

**Luis Miguel Sintra Salvo Paiva**

**Semantic relations extraction from unstructured information for domain ontologies enrichment**

Dissertação para obtenção do Grau de Mestre em

Engenharia Electrotécnica e de Computadores

Orientador: Professor Pedro Maló, FCT-UNL

Co-orientador: Professor Celson Lima, MIT

Júri: *(Font: Arial, 10 pt normal)*

|  |  |
| --- | --- |
| Presidente: | Prof. Doutor(a) [Nome Completo] |
| Arguente(s): | Prof. Doutor(a) [Nome Completo] |
| Vogal(ais): | Prof. Doutor Pedro Maló |
|  | Prof. Doutor Celson Lima |
|  | Mestre Ruben Costa |

*(Tipo de letra: Arial, 10 pt normal)*



**Setembro 2014**



**Luis Miguel Sintra Salvo Paiva**

**Semantic relations extraction in unstructured information for domain ontologies enrichment**

Dissertação para obtenção do Grau de Mestre em

Engenharia Electrotécnica e de Computadores

Orientador: Professor Pedro Maló, FCT-UNL

Co-orientador: Professor Celson Lima, MIT

Júri: *(Font: Arial, 10 pt normal)*

|  |  |
| --- | --- |
| Presidente: | Prof. Doutor(a) [Nome Completo] |
| Arguente(s): | Prof. Doutor(a) [Nome Completo] |
| Vogal(ais): | Prof. Doutor Pedro Maló |
|  | Prof. Doutor Celson Lima |
|  | Mestre Ruben Costa |

*(Tipo de letra: Arial, 10 pt normal)*



**Setembro 2014**

**COPYRIGHT**

**Semantic relations extraction in unstructured information for domain ontologies enrichment**

Luis Miguel Sintra Salvo Paiva – Todos os direitos reservados

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

# Acknowledgements

As this was a long path to walk, I hope I can give the deserved merit to all that help me in some way to achieve this great and fulfilling goal.

# Abstract

(Ver em português para traduzir) Since semantic web appearance, several domain ontologies were developed and delivered in open access repositories. The existing domain ontologies describe semantic elements specific to a particular domain. These elements can be used complementing existing document information. This articulation can be duly empowered if, new methods could be created that in a semi automatic way, could help the ontologic precision. Specifically, the new patterns originating the building of new knowledge will be extracted not only from domain ontologies, but also from unstructured information sources.

Knowing the existence of scientific contributions in knowledge patterns extraction aimed to semantic enrichment, none considers the information inside an ontology. Desta forma, this document contributes with a method proposal of relations quantification between concepts from unstructured information, namely documents, and from knowledge existent in ontology to enrich it. To address this issue, this method is based on data mining techniques, specifically a technique to recognize semantic associations. This technique discovers data, namely concepts, that co-occurs from unstructured information.

To reach the proposed goal, a software support tool was developed, DOKES (*Dynamic Ontology Knowledge Enrichment Support*), that can be applied to any domain, however adapted to building and construction domain. DOKES main goal is to extract relevant knowledge from unstructured information and help its users enrich ontology.

Keywords: Artificial Intelligence, Semantic Web, Knowledge Discovery, Ontology Enrichment, Association Rules, Frequent Pattern

# Sumário

Desde o aparecimento da *web* semântica, várias ontologias de domínio foram desenvolvidas e disponibilizadas em repositórios de acesso aberto. As ontologias de domínio existentes pretendem descrevem elementos semânticos, como conceitos e as relações entre estes, específicos a uma determinada área. Elementos esses que podem ser usados como forma de complemento à informação existente em documentos. Esta complementaridade entre os elementos semânticos e os documentos, poderá ser devidamente potenciada, se forem criados novos métodos que, auxiliem o enriquecimento das ontologias. Mais especificamente, os novos padrões que dão origem a conhecimento novo ou enriquecido poderão ser extraídos não só de ontologias de domínio, como também de fontes de informação não estruturada.

Sabendo que existem bastantes contribuições ciêntíficas na área da extracção de padrões de conhecimento para enriquecimento semântico, destas nenhuma considera a informação contida na ontologia. Desta forma, este documento tem como principal objectivo contribuir com uma proposta de um método de quantificação de relações entre conceitos a partir de fontes de informação não estruturada, nomeadamente documentos, e do conhecimento já existente na ontologia, para enriquecimento das mesmas. De forma a desenvolver tal abordagem, este método tem como base, o recurso a técnicas de *data mining*, especificamente a uma técnica de reconhecimento de associações semânticas. Descrita como uma técnica para descoberta da co-ocorrência de dados, nomeadamente conceitos, a partir de informação não estruturada.

Para atingir o objectivo proposto, foi desenvolvida uma ferramenta de suporte a decisões para enriquecimento de conhecimento na ontologia, chamada DOKES (*Dynamic Ontology Knowledge Enrichment Support*), abrangente a qualquer domínio, mas adaptada ao sector da construção civil. O DOKES tem como principal objectivo a extracção de conhecimento relevante a partir de informação não estruturada e ajudar os seus utilizadores a enriquecer a ontologia.

Palavras-Chave: Inteligência Artificial, Web Semântica, Descoberta de Conhecimento, Enriquecimento de Ontologias, Regras de Associação, Padrões Frequentes

*Dedico a concretização desta etapa, finalizada por esta dissertação aos meus Pais, Mário Luiz e Maria Edite…*

*“Always look on the bright side of life!”*

*Monty Python, in “Life of Brian”*

Contents

[Acknowledgements V](#_Toc398077898)

[Abstract VII](#_Toc398077899)

[Sumário IX](#_Toc398077900)

[Figures XVII](#_Toc398077901)

[Tables XIX](#_Toc398077902)

[Symbols and Notation XXI](#_Toc398077903)

[1 Introduction 1](#_Toc398077904)

[1.1 Motivation 2](#_Toc398077905)

[1.2 Vision 4](#_Toc398077906)

[1.3 Development context 4](#_Toc398077907)

[1.4 Dissertation Structure 5](#_Toc398077908)

[2 State of the Art / Related Work 7](#_Toc398077909)

[2.1 Ontology Learning 7](#_Toc398077910)

[2.2 Association rules 7](#_Toc398077911)

[2.3 Building & Construction 10](#_Toc398077912)

[3 Theoretical and Technical Foundation 13](#_Toc398077913)

[3.1 Association Rules 13](#_Toc398077914)

[3.1.1 Frequent Pattern Growth 16](#_Toc398077915)

[3.1.2 Association Rules Measurement 24](#_Toc398077916)

[3.2 Vector Space Model 30](#_Toc398077917)

[3.2.1 Term Weighting – The TF-IDF 31](#_Toc398077918)

[3.3 Similarity Measure in Information Retrieval 32](#_Toc398077919)

[3.3.1 Cosine Similarity Measure Algorithm 33](#_Toc398077920)

[4 Building & Construction Domain Ontology 35](#_Toc398077921)

[4.1 Ontology 35](#_Toc398077922)

[4.1.1 Construction Methodology 35](#_Toc398077923)

[4.1.2 Ontologic management challenge 37](#_Toc398077924)

[4.2 The E-COGNOS Ontology 38](#_Toc398077925)

[5 Design and Implementation 41](#_Toc398077926)

[5.1 Tools and Technologies 41](#_Toc398077927)

[5.2 Requirements 42](#_Toc398077928)

[5.3 Conceptual & Technical Architectures 44](#_Toc398077929)

[5.4 Implementation and Development 47](#_Toc398077930)

[5.4.1 Document Analysis 47](#_Toc398077931)

[5.4.2 FP-Growth 50](#_Toc398077932)

[5.4.3 Association Rules 50](#_Toc398077933)

[5.4.4 Frequent Itemset Mapping 50](#_Toc398077934)

[5.4.5 Ontology Enrichment 51](#_Toc398077935)

[5.5 Front end 52](#_Toc398077936)

[6 Evaluation 57](#_Toc398077937)

[6.1 Scientific publications 59](#_Toc398077938)

[7 Conclusion and Future Work 61](#_Toc398077939)

[8 Bibliography 63](#_Toc398077940)

[Appendix A 68](#_Toc398077941)

[Appendix B. 69](#_Toc398077942)

[Appendix C 70](#_Toc398077943)

[Appendix D 72](#_Toc398077944)

# Figures

[Figure 1.1 – Darwin’s Evolution Theory (Darwin, 1859) 1](#_Toc398077945)

[Figure 2.1 – Study fields examples 8](#_Toc398077946)

[Figure 3.1 – An FP-Tree example for the items in the transaction Table 3.1. 18](file:///F:\Os%20meus%20documentos\Universidade\Dissertação\Escrita\working\inprocess\Dissertação%20v6%20(Tudo).docx#_Toc398077947)

[Figure 3.2 – The three initial trees at the end of the first three transactions 20](file:///F:\Os%20meus%20documentos\Universidade\Dissertação\Escrita\working\inprocess\Dissertação%20v6%20(Tudo).docx#_Toc398077948)

[Figure 3.3 – Prefix sub-paths for all frequent items 21](file:///F:\Os%20meus%20documentos\Universidade\Dissertação\Escrita\working\inprocess\Dissertação%20v6%20(Tudo).docx#_Toc398077949)

[Figure 3.4 – Conditional FP-Tree for item *professor* 23](file:///F:\Os%20meus%20documentos\Universidade\Dissertação\Escrita\working\inprocess\Dissertação%20v6%20(Tudo).docx#_Toc398077950)

[Figure 3.5 – Interestingness measures types tree (adapted from Silberschatz and Tuzhilin, (1995)) 25](#_Toc398077951)

[Figure 4.1 - Ontology example in civil construction domain 36](file:///F:\Os%20meus%20documentos\Universidade\Dissertação\Escrita\working\inprocess\Dissertação%20v6%20(Tudo).docx#_Toc398077952)

[Figure 4.2 - Knowledge representation technologies - layered approach 37](file:///F:\Os%20meus%20documentos\Universidade\Dissertação\Escrita\working\inprocess\Dissertação%20v6%20(Tudo).docx#_Toc398077953)

[Figure 4.3 – Domains represented in e-COGNOS ontology (adapted from (Lima et al., 2003b)) 38](file:///F:\Os%20meus%20documentos\Universidade\Dissertação\Escrita\working\inprocess\Dissertação%20v6%20(Tudo).docx#_Toc398077954)

[Figure 5.1 - Tools and Technologies adopted 41](file:///F:\Os%20meus%20documentos\Universidade\Dissertação\Escrita\working\inprocess\Dissertação%20v6%20(Tudo).docx#_Toc398077955)

[Figure 5.2 - System requirements 43](#_Toc398077956)

[Figure 5.3 – System Architecture – MVC Methodology 44](#_Toc398077957)

[Figure 5.4 – Entity Relation Model 45](#_Toc398077958)

[Figure 5.5 – Knowledge layer architecture 46](#_Toc398077959)

[Figure 5.7 – Rapidminer Main Process screenshot 47](#_Toc398077960)

[Figure 5.6 – Core Conceptual Architecture 47](file:///F:\Os%20meus%20documentos\Universidade\Dissertação\Escrita\working\inprocess\Dissertação%20v6%20(Tudo).docx#_Toc398077961)

[Figure 5.9 – Vector Creation Rapidminer Process 48](#_Toc398077962)

[Figure 5.8 - Document Analysis Pipeline Block 48](file:///F:\Os%20meus%20documentos\Universidade\Dissertação\Escrita\working\inprocess\Dissertação%20v6%20(Tudo).docx#_Toc398077963)

[Figure 5.10 – Screenshot of AR system home page 52](#_Toc398077964)

[Figure 5.11 - Screenshot from Front-End AR page 53](#_Toc398077965)

[Figure 5.12 - Association Rule Functionality – Exact and Candidates concept visualizer 54](#_Toc398077966)

[Figure 5.13 - AR Case with a new concept discovered 55](#_Toc398077967)

[Figure 5.14 - Association Rules Database view page 55](#_Toc398077968)

[Figure A.1 - USD for Insert Rule Use Case 68](file:///F:\Os%20meus%20documentos\Universidade\Dissertação\Escrita\working\inprocess\Dissertação%20v6%20(Tudo).docx#_Toc398077969)

[Figure B.1 – UML CLASS DIAGRAM 69](#_Toc398077970)

[Figure C.1 - Main requirements UUC 70](#_Toc398077971)

[Figure C.2 - Front-End UUC 70](#_Toc398077972)

[Figure C.3 - Rule DB UUC 71](#_Toc398077973)

# Tables

[Table 2.1 - Examples of Fields of Study with Association Rules Application 9](#_Toc398077974)

[Table 3.1 – Transaction table for frequent items in database 17](#_Toc398077975)

[Table 3.2 – Paths table for frequent items 22](#_Toc398077976)

[Table 3.3 – Frequent itemsets discovered for all items 23](#_Toc398077977)

[Table 5.1 – Numerical to Binomial regulation 49](#_Toc398077978)

[Table 5.2 – Concept matches map for FI *manag* 51](#_Toc398077979)

[Table 5.3 - Concept candidates map for *temperature* FI 51](#_Toc398077980)

[Table 5.4 – Association Rules Database Structure 52](#_Toc398077981)

[Table 6.1 - All unique one-FI sets before AR discovery 57](#_Toc398077982)

[Table 6.2 – AR with FI Manag as premise 57](#_Toc398077983)

[Table 6.3 – AR with FI *Mang* as conclusion 58](#_Toc398077984)

[Table 6.4 - AR - bidirectional rules 58](#_Toc398077985)

[Table 6.5 - AR - unidirectional rules examples 58](#_Toc398077986)

[Table D.1 - List of all the rules discovered in Association Rules Discovery Process 72](#_Toc398077987)

