of type xRy or $R(x, y)$. The Function Learner discovers a set of functional relations F from among the relations in ξ . The Contradiction Detector queries ξ for assertions with a relation R in F , and identifies sets C of potentially contradictory assertions. The system filters out seeming contradictions in C by reasoning about synonymy, meronymy, argument types and argument ambiguity. Each potential contradiction receives a probability that it is a genuine contradiction.

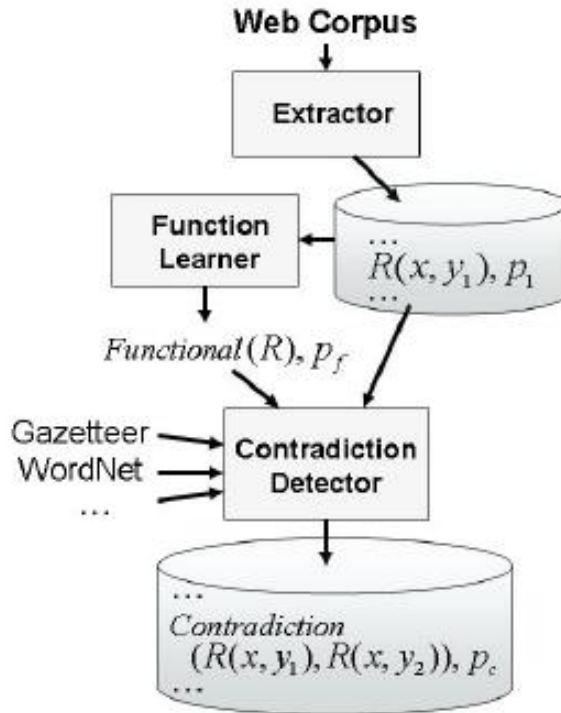


Figure 11. AuCONTRAIRE architecture

2.3 Conclusions

In Artificial Intelligence, the Knowledge Representation model and represent formally the knowledge. It's a very important field in Artificial Intelligence, as many other fields, such as KBS and Semantic Web, depend on how this knowledge is represented and how we can infer conclusions using this symbolic representation (e.g. the Knowledge Base in KBS depends on this knowledge representation). Ontologies are now the most used representation for its expressivity and inference possibilities, but their construction has a bottleneck to tackle: knowledge acquisition. The manual creation of ontologies, which includes domain experts and ontology engineers is an error prone and time consuming method. Many efforts have been made in order to automate ontology learning, but they are mainly focused on learning taxonomic and non taxonomic relations, creating lightweight ontologies in which the inference possibilities are minimized. This lack of automatic and unsupervised approaches to axiomatize ontologies, as seen in **Table 5** (in which the column S means Supervised), is evident and is the main fact that motivated us to aim our efforts into this area.

Table 5. Automatic and Semi automatic tools for axiom learning

Tool	Methodology	Knowledge base	Corpus	S
LExO	Transformation rules	WordNet	Textual resources	Semi
Disjointness	Text patterns	WordNet	Textual resources	No
	Taxonomic Overlap		and World Wide	
	Semantic Similarity		Web	
	OntoClean methodology			
RELExO	Transformation rules + Relational Exploration	WordNet	Textual resources	Semi
HASTI	Templates based on sentence structures	Halex (Hasti Lexicon) and an ontology is built upon an initial kernel	Persian (Farsi) texts	No
AuCONTRAIRE	Extraction patterns	WordNet, Gazetter	World Wide Web	No

On the contrary to related works which are applied in a narrow context (i.e. a predefined document or even a sentence), our approach uses the Web as a massive learning corpus, enabling the discovery of relation axioms in a domain-independent fashion. In these terms, the size and heterogeneity of the Web minimizes data sparseness problems in comparison to approaches like HASTI, in which the extraction of text documents in order to detect conditional or quantified sentences may be a barrier to obtain complete information about the behaviour of the concepts and relations. Unlike approaches such as LExO and RELExO, our approach will not require manual stages which, considering the size of the Web, would compromise its

scalability. Disjointness learning approaches use several methodologies to obtain only this particular axiom, which still lead to an unexpressive ontology.

Our work aims develop to a completely automatic, unsupervised methodology to discover relation axioms by using many linguistic techniques and statistical analysis to analyse and evaluate semantic evidences found in Web resources. In the next section we are going to introduce the Web as a learning corpus, the linguistic techniques and the statistical analysis used in our system.

Section 3

Learning Techniques

In this section we introduce the learning techniques applied in this research in order to discover relation axioms, specifically object property axioms. The learning techniques introduced in this document go from content analysis algorithms to information extraction procedures. For this reason it is important to understand the different techniques that have been used 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 also important to be aware of those characteristics in order to define correctly the set of techniques which will be used to discover, learn and apply the knowledge contained in the sources. This section is structured as follows:

- In §3.1, we introduce the object properties axioms covered in our work, given an initial xRy relation.
- In §3.2, 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.
- In §3.3, we explain how 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. Also we analyse and explain the comparison of different Web search engines from 2 main criterions, technical search engine features and queries properties of each Web search engine in order to find the most appropriate one for our work.
- Pattern-based approaches have been successfully applied in information extraction. In section §3.4 we introduce the main works

using lexico-syntactic patterns and in which type of ontology concepts they can be applied.

- The usage of lightweight analysis over the free text, as shown in §3.5, permits the application of higher-level techniques, which allows us to obtain evidence of new concepts or relations.
- Several scores have been proposed in the past to compute Web-scale statistics, adapting the notion of collocation and mutual information (computed as the probability of joint appearance of concepts in a corpus). In Section §3.6 we explain how search engines can provide Web-scale statistics about information distribution in a scalable and efficient way.
- In Section §3.7 we present the conclusions of this section.

3.1 Axioms covered in this work

The object property axioms studied in our work, related to OWL DL (OWL Full also incorporates them), are presented in **Table 6** together with their OWL notation and how they are expressed given a relation of type xRy (where the brackets $[]$ stand for a unique value, R' an inverse relation and $\neg$ means that the expression should not apply).

Table 6. Object property axioms considered in this work and their OWL notation

Object properties axioms	OWL notation	Expression
Reflexive	owl:ReflexiveProperty	$xRy \gg xRx$
Irreflexive	owl:IrreflexiveProperty	$xRy \gg \neg xRx$
Symmetric	owl:SymmetricProperty	$xRy \gg yRx$
Asymmetric	owl:AsymmetricProperty	$xRy \gg \neg yRx$
Functional	owl:FunctionalProperty	$xRy \gg xR[y]$
Inverse	owl:inverseOf	$xRy \gg yR'x$
Inverse Functional	owl:InverseFunctionalProperty	$xRy \gg yR'[x]$
Transitive	owl:TransitiveProperty	$xRy, yRz \gg xRz$

3.2 The Web as corpus

To establish whether the Web is a corpus we need to define what a corpus is. In [36], the authors provide the following definition:

In principle, any collection of more than one text can be called a corpus. . . . But the term “corpus” when used in the context of modern linguistics tends most frequently to have more specific connotations than this simple definition provides for. These may be considered under four main headings: sampling and representativeness, finite size, machine-readable form, a standard reference.

McEnery and Wilson, 1996

In this context, the Web allows all these main headings:

- It allows the selection, intentionally, of elements as the representation of the population being studied, by using concrete Web search queries.
- The Web has a finite size.
- A standard reference corpus is a "corpus which constitutes a standard reference for the language variety that it presents".

As stated in [1], many classical knowledge acquisition techniques present performance limitations due to the typically reduced corpus used [24]. This idea is supported by current social studies as [32], in which it is argued that collective knowledge is much more powerful than individual knowledge. The Web is the biggest repository of information available [24] 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 [33], [34], [24]. 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 [35]. Redundancy also facilitates the use of simple and fast natural language processing, because it is possible to find almost any written fact to assert it, in an easy way to analyse.

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 [1]. As a result, one may realize the important role that a Web search engine can play in the knowledge acquisition process from the Web.

A Web search engine is designed to search for information on the World Wide Web. The search results are usually presented in a list of results and are commonly called *hits*. The information may consist of Web pages, images, information and other types of files. Some search engines also mine data available in databases or open directories. Unlike Web directories, which are maintained by human editors, search engines operate algorithmically or are a mixture of algorithmic and human input.

3.3.1 Search engine features

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, Ask and Bing have been considered.

Advanced Web Search can help us to find Web sites that match very specific criteria and this fact is very important for our algorithm in order to analyse only valuable information and avoid those that aren't. Each web search engine have different advanced search capabilities, analysed in **Table 7**. For our analysis we took in consideration:

- Default: What happens when multiple terms are entered for a search not using Boolean operators such as OR and NOT.
- Boolean search: Refers to how multiple terms are combined in a search.
 - AND (in some Web search engines is "+") is used to join both terms to be found, *e.g.* ["NASA" AND "Neil"] looks for both terms in a document.

- OR let's either term be found. *e.g.* ["NASA" OR "Neil"] looks for documents that contain at least one of these terms.
- NOT (in some Web search engines is "-") that means any record containing the second term will be excluded. *e.g.* ["NASA" NOT "Neil"] looks for documents that contain "NASA" but do not contain "Neil" is not on the document.
- Exact phrase: Checks if the search engine allows exact lookups. *e.g.* ["Neil was in a NASA project"] retrieves documents which contain exactly these words.
- Proximity search: Looks for documents where two or more separately matching term occurrences are within a specified distance, where distance is the number of intermediate words or characters.
- Wildcards: Allows looking for word(s) where the wildcard is added. *e.g.* ["Columbus * America"] looks for words or characters between Columbus and America. This search can find for example "discovered", "found", or any character such as ",".
- Sorting: By which criteria the indexed Web pages are shown to the user.

Table 7. Query features of Web Search Engines

Engine	Default	Boolean search	Exact phrase	Proximity Search	Wildcard	Sorting
Google	AND	AND, NOT, OR	Yes	No	*	Relevance, site
Yahoo	AND	+, -, OR	Yes	No	*	Relevance, site
Bing	AND	AND, NOT, OR	Yes	No	None	Relevance, site
Ask	AND	+, -, OR	Yes	No	None	Relevance, metasite

As seen in **Table 7**, the advanced search features for the analysed Web search engines are basically the same, except for the wildcards that are available in Google and Yahoo. In this context, the search engine that we can choose to apply in our work will be based on the search engine retrieval comparison.

3.3.2 Search engine retrieval comparison

From the discussion presented in the previous section, it is clear the importance of the search engine retrieval comparison in order to choose the best Web search engine available. This is why we have studied the available

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

In [1], each analysed search engine has been evaluated from different points of view:

- *Access*: some search engines (such as Yahoo) 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, 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 summarized in **Table 8** and **Table 9**. In the first one, we try to analyse basically the size and the scope of each Web search by applying exact phrases queries (which is how we are going to query search engines).

Table 8. Number of results for specific queries

Query	Google	Yahoo	Bing	Ask
"Angelina Jolie received"	138.000	411	308	54
"Breast cancer is affected by"	5.390	328	412	80
"cities like Tarragona"	1.200	65	81	14
"pollution causes diseases"	12.900	203	104	32

In **Table 8**, 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, while Yahoo, Bing and Ask offer less results, especially Ask.

Table 9. Summary of the main characteristics of each Web Search Engine

Search engine	Access	Limitations	Coverage	Response time
Yahoo	API	5000 queries per day, account and IP	<i>Medium</i>	<i>Lowest</i>
	Web access			
Bing	API	No limits	<i>Medium</i>	<i>Fastest</i>
	Web access			
Google	Web access only	N/A	<i>Highest</i>	N/A
Ask	Web access only	N/A	<i>Lower</i>	N/A

Taking those facts in consideration, **Table 5** shows a summary of the analysed features of each search engine.

- **Access:** Bing and Yahoo offer both API and Web access while Google and Ask don't provide any API to retrieve results from queries (they disabled their APIs).
- **Limitations:** The Bing API has no access limitation, so it's possible to query on their engine as long as you want. On the other hand, Yahoo only allows 5,000 queries per day. Google and Ask are not considered, because they don't have any API to access their engine and will not be included in our work.
- **Coverage:** Google has the best Web coverage for general concepts and specific queries. On the other hand, Bing offers a great coverage for specific queries as well as Yahoo. For specific queries, both have great coverage and depending on the Web query any Web search engine can retrieve more pages than the other. Ask in both cases retrieved the lowest quantity of Web pages.
- **Response Time:** Bing API has a faster response time than the Yahoo one. However, Yahoo response time is not too slow.

The results of this empirical study are quite similar to those presented in [37], in which the 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. 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) but unfortunately there is no API available so we can't consider it in our work. However, Bing and Yahoo contain also great quantity of information available for both specific queries and general concepts and their response times are good. We are going to use both (Bing and Yahoo) for our purpose to complement one another to maximize the coverage of Web documents.

3.4 Lexico-syntactic Patterns

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. The use of information extraction techniques, involving lexico-syntactic patterns, has been widely used in order to generate ontological information from unstructured text to enrich ontologies or to create them from scratch. There are several lexico-syntactic patterns that are easily recognizable and occur frequently in natural language texts.

3.4.1 Lexico-syntactic Patterns for Taxonomic relations

Heart's patterns are especially useful for detecting specializations of concepts that can represent is-a (taxonomic) relations [17]. In particular, Hearts defined a collection of patterns indicating hyponymy relations. According to Hearts, the patterns should satisfy these requirements:

- They should occur frequently and in many text genres
- They should accurately indicate the relation of interest
- They should be recognizable with little or no pre-encoded knowledge

Taking these specifications into account, the patterns used by Hearts (where *NP* is a noun phrase) are:

- *NP* such as $\{NP, \}^* \{(and \mid or)\} NP$
- such *NP* as $\{NP, \}^* \{(and \mid or)\} NP$
- *NP* $\{NP, \}^* \{, \}$ or other *NP*
- *NP* $\{NP, \}^* \{, \}$ and other *NP*
- *NP* including $\{NP, \}^* NP \{(and \mid or)\} NP$
- *NP* especially $\{NP, \}^* \{(and \mid or)\} NP$

In **Table 10**, some examples according to the patterns defined by Hearts are presented in order to illustrate how these patterns can be found in natural language text.

Table 10. Examples of Hearts linguistic patterns

Pattern	Example	Relation
<i>NP</i> including $\{NP, \}^* NP \{(and \mid or)\} NP$	... many cancer procedures including radiotherapy, chemotherapy and surgery.	hyponym("radiotherapy", "cancer procedures") hyponym("chemotherapy", "cancer procedures") hyponym("surgery", "cancer procedures")
such <i>NP</i> as $\{NP, \}^* \{(and \mid or)\} NP$	... such cancers as breast cancer and kidney cancer.	hyponym("breast cancer", "cancer") hyponym("kidney cancer", "cancer")
<i>NP</i> such as $\{NP, \}^* \{(and \mid or)\} NP$	... cancers such as breast cancer, and kidney cancer.	hyponym("breast cancer", "cancers") hyponym("kidney cancer", "cancers")
<i>NP</i> especially $\{NP, \}^* \{(and \mid or)\} NP$	... cancer cause, especially heredity, and mutation.	hyponym("heredity", "cancer causes") hyponym("mutation", "cancer causes")

As stated in [1], another pattern-based approach for detecting specializations is the use of noun phrases (*e.g. credit card*) and adjective noun phrases (*e.g. local tourist information office*) in which the phrase has as head a noun or pronoun and is accompanied by one or several modifiers. Concretely, in the English language, the immediate anterior word for a keyword is frequently *classifying* it (expressing a semantic specialization of the meaning), whereas the immediate posterior one represents the *domain* where it is applied [31]. So, on the one hand, the *previous word* for a specific keyword can be used to obtain the taxonomic hierarchy of terms (*e.g. pressure*

sensor is a subclass of *sensor*). If the process is repeated recursively we can create deeper-level subclasses (e.g. *air pressure sensor* is a subclass of *pressure sensor*). On the other hand, the *posterior word* for the specific *keyword* can be used to obtain the context in which the immediate anterior concept is applied (e.g. *colorectal cancer research* will be a domain of application of *colorectal cancer*).

