Chapter 1 – Introduction (planning)

[Flow: Communication Process->Vocabularies->Controlled Vocabularies->Goals of CV->Ontology->Ontology problems->Ontology learning->Pattern Extraction/Knowledge discovery->Approach to enrich an ontology based in data mining techniques.]

1P - Communication process/Languages/Vocabularies

2P - Controlled vocabularies, what are they?

3P - What problems CVs address?

* words with similar meaning and different spelling (Synonym),
* words with the same spelling and different meaning (Homograph)
* Standardization of the vocabulary of a community, one term represent only one concept

4P - Ontology as a form of CV, what is an ontology?

Why use an ontology?

5P - Problems of ontologies.

Static

Require previous agreement on the vocabulary

Require high maintenance

6P - (Automatic maintenance) Ontology learning, to help on maintenance of an ontology

What processes/techniques exist?

7P - Pattern Extraction and Data Mining Techniques to help on ontology learning and knowledge discovery

8P - What will I propose? Approach to discover knowledge in unstructured documents.

Section 1.1 – Challenges

* Lack of existence of a pure approach to quantify relations discovered from unstructured information in documents, without help of an ontology.

Ontology learning is a problem because there are no pure automatic mechanisms. (Explain ontology learning??)

* What can be done to measure a relation and find its meaning?
* This document presents an approach to help discover relations in unstructured information in documents, knowing that there are no real methods to help measure a relation between two or more concepts.

Research question:

How to quantify semantic relations between concepts in a domain ontology, using external sources of non-structured information.

Hypothesis:

Semantic relations between concepts from a domain ontology, can be quantified by applying data mining techniques for pattern extraction into non-structured sources of information.

* Having a set of documents with unstructured information, how could meaning be discovered, in the way of relations between its concepts?
* How to discover the domain of a set of words?

Section 1.2 – Expected outcomes

Present the way that I will propose solutions to research questions.

* How to address the problems?
* What techniques to use?
* Why are these techniques used to solve the problems, and not others?
* Develop a system, proof of concept, to present the results to domain experts.

Section 1.3 – Context of work

* Falar onde foi desenvolvido o trabalho
* A sua ligação com os projetos europeus (e-Cognos e CoSPaces)
* Enquadramento da tese de doutoramento do Ruben e a minha contribuição para a mesma.

Section 1.4 – Document Structure

Chapter 2 – Controlled Vocabularies

* (What are they? What do they represent?)
* What forms of representation of information exist?
* Ontologies (Definition, Construction, relations, concepts)

What is an ontology? What is it utility? How to construct one? Languages to represent it.

* E-cognos (European project for the creation of an ontology in B&C domain)
* Ontology learning
* Relations (meaning)
* Concepts
* Application domain. (Practical cases in building and construction domain)

Chapter 3 – Pattern Extraction from unstructured information sources

* Data mining / Knowledge Discovery. (What is DM/KD? Techniques used today?)
* Association Rules (Definition, Rules)
  + Algorithms to discover [ECLAT, APRIORI, FP-GROWTH]
  + Weaknesses/Strengths between them
  + Why FP-Growth?
* Application domain. (Practical cases where association rules are used)

Chapter 4 – Concept Model

- Explain conceptual model/solution

- Describe an application example

From unstructured information to knowledge representation and ontology structure

- Dimensions included in the model???

- Enrichment process

FP-Growth how to build and FP-Tree

Association rule evaluation

- DER / MVC / UML Diagrams

Chapter 5 – Model Design and Development (Proof of concept)

- Method proposal to address the question.

- What were the technologies used for the solution.

Technologies used,

- Implementation description.

(Present the server / front end solution)

- Include use cases (Relations discovered, new concepts discovered, etc.)

(Discover a relation between two concepts, update a relation between two concepts, and discover new concepts)

- Front end

Brief explanation of the functionality of the front end. Explain in a form of manual??

Chapter 6 – Assessment

* Present list of relations discovered and discuss them
* Present new concepts discovered

Chapter 7 – Conclusion and Future directions

- Evaluate if the goals reached success.

- Evaluate the achievement of the hypothesis

- Present the paper

# Contents

[Contents I](#_Toc418460388)

[Figures III](#_Toc418460389)

[Tables IV](#_Toc418460390)

[Symbols and Notation V](#_Toc418460391)

[1 Introduction 1](#_Toc418460392)

[1.1 Challenges 4](#_Toc418460393)

[1.2 Expected Outcomes 4](#_Toc418460394)

[1.3 Context of work 5](#_Toc418460395)

[1.4 Document Structure 6](#_Toc418460396)

[2 Controlled Vocabularies 9](#_Toc418460397)

[2.1 Controlled Vocabularies – Definition 9](#_Toc418460398)

[2.1.1 Problems Addressed by a CV 11](#_Toc418460399)

[2.1.2 Advantages / Disadvantages of CV 12](#_Toc418460400)

[2.1.3 Types of CVs – Differences, strengths and characteristics 14](#_Toc418460401)

[2.2 Ontology, a Definition 17](#_Toc418460402)

[2.2.1 Ontology Purpose 18](#_Toc418460403)

[2.2.2 Ontology Engineering & Components 19](#_Toc418460404)

[2.2.3 Ontology Languages 21](#_Toc418460405)

[2.3 Ontology Learning 23](#_Toc418460406)

[2.3.1 Problems related to the maintenance of an Ontology 23](#_Toc418460407)

[2.3.2 Definition of Ontology Learning 23](#_Toc418460408)

[2.3.3 Ontology Learning State of the Art 25](#_Toc418460409)

[2.4 Ontologies in Building and Construction Sector – E-Cognos project 25](#_Toc418460410)

[2.4.1 Historical perspective 26](#_Toc418460411)

[2.4.2 Creation of an ontology in B&C – E-Cognos approach 27](#_Toc418460412)

[3 Pattern extraction from non-structured sources of information 31](#_Toc418460413)

[3.1 Data mining / Knowledge Discovery. 31](#_Toc418460414)

[3.1.1 Definition - What is DM/KD? 31](#_Toc418460415)

[3.1.2 Techniques used today? 31](#_Toc418460416)

[3.2 Pattern Discovery 31](#_Toc418460417)

[3.2.1 FP-Growth – definition 31](#_Toc418460418)

[3.2.2 Algorithms to discover patterns Weaknesses/Strengths between them 31](#_Toc418460419)

[3.2.3 Association Rules (Definition, Rules) 33](#_Toc418460420)

[4 Concept Model 41](#_Toc418460421)

[5 Model Design and Development 43](#_Toc418460422)

[6 Assessment 45](#_Toc418460423)

[7 Conclusion and Future directions 47](#_Toc418460424)

[7.1 Work overview 47](#_Toc418460425)

[7.2 Research Contributions 48](#_Toc418460426)

[7.3 Future Directions 51](#_Toc418460427)

[8 Bibliography 55](#_Toc418460428)

# Figures

[Figure 1 - Word search example (Yahoo, 2015) 10](#_Toc418273075)

[Figure 2 - Vocabulary Example 14](#_Toc418273076)

[Figure 3 - Page from a Dictionary (Oxford University Press, 2006) 14](#_Toc418273077)

[Figure 4 - Example of a Species Taxonomy for b) Dog, c) Human and d) Parrot. a) Class Name Hierarchy. 15](file:///D:\Documentos\Universidade\Dissertação\Escrita\Dissertação%20v13.docx#_Toc418273078)

[Figure 5 - Page from Oxford Mini School Dictionary & Thesaurus (Allen and Mannion, 2007) 16](#_Toc418273079)

[Figure 6 - Domain Ontology example (Innovation Ontology adapted from (Stick-iSchool, 2013) ) 16](#_Toc418273080)

[Figure 7 - Some examples of CV-focused initiatives in Europe and worldwide (Lima et al., 2007) 26](#_Toc418273081)

[Figure 8 - The e-COGNOS ontology creation methodology (Lima et al., 2002) 28](#_Toc418273082)

[Figure 9 - e-Cognos taxonomies a) Concepts; b) Relations (Lima et al., 2007) 29](#_Toc418273083)

# Tables

[Table 1 - Examples of a) Homophone, b) Homograph and c) Synonym words 12](#_Toc418273084)

[Table 2 - Examples of relations 20](#_Toc418273085)

[Table 3 - Ontology Languages (Lima, 2004) 21](#_Toc418273086)

# Symbols and Notation

|  |  |  |
| --- | --- | --- |
|  | **API** | **A**pplication **P**rogramming **I**nterface |
|  | **AR** | **A**ssociation **R**ules |
|  | **B&C** | **B**uilding and **C**onstruction |
|  | **CV** | **C**ontrolled **V**ocabulary |
|  | **DB** | **D**ata**b**ase |
|  | **DM** | **D**ata **M**ining |
|  | **DOKS** | **D**ynamic **O**ntology learning with **K**nowledge sources from unstructured data **S**ystem |
|  | **ECLAT** | **E**quivalent **CLA**ss **T**ranformation |
|  | **FI** | **F**requent **I**tem |
|  | **FIM** | **F**requent **I**temset **M**apping |
|  | **FP** | **F**requent **P**atterns |
|  | **HTTP** | **H**yper**T**ext **T**ransfer **P**rotocol |
|  | **IT** | **I**nformation **T**echnology |
|  | **KDD** | **K**nowledge **D**iscovery in **D**atabases |
|  | **OL** | **O**ntology **L**earning |
|  | **OWL** | **W**eb **O**ntology **L**anguage |
|  | **PHP** | **H**ypertext **P**re**P**rocessor |
|  | **RDBMS** | **R**elational **D**ata**b**ase **M**anagement **S**ystem |
|  | **SEKS** | **S**emantic **E**nrichment of **K**nowledge **S**ources |
|  | **XML** | e**X**tended **M**arkup **L**anguage |
|  | **PDF** | **P**ortable **D**ocument **F**ile |
|  | **IFC** | **I**ndustry **F**oundation **C**lasses |
|  | **ISO** | **I**nternational **S**tandard **O**rganization |
|  |  |  |
|  |  |  |

1

# Introduction

The exponential growth of available information in digital format created the need to discover ways to organize it, in order to be easily accessible. First search engines were essentially word-based, meaning that the results provided by the search process could only be achieved if documents had in their bodies exactly the same words being searched for (Lei et al., 2006). The evolution of search engines motivated by the fact that a simple search by term for the information could not be enough, as the set of terms, or vocabulary available in information being searched could be different from the vocabulary being used. Therefore, it was of great importance to discover approaches for the representation of ideas (concepts), and not just the representation of terms, aiming at getting better results for queries (Almeida and Souza, 2011).

Nowadays, computers systems can represent sets of terms or words (also referred to as vocabularies). However, vocabularies themselves, do not represent ideas or concepts, they just represent words. In order to represent concepts and ideas, one approach can be considered. This approach is the use of mechanisms to represent more than pure words, to represent concepts. These mechanisms are referred to as Controlled Vocabularies (CV) (Lima et al., 2007). CVs are defined subsets of terms from a natural language (e.g. Esperanto), or can be pure symbols of any sort (e.g. sequence of digits) used to represent concepts, with some sort of organization. CVs represent the concepts by assigning to each, one or more words, or phrases and some describing properties that both translates its meaning. CVs also describe if or how a concept is related to other concept.

Natural languages are very rich in their vocabulary properties. They can have different meanings represented by the same word (Homograph words), in several contexts. Also, there are words that can be pronounced in the same way, however have different spelling and meaning (Homophone words). Homograph and Homophone words can lead to ambiguity and confusion when using the terms by people. CVs address the problems of Homograph and Homophone words solving them by assigning each term to just one concept, and adding properties to explain and provide a better meaning to each concept. For instance, the word “board” can represent a base used in a classroom to write with chalk, or can represent a platform to use in snow sports to ride on top of a mountain hill covered with snow. The way CVs deal with this, is by adding some properties that will increase the precision of the meaning of each term, reducing the ambiguity when these words are used. (N.I.S.O. (US) and others, 2005)

An Ontology is a type of CV that addresses problems like the consistent representation or word ambiguity in information. According to Gruber (Gruber, 1993) an ontology is “*(...) a formal specification of a shared conceptualization of a domain of interest.*” In other words, an ontology represents a formal agreement, where *formal* implies that it has to be machine readable, and *agreement* implies a shared understanding of meaning on the ontological concepts. An Ontology is used when there is the need to share or exchange knowledge within a given domain. Ontologies can be represented as a hierarchically structured set of concepts describing a specific domain of knowledge.

