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)

knowledgeWhat 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](#_Toc424751413)

[Figures IV](#_Toc424751414)

[Tables VI](#_Toc424751415)

[Symbols and Notation VII](#_Toc424751416)

[1 Introduction 1](#_Toc424751417)

[1.1 Challenges 4](#_Toc424751418)

[1.2 Expected Outcomes 4](#_Toc424751419)

[1.3 Context of work 5](#_Toc424751420)

[1.4 Document Structure 6](#_Toc424751421)

[2 Controlled Vocabularies 9](#_Toc424751422)

[2.1 Controlled Vocabularies – Definition 9](#_Toc424751423)

[2.1.1 Problems Addressed by a CV 11](#_Toc424751424)

[2.1.2 Advantages / Disadvantages of CV 12](#_Toc424751425)

[2.1.3 Types of CVs – Differences, strengths and characteristics 14](#_Toc424751426)

[2.2 Ontology, a Definition 17](#_Toc424751427)

[2.2.1 Ontology Purpose 18](#_Toc424751428)

[2.2.2 Ontology Engineering & Components 19](#_Toc424751429)

[2.2.3 Ontology Languages 21](#_Toc424751430)

[2.3 Ontology Learning 22](#_Toc424751431)

[2.3.1 Problems related to the maintenance of an Ontology 22](#_Toc424751432)

[2.3.2 Definition of Ontology Learning 23](#_Toc424751433)

[2.3.3 Ontology Learning State of the Art 24](#_Toc424751434)

[2.4 Ontologies in Building and Construction Sector – E-Cognos project 25](#_Toc424751435)

[2.4.1 Historical perspective 25](#_Toc424751436)

[2.4.2 Creation of an ontology in B&C – E-Cognos approach 27](#_Toc424751437)

[3 Pattern extraction from unstructured sources of information 31](#_Toc424751438)

[3.1 Definitions 31](#_Toc424751439)

[3.2 Pattern Discovery and Knowledge Extraction 32](#_Toc424751440)

[3.2.1 Data Mining / Knowledge Discovery in Databases. 33](#_Toc424751441)

[3.3 Pattern discovery - Techniques 35](#_Toc424751442)

[3.3.1 Apriori 36](#_Toc424751443)

[3.3.2 ECLAT 37](#_Toc424751444)

[3.3.3 FP-Growth 38](#_Toc424751445)

[3.3.4 Algorithm comparison 40](#_Toc424751446)

[3.3.5 Improvements Attempts 41](#_Toc424751447)

[3.4 Association Rules – Pattern Evaluation 42](#_Toc424751448)

[3.4.1 Association Rules - State of the art 44](#_Toc424751449)

[3.4.2 Interest Evaluation in Association Rules 46](#_Toc424751450)

[3.4.3 Subjective measures 48](#_Toc424751451)

[3.4.4 Objective measures 49](#_Toc424751452)

[4 Concept Model 55](#_Toc424751453)

[4.1 Conceptual & Technical Architectures 55](#_Toc424751454)

[5 Model Design and Development 60](#_Toc424751455)

[5.1 Tools and Technologies 61](#_Toc424751456)

[5.2 Requirements 63](#_Toc424751457)

[5.3 Implementation and Development 64](#_Toc424751458)

[5.3.1 Document Analysis 65](#_Toc424751459)

[5.3.2 FP-Growth 68](#_Toc424751460)

[5.3.3 Association Rules 68](#_Toc424751461)

[5.3.4 Frequent Itemset Mapping 68](#_Toc424751462)

[5.3.5 Ontology Enrichment 70](#_Toc424751463)

[5.4 Front end 71](#_Toc424751464)

[6 Assessment 76](#_Toc424751465)

[7 Conclusion and Future directions 81](#_Toc424751466)

[7.1 Work overview 81](#_Toc424751467)

[7.2 Research Contributions 82](#_Toc424751468)

[7.3 Future Directions 85](#_Toc424751469)

[8 Bibliography 89](#_Toc424751470)

# Figures

[Figure 2.1 - Word search example (Yahoo, 2015) 10](#_Toc424751386)

[Figure 2.2 - Vocabulary Example 14](#_Toc424751387)

[Figure 2.3 - Page from a Dictionary (Oxford University Press, 2006) 14](#_Toc424751388)

[Figure 2.4 - 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%20v14.docx#_Toc424751389)

[Figure 2.5 - Page from Oxford Mini School Dictionary & Thesaurus (Allen and Mannion, 2007) 16](#_Toc424751390)

[Figure 2.6 - Domain Ontology example (Innovation Ontology adapted from (Stick-iSchool, 2013) ) 16](#_Toc424751391)

[Figure 2.7 - Some examples of CV-focused initiatives in Europe and worldwide (Lima et al., 2007) 26](#_Toc424751392)

[Figure 2.8 - The e-COGNOS ontology creation methodology (Lima et al., 2002) 28](#_Toc424751393)

[Figure 2.9 - e-Cognos taxonomies a) Concepts; b) Relations (Costa, 2014) 29](#_Toc424751394)

[Figure 3.1 - Data mining—searching for knowledge (interesting patterns) in data. (Han et al., 2011) 33](#_Toc424751395)

[Figure 3.2 - Data Mining Process – Steps from Data to Knowledge 35](#_Toc424751396)

[Figure 3.3 - Performance comparison of Apriori, Eclat and FP-Growth (Garg and Kumar, 2013). 40](#_Toc424751397)

[Figure 3.4 – Classification of interestingness measures (adapted from Silberschatz and Tuzhilin, (1995)) 48](#_Toc424751398)

[Figure 5.3 – System Architecture – MVC Methodology 56](#_Toc424751399)

[Figure 5.4 – Entity Relation Model 57](#_Toc424751400)

[Figure 5.5 – Knowledge layer architecture 57](#_Toc424751401)

[Figure 5.6 – Core Conceptual Architecture 58](file:///D:\Documentos\Universidade\Dissertação\Escrita\Dissertação%20v14.docx#_Toc424751402)

[Figure 5.1 - Tools and Technologies adopted 61](file:///D:\Documentos\Universidade\Dissertação\Escrita\Dissertação%20v14.docx#_Toc424751403)

[Figure 5.2 - System requirements 64](#_Toc424751404)

[Figure 5.7 – Rapidminer Main Process screenshot 65](#_Toc424751405)

[Figure 5.8 - Document Analysis Pipeline Block 65](file:///D:\Documentos\Universidade\Dissertação\Escrita\Dissertação%20v14.docx#_Toc424751406)

[Figure 5.9 – Vector Creation Rapidminer Process 66](#_Toc424751407)

[Figure 5.10 – Screenshot of AR system home page 71](#_Toc424751408)

[Figure 5.11 - Screenshot from Front-End AR page 72](#_Toc424751409)

[Figure 5.12 - Association Rule Functionality – Exact and Candidates concept visualizer 73](#_Toc424751410)

[Figure 5.13 - AR Case with a new concept discovered 74](#_Toc424751411)

[Figure 5.14 - Association Rules Database view page 74](#_Toc424751412)

# Tables

[Table 2.1 - Examples of a) Homophone, b) Homograph and c) Synonym words 12](#_Toc424751471)

[Table 2.2 - Examples of relations 20](#_Toc424751472)

[Table 2.3 - Ontology Languages (Lima, 2004) 21](#_Toc424751473)

[Table 3.1 - Horizontal and Vertical layout representations 36](#_Toc424751474)

[Table 3.2 - Advantages/Disadvantages for APRIORI algorithm 37](#_Toc424751475)

[Table 3.3 - Advantages/Disadvantages for ECLAT algorithm 38](#_Toc424751476)

[Table 3.4 - Advantages/Disadvantages for FP-GROWTH algorithm 39](#_Toc424751477)

[Table 3.5 - Improvements attempts for Apriori, Fp-Growth and Eclat algorithms 41](#_Toc424751478)

[Table 3.6 - Frequent searches in a search engine from a motor store 42](#_Toc424751479)

[Table 3.7 –Example of Association Rules data type- a) Boolean, b) Quantitative 43](#_Toc424751480)

[Table 5.1 – Numerical to Binomial regulation 67](#_Toc424751481)

[Table 5.2 – Concept matches map for FI *manag* 69](#_Toc424751482)

[Table 5.3 - Concept candidates map for *temperature* FI 69](#_Toc424751483)

[Table 5.4 – Association Rules Database Structure 70](#_Toc424751484)

[Table 6.1 - All unique one-FI sets before AR discovery 77](#_Toc424751485)

[Table 6.2 – AR with FI Manag as premise 77](#_Toc424751486)

[Table 6.3 – AR with FI *Mang* as conclusion 78](#_Toc424751487)

[Table 6.4 - AR - bidirectional rules 78](#_Toc424751488)

[Table 6.5 - AR - unidirectional rules examples 79](#_Toc424751489)

# Symbols and Notation

|  |  |  |
| --- | --- | --- |
|  | **API** | **A**pplication **P**rogramming **I**nterface |
|  | **AR** | **A**ssociation **R**ules |
|  | **ARM** | **A**ssociation **R**ule **M**ining |
|  | **B&C** | **B**uilding and **C**onstruction |
|  | **bcXML** | **B**uilding&**C**onstruction XML |
|  | **BFS** | **B**readth **F**irst **S**earch |
|  | **CV** | **C**ontrolled **V**ocabulary |
|  | **DB** | **D**ata**b**ase |
|  | **DFS** | **D**epth **F**irst **S**earch |
|  | **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 |
|  | **IFC** | **I**ndustry **F**oundation **C**lasses |
|  | **ISO** | **I**nternational **S**tandard **O**rganization |
|  | **IT** | **I**nformation **T**echnology |
|  | **KDD** | **K**nowledge **D**iscovery in **D**atabases |
|  | **NLP** | **N**atural **L**anguage **P**rocessing |
|  | **OL** | **O**ntology **L**earning |
|  | **OWL** | **W**eb **O**ntology **L**anguage |
|  | **PDF** | **P**ortable **D**ocument **F**ile |
|  | **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 |
|  | **TM** | **T**ext **M**ining |
|  | **IDE** | **I**ntegrated **D**eveloping **E**nvironment |
|  |  |  |
|  | **XML** | e**X**tended **M**arkup **L**anguage |

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 knowledge representation. In particular it will be given special attention to 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, as this is the area that deals with the creation and maintenance of an ontology. 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 2.1)

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 2.1, 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 2.1 example would be a sum of the results provided by the terms *“car”*, *“automobile”* and *“vehicle”*.



Figure 2.1 - 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 or any other entity that uses it (e.g. computer). (Refer to Table a) and b) for examples of Homophone and Homograph words respectively). Moreover, when a word is Homophone and Homograph at the same time, it is called Homonym word, that is to say, **Homonym** words are pronounced and spelled the same way although have different meaning.

CVs address the ambiguity problems of Homograph, Homophone and Homonym 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 -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 2.1 - 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 2.1 c)). As a result, a concept can be represented by more than one word. Referring to Figure 2.1 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 2.2 - 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 2.2). 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 2.3 - 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 2.3). 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 2.4 - 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 2.4) 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 2.5**)**. 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 2.5 - Page from Oxford Mini School Dictionary & Thesaurus (Allen and Mannion, 2007)

**Ontology** (Figure 2.6) 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 2.6 - 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 2.2 - 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 2.4 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 c). Several other examples of relations can be found in ontologies. A non-exhaustive example list can be found in Table 2.2 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 2.3 provides a non-exhaustive list just for demonstration purposes, of several languages used in Ontology Engineering.

Table 2.3 - 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 data are valuable and can provide interesting 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 2.7 - 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 2.8):

* 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 2.8 - 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 2.9 can be seen both taxonomies of e-COGNOS ontology, concept taxonomy and relations taxonomy.



Figure 2.9 - 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.

To summarize, this chapter presented an overview about controlled vocabularies. It was discussed what are they, and what are they for. That is to say, that CVs reduce the ambiguity that the words of a language may have when these are used. Moreover, the advantages and disadvantages of a CV were discussed. It was presented also the types of CVs available, specifically, Vocabulary, Dictionary, Thesaurus, Taxonomies and Ontologies, as well as a nouvelle type of vocabulary, Folksonomy, which is an uncontrolled vocabulary, which relies on social tagging.

Secondly, the controlled vocabulary Ontology was discussed in more detail. It was explain its main purpose. Additionally, how to build one and what are the components of an ontology. Specifically, Concepts, Meaning and Relations, Instances and Axioms and Rules. Furthermore, a non-exhaustive list of formalisms to describe an ontology were presented with a brief explanation for each. The ontology will be the CV in which this dissertation will try to propose a method to maintain it.

Thirdly, it was defined what is Ontology Learning, an approach to deal with the cumbersome task of building and maintaining an Ontology. OL aims also to transform these tasks into a semi- or fully automatic processes, which are yet very far to happen. In this sense, OL is a very important task as an ontology can be very difficult to maintain.

Lastly, it was made an overview through the initiatives of CV over history. One of them is the inspiration for this work, specifically, E-Cognos, an ontology from the B&C sector. An adaptation from E-Cognos ontology will be used in work developed under this dissertation.