# Symbols and Notation

|  |  |  |
| --- | --- | --- |
|  | **AEC** | **A**rchitecture, **E**ngineering and **C**onstruction |
|  | **AI** | **A**rtificial **I**ntelligence |
|  | **API** | **A**pplication **P**rogramming **I**nterface |
|  | **AR** | **A**ssociation **R**ules |
|  | **ASP** | **A**ctive **S**erver **P**ages |
|  | **CSS** | **C**ascading **S**tyle **S**heet |
|  | **DB** | **D**ata**B**ase |
|  | **DOKES** | **D**ynamic **O**ntology **K**nowledge **E**nrichment **S**ystem |
|  | **ECLAT** | **E**quivalent **CLA**ss **T**ranformation |
|  | **ERD** | **E**ntity **R**elation **D**iagram |
|  | **FI** | **F**requent **I**tem |
|  | **FP** | **F**requent **P**attern |
|  | **HTML** | **H**yper **T**ext **M**arkup **L**anguage |
|  | **IDE** | **I**ntegrated **D**evelopment **E**nvironment |
|  | **IR** | **I**nformation **R**etrieval |
|  | **IT** | **I**nformation **T**echnology |
|  | **JB** | **J**ava**N**eans |
|  | **JDBC** | **J**ava **D**ata**B**ase **C**onnection |
|  | **JDOM** | **J**ava **D**ocument **O**bject **M**odel |
|  | **JSP** | **J**ava **S**erver **P**age |
|  | **JVM** | **J**ava **V**irtual **M**achine |
|  | **KD** | **K**nowledge **D**iscovery |
|  | **MVC** | **M**odel **V**iew **C**ontroller |
|  | **OWL** | **W**eb **O**ntology **L**anguage |
|  | **RDF** | **R**esource **D**escription **F**ramework |
|  | **RM** | **R**apid**M**iner |
|  | **SQL** | **S**tructured **Q**uery **L**anguage |
|  | **TF-IDF** | **T**erm **F**requency – **I**nverse **D**ocument **F**requency |
|  | **TID** | **T**ransaction **ID**entification |
|  | **KEDHS** | **K**nowledge **E**nrichment **D**ecision **H**elper **S**ystem |
|  | **UML** | **U**nified **M**odelling **L**anguage |
|  | **USD** | **U**ML **S**tate **D**iagram |
|  | **USQD** | **U**ML **S**e**Q**uence **D**iagram |
|  | **UUC** | **U**ML **U**se **C**ase |
|  | **VSM** | **V**ector **S**pace **M**odel |
|  | **W3C** | **W**orld **W**ide **W**eb **C**onsortium |
|  | **XML** | e**X**tensible **M**arkup **L**anguage |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Introduction

Since the ancient times, that mankind evolves in the pace of knowledge. Theory of evolution (Darwin, 1859) provide us with several proven historic facts that turns knowledge one of the most important motors that drive evolution. For instance, when *homo erectus* discovered fire, more than 1.5 million years ago, back in the Stone Age, he acquired new knowledge. Knowledge to use the heat to make him comfortable and provided him with the opportunity to further evolve. He was able to use the fire as a heat source to make the environment warmer, or to be able to eat cooked food, or to use fire as a light source to illuminate its environment as an alternative to the only light source present at that time, the Sun. Similarly, on the importance of knowledge discovery, the wheel discovery was another important knowledge acquired by *Sumerians* in 3500 b.c. that allowed them to evolve even further. When the discovery of the wheel was made it represented a great technological breakthrough, and it was not even invented as we see it today, it was used to serve as an industrial artefact to help make pottery. Only some years later it was used in chariots for vehicle movement. It could be claimed that knowledge itself is one of the supports of mankind evolution.

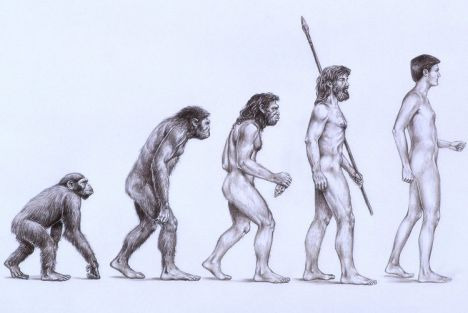


Figure 1.1 – Darwin’s Evolution Theory (Darwin, 1859)

Before information systems appeared, two of the known ways to store knowledge over time was the human brain, the first, or in the books, a later one. Human brains are still a very complex structure that stores lots of information and seems that it does not runs out of space. On the other side are books, as we know them today, were originally used by monks and in the first known universities to store the knowledge and share it between its communities.

In the modern world, knowledge value is huge. Its rising quantity available to everyone is such as the ease of its availability to every person. The global and exponential growth use of information systems by people and businesses, and the growth of the internet changed the way to look to this knowledge and the forms to collect, select and distribute. Every day, in most human daily tasks related to a computer, information records are created and new databases appear. It is thus more important to arrange techniques to represent such information and transform it to knowledge representation understandable for the machine and human at the same time. The discovery and use of these techniques presents a big challenge to system engineers. They have to know that the knowledge must be described in a way that is searchable, with quick access to minimize the time of the access, making this task almost transparent to users. If there were a way to discover the knowledge desired, objectively and efficient, and discover knowledge that it would not be expected, more time would be available to other tasks as or more important to everyday life.

Today is almost impossible to live without information and their several kinds of representation, if one can consider a simple daily task or a simple supermarket visit, or even more complex situations like discover a new treatment for a disease or constructing a building, the information is always present. In all situations, time is also very important, therefore it becomes fundamental for system engineers to create means to reduce the access time to such information through models that respond to these needs like text and data mining models. But to achieve an objective result of a search, the information must have some kind of organization. It is not enough to have it as it is received as raw and unstructured material. It will be necessary to process it in any way. Like separate it by domains or measure its similarity to a central subject, to discover its field but the important is to make something to help achieve the right results faster.

## Motivation

Information is everywhere. Nowadays, every domain has a database or repository with information. As the IT systems grow, and the time passes the information also grows, and its complexity sometimes reach sizes that humans do not imagine, neither can deal alone with them. Although knowing the human brain is a “machine” that can store lots of knowledge inside, there is no one that has all the information in the world.

With the appearance of Internet and computers, arise the opportunity to store information and knowledge and share it with others, making the human more aware of the world around. One can be, for instance in Australia, and get information from Portugal without travelling to the country itself.

Storing the information makes new challenges for the engineers. With the help of the improving of technology, and the massification of knowledge, the issue of storing information get to a point where was necessary to organize it. Additionally to the store challenge, others arise. Sharing the knowledge and creating techniques to treat the information and transform it are two examples of situations that the new era of technologies brought us.

In the competitive engineering and in business world, a good organized system could be a key to reach success. The need of getting objective results from a search may be the difference in making a contract. Similarly, the speed of accessing a knowledge site to get a solution for a problem, for instance, in a construction site could let the engineer with more time to deal with other situations. Each day, engineers work with lots of information in their systems. The importance to have good systems and to reach this information needed quickly grows.

The issue of information retrieval in a society where the organization, and indexing of information itself is very useful, and even sometimes it is critical, it becomes important to develop systems and processes that eases the complication and challenges that information has.

Organizing the information in databases is one of the steps for these challenges. Organizing in a way those information systems can easily retrieve, trying to discover relevant information, related to the search pretended.

Before one can know how to discover new knowledge it is necessary to know how information systems recognize knowledge when this is still unstructured and raw text. There are some knowledge representation techniques, one called *ontology*. Building an ontology arises from the need of detection, extraction and find relations of the concepts from the different fields, through some classification method. Usually, these ontologies are created and maintained in anyway by human interaction. As a result the ontology is a very static environment. Recognize a concept, know if is already in the ontology, define a relation or assign a classification are tasks from an ontology maintenance responsible human. This is a very exhaustive work, and tends to escalate with the size of the knowledge database. Consequently, some questions and doubts come to surface. For instance, how to update an ontology? How to add more knowledge? How to update the existing knowledge? How to know the relation that the concepts share? Additionally, How to know if a concept relation is stronger to concept A than it is to concept B? How can the system understand the meaning of the concept itself? These are just a few that an ontology engineer has to deal when working with ontologies.

One of the motivations of this study and the proof of concept presented is to create some dynamism in the maintenance of an ontology. If there were a system that could answer and in an automatic way the previous questions, the work on the engineer would be even easier. Second it would be very interesting that some system could update the existing knowledge from the ontology. In the context of knowledge discovery, the main objective of the present work is to build a system, through the application of data mining techniques, to structure information extracted from unstructured documents, and transform it in knowledge that could be used. Also construct an automate process to relieve the human exhaustive interaction in the maintenance of a knowledge source as an ontology.

## Vision

In e-COGNOS ontology and opportunity was found that presented a great challenge. As ontologies are somewhat static, or the dynamic existing in them requires an expert work to exhaustively manage them, the opportunity to develop automate processes aimed to this tasks appeared in front. Greater is the challenge to update the Ontology data with new knowledge, new concepts and relations.

The goal of the present document is mainly to receive a set of information, unstructured, by means of digital documents and discover knowledge that it is related to the respective domain and presenting solutions and proposals for the updated of the ontology itself.

## Development context

The civil industry is no exception when the subject of knowledge information appears. Like any other area, the quantity of information is growing in large scale. Every company holds a database, sometimes still on hardcopies that could present a big challenge if anyone would want to find any kind of information necessary. It is thus necessary arrange systems to store information, with some specific characteristics. The information has to be in an understandable way some easy one that can be understandable and reachable by a user and a machine. Text is a good form of knowledge representation that presents search capabilities and easy understanding by people. When a civil engineer starts a project, normally works in a collaborative way with other roles, like constructors, employees, other engineers. Imagine the data available in a construction company, for instance, and it does not even need to be a major company to have several documents of project requirements, modelling plans, plants, human resource details, and this list could go on. It becomes a necessity to have a system that has all information gathered, and at same time can be scalable. This scalability also brings new challenges. How to get the information for a specific project when all projects are in the system? And if the necessity of searching documents arises? How to get the documents that are similar to the subject one search for? The access to this information could be very demanding or hard. It thus became important to get a system that could address all this. Knowledge management is a field of study that deals with these challenges and many more. Ontologies based systems are a good choice for this. A central database with an ontology system could present itself as a solution for this. Ontology is a system to represent knowledge that can support it in the form of a structured tree, composed by levelled concepts and relations. Concepts are arranged by classes and sub-classes as in a hierarchy, for instance, a concept can have parent, grandparent, child or sibling concepts. The relations are the information on how semantically closed are each concept. Using an ontology itself presents some challenges and opportunities. Another challenge is in the maintenance of the information itself. After some time, the knowledge in the ontology could get obsolete. Or the quantity of data to address could get very big.

The present document was inspired by the European collaboration project called e-Cognos ontology. This project, funded by EC was created to serve the BC sector in Europe as an information collaborative knowledge system. (Lima et al., 2004)

The present document follows the SEKS project (Figueiras, 2012) and its thesis presented. This was a collaborative semantic enhance knowledge sources project. And in this context, the present pretends to propose an ontology enhancement through a knowledge extraction technique called Association Rules.

## Dissertation Structure

The present work will be structured as follows: Chapter 2 will present an overview on the State of the Art in the techniques that drive Ontologies. The main objective is to present some insight of the application on knowledge techniques and ontology learning in the present world.

Chapter 3 describes theoretical and technical foundation behind the development of the present work. Discusses the techniques used in the development of this research and the framework developed.

Chapter 4 argues about the definitions and methodologies of an ontology and the techniques for its enrichment or management. Further discuss is presented in the context of framework inspired by the development of e-Cognos ontology project.

In Chapter 5 is proposed a system solution to deal with the problems associated with Ontology Enrichment and Management process.

Chapter 6 further discusses and evaluates the project developed, and the main challenges. Also presents some relevant publications that were written during the development of this thesis.

Chapter 7 will present some conclusions of the author, highlighting the main difficulties on the development and some future challenges and opportunities in the ontology field of study.

# State of the Art / Related Work

## Ontology Learning

One of the main objectives of the present thesis is to develop a system capable of turning the ontology management process into one that could be more dynamic. One of the techniques to achieve this is through Ontology Learning. Ontology Learning is the discovery process of knowledge from some original data, like documents, files, webpages, and consequently evaluate it against the knowledge of an ontology in the same domain. After this, the ontology is then updated with the conclusions of this learning process. Thus update could mean two situations like new concepts or update relationships. Although this process is not easy to automate, similarly is not easy for a human to do it. Ontology Learning is the study field that addresses these challenges. One research that argues with this situation comes from the field of Biomedicine, presented by (Batet et al., 2011). This research discusses the techniques to use in ontology learning that could create some automatism in the process of ontology maintenance. Other point of view in this subject is presented by (Niepert et al., 2007). This is a very interesting perspective discussed to create automate processes in building an ontology and defining and updating knowledge, namely concepts and relations, in the field of Humanities, on the well-known digital collaborative online project, Stanford Encyclopaedia of Philosophy(SEP[[1]](#footnote-1)).

## Association rules

In semantic systems, there are several ways to reach the goals. One presented in this work and also much studied in the scientific community is the Association Rules technique. The main goal of this technique in the present study is to help making conclusions about the relation of data/words in the texts and to help build a new ontology or improving an existing one without any knowledge of the taxonomy present in it. Table 2.1 shows some of the fields of study where association rules were found an interesting technique to discover unknown and interesting knowledge from its data.

Since the appearance of the Association Rules with the study of (Agrawal et al., 1993), this technique was studied and applied in several different areas. For instance, one noble field of the application of this technique is in Medicine, in particular the study from (Mahgoub, 2006) offers a proposal of the use of AR to retrieve knowledge from unstructured documents from a scientific database related to the well known global problem disease, the H5N1 avian influenza virus. The author investigates the problem with the advantage of AR extraction from a scientific database, namely MEDLINE[[2]](#footnote-2). The main purpose is to discover knowledge, through the extraction of concepts and their relations that could help take conclusion on the avian flu. Another interesting study proposed also in the Medicine area, is the one proposed by (Azevedo et al., 2005). The authors propose the use of AR extraction to detect hydrophobic clusters in molecular dynamics protein unfolding simulations, in other words, they argue that the use of this technique in several big sets of data created by the simulation, will help the discovery of important and relevant knowledge, namely, he discovered several hydrophobic residues, that could be important to explore different proteins. Furthermore in Medicine, a different approach were used by (Tan et al., 2009). This is the application of AR to discover knowledge in Medical Images, and thus trying to discover relations between the extracted and earlier marked regions of interest in the images. With this approach, the author demonstrates the usability of the technique in Medical area, helping the doctors diagnose better the information of serious situations that could arise in these images. Also gives a proof of concept of AR application not only in semantic sources, as it was applied using images as their source.

Figure 2.1 – Study fields examples

Another area of application of the association rules mining is in the Culture Heritage. (Tomi Kauppinen, 2009) examined CULTURESAMPO database to propose an ontology enrichment process through the extraction of relations in concepts by means of AR extraction, namely the Finnish General Upper Ontology YSO that stores the ca. 20000 concepts and is maintained by the National Library of Finland. This work assesses the possibility of discovering relations between the concepts, for instance “works of art” and “art of painting”. To present the results, the author also developed the ONKI SKOS browser application that is a rule visualizer, with similarities with the present work developed with this thesis. One of Kauppinen system’s weaknesses is that it does not improve the existent relations between the concepts. The reason proposed is that not to consider the same relations. By the presentation of the present thesis, it will be proposed an enrichment process that tries also to consider the existing relations between concepts, and thus, in those cases evaluates them with stronger relationships values. The fact that a relation is discovered by the AR process that is already present in the ontology, it will only reinforce the idea of such relation between its concepts.