3.4.2 Lexico-syntactic Patterns for Non-Taxonomic relations

As stated in [2], even though many ontology learning techniques have been developed, most of these approaches focus on the automatic acquisition of classes and taxonomic relationships, and often neglect the importance of interlinkage between concepts. Even though taxonomic knowledge is certainly of utmost importance, major efforts must be dedicated to the definition of *non-taxonomic conceptual relationships*.

For non-taxonomic relations, many patterns have been created for general purposes such as part-of terminology giving relevant results using a large corpus and the use of refined statistical measures to rank the candidates [18].

Some of the defined patterns are presented in **Table 11** together with an example of their use.

Table 11. part-of patterns presented by Berland and Charniak

Pattern	Example	Relation
<i>whole</i> NN[-PL]'s POS <i>part</i> NN[-PL]	Building's basement	<i>part-of</i> ("basement ", " building")
<i>part</i> NN[-PL] of PREP {the a} DET mods [JJ NN]* <i>whole</i> NN	Basement of a building	<i>part-of</i> ("basement ", " building")
<i>part</i> NN[-PL] in PREP {the a} DET mods [JJ NN]* <i>whole</i> NN	Basement in a building	<i>part-of</i> ("basement ", " building")
<i>parts</i> NN[-PL] of PREP <i>wholes</i> NN-PL	Basement of buildings	<i>part-of</i> ("basement ", " building")
<i>parts</i> NN[-PL] in PREP <i>wholes</i> NN-PL	Basement in buildings	<i>part-of</i> ("basement ", " building")

Qualia structures [19] and causation [20] have been also used to extract general relations, by using specific language related linguistic patterns (e.g. *X*

consists of Y , X is used for Y , X leads to Y). Even though those approaches may have interest for developing or enriching general purpose semantic networks (such as WordNet), they are able to retrieve only certain specific relationships. However, relations that can be crucial to have a truly expressive ontology may be omitted.

In [2], they used pattern-based approaches towards the extraction of evidences using as starting point non-taxonomic relationships defined in an ontology to validate the existence of relations properties. We can find unstructured text that matches these patterns and as consequence extract valuable information to deduce related concepts or relations. The use of a large corpus as the Web may help to find text that matches these patterns.

3.5 Natural Language Analysis

In the philosophy of language, a natural language (or ordinary language) is any language which arises in an unpremeditated fashion as the result of the innate facility for language possessed by the human intellect. A natural language is typically used for communication, and may be spoken, signed, or written. Natural language is the primary medium by which humans communicate with each other, asking questions, expressing beliefs, desires, attitudes and commands as well as reporting events, actions and states. In general, different syntactic categories are used to refer to different types of ontological entities. Proper nouns are typically used to refer to individuals, whereas verb nouns can be regarded as referring to classes, as stated in [2]. **Table 12**, describes some of the main categories in natural language and the entities they refer to:

Table 12. Natural language categories and entities referred

Natural language category	Referred Entity
Verbs	Events, states, actions, beliefs, attitudes, etc
Proper nouns	Individuals
Nouns	Classes
Adjectives	Class attribute values
Adverbs	Description of manner
Prepositional phrases	Spatio-temporal conditions

Natural Language Processing (referred as NLP) is a field of computer science and linguistics concerned with the interactions between computers and human (natural) languages. This is a subfield of Artificial Intelligence, and by applying NLP it's possible to analyse sentences syntactically. Natural language understanding systems convert samples of human language into more formal representations such as parse trees or first-order logic structures that are easier for computer programs to manipulate.

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 analyzing the 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.
- *Stop Words*: finite list of domain independent words with very general meaning that can be omitted during the analysis. Determinants, prepositions or adverbs are typically contained in this category.
- *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. 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.

3.5.1 Stemming analysis

Stemming usually refers to a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes. In **Table 13**, we present some examples of the use of the Porter stemming algorithm, first introduced in [38].

Table 13. Results of the Porter stemming algorithm

Words	Stemmed word
Submit	Submit
Submits	
Submitted	
Submitting	
Student	Student
Students	
Enhance	Enhanc
Enhanced	
Enhancing	
Partial	Partial
Partially	
Partialized	
Responsibility	Respons
Responsible	

There are many other stemming algorithms such as Lemmatisation algorithm, stochastic algorithm, N-gram algorithm, etc. We used the by using Porter stemming algorithm, which is one of the most common algorithms applied in Information Extraction. By using stemming, we can avoid duplicity of information to analyse by discarding plurals. For example, by using stemming we can treat “Student” and “Students” as the same instance because they have the same stem as root.

3.5.2 Detecting sentences, tokens and part-of-speech

In sentence analysis the first step is to detect sentences from a text. Most natural language processing tools use as default the point (.) as a delimiter to separate sentences. Other delimiters can be used such as comma (,), question mark (?), exclamation (!) and others. Once the sentence detector splits de whole text, the tokenization is the next step. It’s in charge of separating the words for the analysis. For example, “don’t” should be separated to “do” and “not” for a good further text analysis.

To explain sentence analysis after sentence detection, we take as an example the sentence “Today the book is available on Amazon.” In **Figure 11**, we show the result of the analysis of this sentence. The first layer

corresponds to the tokenization of the sentence. Once the tokenization is successfully done, the part of speech tagging or word category disambiguation process is executed. POS tagging is the process of marking up the words in a text (corpus) as corresponding to a particular part of speech such as nouns, verbs, adjectives, etc. In **Figure 11**, the POS tagging results are below the tokenization (second layer). The terms in the second layer are:

- **DT**: Determiner
- **IN**: Preposition or subordinating conjunction
- **JJ**: Adjective
- **NN**: Noun, singular or mass
- **NNP**: Proper noun, singular
- **VBZ**: Verb, 3rd person singular present

The POS tagging, is very useful because we obtain syntactic information of every word in the sentence structure, but it could also be useful in itself when looking for units of meaning in a sentence. Finding phrases is often called “chunking”. The last layer in **Figure 11** shows the result of the “chunking” process. Note that in this case only a noun phrase (**NP**) was generated from more than one word. The terms expressed in the third layer are:

- **ADJP**: Adjective phrase
- **NP**: Noun phrase
- **PP**: Preposition phrase
- **VP**: Verb phrase

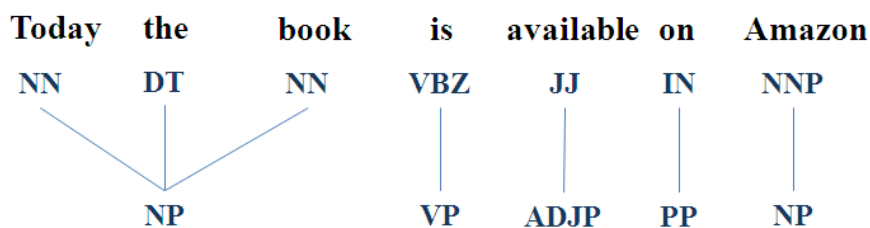


Figure 12. Sentence analysis

Natural language processing is a very powerful tool used in several knowledge acquisition works [1], [2], [7], [12] that can be included in our

work to analyse unstructured knowledge, taking into account that syntactic categories are used to refer to different types of ontological entities as explained before.

3.6 Web-Scale Statistics

The use of statistical measures in knowledge related tasks for inferring the degree of relationship between concepts is a very common technique when processing unstructured text [21]. As stated in [2], some research has hypothesized that the fact that the occurrence of some word implies the occurrence of some other word in the same sentence, paragraph or document hints at a potential directed relation between words. If co-occurrence relations in corpora reflect semantic information, they should appear in a similar way in written and spoken languages [39]. For example, by examining an enormous corpus as the Web we can determine that the words “Spain” and “Europe” keep more relation than the words “Spain” and “Oceania” by detecting that the words “Spain” and “Europe” appear in more documents together than the words “Spain” and “Oceania” America”. As explained before, in our work we used the Web as corpus and in the same way as in [2], [10], [11], [25] several heuristics for exploiting the statistics provided by web search engines.

Information extraction from the web is a noisy process, which can lead statistical techniques to 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 [22] or with a combination of statistics and lexical information, in hybrid approaches [23]. In this sense, some authors [24] have demonstrated the convenience of using a wide corpus in order to improve the quality of classical statistical methods. Concretely, in [25] methods to address the sparse data problem are proposed by using the hugest data source: the Web. Applied to the Web, probabilities p are estimated as the hit count (*hits*) returned by a Web search engine when querying the term. There are many different measures of the degree to which two words co-occur used across many different works.

In [26], Pointwise Mutual Information measure is used measure how much the actual probability of a particular co-occurrence of events $p(x, y)$ differs from what we would expect it to be on the basis of the probabilities of the individual events and the assumption of independence $p(x) p(y)$, in which ‘x’ and ‘y’ are the concepts to relate.

$$PMI(x, y) = \ln \frac{p(x, y)}{p(x) * p(y)} \approx \ln \frac{\left(\frac{\text{hits}("x" \text{ AND } "y")}{\text{total_web_sites}} \right)}{\frac{\text{hits}("x")}{\text{total_web_sites}} * \frac{\text{hits}("y")}{\text{total_web_sites}}} \approx \ln \frac{\text{hits}("x" \text{ AND } "y")}{\text{hits}("x") * \text{hits}("y")} \quad (1)$$

In [27], the implementation of Symmetric Conditional Probability was used to find the most relevant co-occurrent normal units (2). As in (1), 'x' and 'y' represent the concepts to relate.

$$SCP(x, y) = \frac{p(x, y)^2}{p(x) * p(y)} \approx \frac{\left(\frac{\text{hits}("x" \text{ AND } "y")}{\text{total_web_sites}} \right)^2}{\frac{\text{hits}("x")}{\text{total_web_sites}} * \frac{\text{hits}("y")}{\text{total_web_sites}}} \approx \frac{\text{hits}("x" \text{ AND } "y")^2}{\text{hits}("x") * \text{hits}("y")} \quad (2)$$

In [28] [29], the authors implemented Google Normalized Distance function in order to represent the relation between concepts (3). In conditional probabilities the total number M , represents the total number of web pages indexed in the Google search.

$$NGD(x, y) = \frac{\max\{\log p(x), \log p(y)\} - \log p(x, y)}{\log M - \min\{\log p(x), \log p(y)\}}$$

$$NGD(x, y) = \frac{\max\{\log(\text{hits}("x")), \log(\text{hits}("y"))\} - \log(\text{hits}("x" \text{ AND } "y"))}{\log M - \min\{\log(\text{hits}("x")), \log(\text{hits}("y"))\}} \quad (3)$$

The distributional hypothesis in linguistic claims that words that occur in the same context tend to have similar meanings. Due to the fact that the evidences extracted from the corpus shares a common context, we use web-scale statistic to obtain the semantic relatedness between the evidence obtained in respect of the relation analysed from the ontology. For our purpose, we used these three semantic relatedness measures in order to verify the relation between the concepts involved in our search once they are extracted.

3.7 Conclusions

As seen in this section, the Web is a large and valid knowledge repository. As stated by McEnery and Wilson in 1996 [36], the Web provides sampling and representativeness, a finite size and is a standard reference for any domain. Apart from the huge amount of information available, another feature that characterizes the Web is its high redundancy. Thus, there are many Web search APIs available in order to access this information. In this work we used both Yahoo and Bing APIs to complement each other when querying exact phrases, which reduce considerably the quantity of results because it's very restrictive.

The main learning techniques used to support our work are lexico-syntactic patterns and Web-scale statistics. Some lexico-syntactic patterns have been used mainly to generate ontological information from unstructured text to enrich ontologies or to create them from scratch, mainly in taxonomic and non-taxonomic relations. The use of Natural Language Process together with these patterns is very helpful to detect a defined text structure from a corpus and analyse its syntactic information by using processes such as sentence detection, part-of-speech, stemming and many others. On the other hand, the use of statistical measures in knowledge related tasks for inferring the degree of relationship between concepts is a very common technique when processing unstructured text. In this section we presented 3 co-occurrence techniques which are Symmetric Conditional Probability, Pointwise Mutual Information and Normalized Google Distance in order to calculate the semantic relatedness of the concepts or relations extracted from our corpus, the Web.

In the next section we are going to introduce the methodology used in our work in order to discover axioms from the Web, established mainly in the learning techniques introduced in this section.

Section 4

Object Property Axioms Learning Methodology

Up to this point, we have described in detail all the learning methodologies used, so now we are able to describe the object property axioms discovery algorithm that we have developed. In this section we are going to explain step by step the axiom learning methodology process:

This section contains the following content:

- In §4.1, a general description of the algorithm will be done.
- As explained in the previous section, lexico-syntactic pattern has been widely used in many Information Retrieval procedures. In §4.2 we are going to explain the patterns proposed in our work in order to develop our work and how we develop Web search engine in base of this patterns.
- In §4.3, we introduce the natural language analysis performed for the documents obtained from the Web search engine and how we extract the candidates in order to validate if an axiom fulfils or not from a given relation.
- In §4.4 we defined the use of web scale statistic measures in our proposal for the selection process relying on co-occurrence measures computed from the Web search candidates selected in §4.3.
- In §4.5 we present the conclusion of the methodology applied in our work.

4.1 Algorithm Description

The pseudocode of the algorithm we propose is presented, in which a domain ontology is received as input and a set of axioms associated to ontological relationships defined via object properties is obtained as output. During the explanation of the method, we refer to algorithm lines as *<line>*.

```

1. AxiomLearning (Domain Ontology DO)
2. { AXS_OP /* Results variable */
3.   AXIOMS /* Constant holding the set of axioms of interest */
4.   THRESHOLD /* Constant holding the selection threshold */
5.   RELS  $\leftarrow$  get_nontaxonomic_relations(DO)
6.   /* for each relation, analyse each object property axiom */
7.   for all  $RELS_i \in RELS$ 
8.   {   for all  $AXIOMS_j \in AXIOMS$ 
9.   {   QUERY  $\leftarrow$  generate_query( $RELS_i$ ,  $AXIOMS_j$ )
10.      WEBPAGES  $\leftarrow$  retrieve_WebPages(QUERY)
11.      for all  $WEBPAGES_x \in WEBPAGES$ 
12.      { /* for each web page check for matchings */
13.         TEXT  $\leftarrow$  extract_matchings(QUERY,  $WEBPAGES_x$ )
14.         TEXT  $\leftarrow$  remove_stopwords(TEXT)
15.         KEY  $\leftarrow$  stem_word(TEXT)
16.         exists  $\leftarrow$  check_existence(AXS_OP,  $RELS_i$ ,  $AXIOMS_j$ , KEY)
17.         if (!exists)
18.         { /* calculate web scale statistics */
19.            ws = calculate_CoOccurrence( $AXIOMS_j$ ,  $RELS_i$ , TEXT)
20.            /* add info. to result if SCP > THRESHOLD */
21.            if (ws > THRESHOLD)
22.            { AXS_OP  $\leftarrow$  AXS_OP + ( $RELS_i \cup AXIOMS_j \cup KEY \cup TEXT \cup ws$ )
23.            }
24.         }
25.      }
26.   }
27.   Return AXS_OP
28. }
```

In the algorithm presented above, the non-taxonomic relations are considered to be analysed by the algorithm <5> from an ontology used as input <1>. The algorithm presented for non-taxonomic relations axiomatization is basically divided in 3 main processes:

- **Pattern construction & Web search queries:** Depending on the object property axiom to analyse, we construct a query <9> to use in a Web search engine <10>. The construction of this query, depending on the object property axiom, is explained in the Section 4.2. This query will result in a set of Web pages retrieved.
- **Natural language analysis:** The results obtained from the Web search engine are further analysed by natural language processing tools in order to extract candidates that will allow us to determine if, in fact, the relation fulfils or not certain object property axiom. In <13>, we analyse if the extracted document matches the pattern proposed by the object property axiom. If a candidate is extracted, we continue the process by removing stop words on the text extracted <14> and stemming the text same text <15>. This process is explained more in detail in Section

- **Web-scale statistics:** The process ends with the selection of candidates extracted in the previous step depending on its relatedness calculated using web-scale statistics <19>. If this calculation is higher than a threshold established, extraction is considered as a valid candidate for the relation <21, 22>.