3

# Pattern extraction from unstructured sources of information

This chapter will bring to attention techniques and methods about extracting patterns from unstructured sources of information. It will be presented the area that deals with knowledge discovery, specifically Data Mining and why is it important. Moreover, the problem of pattern discovery will be described and presented as well as possible solutions to this problem, specifically, algorithms for the pattern extraction problem. Another point to read in this chapter is the comparison between the most used algorithm and its competitors, describing the advantages and disadvantages of their adoption. Lastly, it will be explained how to extract interesting rules from frequent patterns supported by Association Rules method, as well as how can a rule be evaluated and measured, how to quantify the interest of a rule and which measures are available.

## Definitions

In order to refresh the memory, and to help better understand the contents of this chapter, is important to bring again to attention definitions such as *pattern,* *data*, *information* and *knowledge*. These definitions are presented as follows:

* *Data* is related to individual facts or items which are stored in a computer, in its raw digital form, normally in databases.
* *Information* is defined as sets of organized data in order to provide some sense as well as a context to the data itself. In computer science, information is important or useful facts which results from input processing in a software tool. That is to say, when data is interpreted and associated to a context, it becomes information.
* As stated in Chapter 1, in the area of information retrieval and text mining, a *Pattern* is defined as a predictable occurrence that repeats itself along some (text) data.
* Finally, *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) Knowledge provides meaning to information.

## Pattern Discovery and Knowledge Extraction

The search for patterns aimed to explain and to provide context and meaning for daily common situations is already applied for years by people and is well installed in society motivations. For instance, when a man wants to buy a gift to his girlfriend, tries to remember what she will like. It will search in his mind through all data and information available in it, and try to discover a pattern such as “she likes butterflies” or “she likes red” to make sense of the gift to buy. Similarly, when a farmer wants to know where is the best place to plant seeds and when is the best time to produce a crop, he can search through some data in his mind, or in other place (e.g. a database) to discover. The motivation to discover patterns extends into many more activities, for example, doctors want to discover patterns in the clinical data of their patients, and similarly supermarkets want to discover patterns in the transaction data of their clients and lastly astrophysics want to discover patterns in the data of stellar bodies. One common idea is present in all previous motivations, meaning there is always an entity who wants to discover patterns that could be useful and could originate (produce) knowledge which could help make a good decision.

In the same way, the growth of computer systems through a massification in data storage, with bigger disk drives or online storage, as well as the ease to access a computer by anyone from any area, allow for all data acquired and stored from every field to also grow. Studies estimates a growth of data in world database storages by the double in each 40 month period in the following years (Hilbert and López, 2011). It would be a waste of time, and other resources not to take advantage of the data already stored, to help making better decisions. In this way, new opportunities arise for the discovery of patterns in the data stored, in which could provide interesting facts to data owners. It would be interesting to have methods and techniques that could help easily search a big volume of information and find knowledge that otherwise would be very hard to find through manual processes.

### Data Mining / Knowledge Discovery in Databases.

Several processes can be used for a system be able to discover patterns and further extract knowledge from data and information sources. Data Mining is one of them (Hand et al., 2001) Data Mining (DM), a subfield of Computer Science, is an area that studies methods to discover useful patterns in data sources aimed to help data owners take good decisions. That is to say, the goal of DM is to extract useful patterns from data which could be understandable and later used in the decision making process, in other words, to extract useful information from data sources and transform it into knowledge. Data Mining (DM) allows experts to find knowledge in new data or data they already have. As a result, by adopting DM techniques, decision makers can use newly discovered knowledge that otherwise could be unknown, unavailable or difficult to discover, in order to improve the decision making process. (Witten et al., 2011)

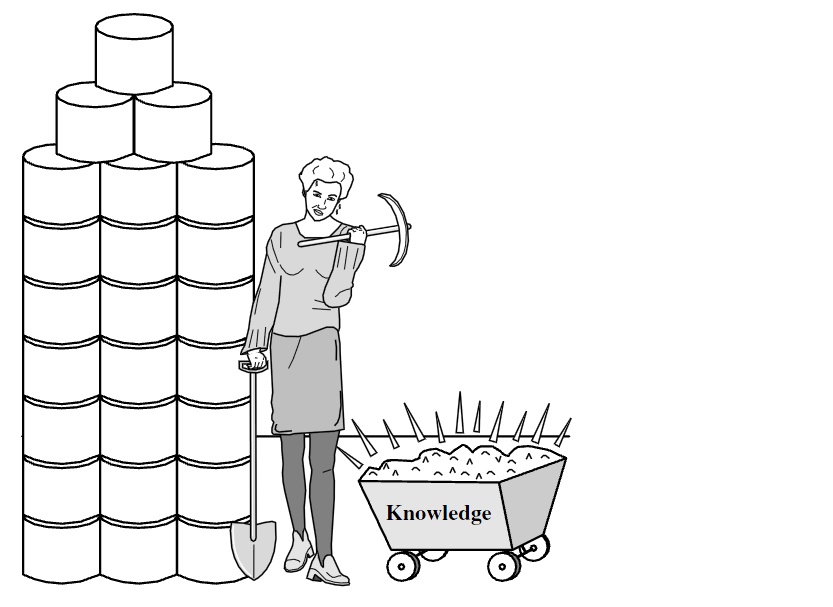


Figure 3.1 - Data mining—searching for knowledge (interesting patterns) in data. (Han et al., 2011)

Data Mining name was inspired in an analogy, easy to understand, from gold mining industry, specifically, in which they search (mine) through raw material, as rocks in order to discover and extract what is really valuable, the gold. In the case of DM, the “gold” is the knowledge that is extracted from the raw data (Figure 3.1). In literature, the process of pattern discovery and extraction from data sources can be found under several names, such as DM. Likewise, it can also be found as Knowledge Discovery in Databases (KDD) (Fayyad et al., 1996), Information Discovery (Steyvers et al., 2004), Information Harvesting (Memon et al., 2007), Data Archaeology (Brachman et al., 1993), Data Pattern Processing (Inmon and Osterfelt, 1991), Predictive Analytics or Data Science (Waller and Fawcett, 2013). Additionally, is also sometimes confused with other areas as Machine Learning or Information Retrieval as these two areas can overlap DM in arranging solutions to data and information problems. Is worth mentioning, that mining data sources in the form of text, is a particular case of Data Mining, and is referred to as Text Mining (TM). All the previous names can mislead the reader. However, in order to make this point clear, in the present document the process of discovery and extraction of patterns will be referred to as Data Mining and when data is specifically related to text, will be referred to as Text Mining.

DM typically uses data sources already collected. Data can be of any type, structured, semi-structured or unstructured, which include databases, data warehouses, web content, text documents, images, sound, etc. Google search engine (Google, 2013), is a software tool from a well-known company which relies on DM techniques to discover patterns from data sources. They take advantage of search engine queries in order to discover knowledge for further use. One example is the use of such query data to discover what people from a specific country commonly looks for, in order to retrieve faster the desired geographically based query answers. Furthermore, other example of DM use is the Page Rank algorithm that Google uses to search pages and rank them according to the ones that are being pointed by more pages (Loukides, 2010).

In order to go from data to knowledge, Data Mining process relies in several steps (Fayyad et al., 1996). Specifically, these include the following three: *Data Preprocessing, Pattern Discovery*and*Pattern Evaluation*(please refer to Figure 3.2 below for an illustration of this process**).** Initially data is not yet ready to use, it needs to be pre-processed and prepared in order to apply techniques for pattern discovery. *Data Preprocessing* is related to the selection and preparation of appropriate data to use, or is related to execution of operations for data size reduction, all without sacrificing its integrity. For instance, specifically in Text Mining, these can include the removal from data sources, of specific words[[5]](#footnote-5) without any significant semantic added value to the process. Another illustration of this is the reduction of the size of a data corpus, by reducing each word to its *lemma* word[[6]](#footnote-6) and removing the duplicates.

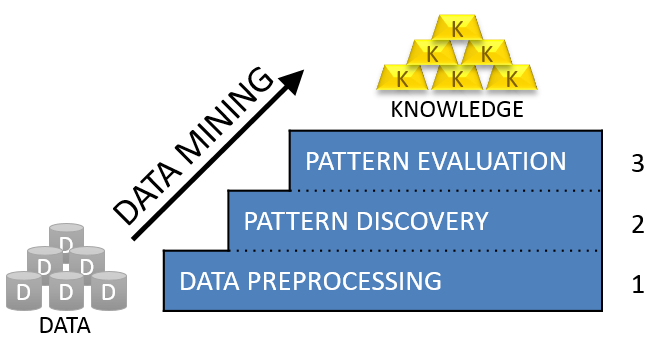


Figure 3.2 - Data Mining Process – Steps from Data to Knowledge

*Pattern Discovery*step relates to the techniques applied to pre-processed data in order to discover patterns. These include techniques as *Apriori*, *FP-Growth* or *ECLAT* algorithms, which will be further discussed in the following section 3.3.

Finally, *Pattern Evaluation***,** the third step in the DM process, is related to the identification of the useful and interesting patterns which can originate knowledge. This is achieved through *Association Rules* and interest evaluation, discussed in the following section 3.4.

## Pattern discovery - Techniques

The typical techniques (algorithms) used for frequent pattern discovery are based in the following three: Apriori, FP-Growth and ECLAT algorithms. These techniques were introduced by its authors in the academic community as being capable of discovering frequent patterns (frequent itemsets) from a data corpus and to serve as a base for Association Rule Mining (ARM). And how can a pattern be considered frequent? A pattern is considered frequent when the frequency in which it occurs in the data sources is above a minimum threshold value called *support* (refer to sub-chapter 3.4.4) which normally is human defined. For instance, if one consider a set of data related to B&C materials such as “*brick, back door, window, front door, door knob, floor, wood*” and consider minimum *support* equal to 2, the algorithm will consider “*door*” as a frequent item as it is discovered in three items: “*back door*, *front door* and *door knob*”.

### Apriori

APRIORI algorithm, was one of the pioneers to address the pattern discovery problem. This algorithm was presented by Agrawal and Srikant (Agrawal and Srikant, 1994) who also brought to attention the association rules problem. It is considered as a starting point for many studies in the academic community about frequent pattern discovery.

Agrawal and Srikant define Apriori algorithm as a procedure for candidate itemset generation and test. To put it in another way, this algorithm generates potentially frequent patterns as candidates and evaluates them one by one in order to discover the frequent ones. It uses a search methodology of breadth first search (BFS)[[7]](#footnote-7) technique as it works in horizontal laid out (each row is a transaction with all items in which they occur) data source representation (refer to Table 3.1a)). One of the characteristics of this algorithm is the property described as: every subset of a frequent itemset is also frequent.

Table 3.1 - Horizontal and Vertical layout representations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TID | Itemset |  | Items | TID set |
| 1 | Bread, Butter, Jam |  | Bread | 1, 4, 5 |
| 2 | Butter, Coke |  | Butter | 1, 2, 3, 4 |
| 3 | Butter, Milk |  | Jam | 1 |
| 4 | Bread, Butter, Coke |  | Milk | 3, 5 |
| 5 | Bread, Milk |  | Coke | 2, 4 |
| a) Horizontal Laid Out Representation | |  | b) Vertical Laid Out Representations | |

The authors explain how this algorithm works, as follows: This algorithm includes two major steps. The first is a read through the data corpus to calculate the *support* value for all items and determine which ones are large items, *i.e.* have its *support* above a *minimum support* defined value, resulting in a new itemset with the large items as candidate sets to the next iteration. This will be done recursively, using the itemset from the previous iteration, until the largest itemset is discovered, and done as many times as the maximum itemset available in the same data sources. The second step of Apriori algorithm is to make another read through the candidate data sources to discover all frequent items, by counting its frequency. As a result, this algorithm obtains a group of frequent patterns prepared for pattern evaluation, that is to say, association rule mining.

Table 3.3 describes a set of advantages and disadvantages identified in Apriori algorithm.

Table 3.2 - Advantages/Disadvantages for APRIORI algorithm

|  |  |
| --- | --- |
| APRIORI | |
| ADVANTAGES | **DISADVANTAGES** |
| Easy to implement | Number of data source scans needed to generate candidate sets – As big as the maximum number of elements in an itemset. |
| Easily parallelized | Assumes transaction database is memory resident. |
| Uses large itemset property |  |

Apriori is an algorithm crossing several scientific areas which deals with data. One example of this is a recent study presented by Kumar and Chadha (Kumar and Chadha, 2012), in which they take advantage of the Apriori algorithm to discover association rules in assessment data from an university students. Through this study they can discover several and revealing facts such as student’s interest, curriculum design; teaching and assessment of methodologies that can affect students who have failed to attain a satisfactory level of performance in the Post-Graduation level.

### ECLAT

ECLAT, which stands for Equivalence CLass Transformation, is an algorithm to find frequent patterns in data sources. This algorithm introduced by Zaki (Zaki, 2000) is aimed to improve the performance problems of Apriori-based algorithms, specifically minimization of I/O costs by reducing the number of database reads or even the reduction of the computational costs with more efficient search procedures. ECLAT needs just a reduced number of reads of the data sources and do not require any hash trees whatsoever as it generates frequent itemsets by only simple intersection operations. It can even handle *support* values lower than Apriori *support* values in large datasets.

ECLAT Algorithm uses a search methodology of depth first search (DFS)[[8]](#footnote-8) technique as it works on vertical laid out (each item stores a list of the transactions in which it occurs) data source representation (please refer to Table 3.1b)).