Table 2.1 - Examples of Fields of Study with Association Rules Application

|  |
| --- |
| FIELD OF STUDY |
| Medicine/Biology |
| Education |
| Cultural Heritage |
| Languages |
| Artificial Intelligence |
| Business & Commerce |
| AEC / Building & Construction |

Recently (Kumar and Chadha, 2012) presented, to the scientific community a study of Association Rule Mining in the Education field. This study proposes the discovery of the elements that leads to a high student success rate by the evaluation of their academic results, consequently improving the quality of the education in this university. This is an interest work that proves the usefulness of the application of AR discovery in improving the referred factors and also the chances of the student’s success.

One more example of the advantage of AR application is presented in the study proposed by (Spruit, 2007), that examines the association between syntactic variables in the several Dutch Dialects. Knowing the existence of 267 Dutch dialects that distributes itself to the north of Belgium and France, the researcher discusses the geographical distribution of the variations of the words in the dialects. When using AR this researcher can find interesting knowledge about, for instance, where a specific word is more used.

Closely to a field of great research and application by commerce and industry is Business Intelligence processes, where the areas of Data Mining, namely Association Rules plays a very important rule to discover knowledge in the data analysis of the companies, with the objective to discover what are the clients preferences, and next buys, based on past transactions. (Brin et al., 1997) take advantage of this concept and used AR to discover interesting knowledge from a supermarket. This work is a use case of the AR, which is very discussed in the community due to the above explained usefulness to the companies, in which presents a practical situation of a market basket data analysis. It was considered, in this research, a dataset of 100 thousand transactions, which also considers the good possibility of the use of AR with large databases. Also in the Business Intelligence field, recently, a research were presented by (Hoque et al., 2011) where it is outlined the good use of knowledge discovery techniques, namely association rules to discover some patterns in the employees behaviour of a company. With the help of AR the company could find interesting knowledge that could help making decision on the employees benefits, for example. Knowing the job experience, age and education, a promotion or some better condition could arise. Other work presented by (Korczak and Skrzypczak, 2012) proposes the process of AR discovery. This process is very similar to the present thesis as it also uses RapidMiner for AR generation. Although the field of study is different, the theory behind it is similar. This article presents the creation of a software solution, DM Cafe, to help discover interesting and surprising knowledge from the clients of the Internet Delicatessen Alma24 store which belongs to the group of stores with the same name in Wroclav, Poland. With this study, the analysts want to better understand the choices of the clients and consequently improve the success of the business.

Association Rules also prove its use in the field of Architecture, Engineering and Construction (AEC) industries. (Hjelseth, 2009) presents an overview on knowledge discovery and how can association rules be implemented into a commercial rule-checking software. He also identifies several problems when dealing with the semantic richness of the languages. For instance, when translating words could produce ambiguous information or the manual verification of the rules could lead to uncertainty in the ontology.

One can easily conclude that Association Rules is a very useful technique in Artificial Intelligence, or specifically, in the Semantic Web. This technique extracts valuable information from semantic texts and data identifying the interest and unexpected knowledge.

## Building & Construction

Likewise, in the same AEC field, and maybe the most interesting field of study, relevant to the present thesis, where association rules were applied to discover some knowledge is specifically in Building & Construction field. (Zhang and El-Diraby, 2012) presented a very interesting study about the use of social networks to support a collaborative environment in this field. Taking advantage of semantic knowledge discovery techniques, they try to make a proposal by the classification of knowledge items and their extraction from the previous referred social network. But one of the strengths of the approach presented can be immediately identified as the existence of a method to evaluate the relations between the concepts. The author argued that these values were based on semantic similarity measure techniques like vector space model. The author developed a proof of concept with a collaborative knowledge system based in a construction information knowledge portal. This portal served as an interaction between an ontology and actors.

Other project in this domain that inspires the present writing and development of this dissertation is the e-COGNOS ontology. This project originates as a part of the e-COGNOS IST project, an European Funded project with the goal of helping Building and Construction sector creating a knowledge collaborative system (Lima et al., 2004). This project was one of the pioneers to build a knowledge management collaborative system to be applied in the professional field. Based on its future opportunities were the challenges of development of this document.

# Theoretical and Technical Foundation

## Association Rules

The process of discovery of rules in data has been a subject of many researches by the community of Artificial Intelligence. In the sub-topic of Knowledge Discovery and Machine Learning, the recognition of rules in associations between items plays an important role. For an AI system be able to recognize and take some conclusions about how the information is related, therefore, Association Rules (AR) is arguably considered as one of the most important tasks in Knowledge Discovery (Marinica and Guillet, 2010), and one of the most studied in the scientific community (Agrawal et al., 1993; Agrawal and Srikant, 1994, 1994; Hoque et al., 2011; Marinica and Guillet, 2010; Paiva et al., 2013; Vo and Le, 2009; Wang et al., 2002; Zaki, 2000; Zeng et al., 2010).

Association Rules is an algorithm that identifies tendencies and relations between frequent items in a database and tries to make predictions over behaviours. Was first introduced by (Agrawal et al., 1993). The problem that AR tries to address is related to the analysis of knowledge in transaction data from a collection of items. The goal is to help information analysts or automate systems making the best decisions. This is a process to improve the quality of those decisions. Additionally, association rules is a process that drives good and easy understanding to an analyst. To demonstrate the problem more clearly, the academic community refers to an example based on market basket analysis and its transactions in a large department store. In this example, the problem presented is to predict the behaviour of the clients, based on the collected data from each of the transactions. For instance, if a client buys some product A, the AR Algorithm, based on the stored transaction data of other clients, AR will give the best common behaviour that this client will have. With a set of products as the *premise*, the AR will predict a *conclusion* that the client will probably buy some product B.

|  |  |  |
| --- | --- | --- |
|  |  | (1) |
|  |  | (2) |

Two tasks are necessary to create Association Rules. First, an algorithm for frequent pattern mining in the database is applied to discover all frequent items that occur in the database, and the second task is the extraction of interesting rules among the frequent items. An AR is an implication rule in the form of equation (1). Two itemsets must be considered, one for the premise and other for the conclusion[[3]](#footnote-3). Itemset A represents the premise, B the conclusion, and the rule is defined by if A happens then B will likely happen also. The intersection of A with B is an empty set. On other words, the transaction A does not have anything in common with the transaction B. The itemsets that are considered in the premise can include one or more items. As for the conclusion it can hold only one.

The Association Rules can have a classification of different data types. Similarly to database attributes, they can be *Boolean* or *quantitative* (Hoque et al., 2011). Boolean association rules are the ones that hold boolean values like true or false, or 0 and 1. They are on the form of if A is true, then B is also true. Quantitative association rules are found when the items are in numerical form of some kind. One can think of when the items are intervals of numbers, like an age for instance. For the purpose of this research, because the items considered are concepts items, only Boolean association rules will be considered and designated by Association Rules.

The transformation of data into knowledge is a very challenging task. Discovering what will be interesting or not is also a good challenge for the association rules discovery process. To achieve an interesting association between two datasets there should be some kind of evaluation. In this field, some considerations and thoughts must be made when considering the evaluation of knowledge and specifically a rule. The first question that one should consider about the evaluation process is what should be evaluated and what should be considered interesting for the problem solution. One should not forget that association rules holds a wealth of information related to a data set, therefore some ways of evaluation were created to extract the best information that is more relevant. As a more broad definition, *evaluation* could be observed as the discovery of results obtained in some process, having in mind the achievement of some goals. The evaluation process is a very broad area, but this study will only be centred in the evaluation of association rules.

When the interest of evaluation has to be considered, one should start with the domain of evaluation. For instance, if the domain is construction and architecture, association rules discovered that include houses or buildings could be more interesting than computers or photography. In contrast, if the idea is to find relations to houses or photographs of buildings, then the interest on photography rises, transforming one uninteresting domain into an interesting one. Therefore, the domain is one important factor and should be carefully chosen to give the best results related to the interest considered. In (Yao et al., 2006) is argued that the user can also play a crucial role. Yao et al. highlights the presence of a judge or someone who benefits from it is also important for the evaluation process. This is true as one can verify also referred by the present study, which a system is built to help this user make better evaluations. However, this approach has some drawbacks. One of them is the subjectivity of a rule. When talking on human beings, different points of view are expressed by different people, the background education can be also different, or even the geographical location can be a factor of difference when evaluating the same rule. Therefore, each of the rules is also dependable on the specific person that would participate in the evaluation process. This could be or not a problem when evaluating the rules. In literature, some approaches have been presented to evaluate the subjectivity of a rule. The subject of subjectivity of the interest of a rule is further discussed in the following sub-chapter 3.1.2.

To overcome the drawback of subjectivity of a rule, some objective measures have been proposed to measure a rule. The methodology to use in the evaluation process depends, as observed above in several factors. Other factors also contribute for the best evaluation as the measurement technique. Much research has been done, since the presentation of the association rules in (Agrawal et al., 1993). Most of them highlight the importance of *support* and *confidence* of a rule as two metrics that can assist the discovery of interest in association rules. This study will reinforce the importance of these two, and demonstrate the existence of more metrics than the former that can be used to enrich the evaluation of a rule. The discussion of these measurement techniques will be illustrated in-depth in the next sub-sections of this chapter.

One interesting approach has been presented in (Hilderman and Hamilton, 2001), where the author outlines some different points of view on how to measure knowledge in general and describes some techniques to measure such interest. Some of the techniques were proposed to be used on association rules. The author discusses the measures application, in which some measures for objective and some for subjective knowledge are debated.

As illustrated above, in the previous lines, AR is a two step procedure. Before a process could discover rules of association between frequent patterns found on data, one big step has to be made. From the pre-processed corpus of data, one has to recognize frequent patterns in the concepts amongst it and transform the processed data into knowledge that could have some semantic significance and interest. This is the first step to achieve AR, and for this, there are several algorithms that propose a solution to this problem. ECLAT (Zaki, 2000), Apriori (Agrawal and Srikant, 1994) and FP-Growth (Han et al., 2004) are the most known and studied. Apriori and FP-Growth are the most used of all three.

The second step of AR discovery is the rules identification and interestingness evaluation. To achieve this one has to define first what interest is and what it finds relevant. This will be once more discussed in the following subsection 3.1.2.

To measure the interest of a rule there are several techniques that help finding the strength of a rule. Some of them will be presented in the following respective subsection 3.1.3.

This next sections are going to examine the foundations of the Association Rules with the description of the algorithm to find frequent patterns used in this study, the FP-Growth. A definition will be illustrated and an explanation of the utility of the frequent pattern procedure. Moreover, the algorithm will be explained and some discussion will be made related to it. It will be compared with the two main competitors presented earlier, and discussed what the best one or the fastest one is, and finally what is the one who develops better performance with small and big data structures.

Furthermore, some discussion around AR will be presented, namely discussion on how this technique works and what is the methodology used. Some other questions will be answered, like what is a rule or how can one define a rule. Subsequently the metrics of a rule will be debated. How can a rule be measured, what metrics are known, what metric is the best, what makes a strong rule and what are the metrics that are most used will be some of the questions argued.

All this questions will be answered by the author of the present work along with some discussion around some other studies in the field of Association Rules and FP-Growth.

### Frequent Pattern Growth

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

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

#### FP-Growth - A Definition

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

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

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

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

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

Table 3.1 – Transaction table for frequent items in database

|  |  |
| --- | --- |
| TID | Items |
| 1 | architect, designer |
| 2 | designer, engineer, analyst |
| 3 | architect, engineer, analyst, professor |
| 4 | architect, analyst, professor |
| 5 | architect, designer, engineer |
| 6 | architect, designer, engineer, analyst |
| 7 | architect |
| 8 | architect, designer, engineer |
| 9 | architect, designer, analyst |
| 10 | designer, engineer, professor |

#### FP-Growth Algorithm

The FP-Growth algorithm is made in two steps. The first is an elimination phase scheme and building of an FP-Tree, and the second step is a frequent itemset generation.

**Step 1 – Infrequent items elimination**

The elimination scheme is where the initial data is mined to separate the frequent from the non frequent items in the database. It uses a recursion process to make a kind of elimination scheme. It compresses the data set by determining the frequent items and deleting all that are not frequent. These infrequent elements are the ones that are found mainly alone in the database, this means, that such frequency is below the above threshold level. On the other side, the frequent items that are above this level are called frequent items, and are the ones that it will be kept, and are the base for the next step.

As already referred, the deleting process is based on a user-defined minimum offset, called support, in which below it, all items are eliminated, and above it the items are the ones to be stored. This value along others is explained in detail in sub section 3.1.2 below when the discussion will be around how to measure an item. The result of this process is a modified set of transactions with only frequent sets of single-items. In the end, the frequent items are ordered for better organization search purpose. Table 3.1 presents an example of a resulting set of transactions after this elimination and ordering process be made. Each line of the table represents a transaction from the database. The resulting items are the ones found in the database that matches the requirements of a threshold value above the user defined value. These items are the ones considered frequent in our initial data.

Figure 3.1 – An FP-Tree example for the items in the transaction Table 3.1.

Following the elimination scheme and frequent itemset filtering, a tree has to be built, the so called FP-tree. This tree is a rooted acyclic graph with vertices not labelled and a root node null valued. It is constructed with transaction scans like the ones on Table 3.1, one transaction at a time. The main idea is to map the graph such that a new path for each unique transaction will be drawn. Each node represents each frequent item found earlier. If the search discovers a common prefix on the item set, it will overlap and remove it, and if a suffix exists, creates a new node in the graph and connect to its previous item. If this common prefix is shared along two transactions, these are merged into the corresponding node. Each node holds a counter. This counter represents the frequency of the node in the respective path and always starts with the value of 1 in each node creation. When a transaction shares a node along the same path, the counter is then incremented, and it goes to the next transaction. This will ensure that each frequent item only needs one path for each item in the tree. The chances of which the common prefixes can be shared are higher if the frequent items have been sorted by its frequency, from top to bottom order.

Example: let’s suppose that there is a database that after applying the elimination and ordering task of FP-Growth presents the following set of items (3), and the previous transaction Table 3.1 with all 10 transactions of this database scan:

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

**Building the FP-Tree**

The algorithm to build the FP-tree will have to deal with each of the transactions, one at a time. First it should be created a node to represent a root with the value of *null*. For the first transaction TID1, the first item is *architect*. This item will create a new node in the tree with the value of *architect* and a counter associated to it with the initial value of 1. The second item in the TID1 is *designer*, as this is also a new one, a node should be created in the next level of *architect* node with counter equal to 1. Figure 3.2a) represents the situation after TID1 where one can see the FP-tree constructed until this moment.