The non-taxonomic relations from the input ontology are obtained using JENA 2.6 <5>, a Java Framework for building Semantic Web applications. Is an open source that provides environment for RDF, RDFS and OWL. By using this API the user can read and write over OWL files (*.owl).

In the next sections we are going to explain the 3 main steps that our algorithm propose.

4.2 Pattern construction & Web Search Queries

Lexico-syntactic patterns, as tools to discover and extract ontological entities, have been used in order to learn ontologies from unstructured text. Our approach bases the axiom learning in the exploitation of specially designed patterns aimed to retrieve (by means of querying a Web search engine) and extract (by means of a linguistic analysis) semantic evidences that support the fulfillment or not of each one of the studied Object Property Axioms. The patterns proposed below are domain-independent and can be adjusted to queries in many Web search engines. In base of these patterns, we have designed queries for the Yahoo! and Bing Web search engines <9>.

4.2.1 Patterns

We defined a set of 3 text patterns in order to discover object properties axioms. To describe the proposed patterns we use the nomenclature *Relation(Subject, Object)* in which *Relation* represents a relation and *Subject/Object* represent individuals given a non taxonomic relation xRy . As an example, in a non-taxonomic relation defined as *Borders(Spain, France)*, Spain is the *Subject*, France is the *Object* and *Borders* represents the *Relation*. In the following sub-sections the proposed patterns and their use to check some object properties are described.

4.2.1.1 RELATION (SUBJECT, ?)

If an individual *SUBJECT* is related to an object *y*, we can try to extract if there exist more objects different from *y* related to *SUBJECT*. If that is not the case, we can conclude that *SUBJECT* cannot be related to other objects

different from y by *Relation*. This pattern can be used to detect both Functional and Inverse Functional Object Properties. Also we can use this pattern to detect Transitive Object Property.

Functional Object Property & Inverse Functional Object Property:

Given a relation xRy , it is possible to detect if the subject x can be related to another object different from y by the relation R to check the *Functionality* of the Object Property xRy and the *Inverse Functionality* of the Object Property $xR'y$ (R' being the inverse of R). In the query language we denote this pattern in (4) for both *Functional* and *Inverse Functional* Object Properties.

$$["x \ R" - "x \ R \ y"] \quad (4)$$

Taking into account that we need extract different objects for a certain relation, a search query for $["x \ R"]$ can fulfill this task. However the object must be different from the original object in the non taxonomic relation. In order to avoid the extraction of the same object in the relation, by adding to the web search query $[-"x \ R \ y"]$, we can avoid the extraction of this same object. From the non taxonomic relation example, to find if exists more individuals y , the example is denoted in web search query language as:

$$["Spain \ borders" - "Spain \ borders \ France"]$$

Transitive Object Property: Given a relation xRy , to detect the *Transitivity* of the property for the subject x , we need first to detect if the object y can be related to another individual z by relation R . In this way we can detect a relation yRz in which the relation R is the same for both xRy and yRz . In query language we denote this pattern in (5) for the first step of Transitive Object Property.

$$"y \ R" - "y \ R \ x". \quad (5)$$

This query may help us to detect if y can be related to an object z by relation R , while $[-"y \ R \ x"]$ avoids the extraction of the original subject x in xRy as object. From the non taxonomic relation example, to find if there exists objects z related to y from the relation xRy , the example is denoted in Web search query language as:

$$["France \ borders" - "France \ borders \ Spain"]$$

4.2.1.2 RELATION (SUBJECT, OBJECT)

From a non-taxonomic relation xRy , the SUBJECT x and the OBJECT y can be used together with the RELATION R in order to check if the non-taxonomic relation is symmetric, as well as we can also use as object y the word “ITSELF” to evaluate the reflexivity in xRy . In basic terms we used this pattern to detect if a relation already established as xRy exists in our corpus.

Symmetric Object Property: Given a relation xRy , by switching the positions of the subject x and the object y we can check if the relation R can be applied from y to x as well, denoted in a Web search query as illustrated in (6).

$$"y \ R \ x". \quad (6)$$

Using the non taxonomic relation example, to find if there exists a relation R between the subject x and the object y , the example is denoted in query language as:

[“France borders Spain”]

Reflexive Object Property: Given a relation xRy , to check its reflexivity we used as object the reflexive pronoun ‘itself’, which is used to refer to things or animals. In query language we denote this pattern as illustrated in (7).

$$"x \ R \ ITSELF". \quad (7)$$

Using the non taxonomic relation example, to find if there exists a relation R between the subject *Spain* to *ITSELF*, the example is denoted in query language as:

[“Spain borders ITSELF”]

Transitive Object Property: As said in the previous pattern **RELATION (SUBJECT, ?)**, we can detect yRz from an initial relation xRy . The second step uses this pattern in order to check that the subject in xRy can be related to the object in yRz . In query language we denote this pattern illustrated in (8).

$$"x \ R \ z". \quad (8)$$

In basic terms, we used this pattern to detect if it is possible to find evidence in the Web of a concrete relationship between two specific individuals. In query language, the example is written as follows:

[“Spain borders *”]

Where * are the candidates extracted from the first step, queried as [“France borders” – “France borders Spain”].

4.2.1.3 ? (SUBJECT, OBJECT)

Relations are commonly defined as a verb that relates two individuals. By searching for verb phrases between two individuals, it is possible to extract a relation between them. This pattern can be used to extract an Inverse Object Property that will be defined as a relation between a subject and an object. This pattern is used in basic terms to extract new relations between two individuals.

Inverse Object Property: Given a relation xRy , we can search for an inverse relationship by looking for a verb that relates y to x . In query language we denoted this pattern as shown in (9).

$$“y * x” \text{ OR } “y * * x” \text{ OR } “y * * * x”. \quad (9)$$

Note that the wildcard (*), which is used in the query in order to force the proximity (up to three words) between the subject and the object, has to correspond to a verb phrase. From the non taxonomic relation example used, this pattern in Web search query language is denoted as:

[“France * Spain” OR “France * * Spain” OR “France * * * Spain”]

Once an Inverse Object Property is detected from a relation xRy , it is possible to detect the Inverse Functional Property as defined in **RELATION (SUBJECT, ?)**.

In **Table 14**, we summarize the patterns proposed and their relation to the object properties studied in our work and how these patterns are constructed into Web search query in base of the example Borders (Spain, France).

Table 14. Queries constructed for each axiom studied

Object Property Axiom	Pattern	Query
Symmetric and Assymetric	"y R x"	"France borders Spain"
Reflexive and Irreflexive	"x R ITSELF"	"Spain borders ITSELF"
Transitivity	1. "y R" – "y R x"	1. "France borders" – "France borders Spain"
	2. "x R [Result in step 1]"	2. "Spain borders [Result in 1rst step]"
Functionality and Inverse Functionality	"x R" – "x R y"	"Spain borders" – "Spain borders France"
Inverse	"y * x" OR "y * * x" OR "y * * * x"	"France * Spain" OR "France * * Spain" OR "France * * * Spain"

4.2.2 Web Search Engine Retrieval

According the non-taxonomic relation and the object properties axioms to analyse, a Web Search query string is generated <9> as seen in the previous sub-section. This Web search engines retrieve ranked lists of web resources, statistics, snippets and html caches. Using these Web Search queries we retrieve the results from the Web Search Engine API <10>. The ranking presented different Web search engines may vary and therefore more quantity of evidences for an axiom may be discovered. Besides, the use of exact phrases in the queries may induce to obtain low quantity of resources to analyse when comparing to non exact phrases, as showed in Section 3.3.2 for Yahoo or Bing. In this sense, we used both Web search engines to compensate this issue:

- **Yahoo Web Search API v2.12:** Yahoo recently provided an API for accessing Yahoo Search services. Similarly, it is also limited to a maximum of 5000 queries per day, account and IP.
- **Bing API v2.0:** The Bing Application Programming Interface (API), Version 2, enables developers to programmatically submit queries to and retrieve results from the Bing Engine. Bing doesn't have any restriction about queries per day, account and/or IP.

The results obtained from this Web search engines contain many fields per Web page retrieved:

- **Description:** Description text of the result.
- **URL:** Contains the URL string of the Web page retrieved.
- **Title:** The title of the Web page.

In **Figure 12**, we illustrate these default fields obtained using Web Search Engines. Within the red box is the title, the black box contains the description and the yellow box the URL.

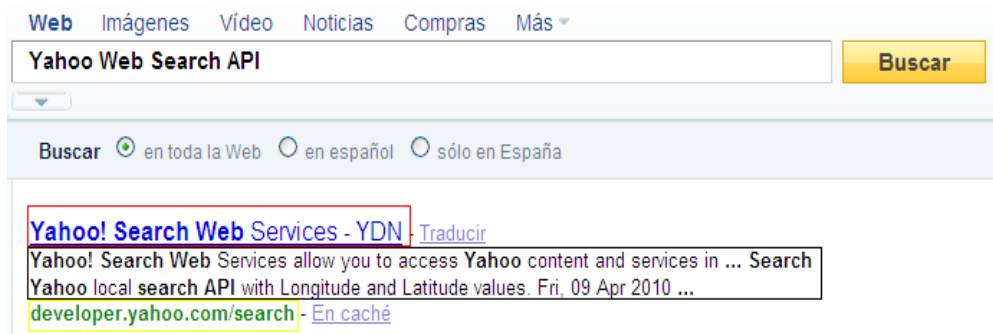


Figure 13. Default fields obtained from Web Search APIs

In our algorithm, we used specifically the URL field. By using HTML Parser we can extract clear text from a Web page. HTML Parser this is a powerful HTML parser that allows processing web content. It has been used to extract automatically clear text contained in a web resource. Taking this tool into account, by using the URL provided by the Web Search result and the HTML Parser functionality, we can analyse the whole content within the Web page, explained in the next section.

4.3 Natural Language Analysis and Candidate Extraction

As stated in Section 3.5, different syntactic categories are used to refer to different types of ontological entities [2]. Taking this into account, we used natural language processing tools to analyse the text extracted from the Web search engine and extract possible candidates that may indicate us if the relation fulfills an axiom or not.

In the following sections we are going to explain the process of natural language analysis and candidate extraction in our algorithm.

4.3.1 Natural Language Processing Analysis

Once we extract clear text from a Web page retrieved, we can perform the text analysis <13> by using natural language processing tools, in our case OpenNLP version 1.4.3, which is a mature Java package that hosts a variety of *Natural Language Processing* tools which perform sentence detection,

tokenization, pos-tagging, chunking and parsing, allowing morphological and syntactical analysis of texts. We integrated these tools to assist in the processing of text.

The first step of the process is using the sentence detector from OpenNLP in order to split the whole text from a Web page into sentences, by using model trained on English data. To illustrate the process, let's analyse the xRy relation "*Tobacco is associated with lung cancer*" in order to verify if this relation is functional, where the relation R is "*is associated with*", the subject x is "*Tobacco*" and the object y is "*Lung cancer*". If we extract from a Web page retrieved the following text from:

"Furthermore, epidemiological studies from various parts of the world show that certain natural chemicals in food may be carcinogenic risks to humans; for example, the chewing of betel nuts with tobacco is associated with oral cancer."

Using the Sentence Detector we can analyse separately these two sentences:

Sentence 1: *"Furthermore, epidemiological studies from various parts of the world show that certain natural chemicals in food may be carcinogenic risks to humans."*

Sentence 2: *"For example, the chewing of betel nuts with tobacco is associated with oral cancer."*

Once the sentences are splitted, the tokenization process is done, splitting each sentence into tokens – that is, words and punctuation. This tokenization process will split words that consist of contractions: for example, it will split "don't" into "do" and "n't", because it is designed to pass these tokens on to the other NLP tools, where "do" is recognized as a verb, and "n't" as a contraction of "not", an adverb modifying the preceding verb "do". Once the tokenization process is performed, we continue with the part-of-speech tagging process, which is the act of assigning a part of speech (POS) to each word in a sentence. Once the POS process is performed, we finish the sentence analysis performing a "chunking" process that will group the tokens of a sentence into larger chunks, each chunk corresponding to a syntactic unit such as a noun phrase or a verb phrase. As we are looking for new candidates that the tobacco is associated to something else besides *Lung Cancer*, we look specifically for "*Tobacco is associated with*" in each sentence. As the sentence 1 doesn't have any match for this structure, only the sentence 2 is analyzed using the natural language processing tool because it contains the structure "*Tobacco is associated with*" in its sentence. In **Table 15**, the tokenization, POS and chunking process are illustrated for the sentence 2.

Table 15. Tokenization and POS process

Tokenization	POS	POS meaning	Chunking	Chunking meaning
For	IN	Preposition or subordinating conjunction	B-PP	Beginning Prepositional Phrase
Example	NN	Noun, singular or mass	B-NP	Beginning Noun Phrase
,	,	,	O	Undefined
The	DT	Determiner	B-NP	Beginning Noun Phrase
Chewing	NN	Noun, singular or mass	I-NP	Intermediate Noun Phrase
Of	IN	Preposition or subordinating conjunction	B-PP	Beginning Prepositional Phrase
Betel	NN	Noun, singular or mass	B-NP	Beginning Noun Phrase
Nuts	NNS	Noun, plural	I-NP	Intermediate Noun Phrase
With	IN	Preposition or subordinating conjunction	B-PP	Beginning Prepositional Phrase
Tobacco	NN	Noun, singular or mass	B-NP	Beginning Noun Phrase
Is	VBZ		B-VP	Beginning Verb Phrase
Associated	VBN		I-VP	Intermediate Verb Phrase
With	IN	Preposition or subordinating conjunction	B-PP	Beginning Prepositional Phrase
Oral	JJ	Adjective	B-NP	Beginning Noun Phrase
Cancer	NN	Noun, singular or mass	I-NP	Intermediate Noun Phrase
.	.	.	O	Undefined

Analyzing the tokenization, POS and chunking process over sentence 2, let's analyze the conditions previously established in Section 3.2.

4.3.2 Extraction Filters

The text is analysed in order to discover and extract the semantic evidences that may lead to the definition (or not) of the corresponding axiom. In order to minimize noise and ambiguity of textual processing, we imposed some syntactic boundaries depending on the used pattern:

- **Relation (Subject, ?)** and **Relation (Subject, Object)**. For these patterns we decided to get rid of interrogative sentences, cardinal candidates and candidates that are superclasses of the initial object. Words between parentheses are deleted.
- **? (Subject, Object)**. For this pattern, we neglected interrogative sentences, cardinal candidates, and verbs in future or gerund tense. Candidates between parentheses are also deleted.

If a text piece fulfills the restrictions defined for the particular pattern, then the evidence is extracted <13> for further analysis. In **Table 16** we illustrate what restrictions are applied for each pattern.

