Ruben Duarte Dias da Costa

Mestre em Engenharia Informática

Semantic Enrichment of Knowledge Sources Supported by Domain Ontologies

Dissertação para obtenção do Grau de Doutor em
Engenharia Electrotécnica e de Computadores

Orientador: Professor Doutor Celson Pantoja Lima,
Professor Visitante, Massachusetts Institute of Technology
Professor Adjunto II, Universidade Federal do Oeste do Pará

Co-orientadores: Professor Doutor Adolfo Sanchez Steiger Garção
Professor Catedrático Jubilado,
Faculdade Ciências e Tecnologia,Universidade Nova de Lisboa

Júri:

Presidente: Professor Doutor Joaquim Pamiés Teixeira

Arguentes: Doutor Alain Zarli

Professor Doutor Paulo José Osório Rupino da Cunha

Vogais: Professor Doutor Manuel Martins Barata

Professor Doutor Ricardo Jorge Silvério de Magalhães Machado

Professor Doutor Ricardo Luís Rosa Jardim Gonçalves

Professor Doutor José António Barata de Oliveira

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Novembro, 2014

Semantic Enrichment of Knowledge Sources Supported by Domain Ontologies

To Ana,

My soulmate. My light.

To my Parents,

For making me who I am today

Acknowledgements

Some people might argue about “why bother in doing a PhD?”. Looking back now, I can think of a million of reasons for not doing it, and very few reasons for doing it. It will not double your monthly salary when you finish it. You will have to stay working overtime, giving up moments where you could be with your family and friends, meaning less and less weekends, less and less holidays. You might even to end up in a situation where you go to bed thinking in your PhD and wake up thinking in your PhD. You have to be well prepared to deal with frustrations – oh yes, they happen! So, it is very plausible to ask: “So why did you bother?”. It is all about challenge yourself, push yourself to new heights, achieve a difficult goal. It is all about, improve yourself and your life, improve your abilities to understand and solve problems, increase your confidence. Like someone said, it is just another challenge in life. But, like any other challenge in life, this one in particular would not be possible with the people that always stood there for me in this journey.

For all the reasons I’ve stated previously, I would like to give a big thank you to my supervisor and friend Professor Celson Lima. Despite the fact that we were not working co-located, Professor Celson support during the development of this thesis, was unconditional. Besides his technical and scientific skills, Professor Celson was much more than a supervisor; he was a mentor to me, which made me a more mature person. He showed me that life as a researcher goes beyond a PhD thesis; it is not only about how you solve problems, but your ability to seek for answers that matters. “Valeu”!

I also would like to thank my co-supervisor Professor Adolfo Steiger Garção, for his wisdom as a teacher at the university. I met Professor Steiger for the first time when I joined the university in 1995. Professor Steiger classes were always full of students, because of his passionate way of teaching. I will always remember his classes and advices to young students.

I wish to thank the members of the thesis accompanying committee, represented by Professor José Barata and Professor Paulo Rupino Cunha, for all the useful comments and suggestions that were provided during the development of this thesis.

I owe a special note of gratitude to my colleague Professor Pedro Maló, who was the responsible for encouraging me to work as a researcher at UNINOVA and start with my MSc and PhD theses. Thank you Pedro for believing in me.

I'm also very grateful to Professor Ricardo Gonçalves, who I worked with for more than 10 years. Professor Ricardo, provided me with valuable advices not only with respect to this thesis, but also at a professional level as a researcher at UNINOVA.

My thanks to Eng. Graham Storer, not only for helping me with English proofreading which I'm very grateful, but much more than that, to our long discussions through skype about this work and your insights about the building & construction sector, which were vital for the accomplishment of this thesis.

I would like to extend my gratitude to all my colleagues at UNINOVA who have worked with me over the years, which I'm very proud of, and especially to Eng. Paulo Figueiras who I had the pleasure to support him during his MSc dissertation and currently working him at UNINOVA. Paulo is the proof that learning can also be derived from young people, and I have learnt a lot by working with Paulo – Keep up the good work!

I would like to thank to the participants of the European research projects in which I was involved in, namely CoSpaces (IST-5-034245), CRESCENDO (FP7-234344) and MobiS (FP7-318452), and specially to Scott Hansen, Terrence Fernando, Mitja Jermol and Luka Bradesko.

My acknowledgement to all my friends, who I had the pleasure to spend wonderful times during the years of a PhD student. They have tried to keep me as sane as possible, even when things were not progressing the way I would like them to be.

I wish to thank my parents for all their unconditional support along my life and always present in providing me so much motivation during this work.

To my most than everything, Ana, for always supporting me and make me belief that this work would become a reality. Thank you for being so comprehensible during the bad moments. Without you, this work would not become a reality.

Resumo

Esta tese introduz um novo conceito para suportar a criação de representações de conhecimento baseadas em vectores semânticos enriquecidos, usando para tal a abordagem clássica do vector space model, extendendo-o com suporte ontológico. Um dos principais desafios de investigação desta tese está relacionado com a tentativa de criar uma abordagem de formalização e de representação do conteúdo de documentos, onde as abordagem tradicionais apenas têm em linha de conta a informação baseada na ocorrência palavras explícitas nos documentos. O trabalho aqui descrito visa explorar a forma na qual representações de conhecimento tradicionais poderão ser enriquecidas, através da incorporação informação implícita extraída através de relações complexas (associações semânticas) modeladas através de ontologias de domínio em conjunto com a informação presente em documentos. Os resultados mais relevantes podem ser descritos da seguinte forma: (i) conceptualização de um modelo que permite o enriquecimento semântico de fontes de conhecimento apoiado por especialistas de domínio; (ii) desenvolvimento de um método para estender o espaço vectorial tradicional, usando conhecimento de ontologias de domínio; (iii) desenvolvimento de um método para suportar a aprendizagem de ontologias, tendo por base a descoberta de novas relações ontológicas em fontes de informação não-estruturadas; (iv) desenvolvimento de um processo para avaliar o enriquecimento semântico; (v) implementação de uma prova de conceito, denominada SENSE (Semantic Enrichment kNowledge SourcEs), que permite validar as ideias desenvolvidas no âmbito desta tese; (vi) publicação de vários artigos científicos e suporte ao desenvolvimento de 4 dissertações de mestrado, no departamento de Engenharia Electrotécnica da FCT/UNL. De se referir ainda que o trabalho sobre o referencial semântico desenvolvido nesta tese inspirou-se e reutilizou

trabalhos já existentes, desenvolvidos no âmbito de projetos Europeus de investigação, a fim de se evitar percorrer caminhos já percorridos.

Palavras-chave: Recuperação da Informação, Representação de Conhecimento, Vocabulários Controlados, Classificação de Documentos não-Supervisionada, *Vector Space Model*

Abstract

This thesis introduces a novel conceptual framework to support the creation of knowledge representations based on enriched Semantic Vectors, using the classical vector space model approach extended with ontological support. One of the primary research challenges addressed here relates to the process of formalization and representation of document contents, where most existing approaches are limited and only take into account the explicit, word-based information in the document. This research explores how traditional knowledge representations can be enriched through incorporation of implicit information derived from the complex relationships (semantic associations) modelled by domain ontologies with the addition of information presented in documents. The relevant achievements pursued by this thesis are the following: (i) conceptualization of a model that enables the semantic enrichment of knowledge sources supported by domain experts; (ii) development of a method for extending the traditional vector space, using domain ontologies; (iii) development of a method to support ontology learning, based on the discovery of new ontological relations expressed in non-structured information sources; (iv) development of a process to evaluate the semantic enrichment; (v) implementation of a proof-of-concept, named SENSE (Semantic Enrichment kNowledge SourcEs), which enables to validate the ideas established under the scope of this thesis; (vi) publication of several scientific articles and the support to 4 master dissertations carried out by the department of Electrical and Computer Engineering from FCT/UNL. It is worth mentioning that the work developed under the semantic referential covered by this thesis has reused relevant achievements within the scope of research European projects, in order to address approaches which are considered scientifically sound and coherent and avoid “reinventing the wheel”.