For TID2 the items *designer, engineer and analyst* are the ones to consider. As the first one is *designer* and in the current tree there is no first level node with such designation, a new node should be created a connected to *null* node for the item *designer*. As this is a new node its counter is set to 1. Then the node for engineer must be created and connected to designer with the counter at 1.

The next item on the transaction is *analyst* that it is also a new one in the path, so a node must be created with its name and the counter equal to 1. The moment at the end of this transaction TID2 is represented in the Figure 3.2b). One can see that there is two individual paths for the transactions, both sharing a node with the same item: *designer*. In this case both should be linked to recognize this situation and further evaluation.

The dotted line represents this linkage. TID3 includes items *architect, engineer, analyst and professor*. One can easily see that the first item of this transaction already is connected to the *null* node. In this case, there is no need to create a new node that would be repeated, instead the respective counter should be incremented by 1, totalling now 2, that represents the two paths that starts with *architect*. For *engineer*, *analyst* and *professor*, the procedure should be similar for new nodes in the same path. As *engineer* and *analyst* already exist in the tree, they should be linked with its equals also in the same way *designer* was above. An illustration of this transaction is Figure 3.2c). One can see that three different paths exists if counting the last leaves of the branches or if totalling the sum of the nodes and counters connected to *null* node.

Figure 3.2 – The three initial trees at the end of the first three transactions

**a) TID = 1**

**b) TID = 2**

**c) TID = 3**

This process shall continue through all transactions until all transactions table lines be evaluated for the actual process. For this example the table has 10 transactions, which are all represented in Figure 3.1 above. From this figure all the paths can be observed from the transaction table. To be noted that node *architect* is the one that starts most of the paths of the itemsets, exactly 8, making it the most frequent item in the database. As we can also infer from the figure, is that *designer* is the second most frequent, but it alternates in its position in the paths, five in the first position and 5 in the second position. It can also be noted in the tree that all items nodes with the same designation in the different paths are linked to each others.

The main purpose of this step is a filtering and organizing step with the objective to facilitate the search of frequent items. With this kind of structure the speed of search will decrease significantly.

**Step 2 – Frequent Itemset Generation**

The next step in FP-Growth algorithm, as noted, is the evaluation and recognition of the most frequent items. It is an extraction process that is called **Frequent Itemset Generation**. The main idea of this technique is to extract frequent items from the earlier built FP-tree. From a Bottom up perspective, the technique will individually process each ending bottom node and separates in an exclusively separate tree which is called **prefix path sub-tree**.

Find an example of this trees in Figure 3.3 where each of the items in the database will have an exclusive prefix path sub-tree. It is the *divide and conquer* method, separating each sub-tree individually for faster performance of the main tree. The paths included in each sub-tree are the ones that have the respective item as a leave node (ending bottom node). Therefore, for a frequent item X and its ancestor Y and Z from an FP-tree, the resulting sub-tree will be used to extract itemsets ending in X, subsequently will extract the ones ending in YX, and after it the ones ZYX, continuing traversing through the path and being processed recursively until it analyses all paths and reaches the most top node, also called null node.

Figure 3.3 – Prefix sub-paths for all frequent items

**a) Prefix path sub-tree for *professor***

**b) Prefix path sub-tree for *designer***

**c) Prefix path sub-tree for *engineer***

**d) Prefix path sub-tree for *architect***

**e) Prefix path sub-tree for *analyst***

Using the previous example, Figure 3.3a) to e) represents the prefix sub-path trees for itemset S. For each of the individual items in S, one sub-path tree was divided for further processing. In a), the prefix path sub-tree for item *professor* presents 3 paths described in Table 3.2. Consequently, the *divide and conquer* approach makes the problem easier to evaluate.

As observed in the first step, for a node be considered frequent it has to hold a support threshold value (*minSup = 2* in previous example). That was a requirement to search the items in the database and eliminate the ones that did not have at least another equal item in it. With this minimum support in mind, one has to traverse from the bottom to find each frequent item. For this task, if one wants to know if X is a frequent item, it has to follow the dotted lines in the prefix sub-tree and sum the counters associated with item X, and thus calculating the support for X, denoted by (Vo and Le, 2009).

Table 3.2 – Paths table for frequent items

|  |  |  |
| --- | --- | --- |
| Item | # | Paths |
| Architect | 1 | architect |
| Professor | 1 | architect, engineer, analyst, professor |
| 2 | architect, analyst, professor |
| 3 | designer, engineer, professor |
| Designer | 1 | architect, designer |
| 2 | designer |
| Engineer | 1 | architect, designer, engineer |
| 2 | architect, engineer |
| 3 | designer, engineer |
| Analyst | 1 | designer, engineer, analyst |
| 2 | architect, engineer, analyst |
| 3 | architect, analyst |
| 4 | architect, designer, engineer, analyst |
| 5 | architect, designer, analyst |

Equation (**4**) represents the mathematical expression to consider an item X a frequent item in the database. If this situation is true, X can be extracted as a frequent item and it can be found frequent items ending in X. For that a table as Table 3.2 has to be considered holding all paths for each of the extracted frequent items.

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

In previous the example, considering the item *professor*, one can easily sees the 3 paths for it. They are *{architect:8, engineer:1, analyst:1}*. *{architect:8, analyst:1}* and *{designer:2, engineer:2}*, these all lead to an ending node *professor*. In case of the third path, the set *{designer, engineer}* appears twice in database, however with *professor*, they only appear once. Similarly, architect in the other two paths is shared among them with a support of , appearing only twice together with *professor*. Therefore, to evaluate the set that appears together with *professor*, the nodes from the corresponding prefix path sub tree have to be updated as follows:

*S1={architect:1, engineer:1, analyst:1}*, *S2={architect:1, analyst:1}* and *S3={designer:1, engineer:1}*. Furthermore, as *designer* has, it's not considered FI with *professor*. Thus *S3={engineer:1}*.

Figure 3.4 – Conditional FP-Tree for item *professor*

These three prefix paths of *professor* *{S1, S2, S3}* constitute the sub pattern-base called, according to (Han et al., 2004), conditional pattern base, consequently, an FP-Tree based in this is called **conditional FP-tree**. A visual representation of the conditional FP-Tree for item *professor* is illustrated in Figure 3.4.

Table 3.3 – Frequent itemsets discovered for all items

|  |  |
| --- | --- |
| Item | Frequent Itemsets discovered |
| professor | {professor}, {analyst, professor}, {architect, analyst, professor}, {engineer, professor}, {architect, professor} |
| analyst | {analyst}, {engineer, analyst}, {designer, engineer, analyst}, {architect, engineer, analyst}, {designer, analyst}, {architect, designer, analyst}, {architect, analyst}, |
| engineer | {engineer}, {designer, engineer}, {architect, designer, engineer}, {architect, engineer} |
| designer | {designer}, {architect, designer} |
| architect | {architect} |

Table 3.3 includes all frequent items found in this example, considering all items in database.

#### FP-Growth versus APRIORI and ECLAT

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

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

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

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

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

### Association Rules Measurement

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

#### Subjectivity and objectivity

To be able to measure interest in the knowledge discovered, two types of measurement of the interest of a rule are identified: Subjective and objective measures. (Mackie, 1977) presents a study where he describes that the subjectivity in evaluation is very common when the evaluation goals are objects, actions or events. The objectivity is used in the measures themselves and their implementations. The subjectivity depends in great factor on the person that is considering the subject. As explained before, it depends in factors like the location or the background. Some other studies also discuss the subjectivity and objectivity of a measure. (Silberschatz and Tuzhilin, 1995) propose a classification for interestingness measurement of a rule. As one could see in Figure 3.5 where this classification is illustrated, it is argued by the author that the interest should be divided in *Objectivity* and *Subjectivity* measures.

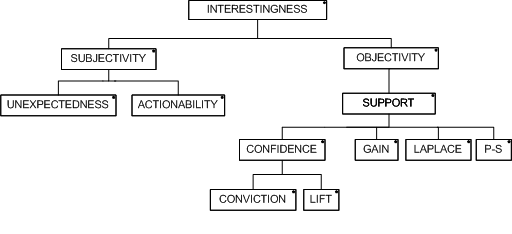


Figure 3.5 – Interestingness measures types tree (adapted from Silberschatz and Tuzhilin, (1995))

Some of the challenges to evaluate the subjectivity of knowledge interestingness, and in this case, a rule, were already discussed in the beginning of this chapter. But it is important to go deeper in this subject. Silberschatz and Tuzhilin (1995) points out in their study that it is important to measure the subjectivity of a measure. In the subjectivity side of the classification tree in Figure 3.5, the author divides this subjectivity of a rule in two concepts, *Unexpectedness* and *Actionability*. The first concept represents the value of some unexpectedness or surprise in a rule when knowledge is discovered. If one could discover a rule that it would not expect, that rule would be interesting. Of course, some knowledge expected, is knowledge that is already known, and thus, not interesting for the user in this sense. The second concept of subjectivity, actionability, represents the usability that a rule could have. In other words, it is the capacity of a rule to be used in an interesting way by its user. One example of these concepts applied to a rule that is presented by Gonçalves (2005) and could help explain these kind of subjective measures in a rule is the association between dippers and beer in a big department store. This example explains that when the transactions are made by young couples on a Thursday, this association is detected. The company analysts would think that the act of buying beer would just be associated with the act of buying appetizers or barbecue meat and other alcohol drinks. Surprisingly, when association rules are discovered, this unexpected knowledge rises to the edge. This is the perfect example of an unexpected and actionable rule, and as a result, for now on, on Thursdays, the department store can use this extraordinary new and unexpected discovered knowledge to move the dippers and the beer closer to each other, so the sales of both could go higher.

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

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

#### The measures

On the next lines, based on the interestingness tree presented in Figure 3.5, the measures will be discussed and presented. All of them will be identified and discussed, namely support, confidence or conviction and lift, or even gain, Laplace and ps. The existence of some other measures will also be presented and discussed.

**Support and confidence**

The majority of the works studied about objective measures of association rules, take advantage of *support* and *confidence*. Hoque et al. (2011) which presents a document on association rules consider these two measures, presented also in Figure 3.5, good examples to help find interest in association rules. Furthermore, Azevedo et al. (2005), Bhujade and Janwe (2011), Brin et al. (1997), Kumar and Chadha (2012) and Spruit (2007) are also examples of researches where support and confidence are used to measure rules of association. They all use at least these two measures to extract knowledge and to evaluate the results of their knowledge discovery processes. Additionally, Bayardo and Agrawal (1999) also considers the use of these two measures as a way to reduce the rules to the most interesting. The former and Azevedo and Jorge (2007), Gonçalves (2005) and Tan et al. (2002) also present some definitions of each one of the measures. The rule *support* (also *frequency* or *coverage*) is equal to the number of occurrences in the corpus of data information where both concept A and concept B evaluates to true. This is presented as *sup(A⇒B)* (also defined in some literature as *σ(A⇒B)*). In other words, the support of a set of items, that is a statistic metric, is defined as a transaction percentage from a database where these items are included. As this is a statistic measure, the values are represented between. The support result is proportional to the frequency. Higher the value, the more frequent are the concepts in the database. The definition for support is represented in the following equation (5).

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

Furthermore, *confidence* represents an estimation of the probability of observation Concept B given Concept A. When a rule is received, one can immediately classify the relationship of the corresponding concepts. The expression to calculate confidence is given in Equation (6) and the result values, as this is also a statistic measure, are enclosed in . One can also identify that the interest rises also with confidence results.

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

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

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

**Conviction and Lift**

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

*Conviction* is an implication measure that quantifies the value of the implication, it is represented as *A⇒B*, meaning that the direction of the rule is important for the interest measurement, hence *A⇒B ≠ B⇒A*. Conviction measure has some very interesting properties, such as if its value is equal to 1, this means that the concepts are considered totally independent from each other. Other property is that this measure considers the value of the antecedent as also the value for the consequent to calculate its value. Other interesting property of conviction is on rules with 100% confidence value, meaning where the antecedent always appears with the consequent, these rules will have the value equal to ∞. To achieve the most interesting rules one can think of as higher the value of Conviction, higher is the interest of that rule. The values of Conviction are included in . Conviction can be defined mathematically as presented in equation (7) or can be also presented as equation (8) dependant of confidence measure.

|  |  |  |
| --- | --- | --- |
|  |  | (7) |
|  |  | (8) |

In contrast to conviction that measures implication, *Lift*[[4]](#footnote-4), (can also be found in literature as *Interest* in (Brin et al., 1997) or as strength in (Dhar and Tuzhulin, 1993)) is a measure that quantifies the co-occurrence of a rule. Lift is not an implication measure, it means it is symmetric in relation to the antecedent and consequent, hence *Lift(A⇒B) = Lift(B⇒A)*, in other words it measures how far from independence are concepts A and B. Lift is defined as a measure to boost (“lift”) the confidence of a rule, this suggests an improvement of the trust of results of rule confidence. Similarly to conviction, if its value is 1 it means they are total independent without any kind of interesting relation, and as far from 1 and as higher the value is, higher will be the interest of the rule and more relation can be found on them. The set of values of this measure are included in . Lift is defined by the following Equation (9) or also defined in Equation (10) where one can see the dependency from confidence measure.

|  |  |  |
| --- | --- | --- |
|  |  | (9) |
|  |  | (10) |

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

**Gain, Laplace and PS**

In Figure 3.5 some more measures are illustrated. As one can see, they are presented as *Laplace*, *PS* and *Gain*. These three are also measures dependable of support.

Laplace is a classifier that is one of the additional measures considered in this research. It can be considered as a confidence estimator that is function of support, and as low as support is, lower is the interest in the rule considered. Laplace is normally used to rank rules by class. The range of values are in . Its mathematical definition is the following Equation (11). The constant *k* represents the number of classes defined when one is defining the respective classification model. Its value is always higher than 1.

|  |  |  |
| --- | --- | --- |
|  |  | (11) |

Another one of these measures is Gain. This is an optimization measure presented by Fukuda et al. (1996) and discussed by Bayardo and Agrawal (1999), and by Brin et al. (2003) as a proposal to solve the optimized gain rules problem. It is defined also as a function of support and given by the following Equation (12). The parameter is defined as a constant fraction with values between 0 and 1. Additionally, if one wants to decrease the subtractive term, it can be only done by decreasing the support of the antecedent. When this happens, the confidence value becomes higher.

|  |  |  |
| --- | --- | --- |
|  |  | (12) |

The last of these three measures presented is *PS*. This measure receives its name from their creators, Piatetsky-Shapiro (1991)[[5]](#footnote-5). It was originally used to classify rules, and later adopted by association rules. This measure is a boost to the support measure. As it gets a value in the range . If its value is equal to 0 it means that A and B are independent. A value below 0 represents a negative dependency and if the value is higher than 0 it is called positive dependent. Higher values represent more interest in the association rules. The definition for PS is presented in the following equation (13).

|  |  |  |
| --- | --- | --- |
|  |  | (13) |

**Other measures**

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

## Vector Space Model

In Text Mining, a Vector Space Model (VSM) is an Information Retrieval statistical model that tries to make the assumption that each document is represented by a point in space in a group of documents. As more near the points are, bigger is the semantic similarity and as more apart the points are, less representative in the semantics they are(Turney et al., 2010).