Although ontologies provide structures for concept representation, they face some challenges (Uschold and Gruninger, 1996). So why use an ontology? Inside an organization people from different domains can have different points of view and different words to communicate. In this sense the benefits of using an ontology is to be able to provide a common ground that can lead to a shared understanding for the same concepts. Additionally, when two IT systems need to exchange knowledge, ontologies provides them inter-operability features in order to ease the integration between them (Pouchard et al., 2000). Furthermore, ontologies are useful when there is the need to reuse its contents and features. There is no need to re-invent the wheel (Gangemi and Presutti, 2009).

Ontology Learning (OL) deals with the creation and maintenance of an ontology, and studies the mechanisms and processes to transform heavy tasks like creation and maintenance of Ontologies, into a semi or complete automatic process. IT is worth noting that relevant literature already presents first results on automatic maintenance of ontologies, but still in a very early stage. Human-based processes are still the current way to update and maintain ontology growth (Zhou, 2007).

One of the motors that drive OL itself is the recognition of patterns in the data that could originate new knowledge to further evaluation. For instance, this could be learned from some information not yet known or unpredictable in a specific domain. A pattern, in the area of information retrieval and text mining, can be defined as a predictable occurrence that repeats itself along some text data. Furthermore, Knowledge is defined as “*awareness, familiarity, or understanding of someone or something (e.g. facts, information, descriptions or skills), acquired through experience or education by perceiving, discovering or learning*.” (Oxford University Press, 2006) Therefore, OL provides techniques to discover knowledge.

Several processes can be used for a system be able to recognize patterns and further extract knowledge from data and information. Data Mining (also referred in literature as Knowledge Discovery in Databases or KDD) is one of them (Hand et al., 2001). Data mining allows experts to find knowledge in new data or data they already have. Additionally, by adopting data mining techniques, it is expected that decision makers can use new knowledge that otherwise could be unknown, unavailable or difficult to discover, to make better decisions. (Witten et al., 2011)

Having settled the context, urge to say that this dissertation aims at proposing an approach to support part of the process of ontology learning. Specifically, the proposed approach adopts a mechanism suitable for the use of data mining techniques for pattern discovery and extraction, and knowledge discovery from unstructured sources of information from a document corpus. Additionally, it is also proposed an approach to help maintain and update CVs, namely domain ontologies, with the previous discovered knowledge. This means: (i) to discover concepts and relations between them; (ii) to propose an approach to quantify these relations; (iii) to discover new concepts; And finally, (iv) to take advantage of (i), (ii) and (iii) results to update a domain ontology. Furthermore, a proof of concept to characterize this approach, referred as DOKS (Dynamic Ontology learning with Knowledge sources from unstructured text System), is also part of the results produced.

## Challenges

One of the biggest challenges in information systems when constructing a CV is to find both meaning and relations among concepts and ideas. Furthermore, how to say that a concept is more related to one, than it is to other concept? How to quantify this relation? Similarly, other challenge is to discover knowledge in sources of information that could be later used, for instance, to update a CV. Moreover, is it possible to fully automate this process? Still, other challenge identified relates to the limited amount of information that is inside a single document. This dissertation proposes an approach to help solving these challenges based in the following guiding question:

**How to formally discover and quantify semantic relations between concepts in a domain ontology, using external sources of non-structured information?**

That question highlights the research path leading the development of this work, as follows:

**Semantic relations between concepts from a domain ontology, can be quantified by applying data mining techniques for pattern extraction and knowledge discovery into unstructured sources of information.**

## Expected Outcomes

When a study is made, there is a need to consider its contribution and applicability that can arise from it. In this sense, the expected outputs to be provided by this work are the following ones:

* To develop a method to describe how to extract concepts and recognize relations between them from a data document corpus, and to find new knowledge sources in order to update a domain ontology.
* To develop a proof of concept, a software platform, based in the previous method in order to reflect the application of the studied techniques.
* Present results of the semi-automatic OL process. Results composed by patterns discovered in the documents, their relations and the new concepts discovered. They should be presented in an understandable way to the user.
* Finally, publication of scientific documents about the work, to be assessed by the academic community.

## Context of work

The context of the present work arisen from three MSc. Dissertations (Antunes, 2010; Figueiras, 2012; Parada, 2010) in the area of Data Mining and Knowledge Sources. These studies provided the background and inspiration for the reasoned path choice of the present work. The setting made through these studies was provided by CoSPaces. CoSPaces was an European Research project aiming to provide digital solutions in a collaborative workspace between individuals, teams and enterprises. The project expected to achieve the former by improving collaboration methods, like human communication and knowledge sharing support, taking advantage and improving existing IT systems.

EU research project E-Cognos was an inspiration in CV domain. Specifically, it provided the insight and methodology needed to build a domain ontology. Also, provided the ground for the structure representation of the semantics in an ontology applied in the B&C sector.

This work takes advantage of the application domain background based in the Building & Construction sector, which provided the knowledge sources, specifically technical documents (e.g. reports and papers) to be used. They were adopted from (Costa, 2014), a PhD Thesis, that also received a contribution from this study. Namely, “*Semantic enrichment of knowledge sources supported by domain ontologies*”, whose main goal was to “*introduce a novel conceptual framework to support the creation of knowledge representations based on enriched Semantic Vectors, using the classical vector space model approach extended with ontological support*”. The respective contribution was the proposal of an ontology learning method based in knowledge discovery techniques.

SEKS (Figueiras, 2012) also provided some resources which were adopted in this work, namely the domain ontology manipulation libraries.

The applicability context of the present work relied in B&C sector, as it was the domain that provided the resources and inspiration. However, in a more abstract sense, the contribution made here can be further used wherever there is the need of a shared communication and understanding of concepts, and in all the fields where knowledge and domain ontologies can be used.

## Document Structure

Following this brief introduction in Chapter 1 with the setting of the problem, the expected outcomes to achieve and the contextualization of the work by the author of the present document, this dissertation will be guided by the following structure.

In Chapter 2, Controlled Vocabularies are the domain of study. Ontology will be the selected CV discussed. It will be explained in more detail what is an Ontology and how to build one. Additionally, it will be presented some existent formalisms to represent them and where are they used.

Chapter 3 will explain what is data mining and knowledge discovery, and describe techniques to discover patterns from unstructured data. One of them, Association Rules will be explained in more detail. FP-Growth, and the concurrent algorithms to discover patterns will be compared, and explained why the former was chosen.

In the following chapter, can be observed the explanation for the solution proposed. Thus, Chapter 4 will present the concept model, an application example describing how to reach from non-structured information to knowledge representation and ontology learning. This chapter also includes the methodology behind FP-Growth and the evaluation of an Association Rule.

With Chapter 5, one can expect to read about the development of a proof of concept. The design and development of a model, with the proposed method to address the question. This will be described with the technologies used, following a description of the implementation and use cases. The framework developed will also be presented in this chapter.

Chapter 6 will be the assessment of the solution proposal, and the evaluation of the results. Chapter 7 will present some conclusions from the author, namely an overview of this dissertation, the achievement of the proposed outcomes, some possible future directions in this area and some scenarios where this work could be an asset.

2

# Controlled Vocabularies

In this chapter it will be presented an introduction to some concepts and definitions about Controlled Vocabularies. Moreover, it will be described forms of information representation. In particular it will be given special attention to the “appealing” and “scary” Ontologies. Furthermore, it will be explained how to represent a concept (or idea) and the relations between them, into an information system and how ontologies use them. Additionally, a more in depth overview of Ontology Learning will be explained in order to better understand what is it and how does it works. Lastly, a brief insight to the project that inspired the idea of Ontology use in the present work, the E-Cognos European project, applied in the Building and Construction sector.

## Controlled Vocabularies – Definition

The exponential growth of available information in digital format created the need to discover ways to organize it, in order to be easily accessible. First search engines were essentially word-based, meaning that the results provided by the search process could only be achieved if documents had in their bodies exactly the same words being searched for (Lei et al., 2006). For instance, if one wanted to do a query on a common search engine (e.g. Yahoo, Google, Bing) for the word “*car*”, each result would need to explicitly contain the word searched for. (Figure 1)

The evolution of search engines was motivated by the fact that a simple search by term for the information could not be enough, as the set of terms, or vocabulary available in information being searched for could be different from the vocabulary being used. Referring to the example from Figure 1, it is shown that if a user could query a search engine for the concept “*road vehicle, typically with four wheels, powered by an internal-combustion engine and able to carry a small number of people”*(Oxford University Press, 2006) represented by the word “*car*”, consequently the results would include the documents containing this search term. Although, the terms “*automobile*” and “*vehicle*” could also describe the same concept. Alternatively, if the term used to search the same concept was “*automobile*”, the results would be a different set of documents. Therefore, it was of great importance to discover approaches for the representation of concepts (ideas), and not just the representation of terms, aiming at getting better results for queries (Almeida and Souza, 2011). In this sense, the results provided by the query of Figure 1 example would be a sum of the results provided by the terms *“car”*, *“automobile”* and *“vehicle”*.



Figure 1 - Word search example (Yahoo, 2015)

Nowadays, sets of terms or words (also referred to as vocabularies) can be represented in computers systems. However, vocabularies themselves, do not represent ideas or concepts, they just represent words. Vocabularies are just word lists with no specific organization. Also, words *per se* are just units of a language, they have the responsibility to be the carriers of meaning. One can only understand an idea represented by a word when a meaning is associated to that word, as the meaning is itself the idea that a person wants to express when using that word.

One approach can be considered, in order to represent concepts and ideas. This approach is the use of mechanisms to represent more than pure words, to represent concepts. These mechanisms are referred to as Controlled Vocabularies (CV) (Lima et al., 2007). CVs are defined subsets of terms from a natural language (e.g. Esperanto), or can be pure symbols of any sort (e.g. sequence of digits) used to represent concepts, with some sort of organization. CVs represent the concepts by assigning to each, one or more words, or phrases and some describing properties that both translates its meaning. CVs can also describe if or how a concept is related to other concept.

*Controlled* refers to (i) the fact that the vocabulary used needs to be organized based in some logic structure (e.g. Alphabetically, Geographically, Hierarchically, etc.) and defined based in the association of each term to just one meaning, reducing the ambiguity and improving the consistency of a CV; (ii) the fact that the constraints in its use are bigger than in a natural language; a CV can only have one meaning per word. (iii) the fact that the access control to the maintenance of the terms of a CV by the people is restricted. It can have different restrictions to different users (e.g. Normal user, Domain Expert user, Admin user). For example, just domain expert users can propose new words for a CV and just admin users can add new words.

### Problems Addressed by a CV

Natural languages are very rich in their vocabulary properties. They can have different meanings represented by the same word, in several contexts, referred to as **Homograph** words. Also, there are words that can be pronounced in the same way, however have different spelling and meaning. These kind of words are referred to as **Homophone** words. Homograph and Homophone words can lead to ambiguity and confusion when using the terms by people. (Refer to Table 1 a) and b) for examples of Homophone and Homograph words respectively)

CVs address the ambiguity problems of Homograph and Homophone words solving them by assigning each term to just one concept, and adding properties to explain and provide a better meaning to each concept. For instance, the homograph word “*board*” (Table 1-b)) can represent a base used in a classroom to write with chalk, or can represent a platform to use in snow sports to ride on top of a mountain hill covered with snow or can even represent a group of directors from a company. Likewise, the homophone words “*made*” and “*maid*” have the same pronunciation, although the first word refers to the conjugation of the verb “to make” in the simple past tense and past participle, and the second word refers to a female servant. How can a search system (e.g. Yahoo!, Bing from Microsoft or Google) deal with homograph words by being able to distinguish two different meanings that can be represented by the same word? How can a voice recognition system (e.g. Cortana from Microsoft, Siri from Apple or Google Voice Search) deal with homophone words by being able to recognize accordingly two words that sounds the same? CVs deal with this by adding some properties that will increase the precision of the meaning of each term, reducing the ambiguity when these words are used. (N.I.S.O. (US) and others, 2005).