Table 16. Conditions to validate the patterns defined

Restrictions	Example	ROLE (SUBJECT,?)	ROLE (SUBJECT, OBJECT)	? (SUBJECT, OBJECT)
Avoid superclasses of the SUBJECT	If SUBJECT is “thermometer”, avoid “pill thermometer”	Yes	Yes	No
Interrogative sentences	It is true that cancer is related to fat diet?	Yes	Yes	Yes
Cardinal candidates	200,000	Yes	Yes	Yes
Parenthesis in candidates	If the candidate is “solar ultraviolet rays (uvr)”, do not extract “(uvr)”.	Yes	Yes	Yes
Future tense verb phrases	The cigarette smoking will affect your lung cancer.	No	No	Yes
Gerund verb tense	distinguishing, studying, etc.	No	No	Yes

- **Avoid superclasses of the SUBJECT:** As the noun to analyze is “*tobacco*”, when analyzing its chunking we can detect that this is the first noun (NN) of the noun phrase (NP) detecting that this word is the beginning of the noun phrase (B-NP).
- **Interrogative sentence:** The last token in the sentence is not “?”, so we cannot determine that the sentence is interrogative.
- **Cardinal sentence:** After the relation “*is associated with*” in the sentence 2 is “*Oral*”, which is not a cardinal number, denoted in POS as CD (Cardinal number).
- **Parenthesis in candidates:** Any parenthesis is found in the sentence.

As none of these restrictions are triggered, we start analysing the words after relation, in this case the words “*is associated with*”, extracting the noun phrase after it. In our case we extracted “*Oral cancer*” as candidate, as the point “.” is not part of the noun phrase.

The natural language process continues with removing stop words <14>. We used JavaSDM 20090423 full version, which is a Java library developed by the Royal Institute of Technology in Stockholm, Sweden. This library contains many natural language processing tools including stemming

capabilities. In concrete we used this library for detecting stop words from a given text.

The last step is the text stemming from the candidate extracted and after this process the result from the stop words removing is stemmed <15>. We used the Snowball library, a small string processing language designed for creating stemming algorithms for use in Information Retrieval by using the Porter Stemming algorithm. This process is used to generate keys from the text and avoid duplicity when evaluating candidates. For example, if we retrieve as candidates from a given relation “*cancer*” and “*cancers*”, by stemming both texts and using this stemming result as key (in this case “*cancer*”) to store the same concept or relation.

4.4 Web Scale Statistics Measures

Applying the mentioned constraints to avoid natural language problems does not mean that the observations extracted for a certain relation are strong enough to grant the definition of an axiom. The next step consists on deciding which of the extractions are reliable and sufficiently related to the property. To perform this selection process we performed a Web-based statistical analysis relying on co-occurrence measures computed directly from Web search engines <19>.

In the next section we explain how we use these co-occurrence measures explained in Section 3.6 for our work. Using for example, the relation “*Tobacco is associated with lung cancer*” we found a possible candidate “*Oral cancer*” to detect if our relation is functional or not. We evaluate the co-occurrence by setting values to the *Discriminator Phrase*, *Discriminator Phrase without Instance* and the *Instance*. The extracted term is the *instance* together with the relation that links it to another individual and the *discriminator phrase without instance* represents the relation without the extracted *instance*.

p(x,y) - *Discriminator Phrase*: “Tobacco” AND “is associated with” AND “Oral Cancer”.

p(x) - *Instance*: “is associated with” AND “Oral cancer”.

p(y) - *Discriminator Phrase without Instance*: “Tobacco” AND “is associated with”.

Taking these 3 factors into account, we can calculate the co-occurrence measure using the hits of each one of these terms. The quantity of hits obtained for these 3 factors are the following:

p(x,y) - Discriminator Phrase: 54,000 hits.

p(x) - Instance: 142,001 hits.

p(y) - Discriminator Phrase without Instance: 1,820,014 hits.

$$\text{PMI}(x, y) = \ln \frac{54,000}{142,001 * 1,820,014} = -15.38$$

$$\text{SCP}(x, y) = \frac{54,000^2}{142,001 * 1,820,014} = 0.011$$

$$\text{NGD}(x, y) = \frac{14.41 - 10.89}{22.81 - 11.85} = 0.32$$

The Web-based statistical evaluation process varies depending on the patterns studied. We are going to analyze how we evaluate this process depending on the pattern in the following paragraphs.

RELATION (SUBJECT, OBJECT)

For the reflexive, symmetric and transitive (in its second step) properties, implies the assessment of the relative relatedness of the extracted evidences (computed by means of the SCP) with respect to a threshold. If the SCP of the extracted evidence is higher than the threshold, we may conclude that the relation is reflexive, symmetric or transitive (in its second step) <21, 22>.

For example, from a given relation “*Bill Clinton is married to Hillary Clinton*”, were:

SUBJECT: “*Bill Clinton*”

OBJECT: “*Hillary Clinton*”

RELATION: “*is married to*”

Symmetric Object Property: By using this pattern we will query to the Web search engine [“*Hillary Clinton is married to Bill Clinton*”]. If there exist this relation, we will have the same concepts and relation, therefore the co-occurrence is the same obtained in the original relation.

Reflexive Object Property: By using this pattern we will query to the Web search engine [“*Hillary Clinton is married to ITSELF*”]. If there exist this relation, we have to calculate the co-occurrence of this relation, by calculating the co-occurrence of the concepts “*Hillary Clinton*” and “*ITSELF*” and the relation “*is married to*”. If evidences of this relation are found in the Web, the

calculation of the measure of this relation is performed and if is higher than the threshold set, we can conclude that the relation is reflexive.

Transitive Object Property (second step): We have to analyse if “Hillary Clinton is married to *”, where * are the candidates obtained from the first step [“Hillary Clinton is married to” – “Hillary Clinton is married to Bill Clinton”]. By using this pattern we will query to the Web search engine [“Bill Clinton is married to *”]. If evidences of this relation are found in the Web, the calculation of the measure of this relation is performed and if is higher than the threshold set, we can conclude that the relation is transitive.

RELATION (SUBJECT, ?) and ? (SUBJECT, OBJECT)

For the functional, inverse, inverse functional and transitive (in its first step) object properties, if the set of extractions contains an item with a co-occurrence higher than the threshold then we may conclude that the extractions are robust for the relation. Depending of these observations, an axiom may be fulfilled or not. In the functional and inverse functional case, if the extractions are robust then we may conclude that the relation is not functional or inverse functional <26, 27>. For the inverse axiom, we may conclude that an inverse relation has been found. In the first step of the transitivity test, we conclude that a good candidate to check for the transitivity in the second step has been found <21, 22>.

Using the same relation from the pattern **RELATION(SUBJECT, OBJECT)** we evaluate the co-occurrence measure as follows:

Functional Object Property: If we are evaluating if this relation is functional, the Web search query generated is [“*Bill Clinton is married to*” – “*Bill Clinton is married to Hillary Clinton*”], we may extract “*Commission member*” and therefore we calculate the co-occurrence that the concepts “*Hillary Clinton*” and “*leading Democratic candidate*” are related by the relation “*is married to*”. If this co-occurrence is higher than a certain threshold, we can conclude that the evidence is valid and therefore the relation is not functional.

Inverse Functional Object Property: The calculation of the Inverse Functional Object Property is the same as applied in Functional Object Property, but using its Inverse Object Property, if applies. If for example, “*Bill Clinton is Married to Hillary Clinton*” is an Inverse relation, the procedure of co-occurrence calculation remains the same.

Transitive Object Property (first step): In its first step, we use the extraction of candidates from the query [“Hillary Clinton is married to” – “Hillary Clinton is married to Bill Clinton”]. If there exists any evidence (denoted as *) that “*Hillary Clinton*” is married to someone else besides “*Bill Clinton*”, we calculate the co-occurrence of the candidate “*”, the concept “*Hillary Clinton*” and the relation “*is married to*”. Once this is applied, the transitivity process continues in its second step for each extraction *.

Inverse Object Property: Let’s use the following example to illustrate how the co-occurrence is measured for the Inverse Object Property.

Given the relation “*Global warming is caused by pollution*”, were:

SUBJECT: “*Global warming*”

OBJECT: “*Pollution*”

RELATION: “*is caused by*”

If we are evaluating if this relation have an inverse relation, the Web search query generated is [“*Pollution* Global warming*” OR “*Pollution* * Global warming*” OR “*Pollution* * * Global warming*”], we may extract the verb “*causes*” and therefore we calculate the co-occurrence that the concepts “*Pollution*” and “*Global warming*” are related by the relation “*is caused by*”. We apply this calculation with any other candidate extracted. If the co-occurrence for each extraction is higher than a certain threshold, we can conclude that the extraction is valid and therefore the relation is not functional.

4.5 Conclusion

As seen in this section, we presented an algorithm to illustrate the whole process applied in our work to discover axioms from the Web based on our main learning techniques. The process is based on 3 main steps which are:

- The pattern construction and Web search queries: We presented 3 patterns used for the object properties axioms studied. These patterns are domain-independent and can be used for any ontology or Web search engine. The process of Web search queries construction is simple and as the patterns are independent to a particular syntactic specification can be adjust to any Web search engine.
- Natural language analysis: The process of how analyse text from Web documents retrieved by using Natural Language processing tools was presented, including sentence detection, tokenization, part-of-speech tagging, stop words removal and stemming.

- Web-scale statistics measures: The calculation of how much two concepts are related to a particular relation is the last step. From the extractions obtained in the steps before, we need to determine if this extractions is in fact valid for the relation to conclude if the axiom analysed fulfils or not. We presented how we calculated the 3 co-occurrence methods presented in base of a given xRy relation and an extraction obtained from the corpus and how this calculation determine if a particular object property axiom fulfils or not.

In the next section we are going to introduce the evaluation process of the results obtained from our algorithm to determine if, in fact, the methodology presented in this work can help us to add expressiveness to an ontology.

Section 5

Empirical Evaluation

Due to the difficulty to carry out automatic evaluations of automatically acquired knowledge, we based them in a manual approach. We developed two manual evaluation procedures. On one hand, the first manual evaluation consists of the selection of the best co-occurrence measure from the 3 measures mentioned, taking into account the best ranking from a set of extractions obtained from the algorithm. On the other hand, we evaluated manually a set of tests with the best extraction ranked by the best co-occurrence measure selected in the first evaluation and the decision taken by the algorithm regarding if the relation fulfils or not the object property analysed and this decision is compared against the decision taken by the expert.

This chapter is structured as follows:

- In §5.1, the first manual evaluation is carried out. The 3 co-occurrence measures are evaluated to check which of 3 fits better for our purpose.
- The second manual evaluation is carried in §5.2. Each object property axiom is evaluated individually by using the co-occurrence measure selected in §5.1. We defined a number of positive and negative relation examples of each of the studied axioms in this work and their results, provided by our algorithm, are compared against the decisions taken by the expert.
- In §5.3 we present the conclusions of the evaluation.

5.1 Discounted Cumulative Gain

To evaluate which of the three semantic relatedness measures defined in Section 3.6 fits better for our purpose, we measured the effectiveness of the extracted candidates by using Discounted Cumulative Gain (DCG), which is a measure of effectiveness, often used in information retrieval. DCG measures the usefulness of a rank based on the position of the extracted candidates in the list of candidates. The premise of DCG is that highly relevant documents appearing lower in a search result list should be penalized as the graded

relevance value is reduced logarithmically in proportion to the position of the result. The DCG accumulated at a particular rank position p , where rel_i is the graded relevance of the result at position i , is defined as:

$$DCG = \sum_{i=1}^p \frac{2^{rel_i} - 1}{\log_2(1+i)} \quad (10)$$

Two assumptions are made in using DCG and its related measures.

- Highly relevant documents are more useful when appearing earlier in a search engine result list (have higher ranks)
- Highly relevant documents are more useful than marginally relevant documents, which are in turn more useful than irrelevant documents.

DCG originates from an earlier, more primitive, measure called Cumulative Gain.

Let's illustrate how we used DCG in order to rank the list. For our example, we punctuate good candidates with 1 point and those that aren't with points. For instance, if we try to detect if the relation "Brazil borders Bolivia" is functional, from the query ["Brazil borders" + "Brazil borders Bolivia"] we obtain the following results:

Table 17. Ranks defined by co-occurrence measures for the relation "Brazil borders Bolivia", checking for its functionality

Candidates extracted	Pt	Rank by PMI	Rank by SCP	Rank by NGD
Adjacent countries	0	Suriname	Peru	Adjacent countries
Ocean	1	Paraguay	Paraguay	Ten countries
Paraguay	1	Peru	Venezuela	Ocean
Peru	1	Venezuela	Suriname	Suriname
Suriname	1	Adjacent countries	Ocean	Paraguay
Ten countries	0	Ocean	Ten countries	Peru
Venezuela	1	Ten countries	Adjacent countries	Venezuela
DCG Calculated		0.98	1	0.64

According to the ranks obtained for PMI, SCP and NGD, the DCG calculated that the SCP rank for the candidates extracted is better than the rank for PMI and NGD. The SCP co-occurrence measure calculated that "Ten countries" and "Adjacent countries" are less related to "Brazil borders" in the Web than the other candidates that the expert classified as good candidates. PMI, for example, ranked the wrong candidate "Adjacent countries" over

“Ocean” and therefore the DCG calculation penalized this rank. In this sense, SCP provided better ranks globally than PMI and NGD.

Table 18. Ranks defined by co-occurrence measures for relation “sociology is the study of society”, checking for its inverse functionality

Candidates extracted	Pt	Rank by PMI	Rank by SCP	Rank by NGD
culture	1	humanity diverse ways	people	symbolically created relationships
different aspects	0	symbolically created relationships	Life	specific societies
group behaviour	1	human social relationships	humans	humanity diverse ways
group life	1	human social interaction	men	human social relationships
human behavior	1	human social life	culture	group behavior
human groups	1	human social behavior	societies	human social interaction
human interaction	1	group life	individuals	group life
human social behavior	1	social rules	relationships	human groups
human social interaction	1	specific societies	meaning	human social life
human social life	1	group behaviour	human behavior	social rules
human social relationships	1	humans	social patterns	human interaction
human society	1	human groups	social life	different aspects
humanity diverse ways	0	social behavior	interactions	human social behavior
humans	1	social action	social behavior	social action
individuals	1	social life	human society	social relationships
interactions	0	People	social groups	social relations
life	1	social groups	human social behavior	social groups
meaning	0	social relationships	social relations	social behavior
men	1	human society	social relationships	human society
people	1	social relations	social action	Interactions
relationships	1	Men	human social life	Meaning
social action	1	human behavior	social rules	social patterns
social behavior	1	Life	human interaction	social life
social groups	1	human interaction	different aspects	Relationships
social life	1	societies	group life	Individuals
social patterns	1	different aspects	human groups	human behavior
social relations	1	Culture	human social interaction	Culture
social relationships	1	individuals	group behaviour	Societies
social rules	1	relationships	humanity diverse ways	Life
societies	1	social patterns	human social relationships	Men
specific societies	1	meaning	specific societies	People
symbolically created relationships	0	interactions	symbolically created relationships	Humans
DCG Calculated		0.82	0.98	0.86

According to the ranks obtained from the **Table 18**, the SCP shows also a better ranking than PMI and NGD.

Having a better ranking by using DCG punctuations means that the extractions obtained are more reliable and therefore the probability of extract good candidates is higher. On the other hand, lower DCG punctuations means that the extractions calculated are more error prone. Taking this into account, we concluded that the SCP provides better results globally than PMI and NGD. On the other hand, globally, NGD had the worst ranking results based on DCG. In the next section we are going to explain how we evaluated our algorithm for every single object property axioms studied in this work.