The VSM is a form to explain to computer systems the semantics of human language. It was created for the SMART information retrieval system by its developer Gerard Salton and his team. (Salton, 1971)

VSM has several properties, one of which is that given a corpus it will extract knowledge automatically. The majority of today search engines use VSM as a model because of its good performance in preparing the raw data to measure the similarity between documents, phrases and words (Manning et al., 2008). Queries made by the engines have a good performance presenting relevant results to the query author. Some of the most known algorithms for semantic relatedness(Pantel and Lin, 2002; Rapp, 2003; Turney et al., 2003) and semantic relation similarity (Lin and Pantel, 2001; Nakov and Hearst, 2008; Turney, 2006) also use VSMs as a base technology for preparation of the data.

There are some hypotheses that VSM tries to answer, they all begin from the main one, the *statistical semantic hypothesis*, that states that if statistical patterns are used on human word syntactic formation and usage of natural language terms, the possibility to understand the meaning of human speech is real.(Turney et al., 2010) The above hypothesis is the converging point of the following ones: bag of words, distributional, extended distributional and latent relation. In the following lines, the author of the present work will give a brief explanation of each of them.

**Bag of words hypothesis** is defined as follows.By representing the documents on the corpus and the query as a bag (or collection) of words, one can estimate the relevance of these documents to a query. The former can be explained as the word frequency that exists in the documents tends to represent the document relevance faced to a query. Each bag of words can be represented by a Term-Document Matrix. (Salton et al., 1975) wrote that this hypothesis is the foundation of VSM application in information retrieval systems; The authors of this hypothesis believe that each column vector of this matrix represents in some way a subject or meaning of the document.

When the subject is similar contexts, the **Distributional Hypothesis** is defined as the words in those contexts are also similar in their meanings.(Harris, 1954) The data is organized in a Word-Concept Matrix. When one wants to measure the word similarity, this hypothesis is the reason for the application of the VSM.

**Extended distributional hypothesis** was proposed by Lin and Pantel (2001), and is described as the co-occurrence of patterns in similar pairs will lead to similar meanings. The co-occurrence of X:Y similar pairs is a tendency of patterns like “X solves Y” or “Y is solved by X”. When this happens one can be lead to think that these kind of patterns present similar meanings. The representation of this pairs results in a pair-pattern matrix.

The last hypothesis, **Latent relation hypothesis,** is the inverse of the extended distributional hypothesis described above. It covers the pairs of words, when these co-occur in similar patterns. In this case one can say they have similar meaning.(Turney, 2006)

### Term Weighting – The TF-IDF

When dealing with large raw data, these can be represented by vectors in a matrix, the Term-Document Matrix. This matrix is prepared in such way that the terms are arranged in row vectors and the documents are arranged in the column vectors. Each entry in the matrix corresponds to a weight of each term in a document. This process orders the terms in a document by their relevance in each document and in corpus by a tf-idf (term frequency – inverse document frequency) weighting normalized scheme. This scheme is presented by the following equation (14):

|  |  |  |
| --- | --- | --- |
|  |  | (14) |

Where represents the term frequency of term *i* in document *j* and represents the number of documents that contains term *i*. The result is the following matrix (15) with the weight or relevance of each term. Through this weighting process, the system may know the relevance of each term in the context and which one is more or less representative.

|  |  |  |
| --- | --- | --- |
|  |  | (15) |

This form of representation is called a bag or multiset, and supports the bag of words hypothesis discussed earlier. This way, one can discover the tendency of the proximity of a document to a subject, by this frequency of words in the document. Salton et al. (1975) states that this hypothesis is the foundation of VSM application in Information Retrieval.

The VSM is not the only way to represent text, as described above the several hypothesis represent more ways to represent and weight relevance of the terms. But this is not exclusively, for instance, another proposal is made for this scenario as another form of weighting the terms and documents in a corpora. This scenario, which is called TWEAK, uses labels to learn the terms weight related to its importance in the subject as a parametric function, where the model parameters are learned from the labelled data. (Yih, 2009). Instead of being an independent weighting scheme like tf-idf, that it does not take consideration the previous analysis or other kind of past similarity calculus, this TWEAK is dependant of the previous analysis as this considers the model parameters in the evaluation. Meaning that the previous labelling and classifying of the terms in the corpus are included in the next weighting, making this process influenced for the actual subject of the text data.

## Similarity Measure in Information Retrieval

Before one can understand what to measure, it is important to understand what similarity is, and what is it role in Information Retrieval (IR). Similarity is the state or the fact of being similar.(Oxford University, 2006). To understand what Similarity means, it is important to understand that each word or concept has a(many) meaning(s)/subject(s) that can be related to. How similar is each word to a subject? Lin even presents a Similarity Theorem to explain it:

“*The similarity between A and B is measured by the ratio between the amount of information needed to state the commonality of A and B and the information needed to fully describe what A and B are."* (Lin, 1998) In other words, a similarity of two words is a quantification of their differences and similarities. It is a measure of how much their meanings are close. How much information do they share, and how much information they do not. Even in words with similar meaning, similarity measures are important to find what the word that best fits a particular context is. Or just to know what the word that is more similar to other is.

In the previous chapters it was presented an approach on how the raw data can be organized for further evaluation. For it there are some ways to do it, several take vector models as starting point. The VSM is one of the best known and applied methods. As VSM states, to better understand the meaning of a word or concept, there must be a measure in the semantic relations of each word, given a set of documents. One must know how similar each word is, therefore it can try to figure out what is the best approximation to the main subject of each document, and thus give the possibility for the machine understand its meaning. For example, given the words *Architect*, *Engineer* and *house*, how can they be related to each other? The reader will obviously know that an *Architect* is more related to a *house* than is an *Engineer*. For the reader is easy to know the meaning but for an AI system, is it? How can it understand the relation to each word? *Similarity Measure*, is a way to measure the semantic relation of each of the words or concepts, and see if they are closer in meaning to ones, than others. Several paths are possible to consider and achieve the previous.

Similarity measure can be used for more than similarity of words and concepts. One of the most known applications of similarity is between a user query and some document pages like a web search engine (“Google.com”, 2013). But this is not an exclusively use of similarities, also for instance, one can measure a document similarity between scientific papers, or measure similarity of unstructured data texts so a context or domain could be found. It can be used to aid ontology construction, fortify relations between concepts, and make ontology dynamic and capable of learning.

To start a similarity measurement some initial thoughts must be made. What is it going to be measured? The corpora documents? Words? Vectors? The type of the initial corpora is important. Other thoughts must be made on if the domain is known, is the data already structured? Is there any ontology to support the process or any previous similarity measurement to help? Are their semantic similarities important? What is the similarity measure process to use? There are several similarity measures in the scientific community that are being discussed. In the following lines some of them will be presented and described, namely Euclidean Distance, Cosine Similarity and Jaccard Coefficient.

### Cosine Similarity Measure Algorithm (Develop further)

The Cosine Similarity Measure Algorithm is one measure used to quantify the similarity between two concepts, defined as the inner product between two vectors, x and y. The mathematical formula can be presented as per the following Equation 16.

|  |  |  |
| --- | --- | --- |
|  |  | (16) |

## Building & Construction Domain Ontology

## Ontology

The term “Ontology” origins from early 18th century, from the modern Latin *ontologia*, from the Greek word *onto*, which means “being”, plus *logia* which means study, science, theory. (Oxford University, 2006). It is, thus, a sub domain of philosophy that studies the nature of a being or existence itself and the way these relates with each other’s.

In information systems, the ontology is the study of the representation of knowledge. It is a form of definition and organization of the knowledge domain. Ontology can be defined as a structure or a form of data organization. For a better understanding of this knowledge, an ontology is represented by concepts and relations, which are organized into *classes*, in a tree structure shape called *taxonomy*. The taxonomy represents the hierarchical characteristics of classes. Also in an ontology are *individuals*. Individuals are the instances of the classes. These classes present the *relations* between its concepts, meaning it represents the semantic distance between each of the concepts in that ontology. Attributes are the last element in an ontology, these are properties to describe each of the objects, or specifically, individuals

One can say though, that ontology is a structured representation of a data set of knowledge that aims for a better understanding of the data presented.

### Construction Methodology

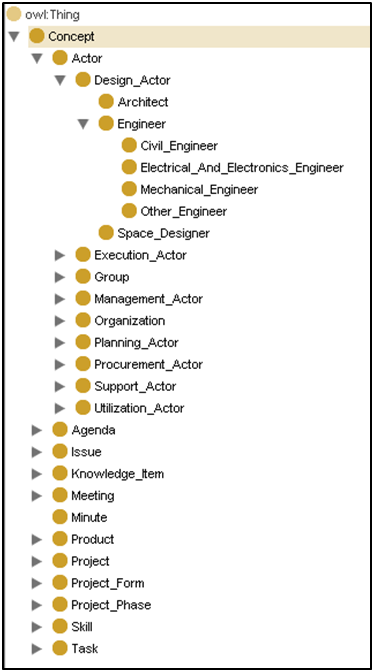
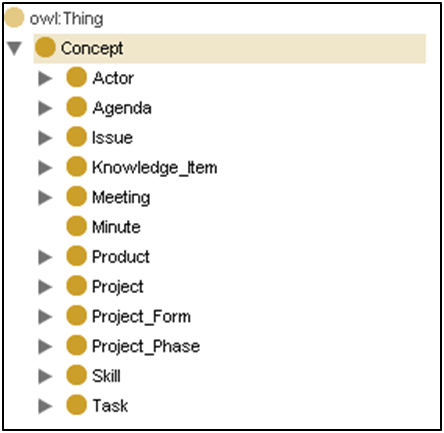
Being an ontology a formal representation of knowledge, its construction should follow some requirements. Ontologies are aimed to collaborative domains, therefore they have to be interoperable as they have to fit different systems without its adaptation into each system. Ontologies thus have to be transparent and represent the same language between the whole participant community to integrate in the system coherent and consistently.

Other concern when an ontology is constructed is to make it understandable and with capacity to interact by human minds and machine intelligent systems. As a result its construction language, OWL, is on top of a very verbose technology as it is RDF-XML. The adoption of XML to build the ontology was a good choice, as it accomplishes this requirement. Every ontology expert understands easily an XML document, so OWL inherits this characteristic from it.

Additionally, ontologies are constructed to suit a specific domain, hence its domain have to be chosen wisely and adopted by all community, and its taxonomy structure should respect this requirement. Ontologies should also allow the possibility of integration in other ontologies. For instance, if one is building an ontology in doors construction domain, that is intended to add in a civil construction domain, the ontologies should be built to make this merge in an easy way. It can be clearly observed in Figure 4.1 the domain represented by the ontology.

Ontologies are made to store knowledge, but its importance and use is very reduced if they do not allow to be searched. Searchability is therefore a characteristic that presents itself as an important one. A user should be able to search through the ontology, or insert queries that the ontology should answer accordingly and in with high reliability. For instance, if the user wants something related to civil construction, like a building process, or some kind of material, if the ontology answers to its query with holydays destinations, something is not quite right.

Figure 4.1 - Ontology example in civil construction domain



a) Concept Sub-Classes

b) Path into Engineer Sub-Classes

Furthermore, an ontology should hold the capacity to be manageable. The maintenance of the contents of an ontology is important to keep it up to date. Adding, updating, removing concepts and relations capabilities should be present in the moment of its construction.

In the next sub-chapters along the following lines, some of the previous characteristics will be explained in more detail.

#### Knowledge representation technologies

In IT systems, W3C created a standard to define a representation and organization of a domain knowledge ontology that can be “easily” read by a human. This standard is named Web Ontology Language (OWL). (W3C, 2004)

Figure 4.2 - Knowledge representation technologies - layered approach

**OWL**

**Interconnections**

**Domain knowledge**

**RDF**

**Objects and relations between them**

**XML**

**Base structure**

One of the main challenges that a set of raw text data presents is the difficult to translate it to a language that computer might better understand and likewise humans also understand. This challenge is somewhat solved with the creation of XML. This technology allows for a structured presentation of information making use of a verbose tag system. This is considered a base structure to many XML based languages that originated from it. RDF is one of these that appeared on top of XML documents. This RDF, meaning Resource Description Framework, allowed the creation of some new structure that could present the relation between objects presented in an XML document. OWL technology appeared on top of the RDF to describe ontologies. They can describe all characteristics of the ontologies, including its domain. Figure 4.2 presents a layered approach of the evolution of the knowledge representation technologies on top XML documents.

### Ontologic management challenge

Maintaining an Ontology could be a very expensive task. The quantity of information inside it could be huge, and thus the necessity of arranging automatic processes arise, and thus the opportunity to embrace new challenges in the Ontology field. These challenges begin with the maintenance of the existing knowledge. As the reader might know, knowledge can become obsolete over time, and thus it is necessary to search through the ontology regularly to see if the ontology is up to date. With the information systems growing day after day, and the appearance of new technologies, processes like removing obsolete concepts or relations, update them, or adding new ones will become natural and compulsory in an ontology. Nowadays these operations are made by human interaction. It is the person dealing with the ontology who has to make decisions. After the building of an ontology, mainly the ontology engineer has to execute this process. Some research has been made in processes that could lead to some automatism in the previous tasks and thus create the called Ontology Dynamic. One very interesting work is presented by (Bedini and Nguyen, 2007) in which are debated several forms of automatic ontology generation.

Other way of making the ontology up to date is to find new knowledge, this maintenance presents some challenges. In this matter, (Lima et al., 2003a) debates and identifies two relevant challenges. The first lies upon the ontology adaptation to the system where it is going to be used. This ontology should be able to receive new knowledge, thus be enriched with new concepts and relations. The second is in a situation for when an ontology already exists, in this case how to assure the consistency of the knowledge already presented in it.

## The E-COGNOS Ontology

E-COGNOS ontology is defined 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., 2003b)Its main goal is to represent construction knowledge in a consistent way.