Table 1 - Examples of a) Homophone, b) Homograph and c) Synonym words

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Homophone |  | Homograph | |  | Synonym | |
| **Words** |  | **Word** | **Meaning** |  | **Word** | **Synonyms** |
| Board, Bored |  | Advocate | Speak or write in support of/Person who supports cause of another person |  | Car | Vehicle, Automobile |
| Dual, Duel |  | Board | Base to write/Platform to ride in snowboard |  | Couch | Sofa, Divan, Chair |
| Loan, Lone |  | Desert | A hot, arid region/To leave |  | Garbage | Trash, Junk, Waste |
| Made, Maid |  | Evening | Late afternoon/Making more even |  | Honest | Honourable, Fair, Sincere, Trustworthy |
| Sloe, Slow |  | Match | Wood stick to ignite fire/Sporting event |  | Intelligent | Smart, Bright, Brilliant, Sharp |
| Peak, Peek, Peke, Pique, Pick |  | Object | Thing to see or touch/Goal |  | Vocabulary | Dictionary, Terminology, Glossary |
| Rain, Reign, Rein |  | Tear | To rip/A drop of water from the eye |  | Woman | Lady, female, girl |
| **a)** |  | **b)** | |  | **c)** | |

Additionally, natural languages have more properties that must be addressed by a CV. In particular, there are words that have the same or similar meaning and have different spelling. These words are referred to as **Synonym** words (refer to Table 1c)). As a result, a concept can be represented by more than one word. Referring to Figure 1 example, one can infer from it that the concept represented by the word “*car*” have also some other words that can represent them, as any of its synonyms “*vehicle*” or “*automobile*”. A CV must allow the use of synonym words, as different people can use different vocabulary for the same concept. And in this case, a query for any of the words from the same concept must return the same or similar results.

### Advantages / Disadvantages of CV

When looking at the advantages from the use of CVs, the following can be enumerated:

* **CVs improve the efficiency and precision of retrieval systems.** By providing more than one possible term to search for a concept, they allow for results that do not explicitly contain the search term and still are somehow related to the concept. For instance, a search by cars would provide results containing the word *“cars”* and also the results containing its synonyms words, like *“automobile”* and *“vehicle”*. Likewise, by limiting the terms that can be used, providing a more objective search through the terms used. For example, if one searches documents about *“football”* would not get documents about *“cars”*.
* **CVs remove the ambiguities from natural languages.** Natural languages associate a word to more than one meaning. Consequently, it is hard for an information system to know what the user wants to search. As a result, each term is associated to a specific and unambiguous meaning.
* **CVs activate semantic search**, meaning that the search will be made by idea and not by word. Through the use of each term associated to the concept. In other words, means that the terms used do not need to explicitly be in the data searched for.
* **CVs improve communication through peers in a community or organization,** in the way that they provide the same name to the same thing in the same domain or working context. . When everybody that uses a CV knows the terms to use when referring to a concept, it allows better communication through all people involved in CV use. For instance, when two civil engineers from the same company, work on the same project, a “bar” will always have the same meaning for them, resulting in better communication.
* **CVs provide its reusability in long-term.** The building of a CV can take a lot of time until it can be ready to be used. Requires preparation, planning, and execution time that can be very exhaustive. In this sense, as CVs are built in a way that can be used in several places, several times, taking advantage of the work that was initially aimed for a specific project, into another different project.

In the Disadvantages side, the following can be found:

* **CVs cost time & money to build for the first time.** Building a CV takes time. First to gather all concepts and vocabulary related to a specific domain that will be necessary to include in a CV; second, to find and associate each term to a specific concept is time-consuming. As a result, companies are reluctant to adopt CVs if they want fast revenue from its investments in short term.
* **CVs allow Human/Domain Expert error.** The concepts are gathered by humans, which should be experts in the CV applicable domain. Although expertise is an asset, the expert is still a human, and humans are prone to errors, even experts. Therefore these errors can lead to imprecise and badly formed CVs.

### Types of CVs – Differences, strengths and characteristics

Nowadays there are several ways to represent information in retrieval systems. One of them are CVs. CVs can be divided by complexity, usability needs and level of control.



Figure 2 - Vocabulary Example

The simplest form of CV is a **Vocabulary**, a list of words or terms without any specific organization logic that gives names to things (Figure 2). Although a vocabulary can have some uses, in a retrieval system, most of the times just the words are not enough for semantic retrieval purposes and are the starting point of a CV use.



Figure 3 - Page from a Dictionary (Oxford University Press, 2006)

When a definition is added to each word from a vocabulary, this vocabulary becomes a **Dictionary** (Figure 3). Dictionary is a vocabulary, or a list of words alphabetically ordered which contains the source of all meaning. Each word has its own meaning described along with some properties. There are several types of dictionaries, in which one of them is a Language Dictionary, which contains all the words that can be used in a particular language (e.g. English Dictionary (Oxford University Press, 2006)). Other type of dictionaries that exist provides the translation of the meaning of every word from one source language to one or more target languages (e.g. Essential Portuguese Dictionary (Oxford University Press, 2012)). This kind of dictionary are used to help when there is the need to communicate between different languages.

Figure 4 - Example of a Species Taxonomy for b) Dog, c) Human and d) Parrot. a) Class Name Hierarchy.

CLASS

SPECIES

GENUS

FAMILY

ORDER

PHYLUM

KINGDOM

HOMO SAPIENS

HOMO

HOMINIDAE

PRIMATES

CHORDATA

ANIMALIA

AVES

PSITTACUS ERITHACUS

PSITTACUS

PSITTACIDAE

PSITTACIFORMES

MAMMALIA

CANIS FAMILIARIS

CANIS

CANIDAE

CARNIVORA



b)

c)

d)

a)

A **Taxonomy** (Figure 4) is a structured vocabulary that introduces a hierarchical and a classification layer to a dictionary. Each term is gathered into groups (or classes) in a parent-child-based structure, from the most abstract to the most specific class. It is through a taxonomy that the association between words can be introduced through the parent-child hierarchy. Each term belonging to the same class shares a common characteristic, meaning that each term is associated through this same characteristic. It provides a structured classification mechanism for the terms from a vocabulary.

Adding another type of layer between terms, a sibling-based (on the same hierarchically level) structure, to a taxonomy, results in a **Thesaurus (**Figure 5**)**. A Thesaurus takes advantage of a taxonomic structure and associative relations, or semantic relations to its terms. These semantic relations are in the form of synonyms.



Figure 5 - Page from Oxford Mini School Dictionary & Thesaurus (Allen and Mannion, 2007)

**Ontology** (Figure 6) is the most complex form of a CV. An Ontology includes a set of words (vocabulary), hierarchical and associative relations (taxonomy/thesaurus), a list of concepts and properties for each concept. An ontology is commonly represented in a graph, where each node represents a concept and each connection represents a relation between two concepts. (Please refer to section 2.2 for in depth overview about ontologies)



Figure 6 - Domain Ontology example (Innovation Ontology adapted from (Stick-iSchool, 2013) )

In brief, one can find some arguments that are common to every type of CV presented above:

CVs are mechanisms to structure, classify and represent terms or concepts;

CVs allow a community to agree and use the same terms in the same way;

CVs can be understandable and readable by machines and humans, as well as be used to exchange information between them.

#### Uncontrolled Vocabularies

Another kind of representation structure is worth mentioning. It is not a controlled vocabulary, however is still a managed vocabulary. Is referred to as **Folksonomy**. and is considered an **uncontrolled vocabulary** (Aquino, 2007). Folksonomy provides a user the possibility to associate any word he/she considers adequate to any information element (e.g. documents). This could be understood as the possibility to customize the information from each entity (e.g. person, company, etc) and adapt to the context of such entity. One of differences between a CV and a folksonomy is the control factor. For a folksonomy there is less control over the vocabulary, meaning that the control is not made by experts as in a CV; on the contrary, the control is made by the people that uses it every day. This gives the possibility for final users that access the information to add words that could have been forgotten by domain experts. Although, a folksonomy can be customized by each user, the terms used in the information are not validated by someone who has the expertise knowledge.

Summing up, a folksonomy is a nouvelle representation mechanism that takes advantage of users and social networks to help classify words and build a vocabulary for a specific purpose. This new form of representation is more user-oriented in contrast to the CVs which are more standard-oriented.

## Ontology, a Definition

The term “Ontology” origins from the early 18th century from the modern Latin word *ontologia*, a composition from the Greek words *onto*, which means “being” and *logia* which means “study” (or science, theory). In Philosophy, ontology is the study of the nature of a being or the existence of things and how these things can be related to each other.

In Artificial Intelligence and Information Systems, the most commonly referred definition for ontology is the one presented by Gruber (Gruber, 1993). In particular, an ontology is “*(...) a formal specification of a shared conceptualization of a domain of interest.*” In other words, an ontology represents a formal agreement, where *formal* implies that it has to be machine readable, and *agreement* implies a shared understanding of meaning on the ontological concepts. An Ontology is a type of CV that addresses problems like the consistent representation or word ambiguity in information. An Ontology is used when there is the need to share or exchange knowledge within a given domain. Ontologies can be represented as a hierarchically structured set of concepts describing a specific domain of knowledge.

### Ontology Purpose

Nowadays, ontologies can be found across several information-system-related areas. Ontologies can be found in the Semantic Web, in Building and Construction, in Medicine, in Libraries, just to name a few. Although ontologies provide structures for concept representation in all these areas, they face some challenges (Uschold and Gruninger, 1996). Therefore, why use an ontology? Inside an organization people from different domains can have different points of view and different words to communicate. In this sense the benefits of using an ontology relies in the ability to provide a common ground that can lead to a shared understanding for the same concepts. If everyone uses the same words to communicate the same ideas, the understanding of meaning is global across all the peers that access the knowledge in an ontology.

Additionally, the need to communicate remotely and through different types of systems rises each day. Often companies work through different sites or work with information that is not physically located in the same place from where it is accessed (as in a library). Also, for a person can be easy to understand an idea that is being communicated by other people, as they can ask questions to each other to clarify possible doubts. On the contrary, IT systems cannot ask questions. An IT system by its nature, can only understand bits[[1]](#footnote-1). As a result, when IT systems need to exchange knowledge, they need to be able to understand more than bits. Ontologies provide inter-operability features in order to ease the integration between IT systems (Pouchard et al., 2000). They provide the necessary formalisms to exchange the exact same idea between both. Ontologies provide formal specifications aiming for machine readability, by explicitly defining concepts through terms (eg. words, images, sounds, etc). Therefore, ontologies provide to systems exchanging knowledge, the capability to understand the exact same ideas. This understanding can be extended to the point of view of human-machine interaction. Indeed, the formalisms used in ontologies are also human readable. This is a requirement in ontologies and allows a human to understand and work with the knowledge from an ontology.

Furthermore, ontologies are useful when there is the need to reuse its contents and features. There is no need to re-invent the wheel (Gangemi and Presutti, 2009). Would not be worth to build an ontology each time anyone would need one. This step is complex and time consuming, so reuse the ontological resources already available is mandatory to motivate the use of the ontologies.

### Ontology Engineering & Components

Specific concepts from a domain are not always easy to understand. Some of them are implicitly understood from other concepts. IT systems do not understand implicit concepts, in contrast they need an explicit conceptualization of ideas in the information in order to be able to understand and work with them easily. Ontology Engineering is a discipline that studies tasks like, Ontology Building (De Nicola et al., 2009; Elsayed et al., 2007) and Ontology Maintenance (Gargouri et al., 2003) which develops approaches for explicit conceptualization of ideas.

One can find ontology tools that can deal with Ontology Engineering tasks (eg. Protegé (Stanford Center for Biomedical Informatics Research, 2011) or OntoEdit (Sure et al., 2002), however these technologies do not have yet sufficient maturation, meaning that the building of an ontology is still a manual, tedious and cumbersome task. Because of this, there is still some reluctance in ontology use. Ontology engineers often face questions and doubts related to ontology development as building time, difficulty, confidence and its maintenance.

In order to help explain why building a domain ontology can be challenging, one first needs to identify the components of an ontology: *Concepts (Ideas), Relations (Meaning), Axioms (Rules)* and *Instances (Individuals).*

#### Concepts (ideas)

A concept is defined as “An abstract idea; a general notion; an idea formed by mentally combining all its characteristics or particulars” (Oxford University Press, 2006). In other words, a concept is an idea that can be difficult to understand and is constructed in the mind of someone. It can be anything, as an object, a place, an image, a task, a reasoning process, etc., it can be whatever would fit on a mind.

#### Meaning and Relations

Meaning is the concept that is represented by a word, phrase. Is the idea that a person desires to express through the use of words, signs, pictures, etc. A relation in an ontology is a connection between two or more concepts, which represents their proximity in meaning. Relations provide more information about concepts related to its meaning. In other words, they help clarify, and position concepts closer to an explicit representation.