5.2 Results Evaluation

A number of positive and negative relation examples of each of the studied axioms are presented, along with the results given by our methodology. In the tables shown in this section the following abbreviations are used:

- SCP O/P: SCP values of the original relation and those obtained for the best extraction from the constructed pattern.
- D: Decision of the algorithm concerning the evaluated axiom relation.
- HD: Decision of a human concerning the evaluated axiom relation.
- NC: Number of extractions for a given axiom and relation.
- BES: Best extraction selected (the one with the highest SCP value), when applicable.

The examples used to evaluate the properties have been selected from different domains, in order to show the genericity of the proposed methodology. In all cases the threshold used to select or discard an extraction has been empirically set to $1E-3$. In order to set this threshold, we took as reference the average of co-occurrence of the analyzed relations, by using the three co-occurrence measures explained in section 3.6. We used two Web search engines and the maximum number of Web pages retrieved for each pattern has been set to 500 per Web search engine.

5.2.1 Symmetric / Assymmetric Object Property Evaluation

The evaluation of the Symmetry of an Object Property has been tested with 10 relations, presented in **Table 19** (5 of them are symmetric and 5 of them are not). The algorithm agrees in 9 out of 10 cases with the human evaluation. Note that, in this case, SCP O and SCP P coincide as the two individuals and the relationship are the same in the original relation xRy and in the relation yRx . In our test, 4 negative examples didn't obtain any result with the given pattern.

Table 19. Test of Symmetry for several Object Properties

Original relation	Pattern	SCP O	SCP P	D	HD
Spain borders France	France borders Spain	0.22	0.22	Y	Y
Obesity is associated with cancer	Cancer is associated with obesity	0.09	0.09	Y	Y
Microsoft competes with Google	Google competes with Microsoft	0.29	0.29	Y	Y
Angelina Jolie is married to Brad Pitt	Brad Pitt is married to Angelina Jolie	0.34	0.34	Y	Y
Cigarettes are linked to cancers	Cancers are linked to cigarettes	0.001	0.001	Y	Y
Deep Blue defeated Gary Kasparov	Gary Kasparov defeated Deep Blue	0.004	0.004	Y	N
Heavy smokers get lung cancer	Lung cancer get heavy smokers	6.5E-4	-	N	N
Michael Jackson was born in Gary	Gary was born in Michael Jackson	0.006	-	N	N
Sociology is the study of society	Society is the study of Sociology	0.009	-	N	N
Madrid is located in Spain	Spain is located in Madrid	0.03	-	N	N

From the "Deep Blue defeated Gary Kasparov" example notice that, as a matter of fact, Gary Kasparov defeated Deep Blue but in a different match. This illustrates a limitation of the automatic process.

5.2.2 Reflexive / Irreflexive Object Property Evaluation

The evaluation of the Reflexivity of an Object Property has been tested with 10 relations, presented in **Table 20** (5 of them are reflexive and 5 of them aren't). A relation is considered reflexive if the SCP score of the pattern exceeds 1E-3. The algorithm agrees in 8/10 cases with the human evaluation.

Table 20. Test of Reflexivity for several Object Properties

Original relation	Pattern relation	SCP O	SCP P	D	HD
Nature regulates CO2	Nature regulates itself	0.04	0.23	Y	Y
Market regulates capital production	Market regulates itself	2.88E-3	0.14	Y	Y
Fire consumes wood	Fire consumes itself	0.15	0.09	Y	Y
Science studies Nature	Science studies itself	0.07	0.03	Y	Y
Immune system attacks viruses	Immune system attacks itself	0.015	0.0048	Y	Y
Testicular cancer has good prognosis	Testicular cancer has itself	1.1E-4	3.6E-5	N	N
People have colon cancer	People have itself	0.002	0.16	Y	N
Pressure sensor measures water pressure	Pressure sensor measures itself	8.4E-3	1.9E-4	N	N
Hysterectomy reduces cancer	Hysterectomy reduces itself	0.012	7.2E-4	N	N
Stress affects cancer	Stress affects itself	0.05	0.046	Y	N

In the evaluated examples, the algorithm found evidences for “people have itself” and “stress affects itself”. The correlations of “*itself*” with their respective subjects are high, but that doesn’t imply they are correct. For these two cases, the human expert evaluated the reflexivity as wrong.

5.2.3 Functional Object Property Evaluation

The evaluation of the Functionality of an Object Property has been tested with 10 relations, presented in **Table 21** (5 of them are functional and 5 of them are not). The algorithm agrees in 8/10 cases with the human evaluation. The table shows the best extraction found with the relation(subject,?) pattern. The algorithm takes the relation to be functional if the best extraction has an SCP lower than the established 1E-3 threshold.

Table 21. Test of Functionality for several Object Properties

Original relation	NC	BES	SCP O	SCP P	D	HD
Brasilia is the capital of Brazil	4	Country	0.004	3.7E-4	Y	Y
Cadmium exposure is associated with prostate cancer	1	reduced pulmonary function	1.28E-4	1.3E-4	Y	N
Da Vinci Code was written by Dan Brown	1	Bart D. Ehrman	0.009	6.5E-4	Y	Y
Comet Halley is named for Edmund Halley	0	-	0.019	-	Y	Y
Microsoft competes with Google	42	Apple	0.29	0.23	N	N
Brazil borders Bolivia	8	Peru	0.14	0.20	N	N
Michael Jackson was born in Gary	3	August	0.005	0.006	N	Y
Raloxifene reduces breast cancer	32	osteoporotic fractures	9.4E-4	0.006	N	N
Spain borders France	3	Portugal	0.229	0.25	N	N
Chichen Itza is known as El Castillo	2	La Iglesia	0.008	8.25E-5	Y	Y

Some of the results of this table are commented with more detail in the following paragraphs.

Cadmium exposure is associated with prostate cancer: Only one candidate was found. The original relation “Cadmium exposure is associated with prostate cancer” scored a SCP of 1.28E-4 and the BES “reduced pulmonary function” scored higher than the original relation with 1.3E-4, but the algorithm considered the relation as not functional. For this case the expert decided that the relation with the best extraction is in fact valid.

Microsoft competes with Google: As illustrated in **Table 22**, there are several evidences with a strong relation that concludes that Microsoft competes with many other companies.

Table 22. Functional candidates extracted for “Microsoft competes with Google”

Evidence	SCP
Apple	0.232
Linux	0.169
IBM	0.165
Oracle	0.129
Yahoo	0.117
Nokia	0.087
Palm	0.064
<i>Filtering products</i>	<i>2.11E-5</i>
<i>Tough rivals</i>	<i>6.65E-6</i>
<i>Traditional brick-and-mortar companies</i>	<i>5.51E-6</i>

Brazil borders Bolivia: Table 23 shows that there are strong countries candidates to border with Brazil, concluding that Brazil borders many other countries aside Bolivia.

Table 23. Functional candidates found for “Brazil borders Bolivia”

Evidence	SCP
Peru	0.244
Paraguay	0.212
Venezuela	0.209
Suriname	0.153
Ocean	0.037
<i>Ten countries</i>	<i>2.106E-4</i>
<i>Adjacent countries</i>	<i>5.531E-5</i>

Michael Jackson was born in Gary: Regarding the relation “Michael Jackson was born in Gary”, the relation “was born in” may be related to either time or place giving unexpected results as seen in Table 24. The expert considered the relation as functional (giving an approach of “place” only) while the algorithm considered it as non functional.

Table 24. Functional candidates found for “Michael Jackson was born in Gary”

Evidence	SCP
August	0.007
Indiana	0.002
<i>Poverty-stricken Gary</i>	<i>7.092E-7</i>

Raloxifene reduces breast cancer: Raloxifene not only reduces breast cancer, but also reduces other problems such as fractures, as seen in **Table 25**.

Table 25. Functional candidates found for “Raloxifene reduces breast cancer”

Evidence	SCP
Osteoporotic fractures	0.006
Bone resorption	0.004
Fracture risk	0.002
Vertebral fracture	0.002
Bone turnover	0.001
Fractures	0.001
<i>Markers</i>	<i>1.39E-4</i>
<i>Studies</i>	<i>1.35E-4</i>
<i>Increase</i>	<i>9.76E-5</i>
<i>Amount</i>	<i>2.99E-5</i>

Chichen Itza is known as El Castillo: Chichen Itza is only known as El Castillo. Our algorithm found as candidate “La Iglesia” from a sentence “The Church in Chichen Itza is known as La Iglesia”, indicating that the church within Chichen Itza is known as La Iglesia, therefore the decision taken by the algorithm is correct, as seen in **Table 26**.

Table 26. Functional candidates found for “Chichen Itza is known as El Castillo”

Evidence	SCP
La Iglesia	8.25E-5
Home	7.092E-7

5.2.4 Inverse Object Property Evaluation

The discovery of Inverse relations for an Object Property has been tested with 10 relations, presented in **Table 27**. The table shows the best extraction (i.e., the best candidate to inverse relation) found with the $?(object, subject)$ pattern. The algorithm takes the relation to have an inverse if the best extraction has an SCP (SCP P column) higher than the established $1E-3$ threshold.

Table 27. Discovery of Inverse relations for several Object Properties

Original relation	NC	BES	SCP O	SCP P	D	HD
Christopher Columbus discovered America	5	was discovered by	0.004	0.017	Y	Y
Market regulates capitalist production	1	is attached to	1.8E-3	5.8E-5	N	N
Global warming is caused by pollution	6	Causes	0.11	0.06	Y	Y
Cigarette smoking is associated with lung cancer	1	is caused by	0.007	0.002	Y	Y
Sensors contain mechanical parts	1	required for	4.7E-3	7.1E-4	N	N
Viruses are associated with liver cancer	2	caused by	5.2E-3	0.003	Y	Y
Kasparov defeated Deep Blue	16	beaten by	0.003	0.001	Y	Y
Stress affects asthma	3	is linked to	0.02	0.025	Y	Y
Meningococcal disease is caused by Neisseria Meningitidis	1	Causes	0.03	0.002	Y	Y
Charles Darwin proposed evolution theory	7	jointly proposed by	0.002	0.001	Y	Y

Global warming is caused by pollution: The best candidate for the inverse relation was found correctly. It's possible to find candidates that in fact don't represent the inverse relation, such as "could fight". Also, the algorithm found "to slow" as a possible verb phrase candidate in order to define the inverse property, so the results depend on a good natural language processing analysis and may lead to a wrong decision from the algorithm.

Table 28. Inverse candidates found for "Global warming is caused by pollution"

Evidence	SCP
Causes	0.111
May boost	0.100
Is driving	0.100
Could fight	0.071
To slow	0.071
Enhanced by	0.022

Charles Darwin proposed evolution theory: The example shows that it is possible to extract incomplete verb phrases. For example, we found evidences for "evolution theory proposed by Charles Darwin", whereas "was proposed by" would be a more complete verb phrase for that relation. Check that some valid candidates received a very low score such as "Suggested by" or "Developed by", while "came from" received particularly a good SCP score.

Table 29. Inverse candidates found for “Charles Darwin proposed evolution theory”

Evidence	SCP
Proposed by	0.002
Came from	0.001
Was founded by	0.001
Suggested by	7.74E-4
Made by	6.24E-4
Developed by	3.80E-4
Published by	3.38E-4

5.2.5 Inverse Functional Object Property Evaluation

The Inverse Functionality of an Object Property has been tested with 5 of the “inverse” relations shown in the previous table. **Table 30** shows the best extraction found with the **RELATION(SUBJECT, ?)** pattern. The algorithm takes the original relation to be inverse functional if the best extraction has an SCP (SCP P column) higher than the established 1E-3 threshold. The algorithm agrees with the human result in 4/5 cases.

Table 30. Test of the Inverse Functionality of some Object Properties

Original relation	NC	BES	SCP O	SCP P	D	HD
Pollution causes global warming	76	acid rain	0.066	0.013	N	N
Asthma is linked to stress	67	allergies	0.026	0.108	N	N
Neisseria Meningitidis causes meningococcal disease	2	inflammation	0.0022	3.08E-5	Y	Y
Evolution theory proposed by Charles Darwin	1	Darwin	0.001	0.001	N	Y
Lung cancer is caused by cigarette smoking	57	second hand smoke	0.003	0.0028	N	N

Evolution theory proposed by Charles Darwin: In the example that states “Evolution theory proposed by Charles Darwin”, notice that a selected extraction is “Darwin”. Both expressions refer to the same individual, but with the algorithm proposed this fact is not taken into account concluding in a wrong decision.

Pollution causes global warming: The algorithm found many consequences as result of the pollution, including acid rain, cancer, stress and many others as seen in **Table 31**. Taking this into account, the decision of our algorithm is correct, as the decision of the expert.

Table 31. Inverse Functional candidates found for “Pollution causes global warming”

Evidence	SCP
Acid rain	0.013
Disease	0.012
Cancer	0.011
Stress	0.009
Health problems	0.006
Heart attacks	0.001
Asthma attacks	0.001
Kidney damage	3.99E-4
Baldness	2.09E-4
Male infertility	5.79E-6
Female snails	5.59E-6

Asthma is linked to stress: As shown in **Table 32**, asthma is not only linked to stress, but it is also linked to pollution, obesity, respiratory tract infections and many others. Taking this into account, the decision of our algorithm is correct.

Table 32. Inverse Functional candidates found for “Asthma is linked to stress”

Evidence	SCP
Allergies	0.108
Obesity	0.06
Antibiotics	0.03
Air pollution	0.03
Pollution	0.026
Weight	0.021
Cleaning products	0.005
Respiratory tract infections	0.003
Breastfeeding	8.14E-4
Acetaminophen use	4.44E-4
Poor indoor air quality	4.57E-4
Paracetamol use	9.87E-5
First-born children	6.88E-6

Neisseria Meningitidis causes meningococcal disease: This bacterium is the only source of the meningococcal disease. As seen in **Table 33**, these candidates (inflammation and whooping cough) are symptoms of the meningococcal disease. Our algorithm didn’t find a strong co-occurrence of this candidates and the bacterium, so it decided that this relation is inverse functional, being correct.

Table 33. Inverse Functional candidates found for “Neisseria Meningitides causes meningococcal disease”

Evidence	SCP
<i>Inflammation</i>	<i>3.08E-5</i>
<i>Whooping cough</i>	<i>2.08E-5</i>

5.2.6 Transitive Object Property Evaluation

To test the Transitivity of an Object Property we have followed two steps. Given a relation xRy , in the first step we look for individuals related to y via R , using the pattern $R(y,?)$. In the second step, we take the best extraction in each of the cases (z), and we try to evaluate whether xRz holds.

Table 34a shows the five relationships used in this test, along with the best extraction found in each case. An extraction is taken as correct by the algorithm if its SCP is above the 1E-3 threshold. From the first four relations given in **Table 34a**, we evaluate in **Table 34b** the best extraction (z) to see if $R(x,z)$ holds. If it does, the algorithm takes the relation to be transitive.

This way to evaluate the transitivity of a given relation presents a critical limitation. Considering that the evaluation of the transitive property in our approach includes two steps, the possibility to find transitivity for a relation is difficult. However, if a transitivity property is found using this approach that is due to the fact that the evaluated relatedness is certainly strong, as in the relations “Berlin is located in Europe” or “Microsoft competes with Google”.