Figure 4.3 – Domains represented in e-COGNOS ontology (adapted from (Lima et al., 2003b))

Root

Actor

Project

Resources

Product

Process

Technical Topics

Related Domains

Each of the domain represented in e-COGNOS are the ones of the previous definition. (See Figure 4.3) *Actor* domain is presented in two sub-domains, namely Individuals and Organizations. For *Project* domain, the definition could be as the procedures made to execute an engineering set of tasks to achieve a bigger solution. The *Resources* domain represent the classes of the materials, employees and equipment considered. *Product* is the domain to represent the materials used in the project. Furthermore, the Process represents the path taken to solve the problems. They can be divided by two sub-domains, Administrative and Engineering processes. The *Technical Topics* are the fields represented in the project. These major domains are inspired by the IFC model.

As being targeted to the Building and Construction domain, this ontology was created with specific characteristics. Some of them already defined during the initial discussion of Ontologies in the previous sub-chapter, namely flexibility and with broad sense to be prepared to be used in different business scenarios. Other characteristic is that it should be easy for a user take advantage of it. More specifically this ontology should be compatible and if necessary include the most recognized and established standards in the field, namely BS6100, UniClass, bcXML and Talo 90.

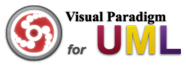
# Design and Implementation

This chapter will illustrate all the information related to the implementation for the solution proposal of the problem presented in this research. Beginning from the modelling of the problem, design, requirement analysis, and building until the development of a final solution. This proposal includes the development of a software system called Dynamic Ontology Knowledge Enrichment System (DOKES). Based on the model proposal of the problem, all the initial requirements were fulfilled in technical solution.

This solution will be illustrated in this chapter starting in the following lines with a substantiation of the tools and technologies adopted for DOKES. Consequently, the initial requirements of the project will be discussed in more detail, following with the architecture conceptualization proposal based on these previous requirements. The chapter will continue presenting the technical implementation of DOKES, and will finish with the presentation of the visualization tool, the Knowledge Enrichment Decision Helper System (KEDHS) created to present the results to users.

## Tools and Technologies

Several technologies and tools could be chosen to developed this project, however, the author chose to use the following based on the initial requirements of the research. These tools and technologies were available free and were competent enough to develop this client-server product (Figure 5.1). To model the problem, it was used UML with Visual Paradigm to design and model all respective diagrams. This software is a good design tool to develop projects in UML as it can make all diagrams necessary, namely the diagrams for Classes, Use Cases and Sequence among others. It is presented with a student version, and other professional ones. The student was used in this work.



**XPATH/XSLT**



Figure 5.1 - Tools and Technologies adopted

To model the database necessary to hold all data from the project the technology chosen was SQL with the use of the MySQL Workbench tool. This is a good tool to use with database modelling. This tool allows the creation of the Entity Relation Diagram that afterwards was translated into SQL with all the code necessary for the database building. The database runs on top of Apache server through XAMPP software package. This package also gives the possibility to maintain the databases through PHPMyAdmin web page.

Rapidminer was the tool used to make all the previous steps necessary to discover association rules. This is a software tool that is indicated to use in machine learning, data mining, text mining, predictive analytics and business analytics processes. All the pre-processing of the unstructured data was developed in this tool. With this powerful tool, it was also used a script in Groovy[[6]](#footnote-6), an agile and dynamic language for the JVM. to prepare the data for export. This script language integrates well with all existing Java classes and libraries in the Rapidminer because it compiles straight to Java byte code so you can use it anywhere you can use Java. The last component used of this tool is an included API that was used to execute the process modelled, (process represented in XML), underneath in the server, this was used to make the interface between rapidminer and the java code.

The core of this project was developed in Java in the SE Platform. The IDE used to make all the coding to hold this product is the NetBeans IDE. This IDE is an open tool that is very complete. It allows the coding of Java, compiling of the code directly to JVM and also includes a debbuger 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 are in JavaBeans interfaces or through XML data files that is serialized with JDOM on the side of the server and in the client with XPATH/XSLT for presentation purposes. The communication with the database that holds the main model is made through JDBC connections (Figure 5.4).

Protegè was the tool to deal with the ontology as a user point of view. 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 Front-End application the technologies used were HTML, CSS to create all components that support the final product presentation. The interface is 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.

## Requirements

As a proof of concept a system was developed based in requirements called Domain Ontology Knowledge System (DOKES), which were divided in three types, Functional, Architectural and Technical (see Figure 5.2). The system developed was integrated in the Business and Construction field of study and the main idea is to receive unstructured information and extract knowledge from it that could help to make automatic enrichment of an ontology. Having this in mind, some of the functional requirements are *frequent items-concept matching*, that searches the ontology and tries to find keywords that are similar to the frequent items, consequently, if found, it retrieves the concept represented by this frequent item. For *new concepts discovery*, this process is based in the previous and is executed when no concept related to the frequent item is found in the ontology. In this case, the new concept found will be added to the Building & Construction domain ontology with the frequent item associated to it. The system must also be able to *show the association rules* with the important information in an easy and understandable way. Another functional requirement is *concept relations enrichment*, which is 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 3.1 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.

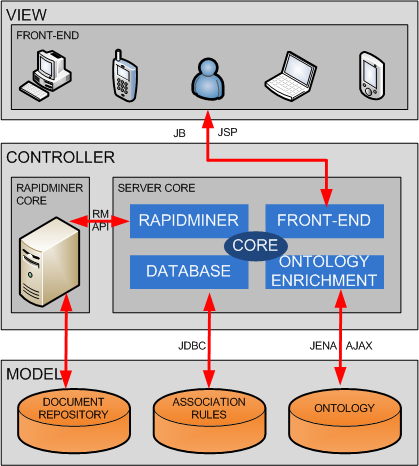


Figure 5.3 – System Architecture – MVC Methodology

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 5.4, 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.

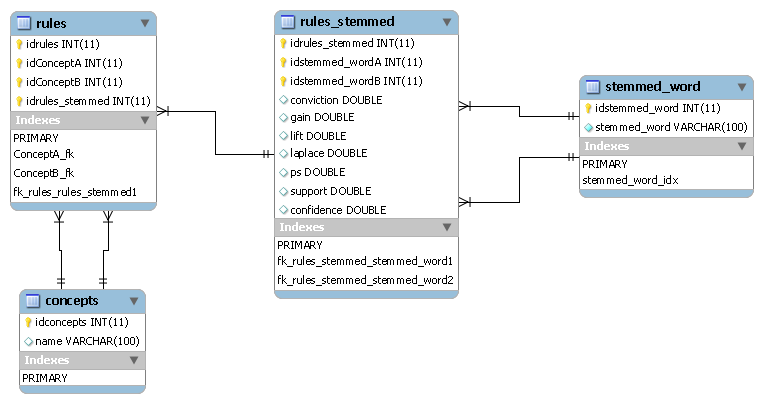


Figure 5.4 – Entity Relation Model

*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 5.5 – 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 5.5 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 5.6. 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 5.6 – Core Conceptual Architecture

ASSOCIATION

RULES

FREQUENT ITEMSET MAPPING

ONTOLOGY

ASSOCIATION RULES DATABASE

DOCUMENT ANALYSIS

FREQUENT PATTERNS

## Implementation and Development

This chapter will describe all steps of the implementation and development proposed. This can be observed in Figure 5.7, 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.

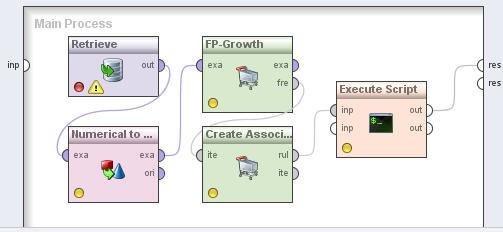


Figure 5.7 – Rapidminer Main Process screenshot

### 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 5.8). This set of procedures is the Document Analysis Block. In fact, is composed by 6 blocks following a specific order.

Figure 5.8 - 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*.

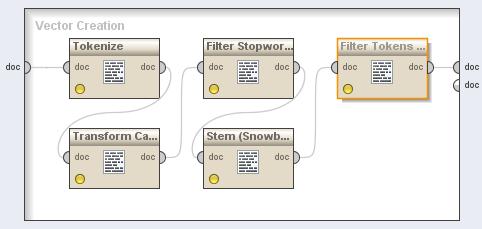


Figure 5.9 – Vector Creation Rapidminer Process

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 4.1.2 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 5.10 – Screenshot of AR system home page

The first screen to appear when the page opens is the Home Page. Figure 5.10 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 5.11. 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.

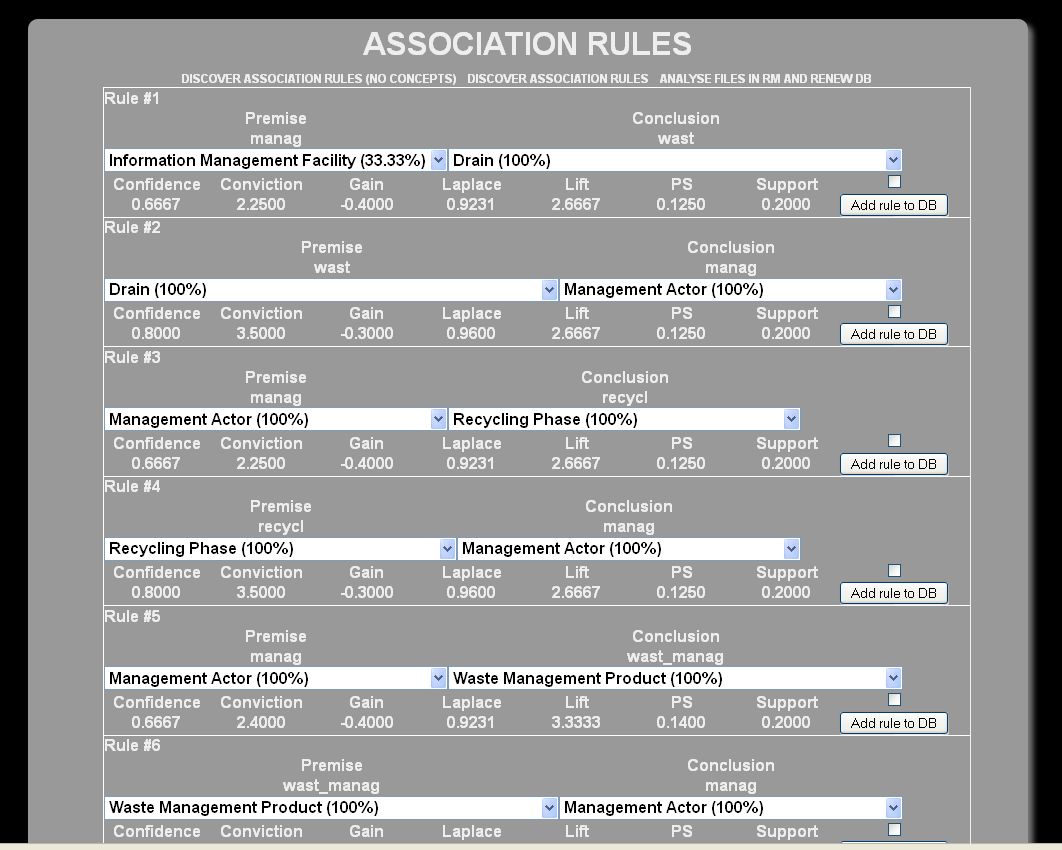


Figure 5.11 - Screenshot from Front-End AR page

After the server had executed the whole process, it delivers the information to the front end for visualization. Figure 5.11 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 3.1.2.2 - The 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 5.14 - 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 5.13. 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.

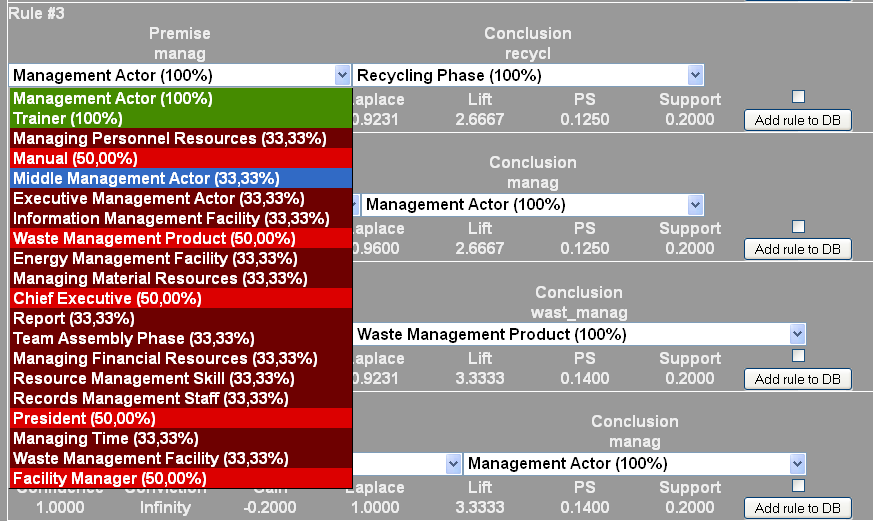


Figure 5.12 - Association Rule Functionality – Exact and Candidates concept visualizer

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 5.12 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.

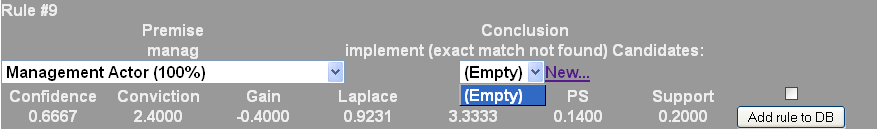


Figure 5.13 - AR Case with a new concept discovered

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 5.12, 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.