Table 2 - Examples of relations

|  |  |
| --- | --- |
| RELATIONS | |
| *is-a* | *has* |
| *part-of* | *is-equal-to* |
| *is-about* | *is-similar-to* |

The relations can be manifested through either or both hierarchical and associative form. Hierarchical relations are in the form of parent-child connections (or with more levels, like grandparent-grandchild, etc.). These relations can be found in taxonomies, in which case they can be referred as “is-a” taxonomic relations. In previous Figure 4 d) from this chapter, it can be seen, as an example, a relation between PRIMATES (parent level) and HOMINIDAE (child level), which can be symbolised in other words as an HOMINIDAE “*is-a*” PRIMATE. Conversely, associative relations are found in connections in the same level, in the form of siblings, called synonyms. This association represents connections to similar or same meaning in a word or concept. One can find some examples for this case in Table 1c). Several other examples of relations can be found in ontologies. A non-exhaustive example list can be found in Table 2 above.

#### Instances (or Individuals)

Instances (or Individuals) are the units that are used to represent a concept. They can be a word, an image, a number, anything that can be represented and can hold the meaning of a specified concept.

#### Axioms (or Rules)

Axioms (or Rules) are formal descriptions of the concepts. They describe additional constraints on the ontology and allow to transform implicit facts into explicit ones. (Maedche and Staab, 2001) Axioms provide descriptions for the characteristics and properties of concepts, and can be seen as the concept definitions. They can include collections of descriptions, as restrictions, classes, boolean combinations of descriptions and one or more individuals. (W3C, 2004)

### Ontology Languages

There are several formalisms defined that can provide representation of information in an ontology. Table 3 provides a non-exhaustive list just for demonstration purposes, of several languages used in Ontology Engineering.

Table 3 - Ontology Languages (Lima, 2004)

|  |  |  |
| --- | --- | --- |
| Language | Description | URL |
| DAML+OIL | DAML+OIL is a semantic markup language for Web resources. It builds on earlier W3C standards such as RDF and RDF Schema, and extends these languages with richer modelling primitives. DAML+OIL provides modelling primitives commonly found in frame-based languages. It is important to emphasise that this language was the basis of OWL. | http://www.w3.org/TR/daml+oil-reference |
| EXPRESS / EXPRESS-G | EXPRESS-G is a standard graphical notation for information models. It is a useful companion to the EXPRESS language for displaying entity and type definitions, relationships and cardinality. Used by the ISO DIS 12006-3. | http://www.steptools.com/support/stdev\_docs/devtools/devtools-8.html |
| OIL | OILS stands for Ontology Inference Layer, a language that was developed in the context of the European IST Ontoknowledge project. It is built on top of RDF(S), using as much as possible RDF(S) constructs in order to maintain backward compatibility. | http://www.ontoknowledge.org/oil/ |
| OWL | The OWL Web Ontology Language is designed for use by applications that need to process the content of information instead of just presenting information to humans. OWL facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema (RDF-S) by providing additional vocabulary along with a formal semantics. | http://www.w3.org/TR/owl-features/ |
| RDF(S) | Resource Description Framework (RDF) defines a language for describing relationships among Web resources in terms of named properties and values. It is particularly intended for representing metadata about Web resources, such as the title, author, copyright and licensing information about a Web document, or the availability schedule for some shared resource. | http://www.w3.org/TR/rdf-schema/ |
| XML | Extensible Markup Language (XML) is a simple, very flexible text format derived from SGML. Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere. XML has been largely used to represent "semantics" in the Web, here including taxonomies, classification systems, etc.. | http://www.w3.org/XML/ |
| Topic Maps | Topic Maps (ISO/IEC **13250)** define a model for the semantic structuring of knowledge networks and are a solution for organising and accessing large and continuously growing information pools. They provide a ‘bridge’ between the domains of knowledge management and information management. They can also be used to generate navigation for a website, and lots of other metadata tasks. A topic map is a collection of topics (a topic is a resource that acts as a proxy for some subject; the topic map system's representation of that subject), associations, and scopes that may exist in one of two forms: (i) a serialized interchange format (e.g. as a topic map document expressed in XTM syntax); or (ii) Some application-internal form, as constrained by the XTM (XML Topic Maps) Processing Requirements. A topic in a topic Map represents a subject inside the computer. | http://www.topicmap.com/  http://www.topicmaps.org |
| KIF | Knowledge Interchange Format (KIF) is a language designed for use in the interchange of knowledge among disparate computer systems. KIF, a particular logic language, has been proposed as a standard to use to describe things within computer systems, e.g. expert systems, databases, intelligent agents, etc.. Moreover, it was specifically designed to make it useful as an "interlingua". This means a language useful as a mediator in the translation of other languages. KIF has declarative semantics; it is logically comprehensive (i.e. it provides for the expression of arbitrary sentences in the first-order predicate calculus); it provides for the representation of knowledge about the representation of knowledge; it provides for the representation of non-monotonic reasoning rules; and it provides for the definition of objects, functions, and relations. When the computer system needs to communicate with another computer system, it maps its internal data structures into KIF. KIF is a programmer-readable language and thereby facilitates the independent development of knowledge-manipulation programs. | http://logic.stanford.edu/kif/kif.html |

## Ontology Learning

### Problems related to the maintenance of an Ontology

The manual creation and maintenance of an ontology is a tedious and cumbersome duty. It is thus desired to reuse and take advantage of the work already done. In this sense, several tasks can be identified when thinking about the creation and maintenance of an ontology.

Lima (Lima et al., 2003a) identifies two main branches that should be concerns when dealing with the ontology maintenance challenges. The first branch refers to the need for an ontology be adapted to the system and domain in which is being modelled. Specifically, this refers to the ability of an ontology to add, update or delete concepts, relations, instances and axioms, through tasks like acquisition of new knowledge, retrieval and matching of concepts and relations, association of terms to concepts, or definition of constraints and axioms.

The second branch relates to the consistency that needs to be assured between the knowledge representations already existing and the necessary evolutions of the ontology. Noy (Noy and Musen, 2004) identifies some specific tasks related to the consistency assurance needs: Import and reuse ontologies; Translate ontologies from one formalism to another; Provide support for ontology versioning; Specify transformation rules between different ontologies and versions of the same ontology; Merge ontologies; Align and map between ontologies; Extract semantically independent parts of an ontology; Support inference across multiple ontologies; Support query across multiple ontologies.

### Definition of Ontology Learning

Ontology Learning (OL) deals with the creation and maintenance of an ontology, and studies the mechanisms and processes to transform heavy tasks like the creation and maintenance of Ontologies, into a semi or complete automatic process. IT is worth noting that relevant literature already presents first results on automatic maintenance of ontologies, but still in a very early stage. Human-based processes are still the current way to create, update and maintain ontology growth (Zhou, 2007).

In fact, manual building of an ontology is an extremely intensive and time consuming process, and because of this, the motivation to automate OL is high. OL provides contributions by offering to the ontology community efficiency and overcoming the bottleneck in content discovery for learning ontologies. (Zhou, 2007)

In literature, commonly OL can be found related to several fields such as *machine learning* (Buitelaar et al., 2005), *knowledge acquisition* (Sánchez, 2010), *natural-language processing* (Liu et al., 2011), *information retrieval* (Zhang et al., 2006), *text mining* (Reinberger and Spyns, 2005) and *artificial intelligence*, just to name a few.

**Unstructured (non-structured), semi-structured and structured data**

The growth of IT systems increased the quantity of data available. This fact created a new challenge, the diversification of the type of information that can be found. These includes, web pages, documents, images and others, holding more or less structure than others.

One of the ways to categorize OL systems is by the data they use to learn ontologies. One can find the data available as structured, semi-structured and unstructured (or non-structured) (Cimiano et al., 2009; Hazman et al., 2011). Consequently, the OL methods to use in each type of data are different.

Structured data is data that is already organized like in databases schemas or in some different type of CVs, like a dictionary or an ontology. As the data is already structured, the main goal in OL from structured data is to find which pieces of structural information are valuable and can provide relevant knowledge. For instance, one can identify concepts and their relations based in a database schema. (Kashyap, 1999)

Semi-structured data is related to text and data that can be found in HTML pages, XML files, etc. This data already includes some structuring, as a schema and also some free text. It takes advantage of learning methods from structured data, although also needs methods applied on unstructured data to process free text.

Finally, unstructured data relates to text or data in its raw form, without any kind of organization nor processing. This kind of data is related to natural language texts and other kinds of data found in e-books, word, pdf documents, web pages, etc. The methods used to retrieve this kind of data does not rely in any kind of structured information, therefore they are supported by statistical or natural language process approaches. (Hazman et al., 2011)

### Ontology Learning State of the Art

The growth of Semantic Web increased the interest to develop methods that could ease the creation and maintenance of semantic resources as ontologies. The automatic learning of ontologies is yet an utopic task, however several researches provided approaches for semi-automatic methodologies for OL.

One example of an approach for OL commonly referred in academic documents is Text2Onto (Cimiano and Völker, 2005). Text2Onto is a tool for ontology learning from unstructured textual sources aimed for the extraction of ontologies from text documents. In particular, this tool targets the components of an ontology (concepts, taxonomical and non-taxonomical relations, and other properties), to whom are applied different algorithms. For instance, Text2Onto relies in machine learning techniques, to learn concepts.

OntoLearn (Velardi et al., 2005) is an OL system that provides a methodology also for ontology extraction from free text sources. Similar to Text2Onto, it likewise targets several steps in the OL cycle, such as term extraction, natural language definitions extraction, expert parsing of knowledge found and ontology mapping.

OntoEdit[[2]](#footnote-2) (Sure et al., 2002) is a tool aimed for OL from the Semantic Web that proposes a method composed by modules that serve different steps in the ontology engineering cycle. The main steps considered in the methodology of OntoEdit relies in requirements specifications, refinement and evaluation of resources extracted from web documents. In particular, this process includes extracting, pruning, refining, applying, importing and reusing data from web documents.

## Ontologies in Building and Construction Sector – E-Cognos project

The following lines will present a perspective related to the development initiatives of semantic sources by European institutions and companies. Special attention is given to the European Project in Knowledge systems aimed for building and construction sector (B&C), specifically the E-Cognos project.

### Historical perspective

Efforts were developed through the last years in Europe, focused in the research and development of controlled vocabularies aimed for B&C sector. Some initiatives include ICONDA®Bibliographic terminology (Fraunhofer, 1986), Industry Foundation Classes (IFC) model from buildingSMART (buildingSMART, 2015), British Glossary for the UK Construction sector (BS6100), bcBuildingDefinitions taxonomy (Lima et al., 2003c) and e-COGNOS ontology (El-Diraby et al., 2005). The initiatives were not limited to Europe. For instance, in North America they included Masterformat™ (Construction Specifications Institute and Construction Specifications Canada, 2015), OmniClass™ (OCCS Development Committee, 2006) standards, the Canadian Thesaurus and the ANSI/NISO Z39.19 standard for CVs from United States of America (N.I.S.O. (US) and others, 2005).

MasterFormat™ originally created in 1963, designed to satisfy the construction sector needs in North America related to a standard for construction specifications, and constructing and procurement requirements. Specifically, it is a list of numbers and titles aimed for organization. Initially, consisted of 16 divisions with 5 digits to represent each item. After 2004 it was heavily updated to 50 divisions with 8 digits representation. (Construction Specifications Institute and Construction Specifications Canada, 2015)



Figure 7 - Some examples of CV-focused initiatives in Europe and worldwide (Lima et al., 2007)

ICONDA®Bibliographic terminology created in 1986, is a publications database for the Construction sector. In 2014 included near 900000 records for international publications related to building domains, and grows with a referred rate of near 20000 records per year. (Fraunhofer IRB, 2015)

Omniclass™ is a standard to organize construction information, whose first version was released in 2006. Developed based in standards from ISO and ICIS subcommittees and workgroups from the early 1990s. The classification framework for Omniclass™ is based in the standard ISO12006-2. The basis for the tables origins from MasterFormat for work results, additionally its elements are derived from Uniformat™. Omniclass™ consists of 15 hierarchical tables representing different facets of construction information. (OCCS Development Committee, 2006)

IFC model, which has been developed by buildingSMART (formally known as International Alliance for Interoperability, IAI) since 1997, is an open standard for exchanging Building Information Model (BIM) data, registered under ISO16739:2013 by ISO. Currently, IFC is now under its fourth version (IFC4), released in 2013.(buildingSMART, 2015)

The bcBuildingDefinitions taxonomy was developed under eConstruct project, the main goal was to present the capabilities of the Building and Construction eXtensible Markup Language (bcXML). This taxonomy contains almost 3000 terms related to *doors* in six different languages. (Lima et al., 2003c)

Finally, the e-COGNOS ontology was a semantic resource developed under the e-COGNOS project, with the goal “*to support the consistent knowledge representation of construction knowledge items considering the e-COGNOS scenario.*” (Lima et al., 2003b) The ontology consists of two taxonomies, one for concepts and other for relations. In the next sub-section this project will be discussed in more detail, with focus on the ontology development, as it was one of the main inspirations for the development of this thesis.