Keywords: Information Retrieval, Knowledge Representation, Controlled Vocabularies, Unsupervised Document Classification, Vector Space Model

Table of Contents

List of Figures

List of Tables

List of Acronyms

AI	Artificial Intelligence
ANN	Artificial Neural Networks
B&C	Building and Construction
bsDD	buildingSMART Data Dictionaries
CoP	Communities of Practice
CSCW	Computer Supported Cooperative Work
CWE	Collaborative Working Environments
DG	Decisional Gate
FI	Frequent Itemset
GUI	Graphical User Interface
HTML	HyperText Markup Language
ICT	Information and Communication Technologies
IFC	Industry Foundation Classes
IP	Integrated Project
IR	Information Retrieval

IT	Information Technology
KB	Knowledge Base
KDD	Knowledge Discovery in Databases
kNN	k-nearest neighbours
KM	Knowledge Management
KR	Knowledge Representation
KS	Knowledge Source
NB	Naives Bayes
NLP	Natural Language Processing
NN	Neural Network
OET	Ontology Equivalent Term
OWL	Ontology Web Language
OWL	Ontology Web Language – Description Logic
PoC	Proof of Concept
POS	Part of Speech
RDF	Resource Description Framework
RSS	Residual Sum of Squares
SD	Sequence Diagram
SECI	Socialization, Externalization, Combination, Internalization
SENSE	Semantic Enrichment of Knowledge Sources
SOA	Service Oriented Architectures
SV	Semantic Vector
SVKB	Semantic Vector Keyword-based
SVM	Support Vector Machine
SVOB	Semantic Vector Taxonomy-based
SVTB	Semantic Vector Ontology-based
TC	Text Categorization
TF-IDF	Term Frequency – Inverse Document Frequency

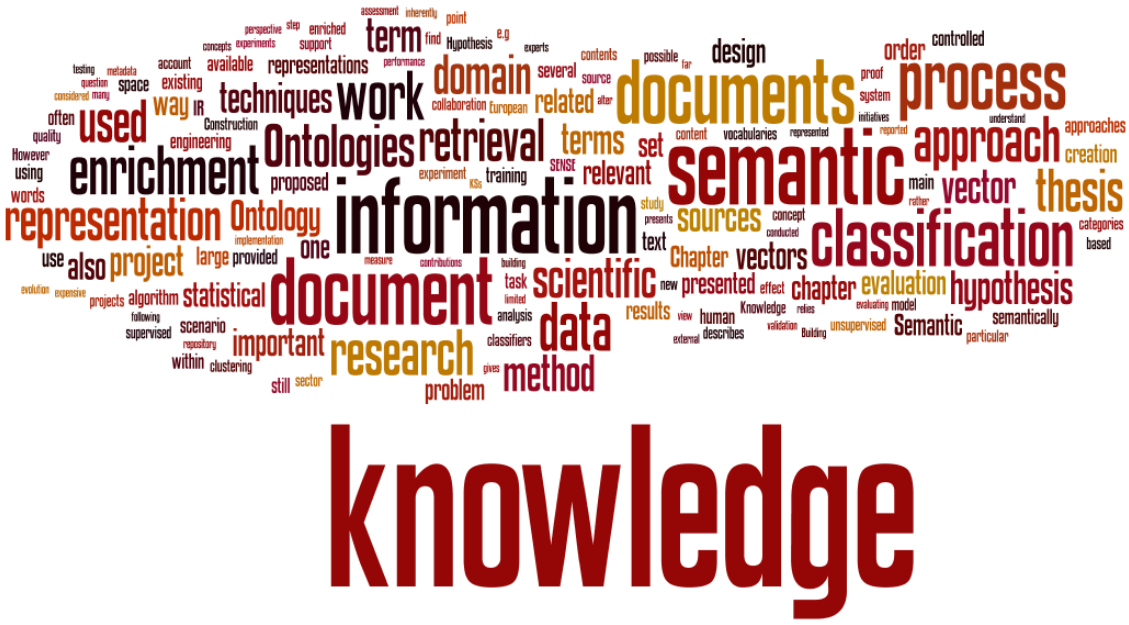
TREC	Text REtreival Conference
UC	Use Case
UML	Unified Modelling Language
VSM	Vector Space Model
XML	Extensible Markup Language
W3C	World Wide Web Consortium
WSD	Word Sense Disambiguation

1

Introduction

"Anybody who has been seriously engaged in scientific work of any kind realizes that over the entrance to the gates of the temple of science are written the words: Ye must have faith. It is a quality which the scientist cannot dispense with."

- Max Karl Ernst Ludwig Planck (1858 – 1947), Nobel Prize in Physics



The way that knowledge might be represented has been an important endeavour since the dawn of the human race. The creation of written and spoken languages is the foremost example of the effort to represent knowledge in such a way as to preserve it and to guarantee that it can be transmitted to future generations. The subject of knowledge representation gained a new dimension with the advent of the computer age. Particularly, with the creation of the World Wide Web, new forms of knowledge representation were needed in order to transmit data from

source to recipient in common data formats, and to aid humans to find the information they want in an easily understandable manner. With the evolution of the Semantic Web, knowledge representation techniques moved into the spotlight, aiming at bringing human understanding of the meaning of data to the world of machines. Such techniques create knowledge representations of Knowledge Sources (KSs), whether they are web pages or documents.

The field of Information Retrieval (IR) is concerned with the retrieval of information content that is relevant to a user's information needs. Information retrieval techniques were primarily designed for the access and retrieval of library documents, and more recently web pages. IR is often regarded as synonymous with document retrieval and text retrieval, though many IR systems also retrieve pictures, audio, and other types of non-textual information. The word "document" is used herein to include not just text documents, but any "clump" of information.

People have the ability to understand abstract meanings that are conveyed by natural language. This is why intermediary reference librarians are useful; they can talk to a librarian about his/her information needs and then find the documents that are relevant. The challenge of information retrieval is to mimic this interaction, replacing the librarian with an automated system. This task is difficult because machine comprehension of natural language is generally still an open research problem.

Ontologies are the foundation of both content-based information access and semantic interoperability over the web. With respect to the work reported in this thesis, it is proposed to use knowledge available in domain Ontologies in order to support the process of representing knowledge sources (e.g. project reports, meeting minutes, descriptions of problems/solutions) thus improving the classification of such knowledge sources. A case study focused on the Building & Construction sector is used. Fundamentally, Ontologies are used to improve communication between people and/or computers. By describing the intended meaning of "things" in a formal and unambiguous way, Ontologies enhance the ability of both humans and computers to interoperate seamlessly and consequently facilitate the development of semantic (and more intelligent) software applications.

Under the scope of this thesis, it is supposed that information contained in Ontologies can be incorporated into many representation schemes and algorithms. This research focuses on a particular representation scheme based on Vector Space Model, which represent documents as a vector of their most important terms (so-called term vector), which is regarded herein as a statistically-based Knowledge Representation (KR). Important terms are those which are considered to be the best discriminators for each document space (i.e. content scope). The aim of the current work is to understand how useful external domain knowledge is to the process of enriching knowledge representations; when it makes sense to bring in such background knowledge and what the pros & cons trade-offs may be. In order to do this, the idea is to intuitively alter basic *tf-idf* (term frequency–inverse document frequency) weighted document

term vectors for all documents to be represented, with the help of already available domain Ontology to generate new “enriched” semantic vectors.

This thesis describes the representation of KSs through the use of Semantic Vectors (SVs) based on the combination of the Vector Space Model (VSM) approach and a domain-specific Ontology. Thus, KSs are represented by SVs which contain concepts and their equivalent terms, weights (statistical, keyword, taxonomical, and ontological), relations and other elements that semantically enrich each SV. The proposed approach takes into account three different but complementary procedures for building up the semantic vector, each of which is considered a more realistic iteration of a given knowledge representation, namely keyword-based, taxonomy-based and ontology-based semantic vectors.

The idea behind a term vector is to represent each document in a collection of documents as a point in a multi-dimensional space (a vector in a vector space). Points that are close together in this space are semantically similar and points that are far apart are semantically distant. The intuition behind this work is to alter term vectors by strengthening the discriminative terms in a document in proportion to how much they are related to other terms in the document (where relatedness includes all possible relationships modelled in an Ontology). A side effect of this process is the weeding out (weakening/removal) of less important terms. Since Ontologies model domain knowledge independently of any particular document corpus, there is also the possibility of introducing new terms to the term vector that are highly related to the document but are not explicitly present in it. The approach adopted for enriching term vectors is therefore based on a combination of statistical information and semantic domain knowledge.

The performance of the proposed “enriched” approach needed evaluation which was done by comparison with an unsupervised document classification algorithm. Document clustering has become one of the main techniques for organizing large volumes of documents into a small number of meaningful clusters. However, there still exist several challenges for document clustering, such as high dimensionality, scalability, accuracy, meaningful cluster labels, overlapping clusters, and extracting the semantics from the texts. Also, performance is directly related to the quantity and quality of information within the Knowledge Base (KB) it runs upon. Until, maybe, Ontologies and metadata (and the Semantic Web itself) become a global commodity, the lack, or incompleteness, of available Ontologies and KBs is a limitation that has to be lived with.

An unsupervised classification algorithm (K-Means clustering) was adopted to evaluate the results of our approach. One of the reasons why unsupervised classification was chosen is that supervised classification is inherently limited by the quality of the information that can be inferred from the training data. The objective here is to use a centroid-based document classification algorithm to assess the effectiveness of the altered vectors since no in-depth

knowledge of the actual contents of the document corpus used was provided (it was largely "blind").

In summary, the reasons why unsupervised classification was chosen over supervised classification were:

- Supervised classification is inherently limited by the information that can be inferred from the training data. Meaning that, the accuracy and the representativeness of the training data, and also the distinctiveness of the classes must be taken into account. This tends to be a problem when dealing with large document corpora, when no previous in-depth (discriminatory) knowledge about the contents of documents is assumed.
- Some documents tend to overlap, even when belonging to different categories. Such situations are quite common when working with documents with an average of 3.500 words each. In general, text classification is a multi-class problem (more than 2 categories). Training supervised text classifiers requires large amounts of labelled data, which annotation can be time consuming and expensive. A common drawback of supervised learning algorithms is that they assume binary classification tasks and thus require the use of sub-optimal (and often computationally expensive) approaches such as "one vs. rest" to solve multi-class problems, let alone structured domains such as strings and trees.
- Manually labelling documents beforehand is not a trivial task and the quality of the task influences the training set of the classification algorithm. The intention of this work is to reduce as far as possible human intervention in the classification task and also to scale up our approach to sets of hundreds of scientific publications.
- The goal of the assessment is to evaluate if the semantic enrichment process improves the measure of similarity among documents, even when such documents were not considered similar using purely statistical approaches whereas they are similar from a semantically enriched perspective.