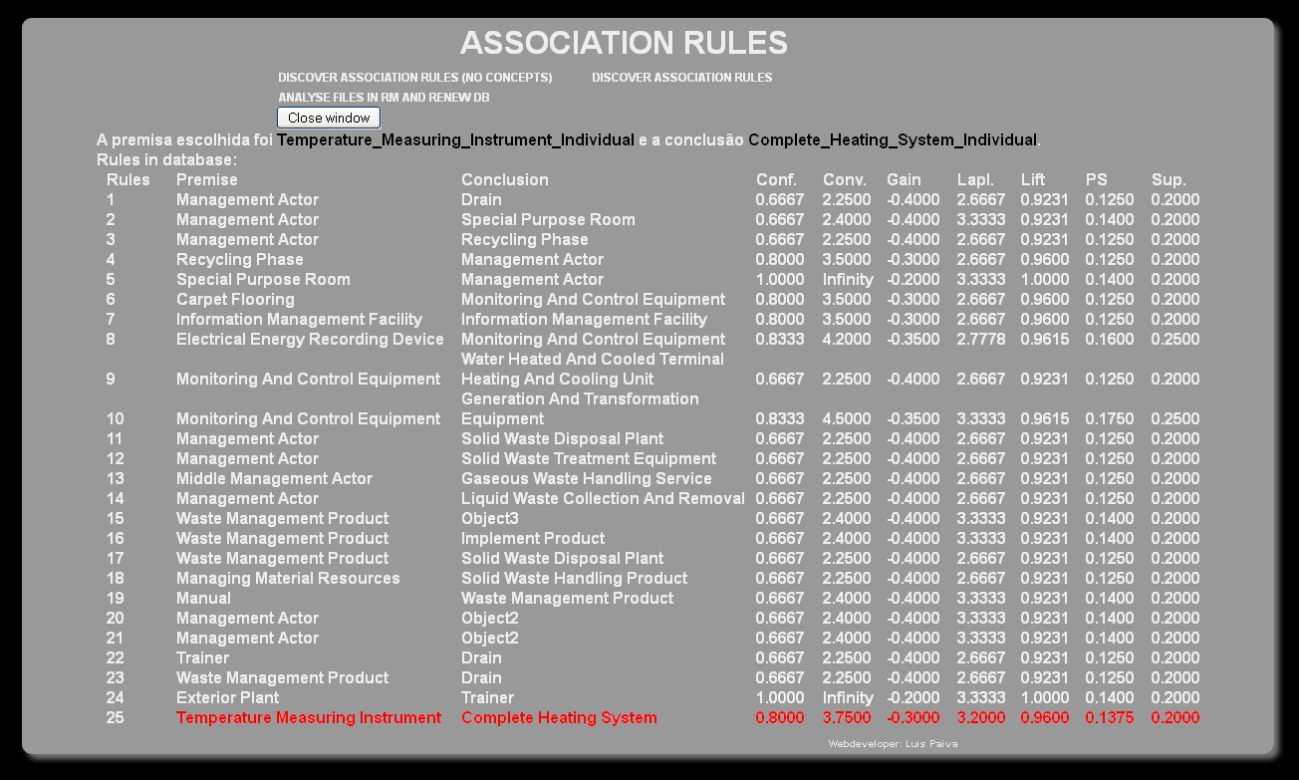


Figure 5.14 - Association Rules Database view page

# Evaluation

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 Figure 4.1 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 5.4.2 and 5.4.3 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 3.1.2.2 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.

## Scientific publications

(Falar da aprovação de publicações deste trabalho pela comunidade científica)

Incluir papers (e talvez incluir referência à ligação com o trabalho do Ruben e do Paulo)

The scenarios proposed by the author of this document were also discussed in some academic publications. For instance, (Paiva et al., 2013) was published in *CISTI'2013 - 8ª Conferência Ibérica de Sistemas e Tecnologias de Informação* a portuguese conference dedicated to information systems technologies. This publication was focused in the discussion of ontology enrichment techniques, namely association rules and frequent itemset mapping.

In addition, (Figueiras et al., 2012) published in the international conference *Knowledge Engineering and Ontology Development Conference 2012* held in Barcelona, Spain, talked focus about knowledge extraction techniques, namely *Vector Space Model*.

Furthermore, (Costa et al., 2012) also a document published and approved by scientific community, presented in a specific conference for the semantic information system techniques, SEMAPRO 2012, The Sixth International Conference on Advances in Semantic Processing. This publication discusses the Representation of Knowledge, namely its capture through semantic relationships.

As already discussed in the present document, this document is a following of the project thesis (Figueiras, 2012) about the construction of a framework to aimed to suppor the representation of knowledge with the help of an ontology.

This development of this project also makes a modest contribution to a PhD about Ontologies by Costa, Ruben, Msc. (needs more context here )

# Conclusion and Future Work

The main goal of this project was to create some dynamic in the management of an ontology. The idea proposed was to use knowledge extraction techniques, namely association rules to discover new knowledge and suggest a more precise ontology management. AR discovered

# Bibliography

Agrawal, R., Imieliński, T., Swami, A., 1993. Mining Association Rules Between Sets of Items in Large Databases, in: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, 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.

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

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

Batet, M., Sánchez, D., Valls, A., 2011. An ontology-based measure to compute semantic similarity in biomedicine. J. Biomed. Inform. 44, 118–125. doi:10.1016/j.jbi.2010.09.002

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

Bedini, I., Nguyen, B., 2007. Automatic ontology generation: State of the art. PRiSM Lab. Tech. Rep. Univ. Versailles.

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

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.

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

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

Costa, R., Figueiras, P., Paiva, L., Jardim-Gonçalves, R., Lima, C., 2012. Capturing Knowledge Representations Using Semantic Relationships An Ontology-based Approach. Presented at the SEMAPRO 2012, The Sixth International Conference on Advances in Semantic Processing, pp. 75–81.

Darwin, C., 1859. On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life. John Murray, London, UK, UK.

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

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

Figueiras, P., Costa, R., Paiva, L., Jardim-Gonçalves, R., Lima, C., 2012. Information Retrieval in Collaborative Engineering Projects - A Vector Space Model Approach: Presented at the Knowledge Engineering and Ontology Development Conference 2012, SciTePress - Science and Technology Publications, Barcelona, Spain, pp. 233–238. doi:10.5220/0004139302330238

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.

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

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

Guillaume Laforge, 2012. Groovy [WWW Document]. URL http://groovy.codehaus.org/

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.

Harris, Z.S., 1954. Distributional structure. Word 10, 146–162.

Hilderman, R., Hamilton, H., 2001. Knowledge discovery and measures of interest. Kluwer.

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.

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.

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., Fies, B., Zarli, A., Diraby, T.E., 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., Schulze, F., Blake, C., Rezgui, Y., Vesa, P., 2004. European Commission : CORDIS : Projects and Results : e-COGNOS ontology [WWW Document]. URL http://cordis.europa.eu/result/rcn/31993\_en.html (accessed 9.6.14).

Lima, C., Stephens, J., Böhms, M., 2003. 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).

Lin, D., 1998. An information-theoretic definition of similarity., in: ICML. pp. 296–304.

Lin, D., Pantel, P., 2001. DIRT@ SBT@ discovery of inference rules from text, in: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp. 323–328.

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

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

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

Manning, C.D., Raghavan, P., Schütze, H., 2008. Introduction to information retrieval. Cambridge university press Cambridge.

Marinica, C., Guillet, F., 2010. Knowledge-Based Interactive Postmining of Association Rules Using Ontologies. IEEE Trans. Knowl. Data Eng. 22, 784–797. doi:10.1109/TKDE.2010.29

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.

Nakov, P., Hearst, M.A., 2008. Solving Relational Similarity Problems Using the Web as a Corpus., in: ACL. Citeseer, pp. 452–460.

Niepert, M., Buckner, C., Allen, C., 2007. A Dynamic Ontology for a Dynamic Reference Work, in: Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries, JCDL ’07. ACM, New York, NY, USA, pp. 288–297. doi:10.1145/1255175.1255230

Oxford University, 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.

Pantel, P., Lin, D., 2002. Discovering word senses from text, in: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp. 613–619.

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.

Rapp, R., 2003. Word sense discovery based on sense descriptor dissimilarity, in: Proceedings of the Ninth Machine Translation Summit. pp. 315–322.

Salton, G., 1971. The SMART retrieval system—experiments in automatic document processing.

Salton, G., Wong, A., Yang, C.-S., 1975. A vector space model for automatic indexing. Commun. ACM 18, 613–620.

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

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

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.

Turney, P.D., 2006. Similarity of Semantic Relations. Comput. Linguist. 32, 379–416. doi:10.1162/coli.2006.32.3.379

Turney, P.D., Pantel, P., others, 2010. From frequency to meaning: Vector space models of semantics. J. Artif. Intell. Res. 37, 141–188.

Turney, P., Littman, M.L., Bigham, J., Shnayder, V., 2003. Combining independent modules to solve multiple-choice synonym and analogy problems.

Vo, B., Le, B., 2009. Mining traditional association rules using frequent itemsets lattice, in: International Conference on Computers Industrial Engineering, 2009. CIE 2009. Presented at the International Conference on Computers Industrial Engineering, 2009. CIE 2009, pp. 1401–1406. doi:10.1109/ICCIE.2009.5223866

W3C, 2004. OWL Web Ontology Language Overview [WWW Document]. OWL Web Ontol. Lang. URL http://www.w3.org/TR/2004/REC-owl-features-20040210/ (accessed 7.7.14).

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

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.

Yih, W., 2009. Learning term-weighting functions for similarity measures, in: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2. Association for Computational Linguistics, pp. 793–802.

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

Zeng, B., Jiang, X.-L., Zhao, W., Luo, C., 2010. The improvement of weighted association rules arithmetic based on FP-tree, in: Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on. IEEE, pp. V4–549.

Zhang, J., El-Diraby, T.E., 2012. Social Semantic Approach to Support Communication in AEC. J. Comput. Civ. Eng. 26, 90–104.

# Appendix A

# F:\Os meus documentos\Universidade\Dissertação\Escrita\working\inprocess\images\insert-rule_usd.bmp

Figure A.1 - USD for Insert Rule Use Case

# Appendix B.

UML Class Diagram

# F:\Os meus documentos\Universidade\Dissertação\Escrita\working\inprocess\images\UMLClassDiagram.bmp

Figure B.1 – UML CLASS DIAGRAM

# Appendix C

UML Use cases

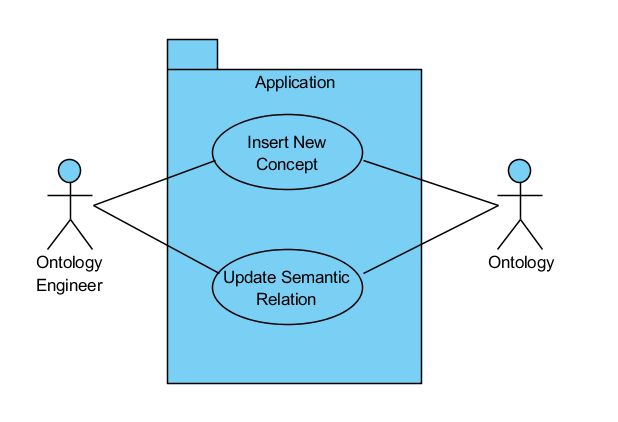


Figure C.1 - Main requirements UUC

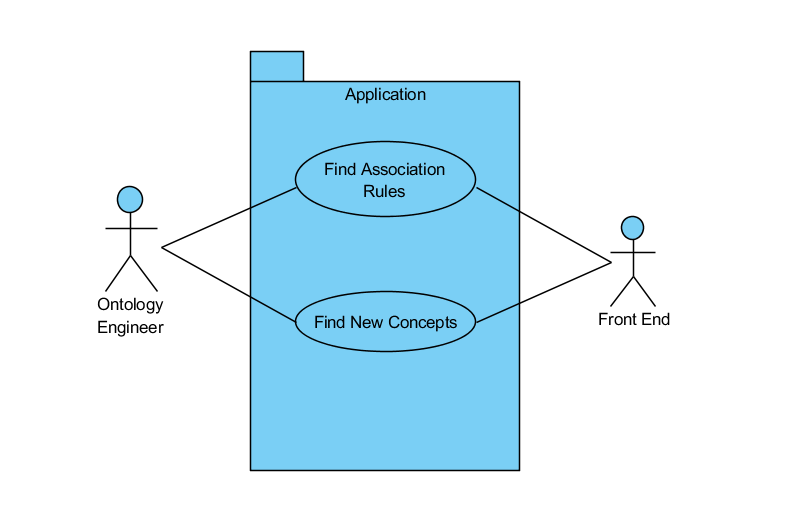


Figure C.2 - Front-End UUC

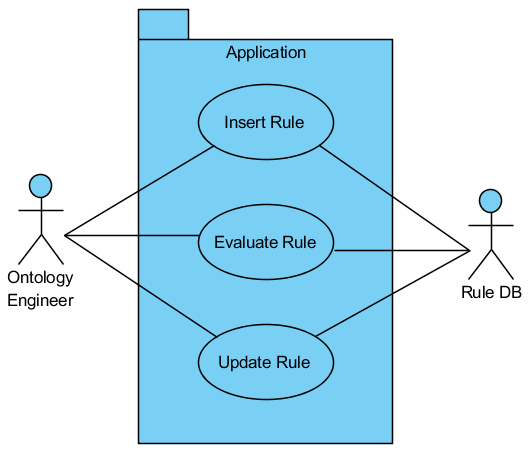


Figure C.3 - Rule DB UUC

# Appendix D

Table D.1 - List of all the rules discovered in Association Rules Discovery Process