It is worth noting that the previous presented initiatives do not, nor it was intention of the author to reflect the complete universe of the development and research projects related to CVs, on the contrary, they just represent a small sample. Nevertheless, these were the ones that directly or indirectly influenced the present study.

### Creation of an ontology in B&C – E-Cognos approach

Developed under a European consortium[[3]](#footnote-3) in 2001, e-COGNOS (Consistent knowledge management across prOjects and between enterpriSes in the construction domain – IST-2000-28671) was a project aimed for the management of Knowledge Resources tailored for the B&C industry sector. This project was created with one vision, particularly by assuming that different information and knowledge sources can be shared and used between several actors, in a co-operative approach.

The e-CKMI was the Knowledge Management Infrastructure developed for the IT-based perspective (two development perspectives were identified, IT and managerial) of e-COGNOS, which motivated the development of several components. One of its components and main pillars was the e-COGNOS ontology. E-COGNOS ontology was developed driven by the following: *a group of* ***Actors*** *uses a set of* ***Resources*** *to produce a set of* ***Products*** *following certain* ***Processes*** *within a work environment (****Related Domains****) and according to certain conditions (****Technical Topics****)*. (Lima et al., 2005)

The methodology proposed for the creation of the e-COGNOS ontology was the following (Figure 8):

* Definition of domain and scope;
* Reuse of ontology-related resources;
* Enumerate the important terms to the taxonomy;
* Define concepts and concept hierarchy based on the relation “*is-a*”;
* Define properties of the concepts;
* Define restrictions;
* Populate the ontology.



Figure 8 - The e-COGNOS ontology creation methodology (Lima et al., 2002)

The components considered in the e-COGNOS ontology were the following: (i) a glossary; (ii) a vocabulary; (iii) a classification system; (iv) a concept taxonomy; and (v) a definition taxonomy. (i) The glossary provides the words from the respective domain. It was adopted from BS6100, because its terms were widely accepted and they include a myriad of synonyms which is very rich. (ii) The vocabulary was an XML vocabulary adopted from bcXML which provided the base to build the bcBuildingDefinitions taxonomy. (iii) The classification needs was provided by ISO 12006-2 classification system. Although this standard did not provide a complete classification system, it provided an identification of classes and their relations, which are necessary for information organization purposes. (iv), (v) both taxonomies, concepts and definitions, were built based on two sources: O’CoMMA[[4]](#footnote-4) ontology and IFC Model. O’CoMMA provided an initial sample of concepts and the IFC Model also provided a list of concepts, to build the ontology. Later, IFC Model provided, as well, more concepts, attributes and relations to improve this ontology. In Figure 9 can be seen both taxonomies of e-COGNOS ontology, concept taxonomy and relations taxonomy.



Figure 9 - e-Cognos taxonomies a) Concepts; b) Relations (Costa, 2014)

The e-COGNOS ontology first version was created with 800 concepts. Later, achieved more than 17000 concepts with the help of bcXML language to import taxonomies into the e-COGNOS ontology.

3

# Pattern extraction from non-structured sources of information

## Data mining / Knowledge Discovery.

Several processes can be used for a system be able to recognize patterns and further extract knowledge from data and information.

Data Mining (also referred in literature as Knowledge Discovery in Databases or KDD) is one of them (Hand et al., 2001).

### Definition

Data mining allows experts to find knowledge in new data or data they already have. Additionally, by adopting data mining techniques, it is expected that decision makers can use new knowledge that otherwise could be unknown, unavailable or difficult to discover, to make better decisions. (Witten et al., 2011)

### Techniques used nowadays?

## Pattern Discovery

### FP-Growth – definition

Before rules of association can be found, the database must be mined to see which of the items are frequent. There are several processes in the academic community for this purpose who, given a set of database transactions can search it and return all the frequent item above some kind of measure to prove that represents the frequency of each item. Apriori, Eclat, FP-Growth are the ones that are most used and discussed by researchers. (Han et al., 2004) is recognized as a first introduction of the FP-Growth approach. It compares FP-Growth with Apriori, one of the initial and most used processes.

There are some characteristics that an algorithm should have to be classified as a good one. Namely time performance, usability in large databases and small databases, scalability, etc. In the next sub sections, the arguments are in favour of FP-Growth, in which the author of the present work identifies, based on the research, as being the best for the present case.

FP-Growth stands for Frequent Pattern Growth, it represents an algorithm to discover frequent patterns in data and specifically used in text mining. This algorithm is currently one of the fastest ones to mine association rules. It can also be defined as the first step in the path of item recognition that appears frequently. These items are called Frequent Patterns, meaning some text that appears in the database of transactions and are considered frequent above some minimum threshold value. This value is chosen by the engineer handling this process. It is more or less an arbitrary choice based on try and error method. More work can be done in this step to improve results, such that an artificial intelligence process could find a method to dynamic choose the best value for the intended use of the algorithm.

Before FP-Growth the processes to discover frequent patterns in databases of text were mainly Apriori-like based algorithms. Such processes are known to be very costly in large databases. Its times to search will exponential grow as the database will also grow. On the other side stands FP-Growth, a *divide-and-conquer method* (Han et al., 2004). It is based in a prefix tree representation, called FP-Tree. This tree holds the frequent patterns found in the transaction database. With the divide and conquer method this can be seen as a recursive elimination process. It will separate the frequent items from the ones that are not frequent inside a database.

Due to the popularity, effectiveness and performance of this algorithm, it was much appreciated in many investigations in the academic and scientific community. Also many changes proposals to the original were studied and presented. For example, in (Wang et al., 2002), the author proposes improvements and upgrades for the algorithm. One of these proposals is the Top Down FP-Growth algorithm. This work’s author debates a different process to search frequent patterns. It searches the FP-Tree from the top to the bottom and not generating conditional FP-Trees to each item. This method processes the nodes of the tree at upper levels before processing the ones on the lower levels. This is different from original FP-Growth, in which it mines the tree from bottom up, from the item to its prefixes, and creating several conditional trees for each item.

(Korczak and Skrzypczak, 2012) illustrate an example of discovering customer frequent patterns in an online store with the help of FP-Growth to discover association rules between the transactions of the customers. In other project, (Bonchi and Goethals, 2004) are inspired by the known small Japanese bonsai tree and tries to apply its broad concept in the FP-Tree of the FP-Growth algorithm. This study examines the reduction of the tree by a technique that is based on pruning specific “leaves” (nodes) resulting in smaller compressed trees.

Another study related to this subject is the one presented in (Zeng et al., 2010). In this paper, the discussion presented is a process to weight association rules based on an FP-tree. It proposes a new method called FP-Weighted Association Rules (FP-WAR) where outlines the importance of getting a technique to weight association rules and give them different *interestingness*.

### Algorithms to discover patterns Weaknesses/Strengths between them

[ECLAT, APRIORI, FP-GROWTH]

Although FP-Growth is a very efficient algorithm to find frequent patterns in databases, it is not the only one, nor even the first one to appear. In the last years, much study was conducted in frequent patterns recognition in data mining subjects of Artificial Intelligence area. Since the presentation of the problem of association rule mining in (Agrawal et al., 1993), many algorithms appeared in researches all claiming to be the best for some reason. For instance, APRIORI, that was one of the pioneers to address this situation and introduced in (Agrawal and Srikant, 1994), is a starting point for many studies in frequent pattern discovery. In this research, Agrawal. defines this algorithm as a procedure for candidate generation. These candidates are used to construct other candidates in the next level and frequent itemsets. One of the main problems recognized in Apriori by the scientific community (Han et al., 2004; Zaki, 2000) is the number of scans it uses to generate the frequent items from the candidate retrieved from the database. It performs as many searches in the database as the maximum element number in an itemset of candidates. Hence, as bigger the candidate sets are, lower is the performance of the algorithm. It starts to be even worse when the size of the database tends to be large, although it could discover the frequent items, it is a little boring to repeatedly search a large set of candidates by pattern matching. In the meantime several other attempts tried to improve Apriori algorithm. Some examples are MSApriori (Liu et al., 1999), A-Close (Pasquier et al., 1999), Apriori-Inverse (Koh and Rountree, 2005), UApriori (Metanat Hooshsadat et al., 2012) and many other Apriori-like based algorithms.

Similarly, ECLAT is another studied algorithm to find frequent itemsets in databases. ECLAT stands for Equivalence CLass Transformation. This algorithm was introduced in (Zaki, 2000) as one that would improve the performance problems of Apriori-based algorithms, like minimization of I/O costs reducing the number of database scans or event the reduction of the computation costs with more efficiently search procedures. ECLAT needs just a reduced number of scans in the database and no hash trees whatsoever as it generates frequent itemsets by only simple intersection operations. It can even handle support values lower than, for instance, Apriori in large datasets.

One of the advantages of FP-Growth, when comparing with the competitors is that it does not create huge amount of frequent itemsets and a small database of transactions. It only needs one scan on the database, along with a minimum support threshold to scan it and discover frequent itemsets. As observed in the previous lines, Apriori and most Apriori derived algorithms are candidate set generation algorithms, on the contrary FP-Growth is not. It does not need to make such a costly operation to generate frequent items, in contrast, it uses mining operations of count accumulation (frequency count) and prefix path count adjustment. These are less costly than candidate set generation and pattern matching operations.

(Borgelt, 2005) presents an interesting study on these three algorithms, Apriori, ECLAT and FP-Growth. It argues that the implementation of the process of frequent discovery in FP-Growth “*clearly outperforms Apriori and ECLAT*”. Even after the previous were improved and optimized.

In brief, one can see that FP-Growth have more strengths than weaknesses, when comparing to others algorithms. It recognizes frequent patterns in data, and needs less time to give the results than its competitors.

## Association Rules (Definition, Rules, Metrics)

### Metrics

As presented before, Association Rules algorithm recognizes associations in frequent patterns resulting from a frequent pattern recognition algorithm like FP-Growth or Apriori. The following step is responsible to evaluate the rules in a way that it will show interest to the subject. Several publications have appeared in the recent years identifying ways to measure the interest in an association rule. In the following lines, it will be discussed how association rules could be measured. It is also discussed what should be thought as interesting to retrieve from the rules, and the ways to do it., in the form of subjective and objective measures.

#### Subjectivity and objectivity

To be able to measure interest in the knowledge discovered, two types of measurement of the interest of a rule are identified: Subjective and objective measures. (Mackie, 1977) presents a study where he describes that the subjectivity in evaluation is very common when the evaluation goals are objects, actions or events. The objectivity is used in the measures themselves and their implementations. The subjectivity depends in great factor on the person that is considering the subject. As explained before, it depends in factors like the location or the background. Some other studies also discuss the subjectivity and objectivity of a measure. (Silberschatz and Tuzhilin, 1995) propose a classification for interestingness measurement of a rule. As one could see in Figure 3.1 where this classification is illustrated, it is argued by the author that the interest should be divided in *Objectivity* and *Subjectivity* measures.



Figure 3.1 – Interestingness measures types tree (adapted from Silberschatz and Tuzhilin, (1995))

Some of the challenges to evaluate the subjectivity of knowledge interestingness, and in this case, a rule, were already discussed in the beginning of this chapter. But it is important to go deeper in this subject. Silberschatz and Tuzhilin (1995) points out in their study that it is important to measure the subjectivity of a measure. In the subjectivity side of the classification tree in Figure 3.1, the author divides this subjectivity of a rule in two concepts, *Unexpectedness* and *Actionability*. The first concept represents the value of some unexpectedness or surprise in a rule when knowledge is discovered. If one could discover a rule that it would not expect, that rule would be interesting. Of course, some knowledge expected, is knowledge that is already known, and thus, not interesting for the user in this sense. The second concept of subjectivity, actionability, represents the usability that a rule could have. In other words, it is the capacity of a rule to be used in an interesting way by its user. One example of these concepts applied to a rule that is presented by Gonçalves (2005) and could help explain these kind of subjective measures in a rule is the association between dippers and beer in a big department store. This example explains that when the transactions are made by young couples on a Thursday, this association is detected. The company analysts would think that the act of buying beer would just be associated with the act of buying appetizers or barbecue meat and other alcohol drinks. Surprisingly, when association rules are discovered, this unexpected knowledge rises to the edge. This is the perfect example of an unexpected and actionable rule, and as a result, for now on, on Thursdays, the department store can use this extraordinary new and unexpected discovered knowledge to move the dippers and the beer closer to each other, so the sales of both could go higher.