1.1 Problem Statement

One of the scientific themes focused on is the information retrieval and especially the evaluation of the quality of the information retrieval process. In order words, how to evaluate the beneficial effect of semantic enrichment from using background knowledge in existing domain Ontologies? The practice of information retrieval evaluation has, however, run well ahead of the theory. It was only at the end of the 1990s that the reliability, efficiency, and interpretability of evaluation results began to be formally investigated. In part the delay was because the datasets needed for a critical investigation of evaluation only became available when large-scale collaborative experiments had been running for several years.

But the same scale of data that makes information retrieval technology necessary, also makes manual assessment costly. Document classification refers to the process of organizing a set of documents, typically a large set, into a set of predefined classes or categories. To perform this function automatically, a document classifier typically relies on some training data, which is often a small but significant fraction of the documents and their correct categories as specified by a human (or some other external and similarly expensive accurate means of classification).

Search technology basically connects simple queries with documents, relieving both the provider and the consumer of knowledge from the complexity of matching knowledge sources to knowledge needs. The result is a panoply of tools that allow “novice” users (and experts) to find relevant information, across billions of documents, in a fraction of a second. But in doing away with precise, formal knowledge representations in favour of approximations, information retrieval faced an important problem. It is simply not possible to conclusively state that a knowledge source matches a knowledge request, even in the terms in which the request is formulated. One can say that a document has been manually assigned a certain classification under a hierarchical taxonomy. However one cannot guarantee that in absence of an effective knowledge representation, a particular knowledge source meets a user’s knowledge need expressed by a set of keywords.

Like many IR tasks, knowledge representation and classification techniques depend on using content independent metadata (e.g. author, creation date) and/or content dependent metadata (e.g. words in the document). However, such approaches tend to be inherently limited by the information that is explicit in the documents, which introduces a very real problem. For instance, in the situation where words like ‘architect’ and ‘design’ do not co-occur frequently, statistical techniques will fail to make any correlation between them. Furthermore, existing IR techniques are based upon indexing keywords extracted from documents and then creating a vector of terms. Unfortunately, keywords or index terms alone often do not adequately capture the document contents, resulting in poor indexation and retrieval performances. Keyword indexing is still widely used in commercial systems because it is by far the most viable way to process

large amounts of text, despite the high computational power and cost required to update and maintain the indexes.

The semantic referential (particularly Ontologies) which been used in information system development as one of the main knowledge representation tools, consists of concepts, a hierarchy, arbitrary relations between concepts, and possibly other axioms. However, Ontology building is a time-consuming process, involving manual work in conceptualizing, formalizing and maintaining it, which sometimes leads to a rather incomplete on inconsistent model. This is why Ontology learning is still one of the topics which deserves the special attention of the scientific community. Although several research streams have been proposed within this area, there are still no automatic mechanisms for creating and maintaining Ontologies from unstructured sources of information efficiently and without human intervention.

1.1.1 Research question

Such challenges provoke the following question:

How to formally alter and add contents to a document's statistical term vector (a basic knowledge representation) and thereby provide classifiers with a semantically enriched vector as opposed to a pure statistical representation?

1.1.2 Hypothesis

The hypothesis which guides this work is that:

Semantic background knowledge from Ontologies can be used for the enrichment of traditional statistical term vectors, by consequently to affect the document term vectors in a way that it is possible to measure the effect of semantic enrichment on existing classifiers*.*

Thus, one of the main contributions of this work is to affect the document term vectors in a way that is possible to use and measure the effect of semantic enrichment on existing classifiers.

1.1.3 Expected Outcomes

The expected outcomes to be delivered by this thesis are the following:

- A step-wise approach for semantic enrichment of knowledge representations.
- Advancement of domain Ontology as a way to externally enrich knowledge sources.
- Developing a semi-automatic method for keeping semantic integrity of domain Ontology harmonized with a knowledge repository.
- An approach for evaluating the performance of the semantic enrichment process.

- A proof-of-concept applicable to the Building & Construction sector which implements the semantic enrichment process.
- A set of relevant scientific publications with peer review accepted.
- A knowledge-search engine enhanced with semantic capabilities.

1.2 Methodological Approach

The methodological approach followed by this thesis has its roots in the scientific method. Aristotle is recognized as the "inventor" of the scientific method, which can be defined as a process by which scientists, collectively and over time, endeavour to construct an accurate (i.e. valid, reasoned, reliable, consistent and non-arbitrary) representation of the world.

When conducting research, scientists observe the scientific method to collect measurable, empirical evidence in an experimental process related to a hypothesis (often in the form of an if/then statement), the results aiming to support or contradict a theory. The scientific method is decomposed into 7 main steps: (i) Research Question / Problem; (ii) Background / Observation; (iii) Formulate Hypothesis; (iv) Design Experiment; (v) Test Hypothesis / Collect Data; (vi) Interpret / Analyse Results; and (vii) Publish Findings.

For illustrative purposes, a description of the instantiation of the scientific method (including steps iv, v and vi) addressing the objectives of this thesis are detailed as follows:

- Design Experiment: The design of the experiment is conducted in order to test a hypothesis, about how a particular process or phenomenon works. The experiment conducted in this body of research and reported in this thesis relies solely on observations of the variables of the system under study, rather than manipulation of just one or a few variables as occurs in other types of experiments such as controlled experiments. The design experiment step incorporates the development of a conceptual framework and system architecture. Herein is developed a proof of concept and a validation scenario illustrated under chapter 4 and chapter 5 "The Semantic Enrichment Conceptual Model" and "Proof of Concept – Design and Implementation".
- Test Hypothesis / Collect Data: Setting up and testing hypotheses is an essential part of the scientific method. Using the experimental design, data can be obtained that will enable conclusions to be drawn. In this step, the proof of concept is implemented and applied into a validation scenario in order to collect data. The testing of the hypothesis is presented in chapter 6 "Evaluation and Analysis".
- Interpret / Analyse Results: Best efforts are made to develop a precise hypothesis to encapsulate the research question, as well as to prepare an appropriate experimental approach to obtain reliable and unbiased data, conduct robust analysis and testing, and culminate in a proper, truthful conclusion. Here, one can decide to: (i) accept the scientific hypothesis, or (ii) reject the hypothesis, depending on the evidence of the experimentation. Even if the hypothesis is incorrect, maybe the experiment had a flaw in its design or implementation; therefore it might lead to a further cycle of research and refinement of the

process. The analysis of the results and final considerations are presented in chapter 7 “Conclusions”.

1.3 Context of the work

The domain context of the work reported in this thesis takes into account a collection of relevant knowledge sources for the Building & Construction sector which are selected and stored in a knowledge base repository. For this case study, they were selected from the ICONDA database, provided by Fraunhofer IRB, which is a large database of technical documents (e.g. reports and papers) related to B&C matters. Such knowledge sources comprise a knowledge repository which the semantic enrichment process works upon.

From an application scenario point of view, it could be stated that the approach proposed here can be applied into an engineering project environment, where collaboration between different disciplines and project teams is the norm. Knowledge creation in engineering projects cannot be analysed in isolation, but rather from a joint perspective between teams of professionals working together to reach the same goal – Project Success. This means that, the success of collaboration in an engineering project, relies on capitalising on existing knowledge across the project team in order to find solutions to problems that are faced. Some examples of closely related research streams in recent years are: the extensive work on knowledge models and knowledge management tools, the rise of so-called knowledge engineering, the myriad of projects around 'controlled vocabularies' (such as ontologies, taxonomies, dictionaries, and thesauri), and the academic knowledge-centred courses (graduation, master, and doctoral).

From an European research perspective, this thesis has re-used as base material some of the scientific results achieved by research projects. The EU CoSpaces project provided contributions to the application scenario (the proof-of-concept) explored by this thesis, specifying a collaborative workspace as a composition of a set of checkpoint decisional gates where issues related to design optimization and risk analysis are taken into account. Each decisional gate is a point where all relevant parties and interests in the collaboration process agree on an approach to problem solving, supported as necessary by inputs from discipline experts. The approach acts as an application scenario to build upon by adding the semantic enrichment dimension to the knowledge collaboration process. The ideas pursued by CoSpaces project establish the practical context of this work.

From a more technological point of view, the EU e-Cognos research project provided a source of inspiration to understand which approaches and methods could be worth a special focus and be improved tackle the domain of work in the scope of this thesis. In this respect, e-Cognos provided insights to important areas: (i) a method to design and develop a domain Ontology with inputs from knowledge experts, which is an important task within the entire process; (ii) the semantic basis for a domain Ontology for the B&C sector and (iii) and some initial considerations for setting-up knowledge representations.

Other relevant initiatives and European projects will also contribute to and impact the context of work to be developed. It is important mention the achievements by the bsDD buildingSMART, formerly known as International Framework for Dictionaries (IFD), which is an Ontology framework for building and construction related Ontologies and dictionaries. Also to be mentioned are the initial initiatives on AEC classification systems, such as the: BS6100, Master Format, and UniClass and OmniClass. The FUNSIEC project reviewed numerous European semantic resources and assembled them into a virtual educational 'Experience Centre' and also conducted a feasibility study into the production of an 'Open Semantic Infrastructure for the European Construction Sector'. Finally, the important eConstruct project, which aimed to support the creation, publication and use of electronic catalogues of construction products.