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # | Premise | Conclusion | Confidence | Conviction | Gain | Laplace | Lift | Ps | Total Support |
| 1 | manag | wast | 0.6667 | 2.2500 | -0.4000 | 0.9231 | 2.6667 | 0.1250 | 0.2000 |
| 2 | wast | manag | 0.8000 | 3.5000 | -0.3000 | 0.9600 | 2.6667 | 0.1250 | 0.2000 |
| 3 | manag | recycl | 0.6667 | 2.2500 | -0.4000 | 0.9231 | 2.6667 | 0.1250 | 0.2000 |
| 4 | recycl | manag | 0.8000 | 3.5000 | -0.3000 | 0.9600 | 2.6667 | 0.1250 | 0.2000 |
| 5 | manag | wast\_manag | 0.6667 | 2.4000 | -0.4000 | 0.9231 | 3.3333 | 0.1400 | 0.2000 |
| 6 | wast\_manag | manag | 1.0000 | Infinity | -0.2000 | 1.0000 | 3.3333 | 0.1400 | 0.2000 |
| 7 | manag | plan | 0.6667 | 2.4000 | -0.4000 | 0.9231 | 3.3333 | 0.1400 | 0.2000 |
| 8 | plan | manag | 1.0000 | Infinity | -0.2000 | 1.0000 | 3.3333 | 0.1400 | 0.2000 |
| 9 | manag | implement | 0.6667 | 2.4000 | -0.4000 | 0.9231 | 3.3333 | 0.1400 | 0.2000 |
| 10 | implement | manag | 1.0000 | Infinity | -0.2000 | 1.0000 | 3.3333 | 0.1400 | 0.2000 |
| 11 | energi | consumpt | 0.8333 | 4.2000 | -0.3500 | 0.9615 | 2.7778 | 0.1600 | 0.2500 |
| 12 | consumpt | energi | 0.8333 | 4.2000 | -0.3500 | 0.9615 | 2.7778 | 0.1600 | 0.2500 |
| 13 | energi | temperatur | 0.6667 | 2.2500 | -0.4000 | 0.9231 | 2.6667 | 0.1250 | 0.2000 |
| 14 | temperatur | energi | 0.8000 | 3.5000 | -0.3000 | 0.9600 | 2.6667 | 0.1250 | 0.2000 |
| 15 | energi | indoor | 0.6667 | 2.2500 | -0.4000 | 0.9231 | 2.6667 | 0.1250 | 0.2000 |
| 16 | indoor | energi | 0.8000 | 3.5000 | -0.3000 | 0.9600 | 2.6667 | 0.1250 | 0.2000 |
| 17 | energi | heat | 0.6667 | 2.2500 | -0.4000 | 0.9231 | 2.6667 | 0.1250 | 0.2000 |
| 18 | heat | energi | 0.8000 | 3.5000 | -0.3000 | 0.9600 | 2.6667 | 0.1250 | 0.2000 |
| 19 | energi | energi\_consumpt | 0.8333 | 4.5000 | -0.3500 | 0.9615 | 3.3333 | 0.1750 | 0.2500 |
| 20 | energi\_consumpt | energi | 1.0000 | Infinity | -0.2500 | 1.0000 | 3.3333 | 0.1750 | 0.2500 |
| 21 | energi | electr | 0.8333 | 4.5000 | -0.3500 | 0.9615 | 3.3333 | 0.1750 | 0.2500 |
| 22 | electr | energi | 1.0000 | Infinity | -0.2500 | 1.0000 | 3.3333 | 0.1750 | 0.2500 |
| 23 | energi | power | 0.6667 | 2.4000 | -0.4000 | 0.9231 | 3.3333 | 0.1400 | 0.2000 |
| 24 | power | energi | 1.0000 | Infinity | -0.2000 | 1.0000 | 3.3333 | 0.1400 | 0.2000 |
| 25 | energi | oper | 0.6667 | 2.4000 | -0.4000 | 0.9231 | 3.3333 | 0.1400 | 0.2000 |
| 26 | oper | energi | 1.0000 | Infinity | -0.2000 | 1.0000 | 3.3333 | 0.1400 | 0.2000 |
| 27 | energi | hvac | 0.6667 | 2.4000 | -0.4000 | 0.9231 | 3.3333 | 0.1400 | 0.2000 |
| 28 | hvac | energi | 1.0000 | Infinity | -0.2000 | 1.0000 | 3.3333 | 0.1400 | 0.2000 |
| 29 | energi | cool | 0.6667 | 2.4000 | -0.4000 | 0.9231 | 3.3333 | 0.1400 | 0.2000 |
| 30 | cool | energi | 1.0000 | Infinity | -0.2000 | 1.0000 | 3.3333 | 0.1400 | 0.2000 |
| 31 | consumpt | temperatur | 0.6667 | 2.2500 | -0.4000 | 0.9231 | 2.6667 | 0.1250 | 0.2000 |
| 32 | temperatur | consumpt | 0.8000 | 3.5000 | -0.3000 | 0.9600 | 2.6667 | 0.1250 | 0.2000 |
| 33 | consumpt | indoor | 0.8333 | 4.5000 | -0.3500 | 0.9615 | 3.3333 | 0.1750 | 0.2500 |
| 34 | indoor | consumpt | 1.0000 | Infinity | -0.2500 | 1.0000 | 3.3333 | 0.1750 | 0.2500 |
| 35 | consumpt | heat | 0.6667 | 2.2500 | -0.4000 | 0.9231 | 2.6667 | 0.1250 | 0.2000 |
| 36 | heat | consumpt | 0.8000 | 3.5000 | -0.3000 | 0.9600 | 2.6667 | 0.1250 | 0.2000 |
| 37 | consumpt | energi\_consumpt | 0.8333 | 4.5000 | -0.3500 | 0.9615 | 3.3333 | 0.1750 | 0.2500 |
| 38 | energi\_consumpt | consumpt | 1.0000 | Infinity | -0.2500 | 1.0000 | 3.3333 | 0.1750 | 0.2500 |
| 39 | consumpt | electr | 0.6667 | 2.2500 | -0.4000 | 0.9231 | 2.6667 | 0.1250 | 0.2000 |
| 40 | electr | consumpt | 0.8000 | 3.5000 | -0.3000 | 0.9600 | 2.6667 | 0.1250 | 0.2000 |
| 41 | consumpt | oper | 0.6667 | 2.4000 | -0.4000 | 0.9231 | 3.3333 | 0.1400 | 0.2000 |
| 42 | oper | consumpt | 1.0000 | Infinity | -0.2000 | 1.0000 | 3.3333 | 0.1400 | 0.2000 |
| 43 | consumpt | hvac | 0.6667 | 2.4000 | -0.4000 | 0.9231 | 3.3333 | 0.1400 | 0.2000 |
| 44 | hvac | consumpt | 1.0000 | Infinity | -0.2000 | 1.0000 | 3.3333 | 0.1400 | 0.2000 |
| 45 | consumpt | cool | 0.6667 | 2.4000 | -0.4000 | 0.9231 | 3.3333 | 0.1400 | 0.2000 |
| 46 | cool | consumpt | 1.0000 | Infinity | -0.2000 | 1.0000 | 3.3333 | 0.1400 | 0.2000 |
| 47 | wast | recycl | 1.0000 | Infinity | -0.2500 | 1.0000 | 4.0000 | 0.1875 | 0.2500 |
| 48 | recycl | wast | 1.0000 | Infinity | -0.2500 | 1.0000 | 4.0000 | 0.1875 | 0.2500 |
| 49 | wast | wast\_manag | 0.8000 | 4.0000 | -0.3000 | 0.9600 | 4.0000 | 0.1500 | 0.2000 |
| 50 | wast\_manag | wast | 1.0000 | Infinity | -0.2000 | 1.0000 | 4.0000 | 0.1500 | 0.2000 |
| 51 | wast | plan | 0.8000 | 4.0000 | -0.3000 | 0.9600 | 4.0000 | 0.1500 | 0.2000 |
| 52 | plan | wast | 1.0000 | Infinity | -0.2000 | 1.0000 | 4.0000 | 0.1500 | 0.2000 |
| 53 | toilet | sanitari | 0.8000 | 4.0000 | -0.3000 | 0.9600 | 4.0000 | 0.1500 | 0.2000 |
| 54 | sanitari | toilet | 1.0000 | Infinity | -0.2000 | 1.0000 | 4.0000 | 0.1500 | 0.2000 |
| 55 | temperatur | indoor | 0.8000 | 3.7500 | -0.3000 | 0.9600 | 3.2000 | 0.1375 | 0.2000 |
| 56 | indoor | temperatur | 0.8000 | 3.7500 | -0.3000 | 0.9600 | 3.2000 | 0.1375 | 0.2000 |
| 57 | temperatur | heat | 0.8000 | 3.7500 | -0.3000 | 0.9600 | 3.2000 | 0.1375 | 0.2000 |
| 58 | heat | temperatur | 0.8000 | 3.7500 | -0.3000 | 0.9600 | 3.2000 | 0.1375 | 0.2000 |
| 59 | temperatur | energi\_consumpt | 0.8000 | 3.7500 | -0.3000 | 0.9600 | 3.2000 | 0.1375 | 0.2000 |
| 60 | energi\_consumpt | temperatur | 0.8000 | 3.7500 | -0.3000 | 0.9600 | 3.2000 | 0.1375 | 0.2000 |
| 61 | temperatur | hvac | 0.8000 | 4.0000 | -0.3000 | 0.9600 | 4.0000 | 0.1500 | 0.2000 |
| 62 | hvac | temperatur | 1.0000 | Infinity | -0.2000 | 1.0000 | 4.0000 | 0.1500 | 0.2000 |
| 63 | temperatur | cool | 0.8000 | 4.0000 | -0.3000 | 0.9600 | 4.0000 | 0.1500 | 0.2000 |
| 64 | cool | temperatur | 1.0000 | Infinity | -0.2000 | 1.0000 | 4.0000 | 0.1500 | 0.2000 |
| 65 | temperatur | climat | 0.8000 | 4.0000 | -0.3000 | 0.9600 | 4.0000 | 0.1500 | 0.2000 |
| 66 | climat | temperatur | 1.0000 | Infinity | -0.2000 | 1.0000 | 4.0000 | 0.1500 | 0.2000 |
| 67 | recycl | wast\_manag | 0.8000 | 4.0000 | -0.3000 | 0.9600 | 4.0000 | 0.1500 | 0.2000 |
| 68 | wast\_manag | recycl | 1.0000 | Infinity | -0.2000 | 1.0000 | 4.0000 | 0.1500 | 0.2000 |
| 69 | recycl | plan | 0.8000 | 4.0000 | -0.3000 | 0.9600 | 4.0000 | 0.1500 | 0.2000 |
| 70 | plan | recycl | 1.0000 | Infinity | -0.2000 | 1.0000 | 4.0000 | 0.1500 | 0.2000 |
| 71 | offic | offic\_build | 0.8000 | 4.0000 | -0.3000 | 0.9600 | 4.0000 | 0.1500 | 0.2000 |
| 72 | offic\_build | offic | 1.0000 | Infinity | -0.2000 | 1.0000 | 4.0000 | 0.1500 | 0.2000 |
| 73 | indoor | heat | 0.8000 | 3.7500 | -0.3000 | 0.9600 | 3.2000 | 0.1375 | 0.2000 |
| 74 | heat | indoor | 0.8000 | 3.7500 | -0.3000 | 0.9600 | 3.2000 | 0.1375 | 0.2000 |
| 75 | indoor | energi\_consumpt | 0.8000 | 3.7500 | -0.3000 | 0.9600 | 3.2000 | 0.1375 | 0.2000 |
| 76 | energi\_consumpt | indoor | 0.8000 | 3.7500 | -0.3000 | 0.9600 | 3.2000 | 0.1375 | 0.2000 |
| 77 | indoor | hvac | 0.8000 | 4.0000 | -0.3000 | 0.9600 | 4.0000 | 0.1500 | 0.2000 |
| 78 | hvac | indoor | 1.0000 | Infinity | -0.2000 | 1.0000 | 4.0000 | 0.1500 | 0.2000 |
| 79 | indoor | cool | 0.8000 | 4.0000 | -0.3000 | 0.9600 | 4.0000 | 0.1500 | 0.2000 |
| 80 | cool | indoor | 1.0000 | Infinity | -0.2000 | 1.0000 | 4.0000 | 0.1500 | 0.2000 |
| 81 | heat | energi\_consumpt | 0.8000 | 3.7500 | -0.3000 | 0.9600 | 3.2000 | 0.1375 | 0.2000 |
| 82 | energi\_consumpt | heat | 0.8000 | 3.7500 | -0.3000 | 0.9600 | 3.2000 | 0.1375 | 0.2000 |
| 83 | heat | hvac | 0.8000 | 4.0000 | -0.3000 | 0.9600 | 4.0000 | 0.1500 | 0.2000 |
| 84 | hvac | heat | 1.0000 | Infinity | -0.2000 | 1.0000 | 4.0000 | 0.1500 | 0.2000 |
| 85 | heat | cool | 0.8000 | 4.0000 | -0.3000 | 0.9600 | 4.0000 | 0.1500 | 0.2000 |
| 86 | cool | heat | 1.0000 | Infinity | -0.2000 | 1.0000 | 4.0000 | 0.1500 | 0.2000 |
| 87 | energi\_consumpt | electr | 0.8000 | 3.7500 | -0.3000 | 0.9600 | 3.2000 | 0.1375 | 0.2000 |
| 88 | electr | energi\_consumpt | 0.8000 | 3.7500 | -0.3000 | 0.9600 | 3.2000 | 0.1375 | 0.2000 |
| 89 | energi\_consumpt | oper | 0.8000 | 4.0000 | -0.3000 | 0.9600 | 4.0000 | 0.1500 | 0.2000 |
| 90 | oper | energi\_consumpt | 1.0000 | Infinity | -0.2000 | 1.0000 | 4.0000 | 0.1500 | 0.2000 |
| 91 | energi\_consumpt | hvac | 0.8000 | 4.0000 | -0.3000 | 0.9600 | 4.0000 | 0.1500 | 0.2000 |
| 92 | hvac | energi\_consumpt | 1.0000 | Infinity | -0.2000 | 1.0000 | 4.0000 | 0.1500 | 0.2000 |
| 93 | energi\_consumpt | cool | 0.8000 | 4.0000 | -0.3000 | 0.9600 | 4.0000 | 0.1500 | 0.2000 |
| 94 | cool | energi\_consumpt | 1.0000 | Infinity | -0.2000 | 1.0000 | 4.0000 | 0.1500 | 0.2000 |
| 95 | electr | power | 0.8000 | 4.0000 | -0.3000 | 0.9600 | 4.0000 | 0.1500 | 0.2000 |
| 96 | power | electr | 1.0000 | Infinity | -0.2000 | 1.0000 | 4.0000 | 0.1500 | 0.2000 |
| 97 | wast\_manag | plan | 1.0000 | Infinity | -0.2000 | 1.0000 | 5.0000 | 0.1600 | 0.2000 |
| 98 | plan | wast\_manag | 1.0000 | Infinity | -0.2000 | 1.0000 | 5.0000 | 0.1600 | 0.2000 |
| 99 | hvac | cool | 1.0000 | Infinity | -0.2000 | 1.0000 | 5.0000 | 0.1600 | 0.2000 |
| 100 | cool | hvac | 1.0000 | Infinity | -0.2000 | 1.0000 | 5.0000 | 0.1600 | 0.2000 |
| 101 | coat\_materi | coat | 1.0000 | Infinity | -0.2000 | 1.0000 | 5.0000 | 0.1600 | 0.2000 |
| 102 | coat | coat\_materi | 1.0000 | Infinity | -0.2000 | 1.0000 | 5.0000 | 0.1600 | 0.2000 |

1. Refer to http://plato.stanford.edu/about.html for more information [↑](#footnote-ref-1)
2. 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-2)
3. in literature, other names can be found to represent this same itemsets like antecedent and consequent (Hoque et al., 2011) in contrast to premise and conclusion. The latter designation will be adapted in the present study. [↑](#footnote-ref-3)
4. As a curiosity, Lift is well known in the scientific community as a measure used in the IBM’s Intelligent Miner (IBM - International Business Machines, 1996). [↑](#footnote-ref-4)
5. 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-5)
6. Groovy builds upon the strengths of Java but has additional power features inspired by languages like Python, Ruby and Smalltalk. (Guillaume Laforge, 2012) [↑](#footnote-ref-6)