Although these two concepts are independent of each other, they can be combined to strengthen even more one rule. Regularly the unexpected rules are also rules that are useful. Similarly, the actionability rules, the ones that an ontology engineer can do something useful with them, are also rules not expected to appear. If one thinks a little deeper, this makes sense. If the object of association rules were to result knowledge that was already known, what would be the point, or at least what should be done with this existing knowledge? Some thoughts on this will be discussed in the Ontology Management Chapter of the present document.

On the other side of the interestingness tree, are the objectivity measures. These measures statistically identify the strength of the association rules. It is important to know some characteristics that one would want in a measure. In this matters, (Tan et al., 2002) describes a list of several measures found in the literature and discuss some properties of a measure. In this work are presented some properties that the author defends that should be desirable and applied to the measure operation of association rules. Three properties are presented in the work as the more relevant, the first one state that if one has concepts A and B that are statistically independent, then the measure is equal to 0. This means that if a rule does not find any relation between the concepts these are not related. The second property presented states that a measure increases with the support of a rule, when probability P(A) and P(B) remains the same. And the third property presented as the considered desirable for the authors, describes that a measure decreases with P(A) (or P(B)) when the other parameters remain unchanged, namely the support, P(B), or P(A) respectively. Several more properties are presented in this work and the author examines each of the measures against each property. This is a good way to justify each of the measure considered.

#### The measures

On the next lines, based on the interestingness tree presented in Figure 3.1, the measures will be discussed and presented. All of them will be identified and discussed, namely support, confidence or conviction and lift, or even gain, Laplace and ps. The existence of some other measures will also be presented and discussed.

**Support and confidence**

The majority of the works studied about objective measures of association rules, take advantage of *support* and *confidence*. Hoque et al. (2011) which presents a document on association rules consider these two measures, presented also in Figure 3.1, good examples to help find interest in association rules. Furthermore, Azevedo et al. (2005), Bhujade and Janwe (2011), Brin et al. (1997), Kumar and Chadha (2012) and Spruit (2007) are also examples of researches where support and confidence are used to measure rules of association. They all use at least these two measures to extract knowledge and to evaluate the results of their knowledge discovery processes. Additionally, Bayardo and Agrawal (1999) also considers the use of these two measures as a way to reduce the rules to the most interesting. The former and Azevedo and Jorge (2007), Gonçalves (2005) and Tan et al. (2002) also present some definitions of each one of the measures. The rule *support* (also *frequency* or *coverage*) is equal to the number of occurrences in the corpus of data information where both concept A and concept B evaluates to true. This is presented as *sup(A⇒B)* (also defined in some literature as *σ(A⇒B)*). In other words, the support of a set of items, that is a statistic metric, is defined as a transaction percentage from a database where these items are included. As this is a statistic measure, the values are represented between. The support result is proportional to the frequency. Higher the value, the more frequent are the concepts in the database. The definition for support is represented in the following equation (1).

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

Furthermore, *confidence* represents an estimation of the probability of observation Concept B given Concept A. When a rule is received, one can immediately classify the relationship of the corresponding concepts. The expression to calculate confidence is given in Equation (2) and the result values, as this is also a statistic measure, are enclosed in . One can also identify that the interest rises also with confidence results.

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

These measures, although, alone present some but not enough information. To get the real interesting rules, one has to consider two additional parameters, *minsup* and *minconf*. These two parameters propose a lower limit on the interest of a rule. For instance, a rule can have a support value of 20%, however, if the defined minsup is 50% this rule is considered uninteresting. Bayardo and Agrawal (1999) argue the definition of some these borders. Their objective is to propose the discovering of the most interesting rules using these borders defined by minsup and minconf.

In the time of this research the author did not find an alternative for an automated process to choose this limits, as a result these values have to be an arbitrary choice. And two problems arise immediately when choosing these values. If the values chosen are too low, it could result in too much rules to analyse, and in redundant information, in contrast, if the values chosen are too high, the interest of the rules could be low as some of the knowledge is already known, resulting in expected and/or useless information. These values have to be wisely chosen, and in a balanced way, so that could select some interest from the data, and at same time select the most interesting knowledge. This choice could be done by an expert, like an ontology engineer, who has the knowledge to make little adjustments until the results are considered a good enough.

**Conviction and Lift**

Although support and confidence can give a real good and trustful interest measure results, they sometimes are not enough, as a result some other measures were studied and used in the scientific community. *Conviction* and *Lift* are other two measures that were proposed to complement the former, and that are also commonly used to strengthen the conclusions obtained from confidence and support measures. Also statically measures, these two depend on their values to be calculated.

*Conviction* is an implication measure that quantifies the value of the implication, it is represented as *A⇒B*, meaning that the direction of the rule is important for the interest measurement, hence *A⇒B ≠ B⇒A*. Conviction measure has some very interesting properties, such as if its value is equal to 1, this means that the concepts are considered totally independent from each other. Other property is that this measure considers the value of the antecedent as also the value for the consequent to calculate its value. Other interesting property of conviction is on rules with 100% confidence value, meaning where the antecedent always appears with the consequent, these rules will have the value equal to ∞. To achieve the most interesting rules one can think of as higher the value of Conviction, higher is the interest of that rule. The values of Conviction are included in . Conviction can be defined mathematically as presented in equation (3) or can be also presented as equation (4) dependant of confidence measure.

|  |  |  |
| --- | --- | --- |
|  |  | (3) |
|  |  | (4) |

In contrast to conviction that measures implication, *Lift*[[5]](#footnote-5), (can also be found in literature as *Interest* in (Brin et al., 1997) or as strength in (Dhar and Tuzhulin, 1993)) is a measure that quantifies the co-occurrence of a rule. Lift is not an implication measure, it means it is symmetric in relation to the antecedent and consequent, hence *Lift(A⇒B) = Lift(B⇒A)*, in other words it measures how far from independence are concepts A and B. Lift is defined as a measure to boost (“lift”) the confidence of a rule, this suggests an improvement of the trust of results of rule confidence. Similarly to conviction, if its value is 1 it means they are total independent without any kind of interesting relation, and as far from 1 and as higher the value is, higher will be the interest of the rule and more relation can be found on them. The set of values of this measure are included in . Lift is defined by the following Equation (5) or also defined in Equation (6) where one can see the dependency from confidence measure.

|  |  |  |
| --- | --- | --- |
|  |  | (5) |
|  |  | (6) |

As can be easily observed in both equations from conviction and lift there is a relation to confidence measure. Therefore, these measures can be understood as measures to help improve or strengthen the trust on confidence results where the confidence itself would not be enough to make the conclusions and find relevant knowledge in the association rules. In the case of Lift, the measure is better for rules with lower support.

**Gain, Laplace and PS**

In Figure 3.1 some more measures are illustrated. As one can see, they are presented as *Laplace*, *PS* and *Gain*. These three are also measures dependable of support.

Laplace is a classifier that is one of the additional measures considered in this research. It can be considered as a confidence estimator that is function of support, and as low as support is, lower is the interest in the rule considered. Laplace is normally used to rank rules by class. The range of values are in . Its mathematical definition is the following Equation (7). The constant *k* represents the number of classes defined when one is defining the respective classification model. Its value is always higher than 1.

|  |  |  |
| --- | --- | --- |
|  |  | (7) |

Another one of these measures is Gain. This is an optimization measure presented by Fukuda et al. (1996) and discussed by Bayardo and Agrawal (1999), and by Brin et al. (2003) as a proposal to solve the optimized gain rules problem. It is defined also as a function of support and given by the following Equation (8). The parameter is defined as a constant fraction with values between 0 and 1. Additionally, if one wants to decrease the subtractive term, it can be only done by decreasing the support of the antecedent. When this happens, the confidence value becomes higher.

|  |  |  |
| --- | --- | --- |
|  |  | (8) |

The last of these three measures presented is *PS*. This measure receives its name from their creators, Piatetsky-Shapiro (1991)[[6]](#footnote-6). It was originally used to classify rules, and later adopted by association rules. This measure is a boost to the support measure. As it gets a value in the range . If its value is equal to 0 it means that A and B are independent. A value below 0 represents a negative dependency and if the value is higher than 0 it is called positive dependent. Higher values represent more interest in the association rules. The definition for PS is presented in the following equation (9).

|  |  |  |
| --- | --- | --- |
|  |  | (9) |

**Other measures**

All the values that are illustrated should be enough to classify any association rule extracted, although, in the academic community several other measures were studied over the years. For instance, the interesting work in Tan et al. (2002) evaluates a list of 21 measures for association patterns, where the measures studied in the present research are also considered and evaluated. Some others like gini, entropy gain and chi-squared are also discussed in Bayardo and Agrawal (1999). Further research can be made in the direction of more measures to improve the association rules process reliability.

* + Algorithms to discover patterns [ECLAT, APRIORI, FP-GROWTH]
  + Weaknesses/Strengths between them
  + Why FP-Growth?
  + Application domain. (Practical cases where association rules are used)
  + Measures/metrics

4

# Concept Model

Chapter 4 – Concept Model

- Explain conceptual model/solution

- Describe an application example

From unstructured information to knowledge representation and ontology structure

- Dimensions included in the model???

- Enrichment process / Ontology learning process

FP-Growth how to build and FP-Tree

Association rule evaluation

- DER / MVC / UML Diagrams

5

# Model Design and Development

Chapter 5 – Model Design and Development (Proof of concept)

- Method proposal to address the question.

- What were the technologies used for the solution.

Technologies used,

- Implementation description.

(Present the server / front end solution)

- Include use cases (Relations discovered, new concepts discovered, etc.)

(Discover a relation between two concepts, update a relation between two concepts, and discover new concepts)

- Front end

Brief explanation of the functionality of the front end. Explain in a form of manual??

6

# Assessment

Chapter 6 – Assessment

* Present list of relations discovered and discuss them
* Present new concepts discovered

7

# Conclusion and Future directions

Through the following lines will be presented an overview of the work developed in this thesis. The objectives were defined in Chapter 1 that intended to guide the path of the study. For these objectives it will be described which ones were achieved and which ones were not, describing also the problems and difficulties found during the development and research, and also, how were these difficulties solved. Similarly, this will also bring to attention some possible future research topics, where achievements addressed by this work can constitute a solid basis.

## Work overview

As presented through this dissertation, it was described the importance of concept representation in contrast to word representation, in the Semantic Web area. It was highlighted the advantage of the use of mechanisms (e.g.. Controlled Vocabularies), as these provides means for semantic representation, which allows more than just simple word representation. These mechanisms provide the possibility to make semantic search in contrast to the currently used search technology, providing results more adapted to what users want.

Similarly, frequent pattern discovery in texts may enhance the recognition of semantic relations between words. As a result this recognition can help discover the meaning associated to a word. The Data Mining techniques adopted to achieve this task were FP-Growth to discover frequent patterns and Association Rules to provide more than the just recognition of relations between the words (refer to chapter XX.XX). Based in the AR algorithm, this work demonstrated that it is possible to measure the strength of a relation.

Pattern recognition by itself can be an indicator of relations between words, however this can be enhanced through the use of a domain ontology. In this sense, this work proposed Frequent Itemset Mapping, a process to match frequent items discovered in a document corpus and keywords associated to concepts from a domain ontology related to B&C (refer to chapter 4).

As explained in Chapter 2.3, Ontology Learning is the area related to the automatic or semi-automatic (meaning without human supervision) maintenance of an ontology. Through newly discovered knowledge sources it is possible to learn a domain ontology, in the sense that one can use this new knowledge that could otherwise be unknown, be difficult to discover or be unavailable to improve concept relations inside the ontology. The method proposed in this dissertation, based in the AR algorithm, provides metrics in the form of numeric values to evaluate the strength of semantic relations between concepts. Through these value is possible to know if a concept A is more related to a concept B than to a concept C, therefore learn or maintain the concepts from a domain ontology, as the one used in this work related to B&C.

## 7.2 Research Contributions

The development of this work proposed four expected outcomes in Chapter 1 as follows:

* To develop a method to describe how to extract concepts and recognize relations between them from a data document corpus, and to find new knowledge sources in order to update a domain ontology.