1.4 Thesis Outline

This thesis is organized in the following chapters:

- Chapter 1: gives an introduction about the relevance of having knowledge representations as a way to facilitate knowledge creation and sharing among users within collaborative engineering environments. It presents the problem statement to be addressed by the thesis. It presents the research method used and the research question and hypothesis which will conduct the scientific work.
- Chapter 2: describes the philosophical underpinnings of knowledge evolution and how controlled vocabularies within contribute to such knowledge evolution. Several initiatives regarding the adoption of controlled vocabularies are presented. This chapter concludes with some trends and future directions for research on controlled vocabularies.
- Chapter 3: gives an historical overview of information retrieval. It describes several techniques for document classification and text cleansing operations. The vector space model method is described in detail. The most relevant classical measures for evaluating information retrieval are also presented. This chapter concludes by presenting a clustering algorithm used for hypothesis evaluation.
- Chapter 4: aims at describing a conceptual approach for semantic enrichment of knowledge sources. An overview of the knowledge sources is presented also with the main actors involved in the process of semantic enrichment. The method for formalizing the semantic referential is detailed. The step-wise approach for enrichment is presented with some examples of each enrichment step.
- Chapter 5: brings in the description of the proof of concept design and its implementation. The proof of concept is instantiated in a software platform named SENSE (*Semantic Enrichment kNowledge SourcEs*). SENSE was conceived using UML notation, where several views are used (functional, architectural and behavioural). A data model is described using an entity-relation diagram. An implementation section is included, presenting an architectural view of SENSE and the technologies used for implementing it are detailed.
- Chapter 6: describes the method used for evaluating the SENSE platform; in other words, to assess into what extent semantic enrichment can bring improvements to representation of knowledge sources when compared to traditional statistical approaches. This chapter starts by introducing the data set used to perform the assessment, followed by the evaluation process including the techniques used for evaluation. Next, techniques for data transformation and cleaning to deal with some inconsistencies in raw data are presented. Finally, the results and analysis of the initial hypothesis validation complete the chapter.

- Chapter 7: summarises the contributions of this PhD research on the field of semantic enrichment of unstructured information, with the support of external knowledge available in domain Ontologies. It gives an overview of the proposed work, presents the thesis outcomes and discusses important future work.

An important remark concerning the assessment of this thesis, relates to the scientific published papers arising from the work, which is *per se* an essential mechanism for evaluating the work carried out here.

Table 1.1. List of scientific publications

1	Ruben Costa, Celson Lima, Intelligent Systems in Accounting, Finance and Management, Wiley	Paper Journal	Submitted, waiting for decision
2	Ruben Costa, Celson Lima, Ricardo Gonçalves, *"Classification of Knowledge Sources Using Vector Space Model Supported by Domain Ontologies –AEC Case Study"*, Automation in Construction, Elsevier	Paper Journal	Submitted, waiting for decision
3	Ruben Costa, Celson Lima, *"Knowledge Representations With Ontology Support for Collaborative Engineering in AEC"*, Journal of Information Technology in Construction: ITcon	Paper Journal	Published
4	Ruben Costa, Celson Lima, João Sarraipa, Ricardo Jardim-Gonçalves. *"Semantic enrichment of building and construction knowledge sources using a domain ontology for classification"*, in 11th international conference of numerical analysis and applied mathematics 2013: icnaam 2013, pp 1381-1384, 20013	Conference Proceedings	Published
5	Ruben Costa ,Paulo Figueiras,Pedro Maló, Celson Lima. "*Classification of Knowledge Representations using an Ontology-based Approach",* International Conference on Knowledge Engineering and Ontology Development: KEOD2013, pp184-191, 2013	Conference Proceedings	Published
6	Ruben Costa, Celson Lima, João Sarraipa, Ricardo Jardim-Gonçalves. *"Facilitating knowledge sharing and reuse in building and construction domain: an ontology-based approach"*, Journal of Intelligent Manufacturing, pp 1-20, Springer, 2013	Paper Journal	Published

7	Luis Paiva, Ruben Costa, Paulo Figueiras, Celson Lima, *"Discovering Semantic Relations from Unstructured Data for Ontology Enrichment - Association rules based approach"*, 8ª Conferência Ibérica de Sistemas e Tecnologias de Informação: CISTI'2013, pp 579-584, 2013	Conference Proceedings	Published
8	Ruben Costa, Celson Lima, *"An Architecture to Support Semantic Enrichment of Knowledge Sources in Collaborative Engineering Projects"*, Knowledge Discovery, Knowledge Engineering and Knowledge Management, pp 276-289, Springer, 2013	Book Chapter	Published
9	Ruben Costa, Paulo Figueiras, Pedro Maló, Celson Lima, *"Representação Semântica de conhecimento no setor da construção"*, Simpósio de Informática e Geotecnologia de Santarém, 2012	Conference Proceedings	Published (best paper award)
10	Ruben Costa, Paulo Figueiras, Luis Paiva, Ricardo Jardim-Gonçalves, Celson Lima, *"Capturing Knowledge Representations Using Semantic Relationships An Ontology-based Approach"*, Sixth International Conference on Advances in Semantic Processing: SEMAPRO 2012, pp 75-81, 2012	Conference Proceedings	Published
11	Paulo Figueiras, Ruben Costa, Luis Paiva, Ricardo Jardim-Gonçalves, Celson Lima, *"Information Retrieval in Collaborative Engineering Projects-A Vector Space Model Approach"*, International Conference on Knowledge Engineering and Ontology Development: KEOD2012, pp 233-238, 2012	Conference Proceedings	Published
12	Ruben Costa, Celson Lima, *"An Approach for Indexation, Classification and Retrieval of Knowledge Sources in Collaborative Environments"*, Fifth International Conference on Advances in Semantic Processing: SEMAPRO 2011, pp 14-20, 2011	Conference Proceedings	Published
13	Celson Lima, Ruben Costa, Pedro Maló, João Antunes, *"A knowledge-based approach to support decision making process in project-oriented collaboration"*, 11th European Conference on Knowledge Management: ECKM 2010, pp	Conference Proceedings	Published

	614-622, 2010		
14	Celson Lima, Paulo Figueiras, Ruben Costa, *"A Knowledge Engineering Approach Supporting Collaborative Working Environments Based on Semantic Services"*, International Conference on Knowledge Engineering and Ontology Development: KEOD2010, pp 123-132, 2010	Conference Proceedings	Published
15	Ruben Costa, Celson Lima, João Antunes, Paulo Figueiras, Vitor Parada, *"Knowledge Management Capabilities Supporting Collaborative Working Environments in a Project Oriented Context"*, European Conference on Intellectual Capital: ECIC 2010, pp 208-216, 2010	Conference Proceedings	Published
16	Celson Lima, Pedro Maló, Ruben Costa, *"Knowledge support for collaborative workspaces: the cospaces approach"*, 5th Conference on Information and Knowledge Management in Building: CIB-w102, pp 59-71, 2009	Conference Proceedings	Published
17	Ruben Costa, Pedro Maló, Colin Piddington, Gilles Gautier, *"Knowledge Enabled Collaborative Engineering in AEC"*, European Conference on Product and Process Modelling: ECPPM 2008, 2008	Conference Proceedings	Published

2

Knowledge in Context

"Knowledge has to be improved, challenged, and increased constantly, or it vanishes."

- Peter F. Drucker (1909 – 2005), Management consultant, educator and author

Knowledge resides inside people's heads, surely, but also in books, electronic media, regulatory documents and many other distributable, human-readable resources. One of today's problems is that we have too many knowledge sources and too little time to browse them! Assistance is needed.

Researchers working with practitioners believe that knowledge can be externalised, captured, formalised, represented and mined to provide targeted, tailored information. This thesis strongly supports this viewpoint and seeks to extend and refine techniques drawing on semantic methods.

Knowledge is the very focus of this thesis and the nature of knowledge needs to be analysed in detail, through basic questions, like: (i) what is knowledge?; (ii) how can it be represented?; (iii) what are the different types or knowledge?; (iv) how can knowledge be classified?, just to put down a short list. It is also a valuable starting point to take an engineering related view of knowledge (as an example), covering tools and knowledge-related applications and initiatives. Special attention is given to the Building & Construction sector due to its usage in the assessment process in this work. However, the approach is in fact generic and would be appropriate to knowledge mining in other fields such as pharmaceutical research and regulatory frameworks.

2.1 Concepts and Definitions

Knowledge is a broad and abstract notion that has generated epistemological debates in Western philosophy since the classical Greek era. In a broad sense, knowledge is information possessed in the mind of an individual; it is personalized or subjective information related to facts, procedures, concepts, interpretations, ideas, observations and judgments (which may or may not be unique, useful, accurate, or structured). Knowledge can refer to physical skills and competencies (e.g., playing tennis or doing carpentry), cognitive/intellectual activity (e.g., problem solving), or both (e.g., surgery which involves both manual skills as well as cognitive elements of human anatomy and medicine) (Holsapple 2003).