Tables 34a and 34b. Two steps in the test of Transitivity for some Object Properties

Original relation	NC	BES	SCP O	SCP P	D	HD
Global warming is caused by pollution	54	Nature	0.11	0.05	Y	Y
Berlin is located in Germany	12	Europe	0.04	0.07	Y	Y
Pollution affects global warming	46	Weather	0.038	0.05	Y	Y
Microsoft competes with Apple	30	Google	0.29	0.18	Y	Y
Arsenic is associated with kidney cancer	5	Obesity	8.1E-5	4.3E-5	N	Y

Relation	SCP O	SCP P	D	HD
Global warming is caused by Nature	0.11	0.05	Y	Y
Berlin is located in Europe	0.04	0.02	Y	Y
Pollution affects weather	0.038	0.02	Y	Y
Microsoft competes with Google	0.29	0.31	Y	Y

Global warming is caused by pollution: Table 34 shows some examples of the candidates extracted, concluding that there are many factors by which the pollution is caused. In its first step, this extractions come from the query [“Pollution is caused by” –“Pollution is caused by Global Warming”], illustrated in Table 33a. The best candidate extracted is “Nature” and to detect the transitivity of the original relation we search for “Global warming is caused by Nature” in Table 33b.

Table 35. Transitive candidates found for “Global warming is caused by pollution” in the first step

Evidence	SCP
Nature	0.05
Rainfall	0.047
Cars	0.03
Contaminants	0.002
Carbon dioxide	0.004
<i>Waste gases</i>	<i>1.29E-4</i>
<i>Private automobiles</i>	<i>2.67E-4</i>
<i>Airplanes</i>	<i>3.36E-4</i>
<i>Maritime transport</i>	<i>3.4E-4</i>
<i>Various industrials</i>	<i>3.58E-5</i>
<i>Loud noise</i>	<i>8.19E-5</i>

Berlin is located in Germany: Table 36 shows that Germany is located in Europe from the query [“Germany is located in” –“Germany is located in Berlin”] in its first step, illustrated in Table 34a. The best candidate extracted is Europe, therefore in Table 34b we look for the transitivity. A problem can arrive when detecting exactly in what part of Europe Germany is located. For example, if we have modeled in our ontology how Europe is divided (Central Europe, West Europe and so on), an expert will not allow the representation that Berlin is located in Central Europe and West-Central Europe.

Table 36. Transitive candidates found for “Berlin is located in Germany” in the first step

Evidence	SCP
Europe	0.073
Central Europe	0.001
West-Central Europe	2.9E-2
<i>Bruschal</i>	<i>7.47E-5</i>
<i>North central Europe</i>	<i>2.77E-5</i>
<i>North-central Europe</i>	<i>2.32E-5</i>
<i>Graitschen</i>	<i>1.99E-7</i>

Microsoft competes with Apple: Table 37 shows that Apple has many competitors in the technology industry, therefore the decision of our algorithm is correct. For the original relation in Table 34a, we used the query [“Apple competes with” – “Apple competes with Microsoft”]. The best candidate extracted is “Google” and in Table 34b we look in the second step the transitivity by using this extraction.

Table 37. Transitive candidates found for “Microsoft competes with Apple” in the first step

Evidence	SCP
Google	0.188
Dell	0.148
Intel	0.09
Yahoo	0.071
IBM	0.068
HP	0.061
Sandisk	0.013
<i>Different companies</i>	<i>2.13E-4</i>
<i>Cheaps pcs</i>	<i>1.7E-4</i>
<i>Small developers</i>	<i>7.72E-5</i>
<i>Google docs</i>	<i>1.93E-6</i>

5.3 Conclusions

The manual evaluation performed by the Discounted Cumulative Gain indicated us that for our purpose, the Symmetric Conditional Probability (SCP) globally is the best co-occurrence measure from the 3 presented and analysed in our work in terms of better ranking.

From the manual evaluation of the results, we obtained very promising results from the different relations analysed, all of them over 70% of effectiveness. Evaluating the object property axioms studied in this work individually we can conclude the following:

- Symmetric / Assymetric: From 10 relations analysed, our algorithm obtained 9 good decisions. A limitation regarding our algorithm for this object property axiom was found in the relation “Deep Blue defeated Gary Kasparov” where in fact Gary Kasparov defeated Deep Blue, but in another game.
- Reflexive: 8 out of 10 decisions were right comparing the algorithm decision against the expert decision. A limitation of our algorithm for reflexive axioms is the fact that “ITSELF” is a very common word in any corpus. If *SUBJECT* and *RELATION* are very common (e.g “people” and “have”), wrong decisions may be taken by the algorithm.

- Functional: 8 out of 10 decisions were right, comparing the algorithm decision against the expert decision. A limitation of our approach consists of verb phrases that may be used for time / place subjects (e.g. “was born in” may extract information about a time as for example month, day or year; and places).
- Inverse: All the decisions were right. However, a limitation of our algorithm is that the verb phrase extracted may not be in fact the inverse relation of the original. Also, not necessarily complete verb phrases can be extracted by the algorithm (e.g. “proposed by” instead of “is/was proposed by”).
- Inverse Functionality: From the 5 relations studied, only 1 relation was wrong. Another limitation of the algorithm proposed is the extraction of same individuals, for example as found in our evaluation “Darwin” and “Charles Darwin”.
- Transitive: We obtained 4 right decisions out of 5. The algorithm applied for the transitive object property is very restrictive because we need to apply 2 steps, but as seen in the relations presented, the transitivity found in 4 relations are very strong.

Section 6

Conclusions and Future Work

Up to this moment, we have described in detail all the learning methodologies applied in our work, its implementation, the results generated by our algorithm and the evaluations performed with an expert for every object property axiom studied. In this final section, we provide a final summary and present the conclusions from a general point of view of the methodology presented in this work and we suggest several lines of future research about different open issues presented in previous sections and give some ideas on how they can be tackled.

6.1 Summary & Conclusions

Axiomatizing ontologies by hand is a hard and time-consuming task which involves ontology engineers and domain experts. The design of automatic axiomatization solutions is fundamental to provide a powerful knowledge representation that will contribute to the success of the Semantic Web and many other fields in Computer Science. However, most of the works related to ontology learning focus their approach in taxonomic and non-taxonomic relations learning, which results in lightweight ontologies with low degrees of expressivity and capability of inference.

The approach that we have proposed is a step forward to fill this gap, by learning axioms for non-taxonomic relations using the Web as corpus, shallow linguistic techniques and Web-scale statistics. The Web can be exploited to aid in the Ontology Learning process by using suitable queries in order to retrieve valuable information. The patterns used in this work are totally generic and can be easily applied to any Web search engine and to any domain. By using natural language processing tools we can extract valuable information retrieved from Web documents, as different syntactic categories in natural language are related to an ontology structure and therefore use co-occurrence measures to verify if the extractions from the Web documents based on a pattern are in fact, related to the relation analysed to determine if an axiom fulfils or not.

Considering the developed methodologies and the evaluated and obtained results, we can conclude the following:

- The web is a valid corpus as it fulfils the four main headings as stated in [36], which are sampling and representativeness, finite size, machine-readable form and a standard reference which are the main considerations to consider a repository as a valid corpus. The Web is the biggest repository of information and its high redundancy can represent a measure of its relevance.
- Pattern-based approaches have been widely used in Information Retrieval. Several patterns have been used in the ontology learning process, and most of them are focused on taxonomic and non-taxonomic relations learning processes. In this work we extend this process by adding generic and domain-independent patterns to improve the expressiveness of ontologies in an automatic fashion way.
- Together with the set of text patterns presented and Web search query construction, the Natural Language Processing and Web-scale statistics are the main steps in our approach. By using NLP tools we are able to analyse syntactically unstructured text and extract valuable information to determine the fulfillment or not of an axiom, and by using Web-scale statistics we can measure the semantic relatedness of this extracted information and the relation under review.
- The evaluations performed for several domains showed very promising results for all the object property axioms studied in this work and are a step forward over the tedious, time-consuming and error prone manual approaches.

6.2 Future Work

The proposed algorithm presents several limitations that can lead to the definition of lines of future work. Some of these shortcomings are the following:

- At the moment we study only the axioms of a particular relationship between two individuals, xRy , e.g. to test the symmetry, we check whether yRx holds. We don't cover the study of the mathematical properties of the R relation in general; for example, to guarantee the symmetry of R , we should check whether, for all x,y such that xRy holds, yRx also holds.

- Regarding the reflexive property, the search for “*xR itself*” may be too restrictive and hard to find explicitly. We can add to the query the words “may” and “can” in order to extend the reflexive property discovery in the form of *Subject may/can Relation itself*.
- The transitive property is checked only with a particular instance. Note that “Italy borders Switzerland” will be qualified as transitive depending on the country which is the best extraction for the “Switzerland borders” pattern (giving different results e.g. for France and Germany, as one borders Italy and the other doesn’t). Thus, we should at least check whether all the neighbours of Switzerland are also neighbours of Italy.
- We can evaluate the results with more specific domains, including domains such as nuclear physics, where less quantity of documents are available on the Internet in order to check if the co-occurrence calculation average for the SCP remains the same for the threshold set.

It is also necessary to study more deeply the effect of many natural language phenomena such as synonyms and polisemy in order to provide better results in certain domains in which these ambiguities are common. This may lead to further research in composing a more concrete set of patterns or adding more constraints in their extraction and selection.

References

1. Sánchez, D., Domain Ontology Learning from the Web. VDM Verlag Dr. Mueller Akt.ges. & Co.KG.ISBN: 386470691 (2008)
2. Cimiano, P., Ontology Learning and Population from Text, Springer Science, 1-2 (2006)
3. Speel, P-H., Schreiber A. Th., van Joolingen W., van Heijst G., Beijer G.J., Conceptual Modelling for Knowledge-Based Systems, University of Amsterdam (2001)
4. Volker J., Haase P., Hitzler P., Learning Expressive Ontologies, in: Proc. of the Conference on Ontology Learning and Population Germany, 45-69 (2008)
5. Sánchez, D., Moreno, A., Pattern-based automatic taxonomy learning from the Web, AI Communications, 21(1) 27-48 (2008)
6. Sánchez, D., Moreno, A., Learning non-taxonomic relationships from web documents for domain ontology construction, Data and Knowledge Engineering 63(3) 600-623 (2008)
7. Volker J., Hitzler P., Cimiano P., Acquisition of OWL DL Axioms from Lexical Resources, in: Proc. of the 4th European conference on The Semantic Web, Germany, 670-685 (2007)
8. S. Rudolph, Exploring relational structures via FLE, in: Wolff, K.E., Pfeiffer, H.D., Delugach, H.S., eds.: Conceptual Structures at Work, 12th International Conference on Conceptual Structures, volume 3127 of LNCS, Springer, pp. 196–212, 2004.
9. Rudolph S., Volker J., Hitzler P., Supporting Lexical Ontology Learning by Relational Exploration, in: Proc. of the 15th International Conference on Conceptual Structures ICCS, Germany, 488-491 (2007)
10. Volker J., Vrande D., Sure Y. and Hotho A., Learning Disjointness, in: Proc. of the 4th European conference on The Semantic Web, Germany, 175-189 (2007)
11. Ritter A., Downey D., Soderland S., Etzioni O., It's a Contradiction-No it's not: A Case study using Functional Relations, in: Proc. of Conference on Empirical Methods in Natural Language Processing, University of Washington, 11-20 (2008)
12. Shamsfard M., Abdollahzadeh A., Learning Ontologies from Natural Language Texts, in: Proc. of the International Journal of Human-Computer Studies, International Journal of Human-Computer Studies, Volume 60, 17-63 (2004)
13. Studer, R., Benjamins, V.R. and Fensel., D.: Knowledge Engineering: Principles and Methods. IEEE Transactions on Knowledge and Data Engineering 25(1-2), 161-197 (1998)

14. 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.
15. 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.
16. Uschold, M. and Gruninger, M.: *Ontologies. Principles, Methods and Applications*. *Knowledge Engineering Review* 11(2), 1996. 93-155.
17. Hearts M., Automatic Acquisition of Hyponyms from Large Text Corpora: In *Proc. of the 14th conference on Computational linguistics - Volume 2*, (1992)
18. Berland M., Charniak E., Finding Parts in Very Large Corpora: In *Proc. of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, Brown University (1999)
19. Cimiano P., Wenderoth J., Automatically Learning Qualia Structures from the Web: In *Proc. of the ACL-SIGLEX Workshop on Deep Lexical Acquisition*, University of Karlsruhe, Germany (2005)
20. Girju R., Moldovan D., Text Mining for Causal Relations: In *Proc of the FLAIRS Conference*, 360-364 (2002)
21. Grefenstette, G., Finding Semantic Similarity in Raw Text: The Deese Antonyms: In *Proc. of Working Notes of the AAAI Fall Symposium on Probabilistic Approaches to Natural Language*, AAAI Press (1992)
22. Lee, J.H., Kim, M.H. and Lee, Y.J., Information Retrieval Based on Conceptual Distance in ISA Hierarchies, *Journal of Documentation*, 1993.
23. Jiang J. and Conrath, D., Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy: In *Proc. of the 10th International Conference on Research on Computational Linguistics*. Taiwan. (1997)
24. Brill E., Processing Natural Language without Natural Language Processing: In *Proc. of CICLing*, (2003)
25. Turney P.D.: Mining the Web for synonyms: PMI-IR versus LSA on TOEFL: In *Proc. of the Twelfth European Conference on Machine Learning*. Freiburg, Germany (2001)
26. Bouma G., Normalized (Pointwise) Mutual Information in Collocation Extraction, *Department Linguistik, Universitat Postdam*
27. Dias G., Santos C., Cleuzious G., Automatic Knowledge Representation using a Graph-based Algorithm for Language-Independent Lexical Chaining: In *Proc. of the Workshop on Information Extraction Beyond The Document*, Sydney (2006)
28. Cilibrasi R., Vitanyi P., Automatic Meaning Discovery Using Google: In *Proc. of the National ICT of Australia*, Sydney (2005)
29. Gligorov R., Aleksovski Z., ten Kate W., van Harmelen F., Using Google Distance to Weight Approximate Ontology Matches: In *Proc. of the International World Wide Web Conference Committee (IW3C2)*, Canada (2007)
30. Schutz A., Buitelaar P., A Tool for Relation Extraction in Ontology Extension. In: *4th International Semantic Web Conference* (2005)
31. Grefenstette G., SQLET: Short Query Linguistic Expansion Techniques: Palliating One-Word Queries by Providing Intermediate Structure to Text, In *Proc. of Information Extraction: A Multidisciplinary Approach to an Emerging*

- Information Technology, LNAI 1299, International Summer School, SCIE-97. Italy, 97-114 (1997)
32. 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)
 33. Ciravegna F., Dingli, A., Guthrie, D. and Wilks, Y: Integrating Information to Bootstrap Information Extraction from Web Sites, In Proc. of the IJCAI Workshop on Information Integration on the Web, 9-14 (2003)
 34. Etzioni O., Cafarella M., Downey, D. Kok, S. Popescu, A. Shaked, T. Soderland, S. and Weld D.S., *WebScale Information Extraction in KnowItAll*, In Proc. of WWW2004, New York, USA (2004)
 35. Navigli R. and Velardi, *Learning Domain Ontologies from Document Warehouses and Dedicated Web Sites*, In Proc. of Computational Linguistics 30(2), 151-179 (2004)
 36. Kilgariff A., Grefenstette G., *Introduction to the Special Issue on the Web as Corpus*, University of Brighton, UK, 333 – 347 (2003)
 37. Dujmovic J. and Bai, *Evaluation and Comparison of Search Engines Using the LSP Method*, ComSIS 3(2), 711-722 (2006)
 38. Porter M.F., *An algorithm for suffix stripping*, 130 – 137 (1980)
 39. Lemaire B. and Denhiere Guy, *Effects on High-Order Co-occurrences on Word Semantic Similarity*, 18, Volume 1 (2006)
 40. Fellbaum C., *Wordnet, an electronical lexical database*, MIT Press (1998)
 41. Guarino N., Welty C. A., *A formal ontology of properties*. In Proc. of Knowledge Acquisition, Modeling and Management, 97 – 112 (2000)
 42. Yang Y., *An evaluation of statistical approaches to text categorization*. In Proc. of Journal of Information Retrieval, 1 (1-2), 67-88 (1999)

Annex I. Article

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

Conference: The *KSEM-10* is an international conference for research on knowledge science, engineering and management, which attracts high quality, state-of-the-art research papers from all over the world. The conference will bring together researchers, practitioners, developers, and academic and industry leaders who are active and interested in KSEM-related research areas and their practices. The conference is specifically focused on becoming a premier forum for prototype and deployed knowledge engineering and knowledge-based systems.