This algorithm works as follows: The first step is to construct a vertical data representation, searching through the data sources and obtaining a list with all items, one per row, each with all transactions in which they occur, as a conditional data source table. The *support* of an item is the count of transactions in which it occurs. In the next iteration, the data source to use is the conditional table from the previous iteration, in which the (n+1)-itemsets are intersected to obtain a list of the transactions in which they co-occur, as the conditional data source table for the (n+1)-itemset. All iterations will continue recursively until all frequent items are discovered or no more candidates are available to test. In each iteration the support is counted. .

Table 3.3 describes a set of advantages and disadvantages identified in ECLAT algorithm.

Table 3.3 - Advantages/Disadvantages for ECLAT algorithm

|  |  |
| --- | --- |
| ECLAT | |
| ADVANTAGES | **DISADVANTAGES** |
| Very fast support counting related to Apriori | Intermediate tid-lists may become too large for memory |
| No complex data structures-no hash trees |  |
| Effective when the discovered frequent itemsets are long. |  |
| *Support* value easily calculated |  |

### FP-Growth

FP-Growth, which stands for Frequent Pattern Growth is an algorithm introduced by Han et al. (Han et al., 2000), that is applied in data mining, aimed to discover frequent patterns (frequent itemsets) in data sources as a base for further evaluation. In other words, this algorithm searches through a set of data to discover patterns, which are considered frequent. Many studies appreciate the popularity and effectiveness of FP-Growth algorithm, moreover some authors claim “this algorithm is currently one of the fastest ones to mine association rules.” (Borgelt, 2005; Han et al., 2004; Paiva et al., 2013)

This algorithm is executed in three steps, firstly the build of a compact data structure in the form of a prefix-tree (meaning the nodes will only be based in frequent patterns of size 1), called FP-Tree, from the data corpus. The FP-tree is aimed to store quantitative information (as each node represent an item, with its support value) about the frequent patterns discovered and to prepare for the next step of this algorithm. It is worth adding, that the FP-tree is arranged in such way, that the chance for node sharing is higher between frequently occurring patterns, than between less frequently ones; secondly, the development of a mining method of pattern fragment growth, based in the previous FP-tree, hence the name FP-Growth. This method is referred as a *divide-and-conquer technique*, meaning it separates frequent patterns from not frequent patterns in data sources; lastly, is the extraction of frequent itemsets directly from the FP-tree. The process will start in the less frequent nodes until it reaches the top node (bottom-up approach).

Table 3.4 describes a set of advantages and disadvantages identified in FP-Growth algorithm.

Table 3.4 - Advantages/Disadvantages for FP-GROWTH algorithm

|  |  |
| --- | --- |
| FP-GROWTH | |
| ADVANTAGES | **DISADVANTAGES** |
| Uses compact data structure (FP-Tree) | FP-Tree may be expensive to build |
| One database read | FP-Tree may not fit in memory |
| No candidate set generation | Rebuilds conditional FP-Trees |
| Simple – Counting items and building a tree structure only operations |  |
| Very fast to discover frequent patterns |  |

Due to its popularity and performance, FP-Growth is an algorithm much appreciated and used. An example of this is a study by Korczak and Skrzypczak (Korczak and Skrzypczak, 2012) which illustrates the use of FP-Growth algorithm for frequent pattern discovery in a transaction database from an online store, aimed to discover association rules between the customers transactions. Based in this algorithm, the authors discovered frequent patterns in transactions whose shopping value was above 50 euros and contained more than 2 items, which made them discover interesting behaviour by the customers, as “*more than 2% of the customers who buys tomatoes and a chicken always buy new potatoes*”, or “*more than 2% of the customers who buy white grape and watermelon always buy tomatoes*”.

### Algorithm comparison

As seen in the previous sub-sections all the referred algorithms have advantages and disadvantages. With this in mind, several comparisons in the academic community were made between them to evaluate their performance and discover if any outperforms the others. Garg and Kumar (Garg and Kumar, 2013) presented an interesting study with a comparison of the algorithms. Specifically, the proposed study compares Apriori, Eclat and FP-Growth under several scenarios, such as testing their behaviour when there is an increase of the transactions quantity, meaning an increase in the data source size (Figure 3.3); moreover, testing their behaviour when there is an increase of attributes and testing their behaviour when last two scenarios are used at the same time. The authors measured the time performance under all the latter scenarios. As a conclusion, is argued that “*FP-Growth is the best among the three, and thus more scalable*”. Moreover, the authors add that “*Apriori is the worst in the performance tests*” especially when the data corpus increase in size.

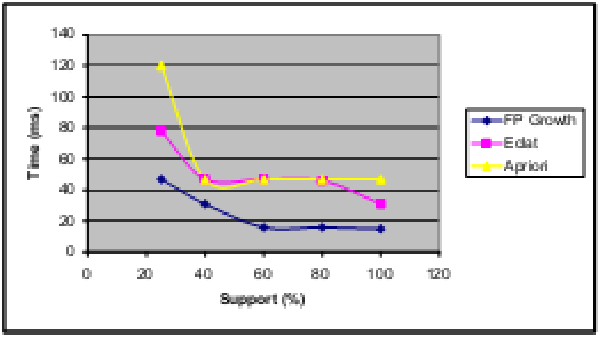


Figure 3.3 - Performance comparison of Apriori, Eclat and FP-Growth (Garg and Kumar, 2013).

Borgelt (Borgelt, 2005) presents another interesting study which also compares Apriori, Eclat and FP-Growth algorithms. The author uses the FP-Tree building preprocessing, which represents the most expensive (in time performance) operation made by FP-Growth, as the compare element. The conclusion was similar to previous study from Greg and Kumar, specifically the author argues, based on the experimental results, that FP-Growth “*clearly outperforms Apriori and Eclat, even in highly optimized versions*”.

### Improvements Attempts

For illustration purpose, Table 3.5 describes a few algorithms found during the research for this dissertation, which are based in the three algorithms discussed, and present themselves as potential improvements. The intention of this table is not to provide an exhaustive list, on the contrary, just a small overview.

Table 3.5 - Improvements attempts for Apriori, Fp-Growth and Eclat algorithms

|  |  |  |
| --- | --- | --- |
| Name | Description | Source |
| APRIORI | | |
| MSApriori | Uses the same procedure as Apriori, but uses more than one minimum support value. | (Liu et al., 1999) |
| A-Close | Uses a closed mechanism to find frequent patterns, specifically it is based in pruning a closed itemset lattice. | (Pasquier et al., 1999) |
| Apriori-Inverse | Discovers sporadic rules (low support but high confidence) by ignoring all candidate itemsets above a maximum support threshold. | (Koh and Rountree, 2005) |
| UApriori | Searches frequent sequences in the presence of existentially uncertainty (considers attribute level (existential) probability) in transactions | (Metanat Hooshsadat et al., 2012) |
| FP-GROWTH | | |
| Top Down FP-Growth | Searches the FP-Tree through the Top-Down approach as opposed to normal FP-Growth (Bottom-Up). Does not generate conditional pattern bases nor sub-FP-trees. | (Wang et al., 2002) |
| FP-Bonsai | Uses a data reduction technique to reduce FP-Tree structure size, based in pruning operations, aimed for performance improvement, specifically memory size reduction and disk writing frequency reduction. Resulting an FP-Tree referred as FP-Bonsai. | (Bonchi and Goethals, 2004) |
| nonordfp | Improves the performance (memory efficiency) problem of conditional FP-Tree rebuilding, by using just one compact structure without the need to rebuild conditional trees. | (Rácz, 2004) |
| ECLAT | | |
| MaxEclat | Improves Eclat, ordering itemsets in transaction lists by their size (long first) based in their support values.  It uses an hybrid search, as opposed to bottom-up search from Eclat. Discovers the long maximal frequent itemsets, and some nonmaximal ones. | (Zaki, 2000) |
| dEclat | Uses an approach called Diffsets Eclat (dEclat), which uses the count difference between transaction itemsets (tids), as opposed to using the complete tids. Improves the computation size problem, reducing the memory necessary for frequent pattern mining. | (Zaki and Gouda, 2003) |

## Association Rules – Pattern Evaluation

Introduced by Agrawal et al.(Agrawal et al., 1993), Association Rules (AR) is an algorithm used after frequent pattern discovery, to evaluate frequent patterns and identify tendencies and relations (or associations) between frequent items in data sources resulting in knowledge, which could be used to try to make predictions over potential scenarios. In other words, the main goal is to use the knowledge discovered from data sources already existent to help improve the decision making process.

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

An AR can be defined as an implication as represented by equation (1). Each AR is composed by two sets of items, *A* and *B*, called *itemsets*. Each itemset can hold one or more items, each belonging to the same data source. Moreover, the intersection of A with B is an empty set, that is to say that A and B are two different itemsets. Additionally, A and B are referred to as premise and conclusion[[9]](#footnote-9), respectively. Premise represents the initial occurrence to evaluate, and conclusion represents the occurrence of what was determined by the rule. Therefore, an Association Rule can be read as follows: If a premise A occurs in a data corpus then conclusion B will likely also occur.

Table 3.6 - Frequent searches in a search engine from a motor store

|  |  |
| --- | --- |
| ID | Search terms |
| 1 | {“car”, “motorcycle”, “race car”, “Yamaha”, “luxury car”, “small car”, “boat”} |
| 2 | {“race car”, “Yamaha”, “luxury car”, “small car”} |
| 3 | {“race car”, “Yamaha”, “small car”, “motorcycle”} |
| 4 | {“race car”, “Yamaha”, “small car”} |
| 5 | {“luxury car”, “boat”, “Yamaha”, “race car”} |
| 6 | {“motorcycle”, “boat”, “small car”} |
| 7 | {“Yamaha”, “luxury car”, “boat”} |

As a practical example of an AR, one can imagine a database with the most frequent queries in a search engine from a store that sells motors as represented in Table 3.6, and considering a rule with two itemsets, A={“race car”, “small car”} and B={“Yamaha”}, one can say when the items “*race car*” and “*small car*” occur in a query, the item “*Yamaha*” also occur. The former example provides knowledge to the data analyser, in the sense it can potentially predict that whenever someone looks for a race car and a small car, it will likely search for a “Yamaha” motor.

Another illustration that also helps to clarify the objective behind AR mining, which is usually referred by the academic community, is an example based on market basket analysis and its transactions in a department store. In such example, the objective is to predict customer behaviour, based on the collected data from several transactions. Specifically, if a customer buys product A, the AR algorithm, based on the stored transaction data of other customers, will be executed and its results will try to predict the behaviour that potential customers will have, or in other words, which product B will the customer likely buy.

Table 3.7 –Example of Association Rules data type- a) Boolean, b) Quantitative

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Month | Humidity | Temperature | Weather |  | ID | Age | Married | Cars |
| 1 | Jan | HIGH | LOW | RAIN |  | **1** | 20 | YES | 1 |
| 2 | Feb | HIGH | LOW | RAIN |  | **2** | 35 | YES | 1 |
| 3 | Mar | HIGH | LOW | RAIN |  | **3** | 23 | NO | 2 |
| 4 | Apr | HIGH | LOW | RAIN |  | **4** | 22 | NO | 1 |
| 5 | May | HIGH | LOW | CLOUD |  | **5** | 50 | NO | 0 |
| 6 | Jun | HIGH | LOW | CLOUD |  | **6** | 28 | YES | 2 |
| 7 | Jul | MEDIUM | HIGH | SUN |  | **7** | 36 | YES | 2 |
| 8 | Aug | LOW | HIGH | SUN |  | **8** | 30 | NO | 0 |
| 9 | Sep | HIGH | MEDIUM | SUN |  | **9** | 31 | YES | 0 |
| 10 | Oct | MEDIUM | MEDIUM | SUN |  | **10** | 45 | YES | 2 |
| 11 | Nov | HIGH | LOW | CLOUD |  | **11** | 41 | NO | 1 |
| 12 | Dec | HIGH | LOW | CLOUD |  | **12** | 37 | YES | 2 |
| 1. Weather / Humidity / Temperature average conditions at day 10 of each month of 2013 | | | | |  | 1. Number of cars owned by people older than 18. | | | |
|  |

Association Rules can be expressed in two different data types, *Boolean* or *Quantitative* (Hoque et al., 2011). Boolean Association Rules are expressed by true/false based values. Consequently, premise and conclusion values are also either true or false, for instance, if A = true then B = true, or if Humidity=HIGH and Temperature=LOW then Weather=RAIN or CLOUD (Table 3.7b)). Quantitative Association Rules are expressed by items that holds a numerical representation with more than two values (e.g. age intervals as [18-35]). An illustration of this is presented by Table 3.7a), where if Age > 34 and Married=Yes then Cars > 1.

### Association Rules - State of the art

Association Rules is a technique which discovers knowledge in order to improve decision making processes. Since its introduction in 1993 by Agrawal et al. (Agrawal et al., 1993) several initiatives were made to take full advantage of their potential with interesting results across several areas around the world. Such as Medicine, Chemical Science, Business, Culture Heritage, Languages, Education, Sociology or Building and Construction. In the following lines, several initiatives will be presented. These initiatives take advantage of the high potential of AR application to discover interesting and surprising knowledge.