From a more engineering-oriented perspective, a comprehensive definition has been given by Alavi and Leidner (Alavi e Leidner 1999), which defined knowledge as: “A fluid mix of framed experience, values, contextual information and expert insight, which provides a framework for evaluating and incorporating new experiences and information”. Knowledge originates only in the mind of knowledge holders, and may be embodied in documents, repositories, organisational routines, processes, practices and norms.

One way to better understand the concept of knowledge is to make the distinction between Data, Information, Knowledge, and Wisdom, which are often confused and overlapping terms, although they embed different meanings. This distinction is referred to as the DIKW (Data, Information, Knowledge, and Wisdom) hierarchy, knowledge hierarchy or the knowledge pyramid (Figure 2.1). The implicit assumption is that data can be used to create information, information can be used to create knowledge, and finally that wisdom is built on knowledge (Rowley 2007). The visual metaphor of the pyramid depicts the fact that, usually, large amounts of data are distilled to a smaller quantity of information. Then, a still rather large amount of information is further distilled to a more limited knowledge (Hey 2004).

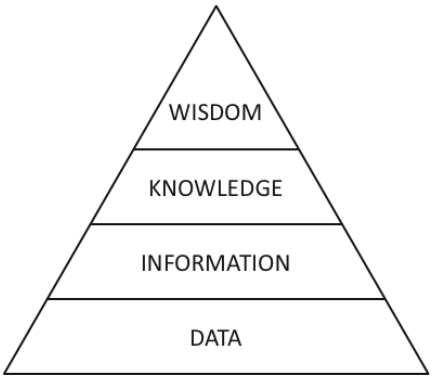


Figure 2.1. The DIKW hierarchy, also known as the knowledge pyramid

In the knowledge management domain, something along the lines of a DIKW hierarchy was alluded to by Zeleny (Zeleny 1987). Also, Ackoff (Ackoff 1989) proposed a similar categorization, where the concepts are defined as follows:

- **Data** represents raw facts and symbols that have no significance or meaning by themselves. Data is unorganised and unprocessed, and has no relation to anything else (Ackoff, 1989; Bellinger, Castro and Mills, 2004). Data deals with the past (Ackoff, 1989) and comes through research, creation, gathering, and discovery. Data is also sometimes defined as unprocessed information (Hey, 2004). When encountering a piece of data, the first action is usually to attempt to find a way to attribute meaning to it.
- **Information** is data that has been processed and given meaning, by relating it and organising it, so that it is useful (Ackoff, 1989; Bellinger, Castro and Mills, 2004; Hey, 2004). Information can be considered as an aggregation of processed data which relates to the What, Who, When, and Where dimensions (Ackoff 1989). Information has context. When information is created from data, sense is made out of the data (Hey 2004). Information has a tendency to be relatively static in time and linear in nature and, while it entails an understanding of the relations between data, it generally does not explain the reason why the data is what it is, nor it gives an indication on how the data is likely to change over time. Analogously to, information is about the past, about what has been (Ackoff 1989).
- **Knowledge** is created when data and information are organised (Hey 2004), accumulated (Nonaka e Takeuchi 1995), and integrated, so that a pattern appears, which describes both a reoccurring problem and the core of solution to such problem. Knowledge is also constructed from achieving an added context and understanding. Knowledge is understood to be personal (Hey 2004), residing in the heads of the people, and is built up from scratch by the learner through utilising his/her experience. The personal aspect also makes it hard to transfer from one person to another. Knowledge comprises strategies, practices, methods and approaches relating to the issue of 'how', knowing how to do something. Data and information are about looking at the past, but with knowledge representing a pattern in the data and information, it is possible to interpolate in this pattern and to deal with the present. Knowledge is basically the first step in the ladder of decisions can be taken with some predictability and accuracy.
- **Wisdom** is the highest level of understanding and it arises when people understand the fundamental principles (Bellinger, Castro e Mills 2004) for the patterns representing the knowledge - why it is what it is. Wisdom represents the highest level of abstraction and it tends to create its own context, embodying principle, insight, moral, or archetype, dealing with what is right and wrong, good and bad (Ackoff 1989). Wisdom is about understanding 'why', knowing why things are the way they are. Wisdom is extrapolative and deals with the future (Ackoff 1989), rather than the just the past and present, as is the case with the lower

levels of the pyramid. With wisdom, you can create a vision of the future and make "educated" forecasts of impacts of future decisions and actions (Ahsan e Shah 2006). A person who exhibits wisdom is knowledgeable, has a longer perspective, is aware of context, is flexible and can change perspective, recognises uncertainty and the limits to the knowledge, and is prepared to be tentative and flexible about solutions (Rowley 2007). This level of the pyramid is purely a human quality and cannot be represented in a computer. Ahsan and Shah (Ahsan e Shah 2006) stress the interrelations between the different levels in the following way: 'Data' is the basic unit of 'information', which in turn is the basic unit of 'knowledge', which itself is the basic unit of 'wisdom'. The whole purpose in collecting data, information and knowledge is to be able to make wise decisions. If the data sources are flawed then in most cases the decisions will be flawed. However, good data sources do not provide any guarantee that decisions will be good!

2.1 Knowledge Cycle

Since the main focus of this thesis relates to aspects regarding knowledge creation and sharing, it is important to mention what kind of methodologies and framework have been proposed by the wider scientific community.

There are several different typologies and classifications that aim to clarify what knowledge 'is' and how it can be created and shared. Ryle (Ryle 1984) observes a distinction between know-what (i.e. facts) and know-how (i.e. skills), where the latter concept can be seen as "…the particular ability to put know-what into practice". Brown and Duguid (Brown and Duguid 2000) and Quinn, Anderson and Finkelstein (Quinn, Anderson and Finkelstein 1996) note that cognitive knowledge (i.e. know-what) is the basic mastery of a discipline that can be achieved through extensive training and certification, that advanced skills (i.e. know-how) is the ability to apply the rules of a discipline to complex real-world problems, and that systems understanding (i.e. know-why) is the deep knowledge of the web of cause-and effect relationships underlying a discipline. Similarly, (Polanyi 1966) uses the concepts of explicit and tacit knowledge to make a distinction between knowledge that can easily be described in documents and knowledge that humans normally have difficulties articulating and codifying but still are able to express in action.

- **Tacit knowledge** was first mentioned by Polanyi (1962, 1964, 1966) as knowledge that is hard to encode and communicate. Another term that Polanyi uses is 'personal knowledge', which stresses the fact that tacit knowledge is highly personal and deeply rooted in the person that holds it. Tacit knowledge is rooted in action, procedures, routines, commitment, ideals, values, emotions and involvement in a specific context (Nonaka, 1994; Nonaka, Toyama and Konno, 2000). This makes it hard to formalise and communicate (Nonaka, 1994; Nonaka and Takeuchi, 1995). Tacit knowledge includes subjective insights, intuition and hunches. Tacit knowledge is created from people's own experiences and is difficult to understand and imitate (Jasimuddin, Klein and Connell 2005). Tacit knowledge is

ambiguous in nature, which makes duplication difficult. There is therefore also a risk of losing the knowledge due to loss of employees (Jasimuddin, Klein and Connell 2005). Transfer of tacit knowledge is best achieved by using shared experiences, such as spending time together in the same environment (Nonaka, Toyama and Konno 2000), for instance apprenticeships, since it relates to personal skills (Nonaka, Toyama and Konno, 2000; Wyatt, 2001). In these apprenticeships, apprentices learn the tacit knowledge needed for their craft through hands on experience (Snowden 2003), under guidance by experts. Tacit knowledge forms the basis of explicit knowledge (Jasimuddin, Klein and Connell 2005), but is often understood to compromise the main body of knowledge. Polanyi proposes that "we know more than we can tell" (Polanyi 1966).

- **Explicit knowledge**, or codified knowledge, is the knowledge that is transmittable in formal, systematic language (Polanyi, 1966; Nonaka, 1994; Hey, 2004). This knowledge includes facts, rules, relationships, and policies that can be codified and shared without need for discussion (Wyatt 2001). Since explicit knowledge is codified it is easy to communicate and store. The risk of losing explicit knowledge due to employee turnover is quite small (Jasimuddin, Klein and Connell 2005). Hey (Hey 2004) relate to tacit and explicit knowledge with the metaphor of viscosity, depicting that explicit knowledge is "sticky" and that tacit knowledge is "leaky" and hard to retain.

Tacit and explicit knowledge are closely related, where one of the main targets of knowledge management is to formalise and codify tacit knowledge, so that it becomes explicit and can be stored in knowledge repositories or other types of databases. Drivers for this process is the knowledge's accessibility (Jasimuddin, Klein and Connell 2005), where explicit knowledge is accessible by everyone as opposed to just available to the owners, and in the fact that explicit knowledge stays in the company even though there is employee turnover.

Further, Jasimuddin, Klein and Connell propose that the tacit and explicit parts are inseparable, and complimentary to each other, where all knowledge has both tacit and explicit parts. Their relationship can be likened to an iceberg, where the visible, explicit part is supported and given meaning by the hidden tacit part. Nonaka, Toyama, and Konno state that "to understand the true nature of knowledge and knowledge creation, we need to recognise that tacit and explicit knowledge are complimentary", where new knowledge is created through interactions between tacit and explicit knowledge.