Discovery of relation axioms from the Web

Luis Del Vasto Terrientes, Antonio Moreno, David Sánchez

Universitat Rovira i Virgili. Departament d'Enginyeria Informàtica i Matemàtiques
Intelligent Technologies for Advanced Knowledge Acquisition (ITAKA)
Av Països Catalans, 26. 43007 Tarragona, Catalonia (Spain)
{luismiguel.delvasto, antonio.moreno, david.sanchez}@urv.cat

Abstract. Given the proven usefulness of ontologies in many areas, the representation of logical axioms associated to ontological concepts and relations has become an important task in order to create an expressive representation of domain knowledge. Manual inclusions of logical axioms into an ontology can be a harsh, time consuming task. As a result, very few ontologies include axioms in their formal definition. From the ontology learning point of view, axiom learning is one of the less tackled and unexplored problems. In this paper we introduce a preliminary methodology to learn axioms associated to ontological relationships in an automatic and unsupervised way using the Web as corpus.

Keywords: Ontologies, Axioms, Knowledge Discovery, Semantic Web

1 Introduction

An *ontology* represents the conceptual model underlying a certain domain, describing it in a declarative fashion and thus cleanly separating it from procedural aspects [2]. Ontologies play a key role in the Semantic Web [3] and other areas like agent communication, intelligent information integration and knowledge-based systems. Research in this area has increased considerably during the last years, bringing the development of modern ontological languages such as OWL 2, which provides a powerful formalism for knowledge representation and reasoning.

Ontologies are typically built by hand, requiring an arduous and time-consuming effort both from knowledge engineers and domain experts. As a result of the bottleneck introduced by knowledge acquisition approaches, manually composed ontologies are, in many cases, flat and lightweight, with very limited expressivity and mainly considering taxonomical knowledge and simple relations [4, 5]. So, available ontologies very rarely exploit the expressiveness of languages like OWL 2 in defining logical constraints (axioms) associated to relations.

Ontology Learning methods, that extract automatically ontological elements from available corpora and build ontologies from them, mostly focus on taxonomic relationships [5] and, more rarely, on non-taxonomic knowledge [6]. Relation axioms involving logical connectives, role restrictions and other semantic features that can provide expressivity to ontologies remain largely unexplored [4].

This paper presents an automatic axiom learning algorithm which, starting from a set of non-taxonomic relations (denoted xRy or $R(x,y)$, in which x and y represent concepts or individuals and R the relation/property) defined in an input ontology, explores and verifies which axioms those relations fulfill. Specifically, the axioms studied in our approach are *symmetry*, *reflexivity*, *functionality*, *transitivity* and *inverse*. Our approach relies on linguistic techniques and statistical analyses to unsupervisedly and automatically extract and evaluate semantic evidences found in Web resources [1], which support the definition of ontological axioms.

The paper is structured as follows. Section 2 presents the related works dealing with automatic ontology axiomatization. Section 3 describes our approach in detail, explaining the different stages of the learning process and the techniques used. Section 4 explains the criteria and measures considered to evaluate the algorithm and provides some preliminary results. Finally, the last section presents conclusions and some lines of future research.

2 Related Work

Several natural language processes have been applied to acquire OWL axioms from lexical resources in tools such as LExO (that stands for Learning Expressive Ontologies), RELExO (Relational Exploration) and HASTI, among others.

The implementation of LExO basically relies on KAON2, an ontology management infrastructure for OWL, and the Minipar dependency parser [8]. Given a natural language definition of a class, LExO starts by analyzing the syntactic structure of the input sentence. The resulting dependency tree is then transformed into a set of OWL axioms (concept inclusion, transitivity, role inclusion, role assertions, concept

assertions and individual equalities) by means of manually engineered transformation rules [4]. The axiomatization is a semi-automatic cyclic process [7], in which each iteration contains an ontology evaluation step to make sure that the axiom candidates are correct. The main problem raised by this approach are the semantic inconsistencies caused by the analysis of natural language expressions. To tackle this problem the relational exploration [15] method was used in the development of the RELExO application. This technique is an adaptation of attribute exploration from Formal Concept Analysis (FCA) to description logics, and is a good mean to overcome the lack of completeness and precision in learned ontologies. It is natural to apply it in order to further specify the knowledge beyond the information extractable from the corpus, which makes relational exploration a perfect complement to automatic approaches for ontology generation based on lexical resources [9]. Another automated attempt was performed in [10], in which lexico-syntactic patterns are used to extract individuals and subclasses, and the OntoClean methodology is used to obtain evidence of strict disjointness between two classes. Pattern-extracted classes are analysed and, if between two classes there aren't any common individuals or subclasses that taxonomically overlap, they are taken to be disjoint. In the OntoClean methodology, a pattern-based approach was used to analyze text taken from the Web specifically to verify that two classes have incompatible unity or identity criteria. Other patterns based in enumerations, defined in [9] for detecting disjointness, were not precise enough in comparison with average human disjointness detection.

HASTI is a system for automatic ontology construction, based on Persian (Farsi) text understanding. It uses an initial kernel, which has the essential metaknowledge (primitive concepts and operators) to build an ontology including axioms [12]. HASTI implements a rule base which contains linguistic rules, inference rules and heuristics to extract lexical elements from texts. Specifically, it contains Semantic Templates to extract taxonomic relations, non-taxonomic relations and axioms. It extracts axioms (disjointness, transitivity and functionality) from conditional and quantified sentences, using axiom templates written in Knowledge Interchange Format (KIF). Following the lexico-syntactic approach, AuCONTRAIRE is a Contradiction Detection (CD) system which relies on the TextRunner system, an open Information Extraction system that uses the Web as corpus to obtain relations between entities [11]. In order to detect functional relations, it checks if a domain and its relation map to a unique variable using probabilistic assessment based on aggregated textual evidence. The fact that a CD system cannot assume that it knows all the functional relations makes determining when a possible candidate for a relation is or not valid an important task to handle natural language phenomena such as synonyms, meronyms and ambiguity.

The lack of automatic and unsupervised approaches to axiomatize ontologies motivated us to aim our efforts into this area. On the contrary to related works which are applied in a narrow context (i.e. a predefined document or even a sentence), our approach uses the Web as a massive learning corpus, enabling the discovery of relation axioms in a domain-independent fashion. The size and heterogeneity of the Web minimizes data sparseness problems in comparison to approaches like HASTI, in which the extraction of text documents in order to detect conditional or quantified sentences may be a barrier to obtain complete information about the behaviour of the concepts and roles. Unlike approaches such as LExO and RELExO, our approach

does not require manual stages which, considering the size of the Web, would compromise its scalability.

3 Object Property Axioms Learning Methodology

Object properties define relations between pairs of ontological concepts. In contrast to hierarchical relations, an object property represents a non-taxonomic relation, which is typically expressed by a verb that relates a pair of concepts [12]. The definition of *object property axioms* allows the meaning of properties to be enriched and, therefore, the possibility to apply inference on them.

The pseudocode of the algorithm we propose is presented below, in which a domain ontology is received as input and a set of axioms associated to ontological relationships defined via object properties is obtained as output. During the explanation of the method, we refer to algorithm lines as *<line>*.

```

1. AxiomLearning (Domain Ontology DO)
2. { AXS_OP /* Results variable */
3.   AXIOMS /* Constant holding the set of axioms of interest */
4.   THRESHOLD /* Constant holding the selection threshold */
5.   RELS ← get_nontaxonomic_relations(DO)
6.   /* for each relation, analyse each object property axiom */
7.   for all RELSi ∈ RELS
8.     { for all AXIOMSj ∈ AXIOMS
9.       { QUERY ← generate_query(RELSi, AXIOMSj)
10.      WEBPAGES ← retrieve_WebPages(QUERY)
11.      for all WEBPAGESx ∈ WEBPAGES
12.        { /* for each web page check for matchings */
13.          TEXT ← extract_matchings(QUERY, WEBPAGESx)
14.          TEXT ← remove_stopwords(TEXT)
15.          KEY ← stem_word(TEXT)
16.          exists ← check_existence(AXS_OP, RELSi, AXIOMSj, KEY)
17.          if (!exists)
18.            { /* calculate web scale statistics */
19.              SCP = calculate_SCP(AXIOMSj, RELSi, TEXT)
20.              /* add info. to result if SCP > THRESHOLD */
21.              if (SCP > THRESHOLD)
22.                AXS_OP ← AXS_OP + (RELSi ∪ AXIOMSj ∪ KEY ∪ TEXT ∪ SCP)
23.            }
24.          }
25.        }
26.      }
27.   Return AXS_OP
28. }
```

3.1 Pattern construction

Lexico-syntactic patterns, as tools to discover and extract ontological entities, have been used in order to learn ontologies from unstructured text. Our approach bases the axiom learning in the exploitation of specially designed patterns aimed to retrieve (by

means of querying a Web search engine) and extract (by means of a linguistic analysis) semantic evidences that support the fulfillment or not of each one of the studied Object Property Axioms. The patterns proposed below are domain-independent and can be adjusted to queries in many Web search engines. In base of these patterns, we have designed queries for the Yahoo! and Bing Web search engines <9>.

To describe the proposed patterns we use the nomenclature *Relation(Subject, Object)* in which *Relation* represents a relation and *Subject/Object* represent individuals. As an example, in a non-taxonomic relation defined as Borders(Spain, France), Spain is the *Subject*, France is the *Object* and Borders represents the *Relation*. In the following paragraphs the proposed patterns and their use to check some properties are described.

Relation (Subject, ?)

Functional Object Property & Inverse Functional Object Property: Given a relation xRy , it is possible to detect if the subject x can be related to another object different from y by the relation R to check the *Functionality* of the Object Property xRy and the *Inverse Functionality* of the Object Property $xR'y$ (R' being the inverse of R). In the query language we denote this pattern as $["x R" - "x R y"]$, in which $["x R"]$ is used to extract new objects while $[- "x R y"]$ avoids the extraction of the initial object y for the relation.

Transitive Object Property: Given a relation xRy , to detect the *Transitivity* of the property for the subject x , we need first to detect if the object y can be related to another individual z by relation R . In this way we can detect a relation yRz in which the relation R is the same for both xRy and yRz . In query language we denote this pattern as $["y R" - "y R x"]$. $["y R"]$ may help us to detect if y can be related to an object z by relation R , while $[- "y R x"]$ avoids the extraction of the original subject x in xRy as object.

Relation (Subject, Object)

Symmetric Object Property: Given a relation xRy , by switching the positions of the subject x and the object y we can check if the relation R can be applied from y to x as well, denoted in a web search query as $["y R x"]$.

Reflexive Object Property: Given a relation xRy , to check its reflexivity we used as object the reflexive pronoun 'itself', which is used to refer to things or animals. In query language we denote this pattern as $["x R \text{ itself}"]$.

Transitive Object Property: As said in the previous pattern Relation (Subject, ?), we can detect yRz from an initial relation xRy . The second step uses this pattern in order to check that the subject in xRy can be related to the object in yRz . In query language we denote this pattern as $["x R z"]$. In basic terms, we used this pattern to detect if it is possible to find evidence in the Web of a concrete relationship between two specific individuals.

? (Object, Subject)

Inverse Object Property: Given a relation xRy , we can search for an inverse relationship by looking for a verb that relates y to x . In query language we denoted this pattern as $["y * x" \text{ OR } "y * * x" \text{ OR } "y * * * x"]$. Note that the wildcard (*), which is used in the query in order to force the proximity (up to three words) between the subject and the object, has to correspond to a verb phrase.

3.2 Extraction

Applying the described patterns over a given relation and querying the resulting expression over a Web search engine, we are able to retrieve Web resources including one or several matchings <10>. The text is analysed in order to discover and extract the semantic evidences that would lead to the definition or not of the corresponding axiom. In order to minimize noise and ambiguity of textual processing, we imposed some syntactic boundaries depending on the used pattern:

- **Relation (Subject, ?)** and **Relation (Subject, Object)**. For these patterns we decided to get rid of interrogative sentences, cardinal candidates and candidates that are superclasses of the initial object. Words between parentheses are deleted.
- **? (Subject, Object)**. For this pattern, we neglected interrogative sentences, cardinal candidates, and verbs in future or gerund tense. Candidates between parentheses are also deleted.

If a text piece fulfills the conditions defined for the particular pattern, then the evidence is extracted <13> for further analysis.

3.3 Evaluating extractions and axiom definition

Applying the mentioned constraints to avoid natural language problems does not mean that the observations extracted for a certain relation are strong enough to grant the definition of an axiom. The next step consists on deciding which of the extractions are reliable and sufficiently related to the property. To perform this selection process we performed a Web-based statistical analysis relying on co-occurrence measures computed directly from Web search engines <23>. Co-occurrence measures are based on distributional hypothesis claiming that words that occur in the same context tend to have similar meanings.

Several scores have been proposed in the past to compute Web-scale statistics, adapting the notion of collocation and mutual information (computed as the probability of joint appearance of concepts in a corpus). Applied to the Web, probabilities (p) are estimated as the hit count (*hits*) returned by a Web search engine when querying the term. Concretely, we used the Symmetric Conditional Probability (SCP) [13] computed from the Web (1) to evaluate the relatedness between two entities.

$$SCP(x, y) = \frac{p(x, y)^2}{p(x) * p(y)} \approx \frac{\left(\frac{\text{hits}("x" \text{ AND } "y")}{\text{total_web_sites}} \right)^2}{\frac{\text{hits}("x")}{\text{total_web_sites}} * \frac{\text{hits}("y")}{\text{total_web_sites}}} = \frac{\text{hits}("x" \text{ AND } "y")^2}{\text{hits}("x") * \text{hits}("y")} \quad (1)$$

To evaluate the relevance of the extracted terms, we consider a *discriminator phrase* that represents the relation as a whole. The extracted term is the *instance* together with the relation that links it to another individual and the *discriminator phrase without instance* represents the relation without the extracted *instance*. In this way, when evaluating if the relation xRy “Spain borders France” is functional when a new object “Portugal” is retrieved, we can denote from (1):

p(x,y) - *Discriminator Phrase*: “Spain” AND “borders” AND “Portugal”

p(x) - *Instance*: “borders” AND “Portugal”

p(y) - *Discriminator Phrase without Instance*: “Spain” AND “borders”

The Web-based statistical evaluation process for the reflexive, symmetric and transitive (in its second step) properties, implies the assessment of the relative relatedness of the extracted evidences (computed by means of the SCP) with respect to a threshold. If the SCP of the extracted evidence is higher than the threshold, we may conclude that the relation is reflexive, symmetric or transitive (in its second step) <21, 22>. For the functional, inverse, inverse functional and transitive (in its first step) object properties, if the set of extractions contains an item with a co-occurrence higher than the threshold then we may conclude that the extractions are robust for the relation. Depending of these observations, an axiom may be fulfilled or not. In the functional and inverse functional case, if the extractions are robust then we may conclude that the relation is not functional or inverse functional <26, 27>. For the inverse axiom, we may conclude that an inverse relation has been found. In the first step of the transitivity test, we conclude that a good candidate to check for the transitivity in the second step has been found <21, 22>.