For instance, in Medicine, Mahgoub (Mahgoub, 2006) applied AR to discover knowledge from MEDLINE[[10]](#footnote-10),.a scientific online database. His goal was to discover knowledge from medical publications related to the H5N1 avian influenza virus outbreak. He achieve it by searching through more than 8000 keywords from the abstracts of 100 publications and discovering a set of associations that could later be evaluated. For instance, he was able to discover that when the keywords “*highly, pathogenic and avianinfluenza*” co-occur in the abstracts, the keyword “H5N1” will also occur.

Coming from the Chemical Science area, an interesting application of AR was the study from Azevedo et al.(Azevedo et al., 2005). The authors analysed a massive quantity of molecular simulations, aimed for the discovery of associations between the constituents of the molecular systems. As a conclusion, through the use of this technique, they were able to successfully discover 28 hidden associations between the 127 constituents from the simulations, thus discovering new biochemically relevant knowledge.

In Medicine, Tan et al. (Tan et al., 2009) presented a study which took advantage of the AR potential by trying to discover knowledge from the contents of medical brain images. This process was aimed to help doctors better diagnose brain threats in medical images (specifically, in previous defined regions). After discussing their results with medical specialists, comparing real diagnosis, the authors concluded that their experience was more than 90% effective, which is a relevant value for the use of AR.

From Culture Heritage area, Tomi Kauppinen (Tomi Kauppinen, 2009) examined a semantic portal database (*viz.* CULTURESAMPO), aimed to propose the enrichment of an ontology (*viz.* Finnish General Upper Ontology - YSO) through the use of AR technique. Specifically through the discovery of interesting associations between the concepts used to describe art objects (*viz.* paintings, photographs, etc.) from the database. Using AR technique, the author was able to analyse more than 4700 concepts and successfully discovered relevant associations, which could be proposed for the ontology. Such as an association between “*musical instruments*” (discovered 385 times) and “*folk music*” (discovered 125 times) as they were discovered together in the same descriptions 125 times. Based in this technique, the author were able to propose to the ontology engineer who was responsible for the ontology itself, 23 out of the top 50 associations discovered to be added to the ontology.

More areas took advantage of AR technique, such as Education. In this area, Kumar and Chadha (Kumar and Chadha, 2012) presented recently a case study aimed for the improvement of the education quality of an university. Through the use of AR, the idea was to discover the factors leading to a high student success rate by analysing academic results of the students. For instance, they discovered knowledge such as “*students who have scored badly in their Graduation have done relatively well in their Post Graduation in the subjects which are common in both Graduate and Post Graduate courses.*” Based in the new knowledge discovered, the university could use it to change methodologies or redesign the curriculums aimed for the benefit of both students and university.

Another interesting example of the AR application was presented by Spruit (Spruit, 2007) in the Languages area. The idea of this study was to examine associations between syntactic variables in the 267 Dutch Dialects (from Netherlands, Northern part of Belgium and a small northwest part of France), aimed to discover the geographical distribution of the variations of the words from the dialects. This study evaluated more than 234000 rules, discovering more than 10000 interesting associations with 90% accuracy. By using AR technique the author discovered interesting knowledge about, for instance, the geographical region where a specific word is more used.

The use of AR was also applied to Sociology area. Specifically, to the demography of Washington DC through a study presented by Brin et al. (Brin et al., 1997). These authors took advantage of AR techniques to discover interesting knowledge from a demography study in the capital of the USA. Based in the census data (more than 30000 entries) from 1990 they were able to discover more than 23700 rules, from which they discover interesting knowledge, such as “*unmarried people have moved in the past five years*” or as “*African-American women are not in the military*”.

One study from the Business area, was presented by Hoque et al. (Hoque et al., 2011) aimed to discover knowledge from a company employee database that could provide clues about the behaviour of their employees, to assist in the decision making process for their benefits. From a database of 10 employees, by using AR technique they were able to discover 35 association rules. Such as “*all the low experienced, with medium salary and master education level are young*”, or as “*all the PHD education level employees have high salaries*”. Through the newly discovered knowledge the organization was able to reflect about some opportunities, as salary increment or a promotion.

Other study from Business area was presented by Korczak and Skrzypczak (Korczak and Skrzypczak, 2012). This study proposes the discovery of interesting knowledge from the analysis of customer transactions in an online store (*viz.* Delicatessen Alma24), aimed to assist decision makers better understand customer behaviour, in order to improve sales performance. Through the application of AR in more than 1000 transactions, the authors were able to discover 32 interesting rules. Based in these rules they were able to discover interesting knowledge, for instance, that “*at least 2% of the customers who buys tomatoes and a quarter chicken, always buy the new potatoes*” or that “*at least 2% of the customers of red pepper and Hojnowka butter always buy tomatoes per kg*”. Based in this new knowledge the decision makers were able to better understand the choices of the customers and consequently improve the success of the business.

All in all, Association Rules is a very useful technique to make sense out of data. It is largely used amongst several areas, by data analysts from companies, institutions, stores, museums, health care, etc.

### Interest Evaluation in Association Rules

*Interest* is the feeling of wanting to know or learn about something or someone (Oxford University Press, 2006). *Evaluation* is the making of a judgement about the amount, number or value of something or someone (Oxford University Press, 2006). The discovery of what will be interesting in a data corpus is the main goal of AR. In order to find interest in useful patterns they need to be evaluated somehow, but how to evaluate an association rule? Moreover, how to know which association rules are interesting? Is an association rule interest in two contexts? The interest of an association rule can be discovered through evaluation of their interestingness, by making a judgment of an association rule based in the context against which is evaluated. In this matter, some considerations and thoughts can be made when evaluating knowledge and specifically an association rule. Firstly, what should be evaluated and what should be considered interesting; lastly, how to evaluate interesting association rules.

In order to discover interest, it is important to know that the first factor to consider is the context in which an association rule needs evaluation. For instance, if the context is building and construction, association rules discovered including houses or buildings could be more interesting than those who include computers or photography. In contrast, if the idea is to discover association rules about houses or photographs of buildings, then the interest on photography rises, transforming one uninteresting context into an interesting one. Therefore, the context is one important factor to consider and should be carefully chosen when evaluating an association rule as it can help to provide the best and results related to the interest considered.

Yao et al. (Yao et al., 2006) argue that there is another factor that could influence the evaluation process of an association rule, and thus its interest. This factor is the user who evaluates an association rule, who can be a person (or an organization or a system). Each user is unique and this can lead to different points of view between different users when discussing about the same idea. Moreover, the background or even the geographical location of a user are factors that can lead to different interpretation or perception about a subject, thus influencing the judgement of the user evaluating an association rule. Therefore, a user can play a crucial role in the judgement of the interest, meaning that the practical results of an evaluation process of association rules could also depend on the subjectivity of a specific user who participates in the evaluation process.

Several studies in the last few years discussed about methods to overcome the drawback of evaluating the *interestingness* of knowledge discovered from association rules. There are two ways to measure the interestingness of an association rule, specifically subjectivity and objectivity measurements (Hilderman and Hamilton, 1999; Silberschatz and Tuzhilin, 1995). Additionally, Mackie (Mackie, 1977) argues that the evaluation of subjectivity is very common when the goal is to evaluate actions or events. Moreover, the author adds that the objectivity is used when there is the need to quantify a value, thus being reflected by the measures themselves. The subjectivity is influenced by the entity that is evaluating a subject. As explained before, it can also be influenced by factors such as the location or the background. Figure 3.4 illustrates, the classification of the types of measures aimed for interestingness evaluation of an association rule.



Figure 3.4 – Classification of interestingness measures (adapted from Silberschatz and Tuzhilin, (1995))

### Subjective measures

To measure the subjectivity, Silberschatz and Tuzhilin (Silberschatz and Tuzhilin, 1995) identify two concepts, specifically *unexpectedness* and *actionability*. The first concept represents the value of something unexpected or a surprise in an association rule when knowledge is discovered. That is to say, if one could discover an association rule that would not expect, that rule would be an interesting one. The authors argue that, the knowledge discovered which was expected, is knowledge that is already known, and thus, not interesting for the user in this sense. The second concept of subjectivity is actionability, it represents the usability that an association rule could have. In other words, it is the capacity of a rule to be used in an interesting way by its user. One practical example of the use of unexpectedness and actionability concepts applied to association rule mining is the study presented by Gonçalves (Gonçalves, 2005). This author explains how these concepts can be used in association rules, by applying them in the transactions of a department store. The author discovered that when the transactions were made by young couples on a Thursday, there was an association between dippers and beer. Initially, the 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 the association rules were discovered, this unexpected knowledge was also discovered. This is a good illustration of what an unexpected and actionable rule is. As a result, the department store, for now on, on Thursdays, can apply this new and unexpected discovered knowledge to move the dippers and beer closer to each other, so the sales of both could increase.

Is worth adding that, although actionability and unexpectedness concepts are independent of each other, they can be combined to strengthen even more an association rule. Although this could not always be true, frequently, the unexpected rules are also actionable rules, the ones that a user can do something useful with them. Similarly, the actionable rules, are also rules unexpected to appear. If the objective of association rules mining was to discover knowledge already known, what would be the point of doing it? Moreover, what can be done to the knowledge that is already known? Could it be used to improve something? Or can it be discarded? More research is needed in order to discover answers to this questions.

### Objective measures

The other type of interestingness measures, are the ones to measure objectivity. These measures quantify the strength of association rules, reflected in a numeric value as a result. Examples of this can be the measures *support*, *confidence*, *conviction*, *lift*, *gain*, *laplace* and *ps*, which will be discussed in this sub-section.

**Support and confidence**

Since the presentation of association rules by Agrawal et al. (Agrawal et al., 1993), many authors illustrate the use of two specific measures that could assist the evaluation of interest in association rules: *support* and *confidence*. In fact, several authors, proposed studies which take advantage of these two measures to quantify the interest of an association rule (Azevedo et al., 2005; Bayardo and Agrawal, 1999; Bhujade and Janwe, 2011; Brin et al., 1997; Hoque et al., 2011; Kumar and Chadha, 2012; Spruit, 2007). All such studies is a proof that these measures are relevant and should be considered when evaluating the objective interestingness in association rules.

The *support* (also *frequency* or *coverage*) of an association rule is a statistic measure, defined as the number of occurrences in a data corpus where both premise A and conclusion B occurs simultaneously. That is to say, the support of an association rule, is defined as the percentage of the transactions containing both itemsets A and B. The values from this measure are enclosed between, and are normally represented in percentage. The support result is proportional to the frequency. That is to say, higher the support value, the more frequent are the itemsets in the data corpus. Support is represented by the following equation (2).

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

*Confidence* represents an estimation of the probability of observing itemset B given itemset A. When this measure is applied in an association rule, one can immediately quantify the strength of the association between both itemsets. The expression to calculate confidence is given in Equation (3) and its values are enclosed in . One can also identify that the interest rises also with confidence results. That is to say that there is more interest in an association rule with higher confidence values. Association rules with value equal to 1 mean that given A, B always occur.

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

Although both these measures individually can provide some interesting rules, this could not be enough. To get the real interesting rules, one has to consider two additional parameters, *minsup* and *minconf* (Bayardo and Agrawal, 1999). Bayardo and Agrawal argues that when there is lower limit defined, the rules with values above the limit are the real interesting ones. For instance, a rule can have a support value of 20%, however, if the defined minsup is 50% this rule is considered uninteresting. In contrast, if a rule have a support of 80% is considered a very interesting rule.

**Conviction and Lift**

Support and confidence can provide a good and trustful interestingness objective measure results, however sometimes they are not enough. As a result some other measures were studied and proposed by the scientific community, *Conviction* and *Lift* are two other statistic measures proposed to complement the objectivity measurement of an AR. These measures are commonly used to increase the strength of the values obtained from confidence and support measures, as these two depend on the confidence and support 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* has some interesting properties, such as:

**Property 1:** If its value is equal to 1, this means that the concepts are considered totally independent from each other. That is to say, they do not have any kind of association.

**Property 2:** On rules with 100% as confidence value, the value of conviction is equal to ∞.

|  |  |  |
| --- | --- | --- |
|  |  | (4) |
|  |  | (5) |

The most interesting rules based in this measure, are reflected by the higher result values, and higher this value higher is the interest of that rule. The result values of *Conviction* are found inside the interval . *Conviction* is represented by equation (4) or alternatively, by equation (5).

*Lift*[[11]](#footnote-11) (also found in literature as *Interest* (Brin et al., 1997) or as *Strength* (Dhar and Tuzhulin, 1993)) is a measure to quantify the co-occurrence of a rule. In contrast to conviction, Lift is not an implication measure as it is symmetric in relation to premise and conclusion, hence *Lift(A⇒B) = Lift(B⇒A)*. That is to say, it measures how far from independence are itemsets A and B. Lift is defined as a measure to boost (“lift”) the confidence of a rule, this suggests a trust increase in confidence results of a rule. This measure have better results in association rules with lower support values. Similarly to conviction, if its value is equal 1 it means the itemsets are totally independent, without any kind of interesting association. Moreover, as far from 1 and as higher the value, higher will be the interestingness of the association rule, which can reflect a higher association between the itemsets. The result values of this measure are included in . Lift is represented by the following equation (6) or equation (7) where this value can be given taking advantage of confidence measure.

|  |  |  |
| --- | --- | --- |
|  |  | (6) |
|  |  | (7) |

As can be easily observed in both equations (5) and (7) from conviction and lift, there is a relation to confidence measure. Therefore, these measures can be understood as measures to increase or strengthen the trust of confidence results when confidence itself would not be enough to make conclusions and evaluate interesting knowledge in the association rules.