Nonaka, Toyama, and Konno suggest that the knowledge creation process consists of three elements namely: (i) the SECI (socialisation, externalisation, combination, and internalisation) model; (ii) the concept of *ba* (shared context); and (iii) knowledge assets (inputs, outputs and moderator of the knowledge creation process). Using these main elements, managers can create conditions to lead the dynamic knowledge creation process and provide a knowledge vision. The SECI model describes the process of converting and recreating explicit and tacit

knowledge. The SECI model is depicted in the form of a spiral (Figure 2.2), since the interaction between explicit and tacit knowledge amplifies knowledge creation, i.e., the SECI model tries to illustrate that knowledge held by individuals is shared with other individuals so it interconnects to a new knowledge. The spiral of knowledge, or the amount of knowledge so to say, grows all the time when more rounds are done in the model. Socialisation is creating new tacit knowledge from existing tacit knowledge through shared experiences. Externalisation is about articulating tacit knowledge into explicit. Combination depicts converting explicit knowledge into more complex and systematic sets of knowledge. Internalisation is “learning by doing” by embodying the explicit into tacit.

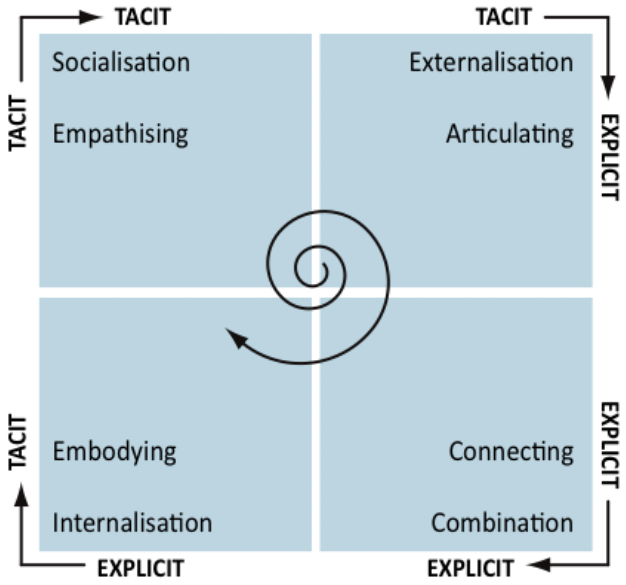


Figure 2.2. The SECI model, depicting modes of knowledge creation and conversion (Nonaka e Takeuchi 1995)

The knowledge creation process proposed by Nonaka and Takeuchi (Nonaka e Takeuchi 1995) seems adequate to adopt into a collaborative engineering environment, where knowledge is: (i) transformed in an evolving way along the time; (ii) managed around problems and solutions in order to be proper capitalised; (iii) better capitalised with the appropriate support of reasoning mechanisms; and (iv) supported by a set of ontology-enabled services to increase the semantic level of knowledge sources.

Within the scope of this thesis, it could be stated that, ontology-enabled services are key mechanisms for enabling the knowledge creation process and therefore it deserves an important focus within this chapter.

2.2 Controlled Vocabularies (Ontologies & Taxonomies)

2.2.1 Definitions

Various definitions of what constitutes an ontology have been formulated and have evolved over time. A good description of these can be found in (Corcho, Fernandez-Lopez and Gomez-Perez 2003). From the authors' perspective, the best definition that captures the essence of an ontology is the one given by Gruber (Gruber 1993): "an ontology is a formal, explicit specification of a shared conceptualization". As elaborated in (Studer, Benjamins and Fensel 1998):

Conceptualization refers to an abstract model of some phenomenon in the world which identifies the relevant concepts of that phenomenon. *Explicit* means that the types of concepts used and the constraints on their use are explicitly defined. *Formal* refers to the fact that the ontology should be machine processable.

People often find it difficult to see clearly how an ''ontology'' differs from what they already recognize as a ''data model'', focusing on the formal nature and structuring mechanisms that seem to be characteristic of both. Certainly, data modelling languages provide the ability to define taxonomies through notations that support classification, generalization and specialization, they support the definition of relationships or associations between concepts, and ideas of aggregation and composition, and in terms of these primitives appear to offer the same support for representing concepts and the relationships between them. However, it should be stated that trying to understand the distinctions in terms of the modelling primitives that are used is a mistake; it is the nature of the models themselves, the way in which they are derived, and the tools that support their use that provides the differentiation. In order to understand this, it is necessary to return to the underlying problems that make it difficult to achieve a single agreed data model for an industry.

Back to Gruber's definition, a key element is the idea of a shared conceptualization (Gruber 1993). Typically, in human endeavour, shared conceptualizations are defined over a lengthy period of time, based on the shared experience of a group of people, sometimes referred to as a community of practice (Wenger e Snyder 2000). They will involve the definition and use of abstractions that are designed to capture the important aspects of some practical context in order to support a particular activity or type of activity. As such, a shared conceptualization is a socially constructed model or reality that is distinct from reality and is optimized to support the goals and activities of the community of practice in which it was defined. Communities engaged in different activities are likely to form shared conceptualizations that are quite different views of reality, and make up shared ''world-views'' (Checkland and Scholes 2000) that provide a basis for highly effective and efficient communications within the respective communities.

In order to understand and formalize the shared worldviews of such communities in the form of ontologies to support the integration of diverse human activities, it is important to consider approaches that derive from an interpretive philosophical standpoint rather than from a positivist, scientific/engineering one (Fitzgerald and Howcroft 1998). In such an approach, it is important to interpret, accommodate, and model what is, rather than trying to change reality to fit a single model. This inevitably results in different ontologies for different communities, but the challenge then is to find ways to allow those communities to collaborate effectively with one another whilst maintaining their existing, efficient, effective separate worldviews. The implication is that the emphasis must be shifted from developing a standard representation of a single ‘‘reality’’, towards providing mechanisms for supporting communication between differing perceptions of reality, focusing our attention on the overlaps at the boundaries and the specific conceptualizations that are required for such communication to happen.

2.2.2 Ontological Structures

Ontologies, having transcended the domain of philosophy, are currently referred as part of many activities from different domains, such as Artificial Intelligence, Knowledge Management solutions or e-commerce/e-business-related processes. Ontologies are very often considered key elements integrated into catalogues, semantic-oriented databases, web-based documents, and so on.

From a general point of view, it could be stated that an ontology is required and (likely) used when talking about precise meaning of things (such as terms, expressions, or products). For instance, when indexing/retrieving documents, ontologies can provide richer indexes linking different documents through terms that are not found within these documents; rather they are "mapped" through the ontology. In other words, the ontology gives access to the knowledge that is implicitly found within the documents. A keyword-based mechanism would never be able to do that. An ontology-based query is not ambiguous as the queries in natural language can be. It is also needed a semantic resource (ontology/taxonomy) when doing e-procurement and searching for products using a very precise technical specification. Unless the technical attributes are available to the search mechanism, this is not possible.

An ontology is required when there is a need to communicate/exchange (transfer and/or share) various sorts of information where the meaning is fundamental. Ontology is also useful when reuse of existing knowledge is required. From a non-exhaustive list of uses, ontology can be used for simple kinds of consistency checking, interoperability support, validation and verification testing, configuration support, help to perform structured, comparative and customised search as well as to exploit generalisation/specialisation of information (McGuinness 2003).

According to McGuiness, one of the simplest notions of a possible ontology is a vocabulary with a finite list of terms. Another potential ontology specification for modelling an ontology starts with the enumeration of relevant concepts that are useful to describe it. Each concept will be labelled with a unique identifier. In order to facilitate the comprehension of the meaning of each concept by human beings, this identifier can be based on a combination of everyday words. This set of identifiers represents a **Vocabulary**. A definition (for instance, in natural language) is attached to each identifier and this produces a dictionary or a glossary.

Identifying and naming the relevant concepts in a given domain is a complex exercise. A good way to proceed is to classify these concepts into a hierarchical structure, creating a **Classification**. This hierarchy, which is actually a tree structure, must enable a multi-inheritance mechanism in order to allow the expression of a multi-dimension space in a 2D diagram.

If the way to classify is based on the use of the relation "**is a**" (for instance the concept "person" "**is a**" "human being"), the tree produced as the result of such a classification is called a **Taxonomy** which is than a special way of classifying things.

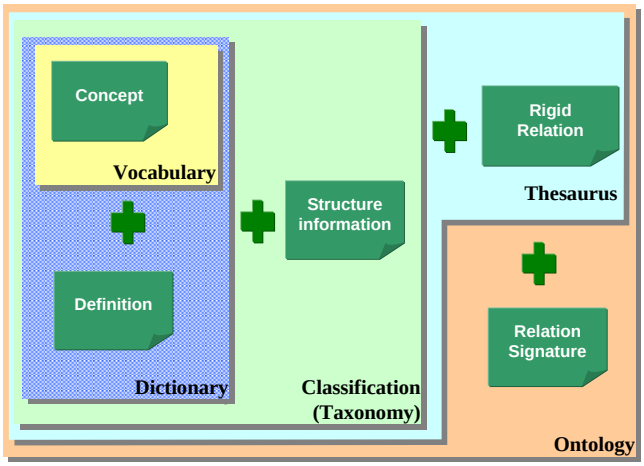


Figure 2.3. Ontology modelling concept (Barresi, et al. 2005)

The use of the unique relation "**is a**" is not enough to model a complex system. Extra relations exist "de facto" between concepts even if these concepts are not closely defined in the taxonomy tree. This leads to the definition of a different structure (more complex than a tree) to express these semantic relations. One can consider that the "**is a**" relation is a semantic relation but to keep things simple, this particular relation will only be called a hierarchical relation. These semantic relations enable the expression / representation of a domain specific knowledge. A relation, called a **Signature**, may bind only two concepts. The notion of signature is very important. It allows the liaison of each concept with any other existing concept within the ontology. The liaison of concepts can be done freely in order to really stick with the domain being represented by the ontology.