The proposed method relies on applying Data Mining techniques to discover knowledge in documents that could be useful to update a domain ontology. Knowledge, meant the discovery of new concepts, relations or the improvement of the relations between the concepts already in a domain ontology (e.g. the ontology used in this work adopted from the B&C domain). The initial resources were a set of documents from the ICONDA[[7]](#footnote-7) database and a domain ontology adopted from SEKS framework developed under a MSc. Dissertation (Figueiras, 2012) both related to B&C domain. The documents were initially processed in the Rapidminer software tool. Rapidminer proved to be a satisfactory tool, as it also allowed to apply the algorithms FP-Growth to discover frequent patterns and Association Rules to discover the relations. The process created for the matching between the frequent items discovered in the documents and the keywords associated to the concepts from the domain ontology was the Frequent Itemset Matching (refer to chapter XX.XX). This process allowed to search through the ontology in order to verify if the frequent items discovered in the documents were associated to any concept inside the ontology, or if it originated new knowledge.

This work tries to develop a method for Ontology Learning where it is possible to turn a domain ontology more up to date. Even with a small sample, this process provided some good results (refer to chapter XX.XX), as it discovered new concepts, and also provided some interesting relations between the concepts. However, OL relies on automatic methods, this work did not intended to provide a full automatic method to learn an ontology. Alternatively, it was intended to develop a semi-automatic method that relies in human interaction to complete the OL task with the knowledge discovered through all the results.

* To develop a proof of concept, a software platform, based in the previous method in order to reflect the application of the studied techniques.

In order to execute all the steps from the method proposed it was developed a software tool, DOKS (refer to chapter XX.XX). DOKS is a client-server application developed using Java technology to implement all the processes and components in this tool. To interact with the ontology, it was used Jena API. The communication to the database was made by JavaBeans technology. The ontology was developed in OWL. Rapidminer provided an API to access its results, and they were exported through a script represented in Groovy. To hold the results for later access, it was created an XML message. Both DBs for the ontology and for the AR results were saved in a MySQL RDBMS.

* Present results of the semi-automatic OL process. Results composed by patterns discovered in the documents, their relations and the new concepts discovered. They should be presented in an understandable way to the user.

To present the results from DOKS, a FrontEnd was implemented in web technology. Here the set of technologies used were: (i) Html5+CSS3 as a base to support the layout; (ii) The communication with the server was made through HTTP requests based on PHP technology to send the results; (iii) To present the results in the web page, the technology chosen was PHP + XPath. The results were presented in a first page, in which the user could choose two concepts, based on the Frequent Itemset Mapping, and the values of the metrics from each association rule presented by the FrontEnd. This way, a relation between two concepts could be chosen for later processing. It is worth mentioning, the creation of a colour scheme for the Frequent Itemset Mapping process, in order to help the user choose the concept from the domain ontology that best matches the frequent item.

* Finally, publication of scientific documents about the work, to be assessed by the academic community.

The following scientific documents were published after assessment by the academic community during the development of this work:

* Luis Paiva, Ruben Costa, Paulo Figueiras, Celson Lima, “Discovering Semantic Relations from Unstructured Data for Ontology Enrichment - Association rules based approach”, 8ª Conferência Ibérica de Sistemas e Tecnologias de Informação: CISTI'2013, pp 579-584, 2013
* Ruben Costa, Paulo Figueiras, Luis Paiva, Ricardo Jardim-Gonçalves, Celson Lima, “Capturing Knowledge Representations Using Semantic Relationships An Ontology-based Approach”, Sixth International Conference on Advances in Semantic Processing: SEMAPRO 2012, pp 75-81, 2012
* Paulo Figueiras, Ruben Costa, Luis Paiva, Ricardo Jardim-Gonçalves, Celson Lima, “Information Retrieval in Collaborative Engineering Projects-A Vector Space Model Approach”, International Conference on Knowledge Engineering and Ontology Development: KEOD2012, pp 233-238, 2012

## 7.3 Future Directions

As this work relates to some areas from Semantic Web and Ontology Engineering, some possible directions can be identified for further work and improvement. Two paths are proposed, one related to the improvement of the presented method, the second related to its applicability and reuse.

Sometimes, the knowledge that results from the method proposed herein can be huge, and if the process is not fully automated it can be an exhaustive task to analyse these results. This suggests further research related to DOKS ability to deal with the size growth of data used in the Ontology Learning process, can be identified in three areas: (i) speed to process large sets of data as it can be really slow. Research can be taken in methods to, for instance, take advantage of multi-core processor technology in order to use parallelization techniques to improve the speed of the matching process; (ii) way to present results for evaluation by an expert, although this work provided a colour scheme to represent the strength of the matching process (refer to chapter XX.XX). This means to improve the way in which the results are shown, by using more graphics (e.g. graphs to represent relations). This will provide a better efficiency of the method itself and allow for faster reasoning of the results; (iii) method to process large/huge and complex sets of data, also known as Big Data. Big Data is the nouvelle sub domain of Data Mining that studies solutions to the problem of big and complex sets of data.

Searching for patterns in a document, was proved by this work that it is not an easy task, although it is possible. The relation between words in a document can lead to the discovery of a central concept or idea that could represent its context or domain, for instance a document including the words “*bridges*” and “*buildings*”. The central concept from this document could be identified as “*Civil Engineering*”. However, how can one discover the central idea in a document? Is it even possible? Can this discovery be done? How to find the central idea? Syntactic Context[[8]](#footnote-8) or Latent Semantic Analysis[[9]](#footnote-9) are areas that tries to address this questions, and can be a promising future direction.

It is worth mentioning that the intention of this research was not to develop a fully functional model to deal with data mining. However, the author thinks that it could be a good contribution to the following areas:

* Information Systems: Search engines like Google, Bing or Yahoo, just to name a few, could use semantic search capabilities to improve its results instead of just statistical ones. For instance, if one would like to search for a car, the search engine could provide the pages where the word “*car”* appears, and also the pages where the word *“automobile”* appears based on the relation between these two words. This means that one could search for a concept, instead of having to know every word that represents it. Additionally, it could also provide suggestions, for instance, related to their brands, turning the search results more close of what the user really searched for.
* Cybersecurity: This is an area of great interest in the present, based on several world events related to terrorism. The method proposed in this work, could help in this area, for instance, if one could use a search engine to search in the web for a person A that could be known as related to Al-Qaeda. After using the method proposed, one could also discover a person B that appears frequently in some pages related to person A, although not directly related to Al-Qaeda. This could be a proof of the relation between both people, and the discovery of the relation of person B to Al-Qaeda.
* Cybersecurity and human rights: MEMEX is a project from DARPA with an initial goal of using search technology to help fight human trafficking, as they identified this as a serious problem to solve. The secondary goal of this project was identified as to improve the search mechanisms and tools that are used today. Semantic search could help in the sense that it could discover pages with terms related to human traffic, for example *human trade* or *modern slavery*, which could represent the same idea.
* Team Sports: GloballCoach is a software tool idealized by former Liverpool and Chelsea coach, Rafael Benitez, that targets Football Teams and their coaches, providing them data analysis capabilities. Amongst others, this software aims to recognize patterns in game data to show to team players and improve their tactical and technical behaviour. This kind of software systems are focused in the individual behaviours of each player. In this sense, the method proposed in this work could be a great aid in the recognition of relations between the players, augmenting the analysis from an individual up to team perspective analysis capabilities. For instance, the coach could analyse which is the best relation in their left side. Meaning that if he wants to select player A for a game, he could know if the relation between players A and B provides more goals than with A and C. Other example could be, to whom a player A provides more assistances[[10]](#footnote-10), meaning that the relation between player A and B provides more goals to the team than a relation between player A and C.

Summing up, the semantic search is here to stay and is spreading along all research areas. In this sense, controlled vocabularies are useful tools to enhance semantic search capabilities in information systems. Ontologies themselves are great mechanisms to provide search capabilities to users, experts or not, in theirs daily search quests. Consequently, knowing how to provide or obtain the best results will throw companies or entities one technological step ahead of their competitors.

8

# Bibliography

Agrawal, R., Imieliński, T., Swami, A., 1993. Mining Association Rules Between Sets of Items in Large Databases, in: SIGMOD ’93. ACM, New York, NY, USA, pp. 207–216. doi:10.1145/170035.170072

Agrawal, R., Srikant, R., 1994. Fast algorithms for mining association rules, in: Proc. of 20th Intl. Conf. on VLDB. pp. 487–499.

Allen, R.E., Mannion, J., 2007. Oxford Mini School Dictionary & Thesaurus. Oxford University Press.

Almeida, T., Souza, R.F. de, 2011. O vocabulário controlado como instrumento de organização e representação da informação na FINEP [WWW Document]. URL http://repositorio.ibict.br/handle/123456789/88 (accessed 2.6.15).

Antunes, J.P.D., 2010. Design and implementation of an autonomous, proactive, and reactive software infrastructure to help improving the management level of projects.

Aquino, M.C., 2007. Hipertexto 2.0, folksonomia e memória coletiva: um estudo das tags na organização da web. Rev. E-Compós 18.

Azevedo, P.J., Jorge, A.M., 2007. Comparing Rule Measures for Predictive Association Rules, in: Kok, J.N., Koronacki, J., Mantaras, R.L. de, Matwin, S., Mladenič, D., Skowron, A. (Eds.), Machine Learning: ECML 2007, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 510–517.

Azevedo, P.J., Silva, C.G., Rodrigues, J.R., Loureiro-Ferreira, N., Brito, R.M.M., 2005. Detection of Hydrophobic Clusters in Molecular Dynamics Protein Unfolding Simulations Using Association Rules, in: Oliveira, J.L., Maojo, V., Martín-Sánchez, F., Pereira, A.S. (Eds.), Biological and Medical Data Analysis, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 329–337.

Bayardo, R.J., Jr., Agrawal, R., 1999. Mining the Most Interesting Rules, in: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’99. ACM, New York, NY, USA, pp. 145–154. doi:10.1145/312129.312219

Bhujade, V., Janwe, N.J., 2011. Knowledge Discovery in Text Mining Technique Using Association Rules Extraction, in: 2011 International Conference on Computational Intelligence and Communication Networks (CICN). Presented at the 2011 International Conference on Computational Intelligence and Communication Networks (CICN), pp. 498–502. doi:10.1109/CICN.2011.104

Borgelt, C., 2005. An Implementation of the FP-growth Algorithm, in: Proceedings of the 1st International Workshop on Open Source Data Mining: Frequent Pattern Mining Implementations. ACM, pp. 1–5.

Brin, S., Motwani, R., Ullman, J.D., Tsur, S., 1997. Dynamic Itemset Counting and Implication Rules for Market Basket Data, in: Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data, SIGMOD ’97. ACM, New York, NY, USA, pp. 255–264. doi:10.1145/253260.253325

Brin, S., Rastogi, R., Shim, K., 2003. Mining optimized gain rules for numeric attributes. Knowl. Data Eng. IEEE Trans. On 15, 324–338.

buildingSMART, 2015. IFC Overview summary — Welcome to buildingSMART-Tech.org [WWW Document]. URL http://www.buildingsmart-tech.org/specifications/ifc-overview (accessed 4.19.15).

Buitelaar, P., Cimiano, P., Magnini, B., 2005. Ontology learning from text: An overview.

Cimiano, P., Mädche, A., Staab, S., Völker, J., 2009. Ontology learning, in: Handbook on Ontologies. Springer, pp. 245–267.

Cimiano, P., Völker, J., 2005. Text2Onto, in: Natural Language Processing and Information Systems. Springer, pp. 227–238.

Construction Specifications Institute, Construction Specifications Canada, 2015. MasterFormat: Introduction & Guides [WWW Document]. URL http://www.masterformat.com/about/history/ (accessed 4.18.15).

Costa, R., 2014. Semantic Enrichment of Knowledge Sources Supported by Domain Ontologies. Faculty of Science and Technology - New University of Lisbon, Lisbon.

De Nicola, A., Missikoff, M., Navigli, R., 2009. A software engineering approach to ontology building. Inf. Syst. 34, 258–275. doi:10.1016/j.is.2008.07.002

Dhar, V., Tuzhulin, A., 1993. Abstract-driven pattern discovery in databases. IEEE Trans. Knowl. Data Eng. 5, 926–938. doi:10.1109/69.250075

El-Diraby, T.A., Lima, C., Feis, B., 2005. Domain taxonomy for construction concepts: toward a formal ontology for construction knowledge. J. Comput. Civ. Eng. 19, 394–406.

Elsayed, A., El-Beltagy, S.R., Rafea, M., Hegazy, O., 2007. Applying data mining for ontology building. Proc ISSR.

Figueiras, P.A., 2012. A framework for supporting knowledge representation – an ontological based approach.

Fraunhofer IRB, 2015. Fraunhofer IRB | ICONDA©Bibliographic - Find and Access Publications on Planning and Building [WWW Document]. URL http://www.irb.fraunhofer.de/iconda/login/ICONDA/iconda-start-info.jsp (accessed 4.19.15).