**Gain, Laplace and PS**

Finally, the remaining measures illustrated in Figure 3.4 are *Laplace*, *PS* and *Gain*. These three measures will be discussed in the following lines.

*Laplace* is defined as a trust estimator which is function of support. This is reflected by the following: as low as support value is, lower is the interest in the rule considered. Laplace is normally used to rank rules by class. The range of their values are in . Laplace is represented by equation (8). The constant *k* represents the number of classes chosen when someone is defining the respective classification model. Its value is always higher than 1.

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

*Gain* is an optimization measure proposed by Fukuda et al. (Fukuda et al., 1996) as a measure to solve the optimized gain rules problem. It is a function of support and given by equation (9). 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 only be done by decreasing the support of the premise. When this happens, the support value of the association rule becomes higher.

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

The last of these three measures presented is *PS*. This measure receives its name from their creators, Piatetsky-Shapiro (Piatetsky-Shapiro, 1991)[[12]](#footnote-12). It was originally used to classify general rules, and later adopted as an association rule measure. As lift, this measure is a boost to the support measure. Its value is included inside the range . When this value is equal to 0 it means that A and B are independent. When this the value is below 0 represents a negative dependency and if the value is higher than 0 it is called positive dependency. Higher values mean more interesting association rules. The definition for PS is represented by equation (10).

|  |  |  |
| --- | --- | --- |
|  |  | (10) |

**Other measures**

All the measures illustrated in this section should arguably provide enough means to evaluate any association rule discovered, although, in the academic community several other measures were proposed over the years. For instance, Tan et al. (Tan et al., 2002) propose a study where they evaluate a list of 21 measures for association rules, which includes the ones previously discussed. Moreover, this study also includes the measures such as *Kappa*, *Gini*, *Added Value*, just to name a few. Further research should be made in the direction of more measures, to improve the association rules process reliability.

To summarise, this chapter presented an overview of data mining concepts, and how its process works. It was explained what is and how to analyse data in order to make sense out of it, in other words, how to discover knowledge from raw data, based in a 3 step model from Data Preprocessing, Pattern Discovery and Pattern Evaluation. As the present study is aimed more to the last two steps, these were explained in more detail.

Pattern Discovery step is responsible for the techniques to extract frequent patterns from data. Three main techniques were presented capable of discovering frequent patterns, specifically, Apriori, ECLAT and FP-Growth. These techniques were starting points to several other studies of potential improvements, which a non-exhaustive was presented. A study which demonstrates which one is the best technique from the former three was also discussed, concluding that FP-Growth is the one who performs better.

Patterns Evaluation, is the step in Data Mining process responsible for the discovery of interesting knowledge. This is achieved through a technique referred to as Association Rules. This technique discovers associations between the patterns and evaluates them. This evaluation of interest is made through specific measures, subjective and objectives. Moreover, the use of this technique crosses several areas, such as Medicine, Education, Culture, Languages, Demography and others. This technique is relevant to use where there is the need to make sense out of data.

In fact, as the present dissertation aims to propose an approach for ontology learning, data mining techniques are going to be used in order to discover knowledge from unstructured sources of information from a document corpus. The former will be achieved based in the concepts discussed in this chapter, specifically, by using FP-Growth algorithm to discover patterns. Moreover, Association Rules technique will be also applied to evaluate the interest of knowledge discovered. Taking advantage of the measures available, as support and confidence. The interesting knowledge will be proposed in order to update a domain ontology.

4

# Concept Model

Chapter 4 – Concept Model

(What content will the conceptual model represent?

How will the conceptual model be presented?

Who will be using or participating in the conceptual model?

How will the conceptual model describe the system?

What is the conceptual models focus of observation?

Will the conceptual model be efficient or effective in describing the system?)

- Explain conceptual model/solution

- Describe an application example

From unstructured information to knowledge representation and ontology structure

- Dimensions included in the model??? Ontologias e Data Mining???

- Enrichment process / Ontology learning process

FP-Growth how to build and FP-Tree

Association rule evaluation

- DER / MVC / UML Diagrams

This chapter will present the conceptual model inspired by the expected outcomes defined in this document (refer to Chapter 1.2). Specifically it will be proposed a model for pattern discovery and knowledge extraction.

De forma a extrair conhecimento de dados não estruturados e usá-lo para melhorar uma ontologia, é necessária a criação de uma aplicação com vista à apresentação dos resultados do seguinte modelo proposto. No entanto, apesar das tentativas da criação de sistemas totalmente automatizados de manutenção de ontologias, ainda estamos longe dessa visão. Desta forma, os sistemas são parcialmente automatizados, o que quer dizer que terão a intervenção humana como factor necessário. Esta ajuda chega na forma de Actores que irão agir com o sistema.

## Actors

As this is a system, it is aimed for someone to use. In this sense it is important to define the actors which are going to use it. It is important Collaborative environment – Knowledge sharing – Engineering collaborative projects. As this is a tool to help improve an ontology, the first is the use of ontology engineer

Esta aplicação tem o objectivo de servir de apoio à descoberta de conhecimento num domínio, num contexto colaborativo (p.ex. contexto empresarial). Apesar das tentativas da criação de sistemas totalmente automatizados de manutenção de ontologias, ainda estamos longe dessa visão. Desta forma, os sistemas existentes são parcialmente automatizados, o que quer dizer que necessitam de intervenção humana. Esta intervenção chega na forma de especialistas que irão interagir com o sistema. Assim, definem-se os actores seguintes que serão quem irá interagir directamente no uso deste aplicativo, através de uma interface criada para o efeito.

Definem-se dois tipos de actores deste sistema: Especialistas de domínio e especialistas de ontologias. Os especialistas de domínio são os que apresentam um conhecimento aprofundado sobre as características do conhecimento relativo ao domínio da ontologia. Apresentam, avaliam e validam novos conceitos ou conceitos já existentes que estão de acordo com o domínio em questão.

Os especialistas de ontologia detêm o conhecimento da melhor forma de construir e manter uma ontologia. Serão eles os responsáveis pela própria manutenção de conceitos e relações presentes na mesma. Detém o conhecimento dos formalismos necessários para a definição de uma ontologia.

## Model

This work aims at propose a method to update the relations of the concepts of an ontology. One of the goals of the present thesis is to propose a model for knowledge discovery and ontology learning. Based in data mining process presented in Chapter 3.2.1 (Figure 3.2) this work presents a model for knowledge discovery and ontology learning as follows:

As already stated in this document, one of the goals is to extract knowledge from unstructured data. And this is achieved through a model inspired in the one presented in chapter three. Specifically all steps will be included, and after it is introduced another process/module to connect the extracted knowledge to the domain ontology, Specifically, Frequent Itemset Matching (FIM). (insert image here).

Architecture & Functional views??

This model will be developed based on a Model View Controller (MVC), an architectural pattern for software engineering design. This architecture clearly abstracts the system in a 3-tier model, allowing for a clear separation of concerns. Firstly, the *controller* which is the cerebrum of the system, is where all semantic related coding is developed. Additionally the controller makes the connection between the other two layers. Secondly, the *model* layer is responsible for all data related content, holding the databases necessary for the system. Lastly, the *view* layer is related to the interface of the system to the actors.

The functional view will be described by UML Use Case Diagram. UML is a graphical language aimed for modelling software systems. The Use Case Diagram reflects how the actors will interact with the system.

## Requirements

As a proof of concept DOKS system was developed based in a set of identified requirements, which were divided in three types, Functional, Architectural and Technical (see Figure 5.2). In the next lines some of this requirements, will be presented in more detail.

Some of the functional requirements are as follows: (i) *frequent items-concept matching*, which is related to the search of the ontology for the frequent items. Additionally, it they are found, it should retrieves the concept matching this frequent item; (ii) *new concepts discovery*, this requirement is based in the previous one and should be executed when no concept matches the frequent item in the ontology. In this case, the new concept discovered will be available to add to the B&C domain ontology associating the frequent item; (iii) *Association Rules Visualization*, refers to the system capacity to be able to show the association rules discovered, as well as the measures results, in an easy and understandable way; (iv) *Concept Relations Enrichment*, which relates to the process of using the unstructured data through the discovery of association rules with the goal to improve the relations between the concepts. *Best rule selection* is a functional requirement to allow the user to select the best concepts that match the FI discovered. *Best rules storage* represents the process of storing the rule in a database table in order to later evaluate it. With this evaluation, the analyst will decide if the relation should or should not be updated.



Figure 5.2 - System requirements

The Architectural requirements were *Ontology Connection*, to allow for the matching of new frequent items with the keywords that represent the concepts in it. *Knowledge Discovery* to be able to discover new concepts and new relations that could be of great use in the improving of the ontology. Should have a *remote and collaborative access* to allow for several users to use the solution system and make their own decisions also. The architecture of the system should be built in a *Modularity* approach. Should allow *system integration*. In addition, should be intuitive in its use.

*FP-Growth application* and *Association Rule Discovery*, are two requirements that were explained in the previous section **Erro! A origem da referência não foi encontrada.** of the present work. *User interface* is one of the technical requirements that is necessary for a user be able to visualize the rules and be able to select the best one for its intents. *Rapidminer Integration* is the use of its API in order that Rapidminer could make the searching and discovery of the rules. One place to save the rules chosen by the user is also a requirement, hence, it a *Rules Database* is a technical requirement.

## Conceptual & Technical Architectures

The system developed to achieve the solution to the proposed scenario is based on a Model View Controller (MVC) methodology. An MVC methodology is a method to present and organize the model in three components.

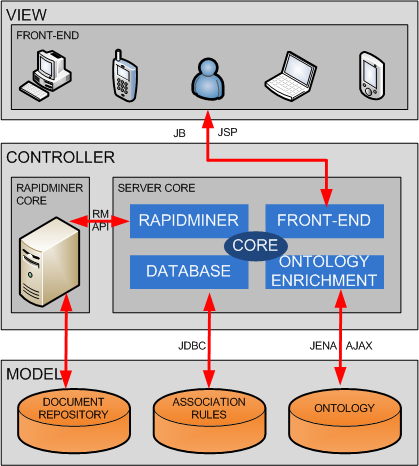


Figure 4.1 – System Architecture – MVC Methodology

In the *Model* component the proposed solution will hold all the databases and repository storing all the data for the system. The first one, Document Repository, is the initial repository with all the initial unstructured data to be analysed. In the second database, namely Association Rules, is the database created to store the rules and all the analysed intermediate data like the concepts, and the metadata related with these associated rules, and all values related to the measures used to classify them. It follows the model of Figure 4.2, where it can be observed a database composed by four tables, namely *concepts* is where all chosen concepts are stored, *rules* this serves to store all rules information of the chosen rules by the user, *rules\_stemmed* is the table that supports all the rule data that arrives from the rapidminer core and *stemmed\_words* is responsible to store all unique stemmed frequent items to further map with the concepts of the ontology. In the third one, called Ontology, is the main database to store the ontology itself. It will hold the concepts, and the relations to all of them.



Figure 4.2 – Entity Relation Model

*View* is the component that is responsible for the front-end. It is the system connection to the world. All the users will access the data by this interface. It represents the Association Rule Visualizer, the web system that presents to the user the rules discovered in the main process. It is responsible to allow the interaction of the user with the system. It receives the requests from the user and delivers them to the controller. As a result delivers the knowledge discovered to the visualizer.



Figure 4.3 – Knowledge layer architecture

*Controller* component is the core of the whole system. It is the artificial intelligent component where all the thinking, analysis and process is performed. In the present system proposal, it can be considered the controller with two cores. It holds the responsibility to receive the unstructured information, and through its processes deliver the knowledge discovered to update the ontology. In Figure 4.3 the reader may see a layered architecture of the various steps that the information is transformed to be able to extract knowledge to the ontology. The first core is the Rapidminer Core, is responsible to make all the steps from the unstructured data until the association rules discovery process. This element is connected to the server core by means of an API provided by Rapidminer. The second is the Server Core where all the logic is developed. It is where all computation to deliver the requests from the users is made, providing all the information from the database to the users.

The global operation of this controller tries to follow the conceptual architecture in Figure 4.4. As it can be observed, these steps are as follows: *Document Analysis* block represent the pre-process of the unstructured documents and transforms it in processed information. This processed information is received by the *Frequent Patterns* Block, and in this processor from the processed information, frequent patterns are extracted and delivered to the next block. The following block responsible to receive the frequent patterns is the *Association Rules* Block that receives these patterns and from it, discovers rules of association. Also in this step, one more block uses this processed information, namely the *Frequent Itemset Mapping* Block. This is responsible to find correspondence of the FI and the Concepts in the ontology. These last two blocks are the ones responsible to support the View component with the knowledge to the ontology expert user be capable to enrich the ontology with new knowledge.

Figure 4.4 – Core Conceptual Architecture

ASSOCIATION

RULES

FREQUENT ITEMSET MAPPING

ONTOLOGY

ASSOCIATION RULES DATABASE

DOCUMENT ANALYSIS

FREQUENT PATTERNS

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??