A **Thesaurus** can be viewed as a subset of an ontology, where the whole structure (hierarchical and semantic relations) is rigidly defined. The first consequence is that the semantic content in the thesaurus is not so rich because of this rigid structure of relations applicable to the concepts. Only high level relations such as the notions of close or far neighbourhood can be represented. The addition of a specific relation to link two given concepts is not allowed.

The word **Ontology** has been used to refer to all of the above things. When used in the Knowledge Representation community, it tends to refer to things that have a rich and formal logic-based language for specifying meaning of the terms. Both a thesaurus and a taxonomy can be seen as having a simple language that could be given a grammar, although this is not normally done. Usually they are not formal, in the sense that there is no formal semantics given for the language. However, one can create a model in UML and a model in some formal ontology language and they can have identical meaning. It is thus not useful to say one is an ontology and the other is not because one lacks formal semantics. The truth is there is a fuzzy line connecting these things.

The bottom line is, taxonomies and thesauri may relate terms in a controlled vocabulary via parent-child and associative relationships, but do not contain explicit grammar rules to constrain how to use controlled vocabulary terms to express (model) something meaningful within a domain of interest.

2.2.3 Language & Representations

Despite the fact that there is a long list of languages used to create and represent ontologies, this section provides a summarised list of the languages/representation formats that are considered more standard-oriented (Table 2.1), by their very nature (i.e. promoted by an standardisation body) or by their acceptance and usage by the research community.

Table 2.1. Ontologies Languages (Lima 2004)

Language	Description	URL
DAML+OIL	DAML+OIL is a semantic markup language for Web resources. It builds on earlier W3C standards such as RDF and RDF Schema, and extends these languages with richer modelling primitives. DAML+OIL provides modelling primitives commonly found in frame-based languages. It is important to emphasise that this language was the basis of OWL.	http://www.w3.org/TR/daml+oil-reference
EXPRESS / EXPRESS-G	EXPRESS-G is a standard graphical notation for information models. It is a useful companion to the	http://www.steptools.com/support/stde

	EXPRESS language for displaying entity and type definitions, relationships and cardinality. Used by the ISO DIS 12006-3.	v_docs/devtools/devtools-8.html
OIL	OILS stands for Ontology Inference Layer, a language that was developed in the context of the European IST Ontoknowledge project. It is built on top of RDF(S), using as much as possible RDF(S) constructs in order to maintain backward compatibility.	http://xml.coverpages.org/OIL-ecai00.pdf
OWL	The OWL Web Ontology Language is designed for use by applications that need to process the content of information instead of just presenting information to humans. OWL facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema (RDF-S) by providing additional vocabulary along with a formal semantics.	http://www.w3.org/TR/owl-features/
RDF(S)	Resource Description Framework (RDF) defines a language for describing relationships among Web resources in terms of named properties and values. It is particularly intended for representing metadata about Web resources, such as the title, author, copyright and licensing information about a Web document, or the availability schedule for some shared resource.	http://www.w3.org/TR/rdf-schema/
XML	Extensible Markup Language (XML) is a simple, very flexible text format derived from SGML. Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere. XML has been largely used to represent "semantics" in the Web, here including taxonomies, classification systems, etc..	http://www.w3.org/XML/
Topic Maps	Topic Maps (ISO/IEC 13250) define a model for the semantic structuring of knowledge networks and are a solution for organising and accessing large and continuously growing information pools. They provide a 'bridge' between the domains of knowledge	http://www.topicmap.com/ http://www.topicmaps.org

	management and information management. They can also be used to generate navigation for a website, and lots of other metadata tasks. A topic map is a collection of topics (a topic is a resource that acts as a proxy for some subject; the topic map system's representation of that subject), associations, and scopes that may exist in one of two forms: (**i**) a serialized interchange format (e.g. as a topic map document expressed in XTM syntax); or (**ii**) Some application-internal form, as constrained by the XTM (XML Topic Maps) Processing Requirements. A topic in a topic Map represents a subject inside the computer.	
KIF	Knowledge Interchange Format (KIF) is a language designed for use in the interchange of knowledge among disparate computer systems. KIF, a particular logic language, has been proposed as a standard to use to describe things within computer systems, e.g. expert systems, databases, intelligent agents, etc.. Moreover, it was specifically designed to make it useful as an "interlingua". This means a language useful as a mediator in the translation of other languages. KIF has declarative semantics; it is logically comprehensive (i.e. it provides for the expression of arbitrary sentences in the first-order predicate calculus); it provides for the representation of knowledge about the representation of knowledge; it provides for the representation of non-monotonic reasoning rules; and it provides for the definition of objects, functions, and relations. When the computer system needs to communicate with another computer system, it maps its internal data structures into KIF. KIF is a programmer-readable language and thereby facilitates the independent development of knowledge-manipulation programs.	http://logic.stanford.edu/kif/kif.html

2.3 Knowledge in Building & Construction

This section presents the adoption/development of European Ontologies focused on the Construction sector. It discusses first the philosophical underpinnings of product data and ontology, and afterwards analyses the state of the art about the development of Construction-related semantic resources (e.g. ontologies, taxonomies, dictionaries). It presents an overview of the several initiatives on controlled vocabularies available for the development or adoption of new ontologies in the construction sector. The purpose for identifying several initiatives within the B&C sector is related with the fact that, the evaluation of this thesis will be relying on the usage of B&C knowledge sources.

In the last decades, the development of Controlled Vocabularies such as dictionaries, classifications, taxonomies, and of course the scary and "appealing" ontologies, has been the focus of many research projects in Europe. A non-exhaustive list of well-known efforts in this area is the following: ISO12006 parts 2 and 3, LexiCon (the Netherlands), Barbi (Norway), bcBuildingDefinitions taxonomy (Lima, Stephens and Böhms 2003), ICONDA terminology (IRB, ICONDA®Bibliographic 1986), BS6100 and UNICLASS (British Standards), e-COGNOS ontology (El-Diraby, Lima and Fiès 2005), Standard Dictionary for Construction in France (SDC) and the International Framework for Dictionaries (IFD). It is worth recalling that in other continents similar efforts were also conducted, such as the SI/SfB, Masterformat, Omniclass, and the Canadian Thesaurus, just to name a few. Many others European projects (research-oriented, standards-biased, etc.) were performed. A brief list includes: CEN/ISSS eConstruction series of Workshops (Böhms, et al. 2004), FUNSIEC (Lima, Silva, et al. 2005), CONNIE (Cerovsek, Gudnason and Lima 2006), and SEAMLESS (Lima, Bonfatti, et al. 2006) projects.

A quick analysis on the above listed projects/initiatives allows to imagine how much effort has been devoted to this area around the world, (likely) guided by one single aim: to put the Construction sector firmly on the front line considering the latest advances of semantic-related ICT resources. Preliminary thoughts were about developing useful e-Commerce/e-Business related tools and resources helping construction companies to publish their own catalogues using their own languages and, at the same time, become actors in the *eConstruction* arena.

2.3.1 Major reasons behind the development of Controlled Vocabularies in Construction

In simple terms, vocabularies give names to things that have meaning at a certain level of detail. In this sense, vocabularies can be seen as a convenient mechanism for exchanging information. For example, "dog" means "a domestic carnivorous animal with four legs that typically has a long muzzle, pointed ears, a fur coat, a long fur-covered tail, and whose characteristic call is a bark". It is certainly different to "elephant" or "bicycle". So, if someone

says, “Where is my dog?” the kind of thing to look for is already known. But there are many dogs. It could be added adjectives like “small”, “long”, “short-legged”, “drooping eared”, “German” (which adjectives must have agreed meaning in the dog context) or it could be simply used another name “dachshund” or “sausage dog”. These need to have agreed meaning, not least because to the English or French the word dachshund is foreign and the other is a descriptive nick-name. The deeper the analysis goes with meaning to add detail or to differentiate, the more control there needs to be in the use of the language. Between specialists in one discipline there can be quite precise understanding of words (in this case zoologists who might even use Latin names) but between experts and non-experts and different kinds of expert there can be misunderstanding. To change to a construction example, what is the difference between a “brick pillar” and a short length of thick wall made from brick? A bricklayer and a cost estimator might use different terms. The answer (in UK at least) is that the difference is defined by rules related to the dimensions.

Vocabularies are important to conveying human thought in a concise way and with precision in a given working context. There must be as much preciseness as possible although in human exchanges, sometimes one can say that something is like something else e.g. the dog is like a dachshund but with longer legs. Questions can be asked in order to refine meaning and (perhaps finally) identify the breed of dog.