4 Empirical evaluation

Due to the difficulty to carry out automatic evaluations of automatically acquired knowledge, we based them in a manual approach in which a human determines if a relation fulfills a certain axiom. A number of positive and negative relation examples of each of the studied axioms are presented, along with the results given by our methodology. In the tables shown in this section the following abbreviations are used:

- SCP O/P: SCP values of the original relation and those obtained for the best extraction from the constructed pattern.
- D: Decision of the algorithm concerning the evaluated axiom relation.
- HD: Decision of a human concerning the evaluated axiom relation.
- NC: Number of extractions for a given axiom and relation.
- BES: Best extraction selected (the one with the highest SCP value), when applicable.

The examples used to evaluate the properties have been selected from different domains, in order to show the genericity of the proposed methodology. In all cases the threshold used to select or discard an extraction has been empirically set to 1E-3. We used two Web search engines and the maximum number of Web pages retrieved for each pattern has been set to 500 per Web search engine.

Symmetric Object Property. The evaluation of the Symmetry of an Object Property has been tested with 10 relations, presented in Table 3 (5 of them are symmetric and 5 of them are not). The algorithm agrees in 9 out of 10 cases with the human evaluation. Note that, in this case, SCP O and SCP P coincide as the two individuals and the relationship are the same in the original relation xRy and in the relation yRx . In our test, 4 negative examples didn't obtain any result with the given pattern.

Table 3. Test of Symmetry for several Object Properties.

Original relation	Pattern	SCP O	SCP P	D	HD
Spain borders France	France borders Spain	0.22	0.22	Y	Y
Obesity is associated with cancer	Cancer is associated with obesity	0.09	0.09	Y	Y
Microsoft competes with Google	Google competes with Microsoft	0.29	0.29	Y	Y
Angelina Jolie is married to Brad Pitt	Brad Pitt is married to Angelina Jolie	0.34	0.34	Y	Y
Cigarettes are linked to cancers	Cancers are linked to cigarettes	0.001	0.001	Y	Y
Deep Blue defeated Gary Kasparov	Gary Kasparov defeated Deep Blue	0.004	0.004	Y	N
Heavy smokers get lung cancer	Lung cancer get heavy smokers	6.5E-4	-	N	N
Michael Jackson was born in Gary	Gary was born in Michael Jackson	0.006	-	N	N
Sociology is the study of society	Society is the study of Sociology	0.009	-	N	N
Madrid is located in Spain	Spain is located in Madrid	0.03	-	N	N

From the "Deep Blue defeated Gary Kasparov" example notice that, as a matter of fact, Gary Kasparov defeated Deep Blue but in a different match. This illustrates a limitation of the automatic process.

Reflexive Object Property. The evaluation of the Reflexivity of an Object Property has been tested with 10 relations, presented in Table 4 (5 of them are reflexive and 5 of them aren't). A relation is considered reflexive if the SCP score of the pattern exceeds 1E-3. The algorithm agrees in 8/10 cases with the human evaluation.

Table 4. Test of Reflexivity for several Object Properties.

Original relation	Pattern relation	SCP O	SCP P	D	HD
Nature regulates CO2	Nature regulates itself	0.04	0.23	Y	Y
Market regulates capital production	Market regulates itself	2.88E-3	0.14	Y	Y
Fire consumes wood	Fire consumes itself	0.15	0.09	Y	Y
Science studies Nature	Science studies itself	0.07	0.03	Y	Y
Immune system attacks viruses	Immune system attacks itself	0.015	0.0048	Y	Y
Testicular cancer has good prognosis	Testicular cancer has itself	1.1E-4	3.6E-5	N	N
People have colon cancer	People have itself	0.002	0.16	Y	N
Pressure sensor measures water pressure	Pressure sensor measures itself	8.4E-3	1.9E-4	N	N
Hysterectomy reduces cancer	Hysterectomy reduces itself	0.012	7.2E-4	N	N
Stress affects cancer	Stress affects itself	0.05	0.046	Y	N

In the evaluated examples, the algorithm found evidences for “people have itself” and “stress affects itself”. The correlations of “*itself*” with their respective subjects are high, but that doesn’t imply they are correct. For these two cases, the human expert evaluated the reflexivity as wrong.

Functional Object Property. The evaluation of the Functionality of an Object Property has been tested with 10 relations, presented in Table 5 (5 of them are functional and 5 of them are not). The algorithm agrees in 8/10 cases with the human evaluation. The table shows the best extraction found with the relation(subject,?) pattern. The algorithm takes the relation to be functional if the best extraction has an SCP lower than the established 1E-3 threshold.

Table 5. Test of Functionality for several Object Properties.

Original relation	NC	BES	SCP O	SCP P	D	HD
Brasilia is the capital of Brazil	4	country	0.004	3.7E-4	Y	Y
Cadmium exposure is associated with prostate cancer	1	reduced pulmonary function	1.28E-4	1.3E-4	Y	N
Da Vinci Code was written by Dan Brown	1	Bart D. Ehrman	0.009	6.5E-4	Y	Y
Comet Halley is named for Edmund Halley	0	-	0.019	-	Y	Y
Microsoft competes with Google	42	Apple	0.29	0.23	N	N
Brazil borders Bolivia	8	Peru	0.14	0.20	N	N
Michael Jackson was born in Gary	3	August	0.005	0.006	N	Y
Raloxifene reduces breast cancer	32	osteoporotic fractures	9.4E-4	0.006	N	N
Spain borders France	3	Portugal	0.229	0.25	N	N
Chichen Itza is known as El Castillo	2	La Iglesia	0.008	8.25E-5	Y	Y

It can be observed that the functionality test presents some limitations. The original relation “Cadmium exposure is associated with prostate cancer” scored a SCP of 1.28E-4 and the BES “reduced pulmonary function” scored higher than the original relation with 1.3E-4, but the algorithm considered the relation as not functional. For this case the expert decided that the relation with the best extraction is in fact valid.

Regarding the relation “Michael Jackson was born in Gary”, the relation “was born in” may be related to either time or place giving unexpected results. The expert considered the relation as functional (giving an approach of “place” only) while the algorithm considered it as non functional.

Inverse Object Property. The discovery of Inverse relations for an Object Property has been tested with 10 relations, presented in Table 6. The table shows the best extraction (i.e., the best candidate to inverse relation) found with the ?(object, subject) pattern. The algorithm takes the relation to have an inverse if the best extraction has an SCP (SCP P column) higher than the established 1E-3 threshold.

Table 6. Discovery of Inverse relations for several Object Properties.

Original relation	NC	BES	SCP O	SCP P	D	H
Christopher Columbus discovered America	5	was discovered by	0.004	0.017	Y	Y
Market regulates capitalist production	1	is attached to	1.8E-3	5.8E-5	N	N
Global warming is caused by pollution	6	causes	0.11	0.06	Y	Y
Cigarette smoking is associated with lung cancer	1	is caused by	0.007	0.002	Y	Y
Sensors contain mechanical parts	1	required for	4.7E-3	7.1E-4	N	N
Viruses are associated with liver cancer	2	caused by	5.2E-3	0.003	Y	Y
Kasparov defeated Deep Blue	16	beaten by	0.003	0.001	Y	Y
Stress affects asthma	3	is linked to	0.02	0.025	Y	Y
Meningococcal disease is caused by Neisseria Meningitides	1	causes	0.03	0.002	Y	Y
Charles Darwin proposed evolution theory	8	proposed by	0.002	0.001	Y	Y

The examples show that it is possible to extract incomplete verb phrases. For example, we found evidences for “liver cancer caused by viruses”, whereas “is caused by” would be a more complete verb phrase for that relation.

Inverse Functional Object Property. The Inverse Functionality of an Object Property has been tested with 5 of the “inverse” relations shown in the previous table. Table 7 shows the best extraction found with the relation(subject,?) pattern. The algorithm takes the original relation to be inverse functional if the best extraction has an SCP (SCP P column) higher than the established 1E-3 threshold. The algorithm agrees with the human result in 4/5 cases.

In the example (shown in the following table) that states “Evolution theory proposed by Charles Darwin”, notice that a selected extraction is “Darwin”. Both expressions refer to the same individual, but with the algorithm proposed this fact is not taken into account concluding in a wrong decision.

Table 7. Test of the Inverse Functionality of some Object Properties

Original relation	NC	BES	SCP O	SCP P	D	HD
Pollution causes global warming	76	acid rain	0.066	0.013	N	N
Asthma is linked to stress	67	allergies	0.026	0.108	N	N
Neisseria Meningitides causes meningococcal disease	2	inflammation	0.0022	3.08E-5	Y	Y
Evolution theory proposed by Charles Darwin	1	Darwin	0.001	0.001	N	Y
Lung cancer is caused by cigarette smoking	57	second hand smoke	0.003	0.0028	N	N

Transitive Object Property. To test the Transitivity of an Object Property we have followed two steps. Given a relation xRy , in the first step we look for individuals related to y via R , using the pattern $R(y,?)$. In the second step, we take the best extraction in each of the cases (z), and we try to evaluate whether xRz holds.

Table 8a shows the five relationships used in this test, along with the best extraction found in each case. An extraction is taken as correct by the algorithm if its SCP is above the 1E-3 threshold. From the first four relations given in Table 8a, we evaluate in Table 8b the best extraction (z) to see if $R(x,z)$ holds. If it does, the algorithm takes the relation to be transitive.

Tables 8a and 8b. Two steps in the test of Transitivity for some Object Properties

Original relation	NC	BES	SCP O	SCP P	D	HD
Global warming is caused by pollution	54	Nature	0.11	0.05	Y	Y
Berlin is located in Germany	12	Europe	0.04	0.07	Y	Y
Pollution affects global warming	46	Weather	0.038	0.05	Y	Y
Microsoft competes with Apple	30	Google	0.29	0.18	Y	Y
Arsenic is associated with kidney cancer	5	Obesity	8.1E-5	4.3E-5	N	Y

Relation	SCP O	SCP P	D	HD
Global warming is caused by Nature	0.11	0.05	Y	Y
Berlin is located in Europe	0.04	0.07	Y	Y
Pollution affects weather	0.038	0.05	Y	Y
Microsoft competes with Google	0.29	0.18	Y	Y

This way to evaluate the transitivity of a given relation presents a critical limitation. Considering that the evaluation of the transitive property in our approach includes two steps, the possibility to find transitivity for a relation is difficult. However, if a transitivity property is found using this approach that is due to the fact that the evaluated relatedness is certainly strong, as in the relations “Berlin is located in Europe” or “Microsoft competes with Google”.

5 Conclusion and Future Work

Axiomatizing ontologies by hand is a hard and time-consuming task which involves ontology engineers and domain experts. The design of automatic axiomatization solutions is fundamental to provide a powerful knowledge representation that will contribute to the success of the Semantic Web and many other fields in Computer Science.

The approach that we have proposed is a step forward to fill this gap, by learning object properties axioms for non-taxonomic relations using the Web as corpus, shallow linguistic techniques and Web-scale statistics. The Web can be exploited to aid in the Ontology Learning process by using suitable queries in order to retrieve valuable information. The patterns used in this work are totally generic and can be easily applied to any Web search engine. Besides, the proposed method is domain-independent.

However, being completely automatic and unsupervised, the proposed algorithm presents several limitations, that can lead to the definition of lines of future work. Some of these shortcomings are the following:

- At the moment we study only the axioms of a particular relationship between two individuals, xRy , e.g. to test the symmetry, we check whether yRx holds. We don't cover the study of the mathematical properties of the R relation in general; for example, to guarantee the symmetry of R , we should check whether, for all x, y such that xRy holds, yRx also holds.
- Regarding the reflexive property, the search for " xR itself" may be too restrictive and hard to find explicitly. We can add to the query the words "may" and "can" in order to extend the reflexive property discovery in the form of *Subject may/can Relation itself*.
- The transitive property is checked only with a particular instance. Note that "Italy borders Switzerland" will be qualified as transitive depending on the country which is the best extraction for the "Switzerland borders" pattern (giving different results e.g. for France and Germany, as one borders Italy and the other doesn't). Thus, we should at least check whether all the neighbours of Switzerland are also neighbours of Italy.

It is also necessary to study more deeply the effect of many natural language phenomena such as synonyms and polisemy in order to provide better results in certain domains in which these ambiguities are common. This may lead to further research in composing a more concrete set of patterns or adding more constraints in their extraction and selection. The empirical tests also show that in some cases the semantic relatedness acquired from the extractions has a similar value than the one of the original relation presented in the ontology. Starting from the assumption that the relations in the ontology to analyse are correct, we can use the SCP of the relations as a criterion to evaluate the co-occurrence of the evidence extracted instead of using a predefined fixed threshold. In this way, the algorithm would be more adaptative to how strong the co-occurrence of the original relation is.

References

1. Sánchez, D., Domain Ontology Learning from the Web. VDM Verlag Dr. Mueller. ISBN: 386470691 (2008)
2. Cimiano, P., Ontology Learning and Population from Text, Springer Science, 9-10 (2006)
3. Antipolis S., Research Challenges and perspectives of the Semantic Web, Report of the EU-NSF strategic workshop, France, 86-88 (2001)
4. Volker J., Haase P., Hitzler P., Learning Expressive Ontologies, in: Proc. of the Conference on Ontology Learning and Population Germany, 45-69 (2008)
5. Sánchez, D., Moreno, A., Pattern-based automatic taxonomy learning from the Web, AI Communications, 21(1) 27-48 (2008)
6. Shamsfard M., Abdollahzadeh A., Learning Ontologies from Natural Language Texts, in: Proc. of the International Journal of Human-Computer Studies, International Journal of Human-Computer Studies, Volume 60, 17-63 (2004)

7. Rudolph S., Volker J., Hitzler P., Supporting Lexical Ontology Learning by Relational Exploration, in: Proc. of the 15th International Conference on Conceptual Structures ICCS, Germany, 488-491 (2007)
8. Volker J., Hitzler P., Cimiano P., Acquisition of OWL DL Axioms from Lexical Resources, in: Proc. of the 4th European conference on The Semantic Web, Germany, 670-685 (2007)
9. Hearts M., Automatic Acquisition of Hyponyms from Large Text Corpora, in: Proc. of the 14th Conference on Computational linguistics - Volume 2, France, 539 -545 (1992)
10. Ritter A., Downey D., Soderland S., Etzioni O., It's a Contradiction-No it's not: A Case study using Functional Relations, in: Proc. of Conference on Empirical Methods in Natural Language Processing, University of Washington, 11-20 (2008)
11. Volker J., Vrande D., Sure Y. and Hotho A., Learning Disjointness, in: Proc. of the 4th European conference on The Semantic Web, Germany, 175-189 (2007)
12. Schutz, A., Buitelaar, P., A Tool for Relation Extraction in Ontology Extension, in: Proc. of the 4th International Semantic Web Conference, 593-606 (2005)
13. Dias G., Santos C., Cleuzious G., Automatic Knowledge Representation using a Graph-based Algorithm for Language-Independent Lexical Chaining, in: Workshop on Information Extraction Beyond The Document, Sydney, 36-47 (2006)
14. Sánchez, D., Moreno, A., Learning non-taxonomic relationships from web documents for domain ontology construction, Data and Knowledge Engineering 63(3) 600-623 (2008)
15. S. Rudolph, Exploring relational structures via FLE, in: Wolff, K.E., Pfeiffer, H.D., Delugach, H.S.,eds.: Conceptual Structures at Work, 12th International Conference on Conceptual Structures, volume 3127 of LNCS, Springer, pp. 196–212 (2004)