This chapter will illustrate all the information related to the implementation for the solution of the problem presented in this research. From the modelling of the problem, design, requirement analysis, and building until the development of a final solution. Based on the model proposal of the problem, all the initial requirements were fulfilled in the technical solution. This solution will be illustrated in the following lines with a description and explanation of the technologies adopted.

## Tools and Technologies

Several technologies and tools could be chosen to developed the present project, however, the author chose to use the following (Figure 5.1).

*Visual Paradigm for UML,* which is a visual design and modelling software tool, was used to develop all UML diagrams. Specifically, this software was used to make the Class Diagrams, Use Case Diagrams and Sequence Diagrams. Currently, two versions are available, student and professional. The chosen was the student version, which is free to use in academic projects.



**XPATH/XSLT**



Figure 5.1 - Tools and Technologies adopted

The database was modelled in SQL technology and implemented through the use of *MySQL Workbench* software tool. This tool which is normally used for database modelling, allowed for the creation of the Entity Relation Diagram (ERD), which was later translated into SQL with all statements necessary to build the database. The database runs on top of an Apache server provided by *XAMPP* software package. This package also provides an admin web portal aimed for database maintenance, specifically PHPMyAdmin.

*Rapidminer* was the tool used for the Data Mining process. Namely, Data Preprocessing, Pattern Discovery, Pattern Evaluation steps. This is a software tool commonly used in machine learning, data mining, text mining, predictive analytics and business analytics processes, just to name a few. All the pre-processing of the unstructured data was developed in this tool, as well as FP-growth and Association Rules algorithms. In order to provide the results adapted to DOKS system, a script based in Groovy[[13]](#footnote-13) technology was implemented as a Rapidminer module. This script language integrates well with all existing Java classes and libraries in the Rapidminer because it compiles straight to Java byte code so it can be used where Java is used. Additionally, Rapidminer provides a java based API, which was used to execute RM process aimed to obtain the results from the Data Mining process.

The core of this project was developed in Java technology in the NetBeans SE Platform. The IDE used to make all the coding of this project was NetBeans IDE. This IDE is an open tool that is very complete, which allowed the use of Java language, compiling of the code directly to JVM and also includes a debugger to check all the possible errors in the code or in the system run itself. All the necessary processes and methods to communicate with the Front-End based in JavaBeans interfaces. The communication with the database that holds the main model is made through JDBC connections (Figure 4.2).

XML technology was used for the ontology, and for the results of Rapidminer process, and were serialized with JDOM on the server side and on the client side they were accessed with XPATH/XSLT for presentation purposes.

Protegè was the tool to deal with the ontology as a user point of view. This tool provides a visual graphical interface for an ontology. It allows opening, visualizing and adding new or update concepts and relations manually. In the code of server side, this functionality was developed with the help of a Java API, Jena Semantic Framework Ontology that supports OWL language, inspired by XML, in which the ontology was created.

In the DOKS Front-End application developed the technologies used were HTML and CSS to create all pages supporting the presentation of the results, specifically association rules discovered and measured, as well as the association rules chosen. The interface was created in JSP to be able to integrate with the Java code. The final user client can be used in all modern browsers in the market, like Google Chrome, Mozilla Firefox or Internet Explorer.

## Implementation and Development

This chapter will describe all steps of the implementation and development proposed. This can be observed in Figure 5.3, a screenshot of the process of association rule generation. Although this process is delivered by the Rapidminer API, with its graphical tool was easier to see and build the main process. The boxes shown here represent the process until the discovery of Association Rule. In the following lines this main process will be explained in more detail.



Figure 5.3 – Rapidminer Main Process screenshot

### Document Analysis

Before one can discover the earlier ontologic relations, the source text documents with the unstructured information must be prepared in such way, so that can be understandable by this architecture. Some organization is necessary, along with some processing as the text is received in a raw state. To achieve to a processed information, the documents go through a procedure pipeline to pre-process them through rapidminer API (see Figure 5.4). This set of procedures is the Document Analysis Block. In fact, is composed by 6 blocks following a specific order.

Figure 5.4 - Document Analysis Pipeline Block

DOCUMENT ANALYSIS

Tokenize

Transform Cases

(lower cases)

Filter

Stopwords

Stemming

(Snowball)

Filter Tokens

(4-50)

Generate

n-grams (n≤3)

The first step of the preparation of the unstructured information is *Tokenize* process. The tokenize process is responsible for the separation of the full text into a sequence of tokens. Tokens can be understood in several ways, for the purpose of this work, one can consider token as a set of letters. Everything that does not have letters is discarded from this process, as punctuation or strange chars.

The second step in the pipeline is the *Transform Case* process and its main objective is to transform all tokens to lower case. This is a necessary step so that all tokens that differ in a letter case would be considered the same thus considering in the same respective frequency. For instance, the tokens *Token* and *token* are considered the same, as they have exactly the same letters in the same order, having just a capital letter different, consequently the one with the capital letter will be case lowered to *token*.



Figure 5.5 – Vector Creation Rapidminer Process

Next process in line is *Filter Stopwords* that is responsible for the stopwords filtration. These stopwords are words that have no semantic importance for the context, like *the*, *each*, *a*, etc. All these stopwords are removed of the set of tokens delivering them to the next process of the pipeline.

*Stem (Snowball)*, is the next process and corresponds to the execution of the stemming algorithm. This algorithm has the responsibility to transform the word in its stem, in other words, it will transform the word in its common morphological root. In this project the stemming algorithm used is the Snowball variation algorithm. This process can be optional, however one thinks it is of a great value, as it reduces the words to its stem, gathering them to the same word family to enrich its value in a document. Meaning that as more words are grouped for their stems, more representative is its stem in the document, consequently, reinforcing a better context. Another good advantage of the stemming process is the reduction of the data size augmenting each stem precision.

After all words are in their stem form, these set enters in the *Filter Tokens* process. This process makes a pruning operation in all tokens (words) that are lower than 4 and higher than 50 characters. This process is necessary to remove all unnecessary tokens that have no taxonomic relevance for the study, like chain of random letters, thus the author chose this interval as a fair number. Although one can consider relevant words with a minimum of 4 characters the upper value can be reduced to the minimum number of chars in each language. The value chosen of 50 chars as the biggest word is probably high for a word that also is stemmed, however this value was maintained for testing purpose. As an example, one can consider one of these tokens, for instance, the token “aaaaaaa”. It is bigger than 4 chars, and don’t have any relevant semantic value for this project.

The last step of the document analysis is the *Generation of n-grams*. The n-grams generation is the creation of sequences of 1 to N words, being for this case N=3, using unigrams, bigrams (eg. Waste Management) and trigrams (e.g. Electric Power Product). The purpose for this generation is a first try to find concepts and groups of words that represents concepts. The output of the analysis is saved into a temporary database for easement of processes.

Table 5.1 – Numerical to Binomial regulation

|  |  |  |  |
| --- | --- | --- | --- |
| NumBinMax | Support | Confidence | Association Rules |
| 0,012 | 0,25 | 0,01 | 18 |
| 0,012 | 0,25 | 0,60 | 18 |
| 0,012 | 0,25 | 0,70 | 18 |
| 0,013 | 0,20 | 0,01 | 102 |
| 0,013 | 0,25 | 0,70 | 12 |
| 0,014 | 0,20 | 0,01 | 92 |
| 0,015 | 0,20 | 0,01 | 92 |

In the interface between the analysis and the FP-growth process, the results enters afterwards in a sub-auxiliary process *Numerical to Binomial*, whose function is to change the nominal values of the vector to binomial values, which changes to false every value inside an interval, and to true the ones outside. This means that words that have no significant ontologic meaning are filtered out of the document corpus.

For the purpose of this work, the interval values were choose as follows: Min – 0.0 Max 0.013, as the Table 5.1 shows, some tests that were made to get the configuration of this module that gives a wider number of Association Rules to examine.

### FP-Growth

This block is responsible to find Frequent Patterns from the source documents that the author is analysing. The process uses the FP Growth algorithm, described earlier in this document. This process is the base for the recognition of frequent items that appear in the text more than others. The minimum frequency or support value, *minSup* for each FI in the database was chosen as 20% (0.20). All the FI that were below this frequency were not considered frequent and as a result pruned from the word set.

### Association Rules

This is the block that is responsible to find all the association rules. It receives the frequent items from the FP-Growth block and generates all AR based in the item set. The result is a new set with the rules discovered and six metrics used to classify them. This process takes advantage of the following measures: *Confidence*, *Conviction*, *Gain*, *Laplace*, *Lift*, *Ps* and *Total Support* (or just *Support*). This is the last process in the rapidminer main process. For a bigger number of rules discovered, the *minConf* value to filter the Confidence was made 1%. It was concluded that could also go higher until 65% as the lowest value of confidence in the result set was 66,7%.

### Frequent Itemset Mapping

Frequent Itemset Mapping Block is a module that is executed after the rapidminer processing. Applied to the FI discovered by FP-Growth. The main objective of this procedure is to map the FI discovered with the keywords associated to the concepts in ontology. The mapping is processed by the execution of the cosine similarity algorithm between frequent itemsets and the ontology concept keywords. Some considerations have to be made about this mapping. The FI is in the stem form and the keyword is a complete word, this suggests some extra logic in the comparison. The solution found is to compare the stem with the beginning of the word. For instance, if the FI stem discovered is *manag*, this will be compared to all keywords words started by this stem, namely *Management*, *Manager*, etc.

Table 5.2 gives an example of the results of a FI to Ontology Concept Keyword comparison where it can be observed a list with all the concepts that have associated keywords starting with the FI *manag*. It should be noted that these procedure finds the exact matches as well as candidate concepts. These are defined by the result of the Cosine Sinilarity Distance applied as obviously follows: 100% corresponds to exact matches, and below that as lower the value, farther is the concept from the FI.

Table 5.2 – Concept matches map for FI *manag*

|  |  |
| --- | --- |
| Concept match for manag | Cosine similarity distance (%) |
| Management Actor | **100** |
| Trainer | **100** |
| Manual | **50.00** |
| Waste Management Product | **50.00** |
| Chief Executive | **50.00** |
| President | **50.00** |
| Facility Manager | **50.00** |
| Managing Personnel Resources | **33.33** |
| Middle Management Actor | **33.33** |
| Executive Management Actor | **33.33** |
| Information Management Facility | **33.33** |
| Energy Management Facility | **33.33** |
| Managing Material Resources | **33.33** |
| Report | **33.33** |
| Team Assembly Phase | **33.33** |
| Managing Financial Resources | **33.33** |
| Resource Management Skill | **33.33** |
| Managing Time | **33.33** |
| Waste Management Facility | **33.33** |

Other example of FI mapping is given in Table 5.3. This is for *temperature* FI, and this presents some significant differences from the previous example. There are no exact matches found in the ontology, however there are candidates that can be chosen.

Table 5.3 - Concept candidates map for *temperature* FI

|  |  |
| --- | --- |
| Concept match for temperature | Cosine similarity distance (%) |
| Monitoring and Control of Internal Climate | **50.00** |
| Industrial Plant Performance Control | **50.00** |
| Environmental Detection And Registration | **50.00** |
| Temperature Measure Instrument | **33.00** |

#### When a table representing this operation is empty, it means that the FI is representative of a new concept. In that situation, the user is presented with an opportunity to enrich the ontology with a new concept, holding the FI as keyword related to it.

Moreover, it should also be observed that FIs are extracted in sets of more than 1 element. For the purpose of the present work, as already argued, the sets considered only included one. The main reason for this is to map directly the FI with one concept itself, and this situation rises an opportunity to improvement.

### Ontology Enrichment

Ontology management, and precisely ontology enrichment is a great challenge for an ontology.

These challenges and opportunities already discussed in Sub-Chapter **Erro! A origem da referência não foi encontrada.** present the goal to better aid the ontology administrator to enrich its ontology. As a result, this document proposes two scenarios.

The first scenario is the discovering of rules of association to propose some update or creation of the relations of the present Ontology. Furthermore, for a better organization of this knowledge, the author proposes a table structure presented in Table 5.4. This table will store each of the rules, one in each line, with all the associated information, namely the concepts, premise and conclusion already mapped, and the measure results of the respective rule. The administrator have a good organized way to see the association discovered, and thus choose to update the existing relations, based on this new knowledge.

Table 5.4 – Association Rules Database Structure

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # | Premise | Conclusion | Confidence | Conviction | Gain | Laplace | Lift. | Ps | Total Support |
| 1 | ConceptA | ConceptB | ValA | ValB | ValC | ValD | ValE | ValF | ValG |

The second scenario is the mapping of frequent items with keywords associated with concepts of the ontology to discover new concepts to enrich it. This scenario was explained in the previous sub-chapter.

These two scenarios are the proposed to enrich the ontology.

## Front end

This chapter will present the interface that users will use to discover knowledge in the unstructured data. It will be described with the help of screenshots representing each of the screens that represents the Front-End.