Fraunhofer, I.R.B., 1986. ICONDA Bibliographic-Find and Access Publications on Planning and Building.

Fukuda, T., Morimoto, Y., Morishita, S., Tokuyama, T., 1996. Data mining using two-dimensional optimized association rules: Scheme, algorithms, and visualization. ACM SIGMOD Rec. 25, 13–23.

Gandon, F., Dieng-Kuntz, R., Corby, O., Giboin, A., 2002. Semantic Web and Multi-Agents Approach to Corporate Memory Management, in: Musen, M.A., Neumann, B., Studer, R. (Eds.), Intelligent Information Processing, IFIP — The International Federation for Information Processing. Springer US, pp. 103–115.

Gangemi, A., Presutti, V., 2009. Ontology Design Patterns, in: Staab, S., Studer, R. (Eds.), Handbook on Ontologies, International Handbooks on Information Systems. Springer Berlin Heidelberg, pp. 221–243.

Gargouri, Y., Lefebvre, B., Meunier, J., 2003. Ontology maintenance using textual analysis, in: Proc. 7TH World Multi Conference on Systemics, Cybernetics and Informatics, USA. List of Figures Figure. Citeseer.

Gonçalves, E.C., 2005. Regras de associação e suas medidas de interesse objetivas e subjetivas. INFOCOMP J. Comput. Sci. 4, 26–35.

Gruber, T.R., 1993. A translation approach to portable ontology specifications. Knowl. Acquis. 5, 199–220. doi:10.1006/knac.1993.1008

Hand, D.J., Mannila, H., Smyth, P., 2001. Principles of Data Mining. MIT Press.

Han, J., Pei, J., Yin, Y., Mao, R., 2004. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. Data Min. Knowl. Discov. 8, 53–87.

Hazman, M., El-Beltagy, S.R., Rafea, A., 2011. A survey of ontology learning approaches. database 7, 6.

Hoque, A.M.S., Mondal, S.K., Zaman, T.M., Barman, P.C., Bhuiyan, M.., 2011. Implication of association rules employing FP-growth algorithm for knowledge discovery, in: 2011 14th International Conference on Computer and Information Technology (ICCIT). Presented at the 2011 14th International Conference on Computer and Information Technology (ICCIT), pp. 514–519. doi:10.1109/ICCITechn.2011.6164843

IBM - International Business Machines, 1996. IBM Intelligent Miner User’s Guide, Version 1 Release 1. SH12-6213-00 edition, July.

Kashyap, V., 1999. Design and creation of ontologies for environmental information retrieval, in: Proceedings of the 12th Workshop on Knowledge Acquisition, Modeling and Management. Citeseer, pp. 1–18.

Koh, Y.S., Rountree, N., 2005. Finding Sporadic Rules Using Apriori-Inverse, in: Ho, T.B., Cheung, D., Liu, H. (Eds.), Advances in Knowledge Discovery and Data Mining, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 97–106.

Kumar, V., Chadha, A., 2012. Mining association rules in student’s assessment data. Int. J. Comput. Sci. Issues 9, 211–216.

Lavrač, N., Flach, P., Zupan, B., 1999. Rule evaluation measures: A unifying view. Springer.

Lei, Y., Uren, V., Motta, E., 2006. SemSearch: A Search Engine for the Semantic Web, in: Staab, S., Svátek, V. (Eds.), Managing Knowledge in a World of Networks, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 238–245.

Lima, C., 2004. Final draft CWA4 proposal “European eConstruction Ontology “version 2004–03–26, in: Workshop on eConstruction N.

Lima, C., El-Diraby, T., Stephens, J., 2005. Ontology-based optimization of knowledge management in e-construction. J. IT Constr. 10, 305–327.

Lima, C., Fiès, B., Lefrançois, G., Diraby, T., 2003a. The challenge of using a domain Ontology in KM solutions: the e-COGNOS experience. Presented at the International Conference on Concurrent Engineering: Research and Applications, Funchal - Portugal, pp. 771–778.

Lima, C., Fiès, B., Zarli, A., Bourdeau, M., Wetherill, M., Rezgui, Y., 2002. Towards an IFC-enabled ontology for the Building and Construction Industry: the e-COGNOS approach. Presented at the European Conference of Information and Communication Technology Advances and Innovation in the Knowledge Society, Salford, UK, pp. 254–264.

Lima, C., Fies, B., Zarli, A., El-Diraby, T., Ferneley, E., 2003b. The E-Cognos Project: Current Status and Future Directions of an Ontology-Enabled IT Solution Infrastructure Supporting Knowledge Management in Construction, in: Construction Research Congress. American Society of Civil Engineers, pp. 1–8.

Lima, C., Stephens, J., Böhms, M., 2003c. The bcXML: supporting eCommerce and knowledge management in the construction industry [WWW Document]. URL http://www.itcon.org/cgi-bin/works/Show?2003\_22 (accessed 9.6.14).

Lima, C., Zarli, A., Storer, G., 2007. Controlled Vocabularies in the European Construction Sector: Evolution, Current Developments, and Future Trends, in: BSc, G.L., MSc, BEng, R.C. (Eds.), Complex Systems Concurrent Engineering. Springer London, pp. 565–574.

Liu, B., Hsu, W., Ma, Y., 1999. Mining Association Rules with Multiple Minimum Supports, in: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’99. ACM, New York, NY, USA, pp. 337–341. doi:10.1145/312129.312274

Liu, K., Hogan, W.R., Crowley, R.S., 2011. Natural Language Processing methods and systems for biomedical ontology learning. J. Biomed. Inform., Ontologies for Clinical and Translational Research 44, 163–179. doi:10.1016/j.jbi.2010.07.006

Mackie, J., 1977. Ethics: Inventing right and wrong. Penguin UK.

Maedche, A., Staab, S., 2001. Learning Ontologies for the Semantic Web. Presented at the Semantic Web Workshop 2001, Hong Kong.

Metanat Hooshsadat, SAMANEH BAYAT, PARISA NAEIMI, MAHDIEH S. MIRIAN, OSMAR R. ZA?ANE, 2012. UAPRIORI: AN ALGORITHM FOR FINDING SEQUENTIAL PATTERNS IN PROBABILISTIC DATA, in: Uncertainty Modeling in Knowledge Engineering and Decision Making, World Scientific Proceedings Series on Computer Engineering and Information Science. WORLD SCIENTIFIC, pp. 907–912.

N.I.S.O. (US), others, 2005. Guidelines for the construction, format, and management of monolingual controlled vocabularies. NISO Press.

Noy, N.F., Musen, M.A., 2004. Ontology versioning in an ontology management framework. IEEE Intell. Syst. 19, 6–13. doi:10.1109/MIS.2004.33

OCCS Development Committee, 2006. OmniClass: A Strategy for Classifying the Built Environment [WWW Document]. URL http://www.omniclass.org/index.asp (accessed 4.18.15).

Oxford University Press, 2012. Oxford Essential Portuguese Dictionary. Oxford University Press.

Oxford University Press, 2006. Oxford Dictionary of English. Oxford University Press, London.

Parada, V.M.M., 2010. Desenho e implementação de um sistema computacional para apoiar a gestão de projectos utilizando técnicas de data mining.

Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L., 1999. Discovering Frequent Closed Itemsets for Association Rules, in: Proceedings of the 7th International Conference on Database Theory, ICDT ’99. Springer-Verlag, London, UK, UK, pp. 398–416.

Piatetsky-Shapiro, G., 1991. Discovery, analysis and presentation of strong rules. Knowl. Discov. Databases 229–238.

Pouchard, L., Ivezic, N., Schlenoff, C., 2000. Ontology engineering for distributed collaboration in manufacturing, in: Proceedings of the AIS2000 Conference. Citeseer.

Reinberger, M.-L., Spyns, P., 2005. Unsupervised text mining for the learning of dogma-inspired ontologies. Ontol. Learn. Text Methods Appl. Eval. 29–43.

Sánchez, D., 2010. A methodology to learn ontological attributes from the Web. Data Knowl. Eng. 69, 573–597.

Semafora Systems, 2012. Semafora systems: OntoStudio [WWW Document]. URL http://www.semafora-systems.com/en/products/ontostudio/ (accessed 4.18.15).

Silberschatz, A., Tuzhilin, A., 1995. On subjective measures of interestingness in knowledge discovery., in: KDD. pp. 275–281.

Spruit, M., 2007. Discovery of association rules between syntactic variables, in: Proceedings of the 17th Meeting of Computational Linguistics in the Netherlands. Citeseer.

Stanford Center for Biomedical Informatics Research, 2011. The Protègè Ontology Editor and Knowledge Acquisition System [WWW Document]. URL http://protege.stanford.edu/ (accessed 4.4.15).

Stick-iSchool, 2013. Innovation Ontolgy [WWW Document]. URL http://stick.ischool.umd.edu/newsite/innovation\_ontolgy (accessed 3.24.15).

Sure, Y., Erdmann, M., Angele, J., Staab, S., Studer, R., Wenke, D., 2002. OntoEdit: Collaborative Ontology Development for the Semantic Web, in: Horrocks, I., Hendler, J. (Eds.), The Semantic Web — ISWC 2002, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 221–235.

Tan, P.-N., Kumar, V., Srivastava, J., 2002. Selecting the Right Interestingness Measure for Association Patterns, in: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’02. ACM, New York, NY, USA, pp. 32–41. doi:10.1145/775047.775053

Uschold, M., Gruninger, M., 1996. Ontologies: Principles, methods and applications. Knowl. Eng. Rev. 11, 93–136.

Velardi, P., Navigli, R., Cucchiarelli, A., Neri, F., 2005. Evaluation of OntoLearn, a methodology for automatic learning of domain ontologies. Ontol. Learn. Popul.

W3C, 2004. OWL Web Ontology Language Semantics and Abstract Syntax Section 2. Abstract Syntax [WWW Document]. URL http://www.w3.org/TR/owl-semantics/syntax.html#2.3 (accessed 4.12.15).

Witten, I.H., Frank, E., Hall, M.A., 2011. Data Mining: Practical Machine Learning Tools and Techniques, 3rd ed. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Yahoo, 2015. Yahoo [WWW Document]. Yahoo. URL https://www.yahoo.com/ (accessed 3.22.15).

Zaki, M.J., 2000. Scalable algorithms for association mining. IEEE Trans. Knowl. Data Eng. 12, 372–390. doi:10.1109/69.846291

Zhang, G.-Q., Troy, A.D., Bourgoin, K., 2006. Bootstrapping ontology learning for information retrieval using formal concept analysis and information anchors, in: Proc. 14th Int’l Conf. Conceptual Structures (ICCS’06). Citeseer.

Zhou, L., 2007. Ontology learning: state of the art and open issues. Inf. Technol. Manag. 8, 241–252.

1. Bits (also referred as binary digits) are the basic units in a digital system. They commonly can have values of 0 or 1 in which are used to represent data. [↑](#footnote-ref-1)
2. OntoEdit is now OntoStudio, a commercial product from Semafora Systems (Semafora Systems, 2012)

   [↑](#footnote-ref-2)
3. The Consortium included R&D organizations, namely University of Salford (UK) and Centre Scientific et Technique du Bâtiment - CSTB (France), as well as end users, particularly, HOCHTIEF (Germany), OTH (France), YIT (Finland) and Taylor Woodrow (UK). [↑](#footnote-ref-3)
4. O’CoMMA is a public ontology from CoMMA project. This ontology includes 470 concepts in a taxonomy, 79 relations in a taxonomy, 715 terms in English and 699 in French to label the primitives, and finally 550 and 547 definitions in English and French respectively. (Gandon et al., 2002) [↑](#footnote-ref-4)
5. As a curiosity, Lift is well known in the scientific community as a measure used in the IBM’s Intelligent Miner (IBM - International Business Machines, 1996). [↑](#footnote-ref-5)
6. In the literature *PS* is also found under different designations, for instance, *Leverage* (Azevedo and Jorge, 2007), *Rule Interest* (Gonçalves, 2005) or *novelty* (Lavrač et al., 1999). [↑](#footnote-ref-6)
7. ICONDA is a large database of technical documents related to B&C domain. [↑](#footnote-ref-7)
8. Syntatic Context relates to the order of the words in a sentence, and states that through language rules, one can infer the context of a sentence. [↑](#footnote-ref-8)
9. Latent Semantic Analysis is the area that analyse the relations between documents, trying to find correspondence between its terms and concepts in order to infer its context. [↑](#footnote-ref-9)
10. Assistance is the word used in a football game and represents the moment when a player passes the ball to a teammate, and this teammate scores a goal. [↑](#footnote-ref-10)