Figure 5.6 – Screenshot of AR system home page

The first screen to appear when the page opens is the Home Page. Figure 5.6 presents this page. This page is composed by a menu with three options, namely *Discover Association Rules (No Concepts)*, *Discover Association Rules* and *Analyse in RM and Renew DB*. The first option, discover association rules (no concepts) is to, as it name describes, find all association rules in the repository. The special factor in this operation is the *no concepts* element. This means that the rules are discovered and results are presented without the execution of the operation block *frequent itemset mapping*, in other words, the rules are presented with only the FI as Premise and Conclusion. The second option is the normal discovery of association rules. This operation includes the AR discovery and the frequent itemset mapping. The result presented by this is represented by Figure 5.7. One can see a screenshot of the first 5 rules. Finally, *analyse files in rm and renew db* is the operation responsible execute the rapidminer main process discussed in the previous chapter, and to make an operation of renovation of the association rules included in the respective database. Upon a new set of unstructured information received and placed in the right server folder, it executes the main process all over to find more interesting knowledge.

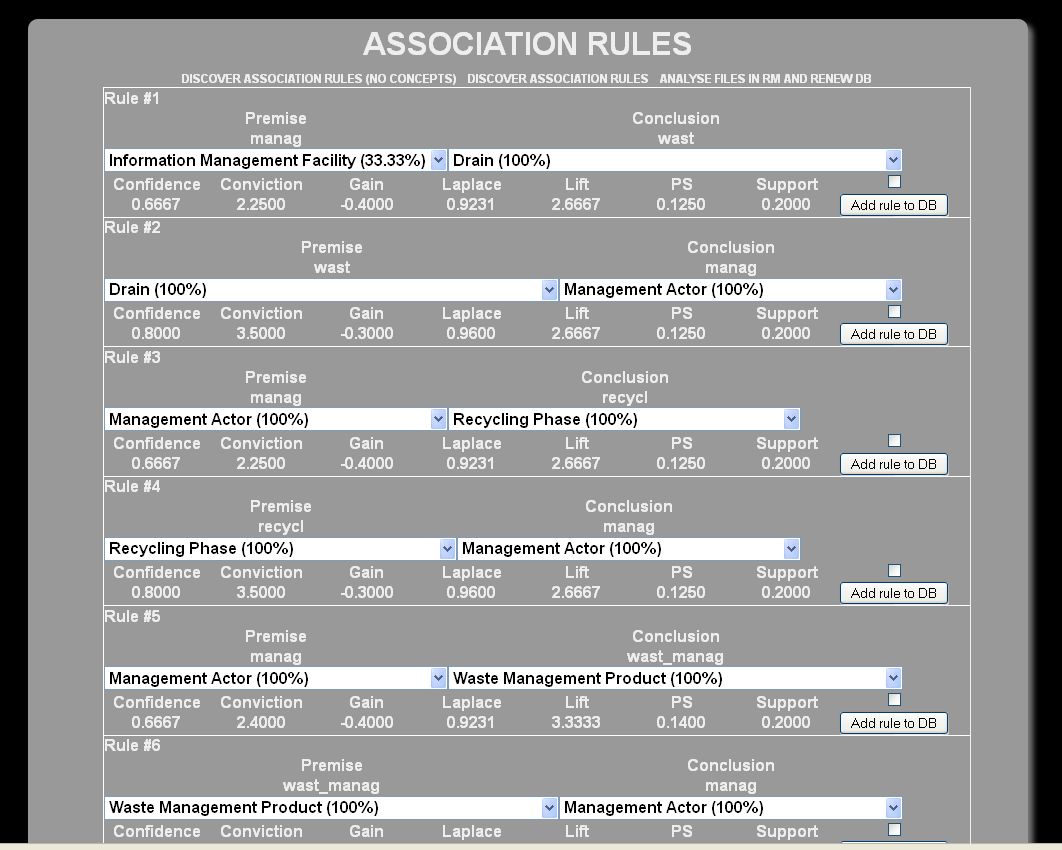


Figure 5.7 - Screenshot from Front-End AR page

After the server had executed the whole process, it delivers the information to the front end for visualization. Figure 5.7 shows the page with this information received from the server. All the rules discovered are delivered for presentation and divided in individual boxes for each one of them. Each box holds the information as follows: The premise and conclusion FI (for instance, in rule #1 are *manag* and *wast*), a dropdown list with all the results from the mapping process is available to choose the concept that best suits the user logic. Furthermore, the measures of the rule are presented for best evaluation. The ones expressed are the ones discussed on the earlier chapter 3.4.4 - Objective measures. As a result, one can see the following metrics presented for each rule, *Confidence*, *Conviction*, *Gain*, *Laplace*, *Lift*, *PS* and *Support*. Finally, a checkbox to select the rule to save, and a button to Add it to the database. When the button is pressed, the information with the selected concepts, for premise and conclusion, and all related measure values are added into the association rules database, and are presented in the Association Rule View Page as presented in Figure 5.10 - Association Rules Database view page. In the cases where the FI is not found in the ontology, the dropdown list will appear empty, with a link to insert the new concept in the ontology in that moment. This will be added to the ontology as an Individual with the respective FI as its first keyword, through a popup window that will request the user to choose the class where this new concept should be added. This case can be followed in Figure 5.9. This screenshot shows an example where the ontology mapping procedure does not find any keyword that match the FI, and the concepts set it returns is empty.

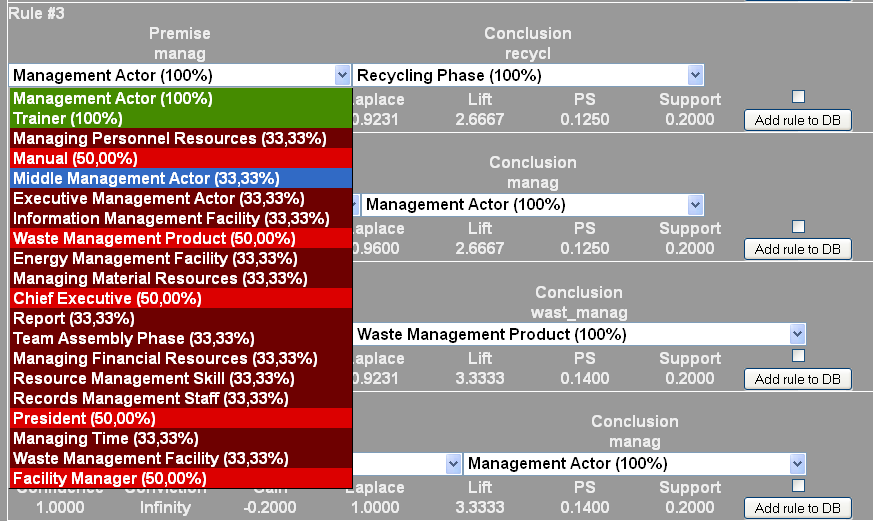


Figure 5.8 - Association Rule Functionality – Exact and Candidates concept visualizer

To choose a concept there are some considerations worth be noted. After each FI has been mapped into a list of candidate concepts, this list is presented with some information that could be relevant when choosing it. The first information is the value of its distance. This distance, as already presented in this document, refers to result of the Cosine Similarity Algorithm between the FI and the keyword that represents the concept in the database. The result of this value is presented in the form of percentage. If the value is 100% means that is an exact match, if the value is below it means this concept is similar to the original FI in means of its value, for instance, in Figure 5.8 one can see that Facility Manager presents 50% of similarity to *manag*. The other information worth notice is the color code presentation. A green color represent an exact match, the red colors represent candidates concepts. As lower the value of similarity becomes, more dark is the red color. Through this intuitive way of better presentation of the results, the user can quickly see which one better represents the FI.

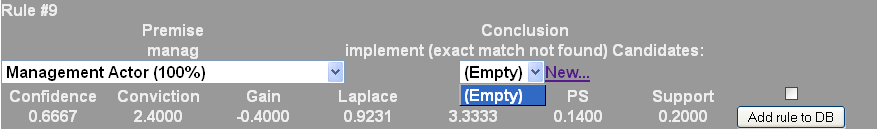


Figure 5.9 - AR Case with a new concept discovered

In order to see what the rules already chosen by the ontology administrator were, it was created a table to store them. This table is included in the Association Rules database. As a result, for the administrator be able to see this new chosen knowledge, it was created a visualization page. This page works with two objectives. The first, as already stated as visualization of the knowledge already stored in the database. The second is a status for the administrator so he can be able to know that the chosen rule was stored with the other knowledge. From this page, the administrator can update the ontology with this new knowledge accordingly. A screenshot of this situation is presented in Figure 5.8, where an example can be observed. Before the list of the included rules, is the status line to give the administrator the information of the stored rule, in this case he chosen the association rule between concept *Temperature Measuring Instrument* and concept *Complete Heating System*. This is also highlighted in the list in red.

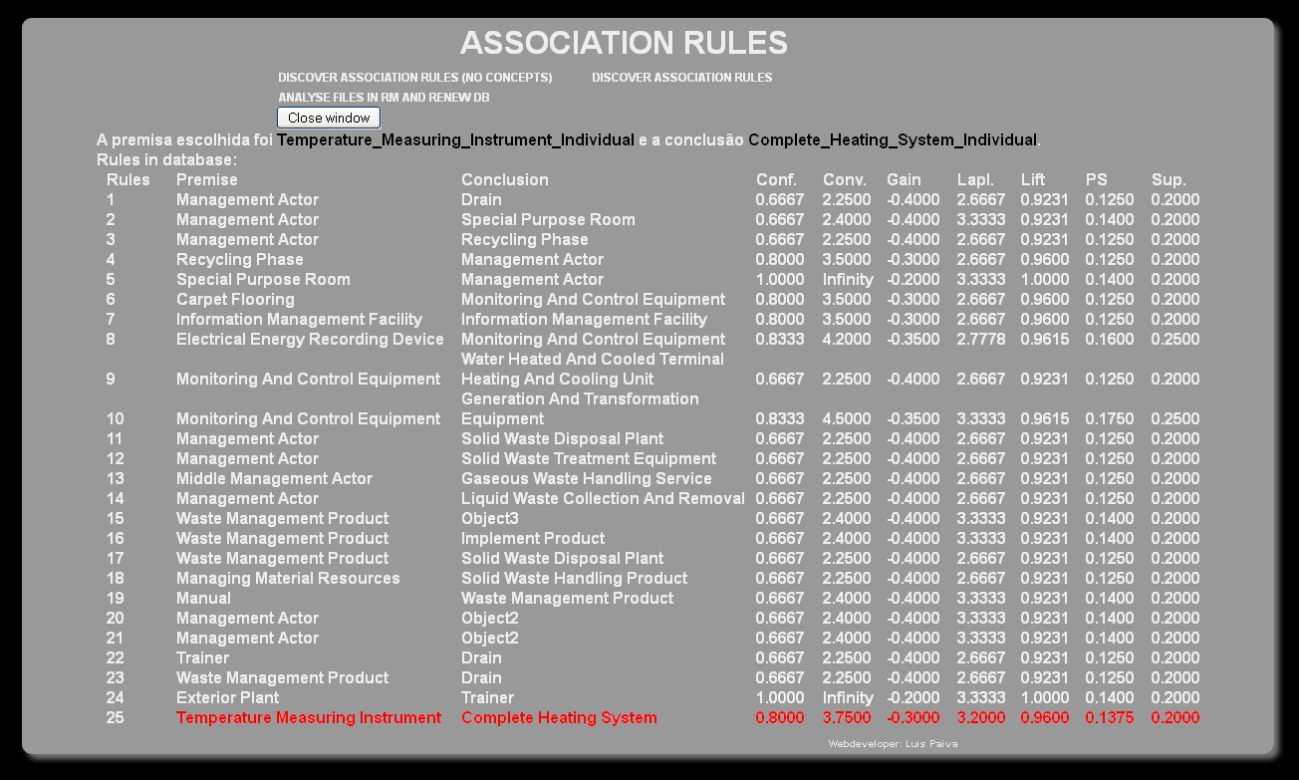


Figure 5.10 - Association Rules Database view page

6

# Assessment

Chapter 6 – Assessment

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

In this section the present work is going to be evaluated with some discussions on the respective domains, in fact, if the results were as expected.

This project received more nearly 20 files to evaluate. They were from sub-domains of Building and Construction Sector, in this case these were *Climate Control*, *Lightning*, *Coating*, *Sanitary* and *Waste Management*. The initial unstructured information included 62568 words to be evaluated. These were the initial data to be applied in this project.

The ontology was adopted from (Figueiras, 2012) project. Besides, this project was also a sample adapted from e-COGNOS project from the BC sector. A representation of Figueiras ontology can be observed in **Erro! A origem da referência não foi encontrada.** from this document, where can be recognized some inspiration in the e-COGNOS ontology itself. For instance, classes like *Actor*, *Product* and *Process* were cases of this inspiration.

Table 6.1 - All unique one-FI sets before AR discovery

|  |  |  |  |
| --- | --- | --- | --- |
| Unique FI | | | |
| manag | **Wast** | **recycl** | **wast\_manag** |
| plan | **implement** | **energi** | **consumpt** |
| temperatur | **Indoor** | **heat** | **energi\_consumpt** |
| electr | **Power** | **oper** | **hvac** |
| cool | **Toilet** | **sanitari** | **climat** |
| offic | **offic\_build** | **coat\_materi** | **coat** |

The resulting AR were very interesting and allows the author to some interesting discussion. With the previous argued values of minsup and minconf (refer to sub-section 5.3.2 and 5.3.3 respectively) the total discovered AR to evaluate were 102 rules, with the FP-Growth algorithm discovering 24 sets of one FI to provide for rule discovery premises and conclusions. These sets can be observed in Table 6.1. These rules only considered a set of frequent items in the premise with one FI. If this set could be expanded to 2 or more, the rules number would probably be higher and more knowledge would be discovered. Conversely, the case could be different, if the number of FI in the sets were too high there would be also an enormous number of rules to evaluate that could not help at all the ontology administrator. The quantity of ARs discovered were considered by the author of the present document as a good value for the administrator could proceed several good ontology enrichment.

Table 6.2 – AR with FI Manag as premise

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # | Premise | Conclusion | Confidence | Conviction | Gain | Laplace | Lift. | Ps | Total Support |
| 1 | Manag | Wast | 0.667 | 2.25 | -0.4 | 0.9231 | 2.667 | 0.125 | 0.20 |
| 2 | Manag | Recycl | 0.667 | 2.25 | -0.4 | 0.9231 | 2.667 | 0.125 | 0.20 |
| 3 | Manag | Wast\_Manag | 0.667 | 2.40 | -0.4 | 0.9231 | 3.333 | 0.140 | 0.20 |
| 4 | Manag | Plan | 0.667 | 2.40 | -0.4 | 0.9231 | 3.333 | 0.140 | 0.20 |
| 5 | Manag | Implement | 0.667 | 2.40 | -0.4 | 0.9231 | 3.333 | 0.140 | 0.20 |

Some interesting knowledge can be observed in the result set, for instance, in Table 6.2 FI *Manag* appears in 5 rules as a premise, as a result it originates 5 different conclusions. Moreover, when these rules conclude to *Manag*, the total sum of the rules is also 5 (refer to Table 6.4). Although *Manag* is present in the same number of sets in premise and conclusion, it can be argued that the rules are not directional. From sub-section 3.4.4 proven with the Conviction value different from each direction, the former situation can be easily observed. In fact, the conviction values for the each premise set, comparing to its conclusion are not the same. In other words, the semantic strength of the relation between *Manag* and *Wast*, for instance, depends on the direction of the rule.

Table 6.3 – AR with FI *Mang* as conclusion

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # | Premise | Conclusion | Confidence | Conviction | Gain | Laplace | Lift. | Ps | Total Support |
| 1 | Wast | Manag | 0.80 | 3.5000 | -0.30 | 0.96 | 2.67 | 0.125 | 0.20 |
| 2 | Recycl | Manag | 0.80 | 3.5000 | -0.30 | 0.96 | 2.67 | 0.125 | 0.20 |
| 3 | Wast\_Manag | Manag | 1.00 | ∞ | -0.20 | 1.00 | 3.33 | 0.140 | 0.20 |
| 4 | Plan | Manag | 1.00 | ∞ | -0.20 | 1.00 | 3.33 | 0.140 | 0.20 |
| 5 | Implement | Manag | 1.00 | ∞ | -0.20 | 1.00 | 3.33 | 0.140 | 0.20 |

Although the rules with different conviction values from the same elements as premise and conclusion, could present some interesting knowledge, there are cases where the rules are bidirectional. In other words, this means that both FI are related to each other in the same sense. Consequently, this could arguably mean that these concepts belong to the same knowledge domain. As a result, these could be good cases to update relations in ontology or create new ones. Although the support value could be in the lower bound of minsup, the Confidence of this association rule discovery proves a 100% relation. Refer to Table 6.4 for examples of this knowledge discovered in the rule set.

Table 6.4 - AR - bidirectional rules

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # | Premise | Conclusion | Confidence | Conviction | Gain | Laplace | Lift. | Ps | Total Support |
| 1 | Recycl | Wast | 1.00 | ∞ | -0.25 | 1.00 | 4.00 | 0.187 | 0.25 |
| 2 | Wast | Recycl | 1.00 | ∞ | -0.25 | 1.00 | 4.00 | 0.187 | 0.25 |
| 3 | Hvac | Cool | 1.00 | ∞ | -0.20 | 1.00 | 5.00 | 0.160 | 0.20 |
| 4 | Cool | Hvac | 1.00 | ∞ | -0.20 | 1.00 | 5.00 | 0.160 | 0.20 |
| 5 | Wast\_Manag | Plan | 1.00 | ∞ | -0.20 | 1.00 | 5.00 | 0.160 | 0.20 |
| 6 | Plan | Wast\_Manag | 1.00 | ∞ | -0.20 | 1.00 | 5.00 | 0.160 | 0.20 |

Nevertheless, the rules considered unidirectional, or in other words, the rules that do not share the same evaluation in both direction (refer Table 6.5 for examples of this situation) could also present unexpected and relevant knowledge. For instance, whenever *Plan* occurs, *Wast* also occurs with 100% sure. In contrast, whenever *Wast* occurs, *Plan* occurs with 80%.

Table 6.5 - AR - unidirectional rules examples

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # | Premise | Conclusion | Confidence | Conviction | Gain | Laplace | Lift. | Ps | Total Support |
| 1 | Plan | Wast | 1.00 | ∞ | -0.20 | 1.00 | 4.00 | 0.150 | 0.20 |
| 2 | Wast | Plan | 0.80 | 4.00 | -0.30 | 0.96 | 4.00 | 0.150 | 0.20 |
| 3 | Toilet | Sanitari | 0.80 | 4.00 | -0.30 | 0.96 | 4.00 | 0.150 | 0.20 |
| 4 | Sanitari | Toilet | 1.00 | ∞ | -0.20 | 1.00 | 4.00 | 0.150 | 0.20 |
| 5 | Electr | Power | 0.80 | 4.00 | -0.30 | 0.96 | 4.00 | 0.150 | 0.20 |
| 6 | Power | Electr | 1.00 | ∞ | -0.20 | 1.00 | 4.00 | 0.150 | 0.20 |

The present work presents several association rules measures to help an ontology administrator update the ontology. For instance, considering Confidence, when this value results in 100%, one can see that Conviction value equals to ∞. Even with support values of 20% as can be observed in rule #4 of Table 6.5.

Furthermore, the mapping of the FI with the concepts of the ontology can provide great and valuable knowledge for its enrichment. If a rule does not presents a confidence value of 100%, still creates doubts, the correspondence of the FI with the ontology keyword, might help to dissipate, or at least reduce this doubt. Furthermore, as higher the mapping value, higher is the certainty of the presence of that concept in the rule, in other words, the semantic distance of the two concepts evaluated in a rule is higher, and the uncertainty is lower, with the presence of this mapping.

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 chapter3.3.3 and 3.4 respectively). 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[[14]](#footnote-14) 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[[15]](#footnote-15) or Latent Semantic Analysis[[16]](#footnote-16) 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[[17]](#footnote-17), 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. 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. Presented at the 2011 International Conference on Computational Intelligence and Communication Networks (CICN), pp. 498–502. doi:10.1109/CICN.2011.104

Bonchi, F., Goethals, B., 2004. FP-Bonsai: The Art of Growing and Pruning Small FP-Trees, in: Dai, H., Srikant, R., Zhang, C. (Eds.), Advances in Knowledge Discovery and Data Mining, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 155–160.

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.

Brachman, R.J., Selfridge, P.G., Terveen, L.G., Altman, B., Borgida, A., Halper, F., Kirk, T., Lazar, A., McGuinness, D.L., Resnick, L.A., 1993. Integrated support for data archaeology. Int. J. Intell. Coop. Inf. Syst. 2, 159–185.

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

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.

Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., 1996. The KDD process for extracting useful knowledge from volumes of data. Commun. ACM 39, 27–34.

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.

Garg, K., Kumar, D., 2013. Comparing the Performance of Frequent Pattern Mining Algorithms. Int. J. Comput. Appl. 69, 21–28.

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.

Google, 2013. Google.com [WWW Document]. URL https://www.google.com/ (accessed 7.7.14).

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., Kamber, M., Pei, J., 2011. Data mining: concepts and techniques, 3rd edition. ed. Elsevier.

Han, J., Pei, J., Yin, Y., 2000. Mining frequent patterns without candidate generation, in: ACM SIGMOD Record. ACM, pp. 1–12.

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.

Hilbert, M., López, P., 2011. The world’s technological capacity to store, communicate, and compute information. science 332, 60–65.

Hilderman, R.J., Hamilton, H.J., 1999. Knowledge discovery and interestingness measures: A survey. Department of Computer Science, University of Regina.

Hjelseth, E., 2009. Foundation for development of computable rules. Nor. Univ. Life Sci. UMB Dept Math. Sci. Technol. Nor.

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.

Inmon, W.H., Osterfelt, S., 1991. Understanding data pattern processing: the key to competitive advantage. QED Information Sciences, Inc.

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.

Korczak, J., Skrzypczak, P., 2012. FP-Growth in Discovery of Customer Patterns, in: Aberer, K., Damiani, E., Dillon, T. (Eds.), Data-Driven Process Discovery and Analysis, Lecture Notes in Business Information Processing. Springer Berlin Heidelberg, pp. 120–133.

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

Loukides, M., 2010. What is data science.

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.

Mahgoub, H., 2006. Mining association rules from unstructured documents, in: Proc. 3rd Int. Conf. on Knowledge Mining, ICKM, Prague, Czech Republic. pp. 167–172.

Memon, N., Hicks, D.L., Larsen, H.L., 2007. Notice of Violation of IEEE Publication PrinciplesHarvesting Terrorists Information from Web, in: Information Visualization, 2007. IV ’07. 11th International Conference. Presented at the Information Visualization, 2007. IV ’07. 11th International Conference, pp. 664–671. doi:10.1109/IV.2007.60

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.

Paiva, L., Costa, R., Figueiras, P., Lima, C., 2013. Discovering Semantic Relations from Unstructured Data for Ontology Enrichment - Asssociation rules based approach. Presented at the CISTI’2013 - 8a Conferência Ibérica de Sistemas e Tecnologias de Informação, AISTI, Lisboa, pp. 579–584.

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.

Rácz, B., 2004. nonordfp: An FP-growth variation without rebuilding the FP-tree., in: FIMI.

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. 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).

Steyvers, M., Smyth, P., Rosen-Zvi, M., Griffiths, T., 2004. Probabilistic Author-topic Models for Information Discovery, in: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’04. ACM, New York, NY, USA, pp. 306–315. doi:10.1145/1014052.1014087

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

Tan, X., Pan, H., Han, Q., Ni, J., 2009. Domain knowledge-driven association pattern mining algorithm on medical images, in: Internet Computing for Science and Engineering (ICICSE), 2009 Fourth International Conference on. IEEE, pp. 30–35.

Tomi Kauppinen, H.K., 2009. Extending an Ontology by Analyzing Annotation Co-occurrences in a Semantic Cultural Heritage Portal.

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).

Waller, M.A., Fawcett, S.E., 2013. Data Science, Predictive Analytics, and Big Data: A Revolution That Will Transform Supply Chain Design and Management. J. Bus. Logist. 34, 77–84. doi:10.1111/jbl.12010

Wang, K., Tang, L., Han, J., Liu, J., 2002. Top down FP-Growth for association rule mining. Springer.

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).

Yao, Y., Chen, Y., Yang, X., 2006. A Measurement-Theoretic Foundation of Rule Interestingness Evaluation, in: Lin, P.T.Y., Ohsuga, P.S., Liau, D.C.-J., Hu, P.X. (Eds.), Foundations and Novel Approaches in Data Mining, Studies in Computational Intelligence. Springer Berlin Heidelberg, pp. 41–59.

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

Zaki, M.J., Gouda, K., 2003. Fast vertical mining using diffsets, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp. 326–335.

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. These words are referred to, in NLP literature as stop words, and can include articles (e.g. “the”, “that”, “a”, “an”) or prepositions (e.g. “by”, “from”, “in”). [↑](#footnote-ref-5)
6. *Lemma* word – a lemma is the canonical form of a set of words from the same family. E.g. Words *friend*, *friendship*, *friendly* have its lemma as *friend*. [↑](#footnote-ref-6)
7. **Breadth-first search** (**BFS**) is an algorithm for traversing or searching data structures. It starts through the initial elements and explores its neighbours before advance to the next element. [↑](#footnote-ref-7)
8. **Depth-first search** (**DFS**) is an algorithm for traversing or searching data structures. It starts at an initial arbitrary element and explores as far as possible along each branch before advance to a neighbour element. [↑](#footnote-ref-8)
9. Although, these two itemsets can be found in literature with other names as antecedent and consequent (Hoque et al., 2011), this thesis adopts the nomenclature of premise and conclusion (Costa, 2014). [↑](#footnote-ref-9)
10. Medline (PubMed) – Wide library of scientific publications in the medicine research field. Please refer to http://www.ncbi.nlm.nih.gov/pubmed/ for more information. [↑](#footnote-ref-10)
11. As a curiosity, Lift is known in the scientific community as a measure used in the Intelligent Miner from IBM (IBM - International Business Machines, 1996).. [↑](#footnote-ref-11)
12. 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-12)
13. Groovy, a technology for JVM, builds upon the strengths of Java but has additional power features inspired by languages like Python, Ruby and Smalltalk. (Guillaume Laforge, 2012) [↑](#footnote-ref-13)
14. ICONDA is a large database of technical documents related to B&C domain. [↑](#footnote-ref-14)
15. 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-15)
16. 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-16)
17. 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-17)