Controlled vocabularies are even more important to electronic information exchange in any form. Whilst humans can ask clarifying questions based on their experience and knowledge, computers do not have yet such as a general capability (though in limited contexts artificial intelligence may enable that). So there needs to be precision built into the language of computer communication used. There is much less possibility for confusion if an object is referred to by its catalogue/part number and as a buyer one uses that to describe its needs to the supplier, but not everything can be conveyed that simply. Architectural details, a building frame, and a plumbing system are usually designed to result in requirements that facilitate choice of components to satisfy the need. Therefore, generic types like wall and pump are then specialised according to several properties (such as dimensions, material, colour, and strength) which themselves must have precise (i.e. agreed) meanings. Although codes could be used to identify components and systems, it is far more convenient that the codes take the form of the names, humans use “Pump” not A254GHT7 unless when buying from a catalogue.

Taking the examples described previously, it could be stated that, a **controlled vocabulary** is a list of terms that have been enumerated explicitly. This list is controlled by and is available from a controlled vocabulary registration authority. All terms in a controlled vocabulary should have an unambiguous, non-redundant definition. This is a design goal that may not be true in practice. It depends on how strict the controlled vocabulary registration authority is regarding

registration of terms into a controlled vocabulary. At a minimum, the following two rules should be enforced:

- If the same term is commonly used to mean different concepts in different contexts, then its name is explicitly qualified to resolve this ambiguity.
- If multiple terms are used to mean the same thing, one of the terms is identified as the preferred term in the controlled vocabulary and the other terms are listed as synonyms or aliases.

2.3.2 Tools

There are different types of tools available to support the work related to ontologies, such as ontology development, ontology merge and integration, ontology querying, ontology demonstrators, and so on. Most of them are domain independent tools that could also be used to support the needs existing in the construction sector. For illustrative purposes only, the following are listed on Table 2.2:

Table 2.2. Ontology supporting tools

Domain Independent	
OilED	a graphical ontology editor to support the creation of DAM+OIL based ontology.
Protégé Editor	provides a graphical and interactive ontology-design and knowledge-base–development environment. Protégé disposes of an OWL plug-in that allows producing OWL compliant ontologies.
Jena	a Java API for manipulating RDF models. It includes a OWL API that provides support for loading and OWL ontologies into Jena RDF(S) models, for managing the ontology structure, as well as for writing ontologies in OWL format.
OntoEdit	a graphical based environment supporting the development and maintenance of ontologies.
B&C Related	
LexiCon Explorer	a browser and editor, showing the class structure and the class attributes of the LexiCon. The LexiCon is a vocabulary of terms of interest for the construction industry and as such an implementation of ISO DIS 12006-3.
The eConstruct	a set of bcXML-based tools supporting the creation and publishing of e-catalogues, the management of catalogue servers, the search of construction

Tools	products supported by query language taxonomy-based
The e-COGNOS Ontology Server	a Java based application supporting on one hand the management of an DAML+OIL ontology (creation, importing, etc.) and on the other hand the calculation of the respective ontological weights for keyword-based queries.
IFD Tools	The IFD/bSDD Online Browser is a basic browsing application for IFD Library/buildingSMART Data Dictionary. The IFD Library Propertylizer is a content management tool designed to handle input of new content and maintenance of existing content.

The CEN/ISSS Workshop on eConstruction has also made a benchmark regarding ontology software tools. The main conclusions from such report relate to the fact that most of the time, the frontiers between querying and inferring capabilities offered by languages and tools are a bit fuzzy. In general, inference is limited to recursive navigation through the ontology class/property hierarchies as well as of data paths involving transitive properties. The majority of the tools identified are based on open source licenses, confirming the willingness to share the tools and get volunteered support from developers.

In most of the cases, one can say that storage and query tools are (academic) prototypes implementing parts of the query language they aim to support, while they do not provide the necessary programming/administration facilities in order to make them really operational in a working environment. Moreover, exhaustive scalability and performance figures are not always available, which prevents having a better evaluation of the tools.

Taking into account the list of tools presented in Table 2.2, in terms of infrastructure support, Jena has proven to be a good option especially due to its compliancy with W3C recommendations. Jena is a free open source Java framework for building semantic web and Linked Data applications. It provides several APIs interacting together to process RDF and OWL data.

Protégé provides a graphical and interactive ontology-design and knowledge-base development environment, with several third party plug-ins. It is freely available and has thousands of users all over the world who use the system for many different projects and applications. It helps knowledge engineers and domain experts perform knowledge-management tasks. Ontology developers can access relevant information quickly whenever they need it, and can use direct manipulation to navigate and manage the ontology. The system is constructed in an open, modular fashion. Its component-based architecture enables system builders to add new functionality by creating appropriate plug-ins. Protégé is the recommended tool for many reasons: it is OWL-compliant, is a freeware tool, and it has a good base of developers supporting it around the world.

In terms of use/adoption in the construction sector, small ontologies can be created and managed using the e-COSer tool, a Web-based and open source tool produced by the e-COGNOS IST project. Richer, morecomplex ontologies can be produced using the LexiCon Explorer but with the constraint of not being exportable to other applications.

The IFD library was renamed to buildingSMART Data Dictionary (bsDD) in 2011. It provides and API enabling software developers to use buildingSMART Data Dictionary in their applications. It also provides an online browsing application for basic search filtering on context and language. Basic information (Fullname, Definition, Comments, Shortname, Relationships and Details) about any selected concept are visible. Relationships are visible in a tree view. The IFD Library Propertylizer provides a graphical relationship browser that is used to assign properties to specific materials, products, or equipment systematically.

The B&C related tools will be further described in the following sections.

2.3.3 Relevant initiatives on Controlled Vocabularies in B&C

As previously pointed-out, a large effort has been put regarding the creation and use of CVs around the world. This section briefly summarises a suite of relevant research projects, and pan-European & international initiatives (Figure 2.4) and provides some details about the most relevant ones. It is worth saying that this panorama is not presented as exhaustive or complete; rather, the idea here is to simply describe the importance of controlled vocabularies within the building and construction sector.

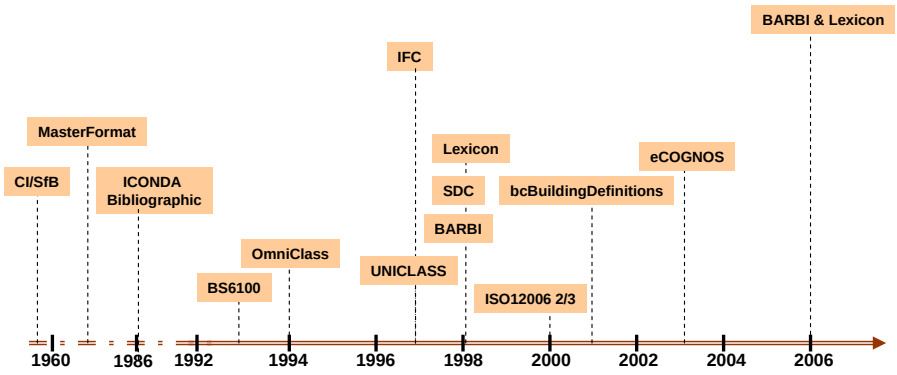


Figure 2.4. Some examples of CV-focused initiatives in Europe and worldwide (Lima, Zarli and Storer 2007)

Starting with the CI/SfB (Construction Index/Samarbetskommitten for Byggnadsfragor), a Scandinavian system of classification originally set up in 1959 and specially designed for the construction sector. It claimed to be in use worldwide for any technical and trade literature in the broad construction area. The CI/SfB was used in North America as the basis of the MasterFormat™, which is the specification-writing standard for most commercial building design and construction projects.

MasterFormat is a master list of numbers and titles intended for use in the organizing of specifications, and contracting and procurement requirements initially started with 16 divisions coded with 5 digits. In order to cope with the changes required by the modern Construction industry, in 2004 MasterFormat was heavily updated; new sections were added (the initial 16 were extended to 50) and the number codes are composed now of 8 digits (instead of the initial 5). MasterFormat targets the standardised communication of projects for all actors involved.

MasterFormat works together with Uniformat. Uniformat is an arrangement of construction information based on physical parts of a facility called systems and assemblies, aiming to: (i) achieve consistency in economic evaluation of projects; (ii) enhance reporting of design program information; and (iii) promote consistency in filing information for facility management, drawing details and construction market data. Masterformat tells what the construction item is, whilst Uniformat says where the construction item is.

In the beginning of the 90's the British Standard 6100 (BS6100, the pioneer in UK) has appeared; this is a glossary of the terminology used in the UK Construction sector, aimed to provide a comprehensive list of terms that will promote better understanding between various sections of the construction industry, facilitate trade and provide better tools for improving handling of information.

The use of BS6100 was combined with the Unified Classification for the Construction Industry (UNICLASS, published in 1997 as a substitute for the widely accepted but increasingly out of date CI/SfB), which is a Construction specific information classification system that covers information generated from all phases of a construction project. Uniclass is structured with a faceted classification system rather than a hierarchical one.

The ISO 120063 family (part 2 and part 3) came from another level of concern: the International Organisation for Standardisation. ISO was also targeting the development of standard CVs for the Construction sector in a world-wide scale. On one hand, ISO12006-2 targeted the definition of a model for classification systems (it is not a classification system in itself); rather it sets out an approach whereby particular classification systems that meet regional or national requirements can be developed according to a common international approach. On the other hand, ISO 12006-3 defines a schema for a taxonomy model, providing the ability to define concepts by means of properties, to group concepts, and to define relationships between concepts. Objects, collections and relationships are the basic entities of the model.

The ISO foundation work was adopted and used by some institutions around the world. Among them, it can be cited the Stabu (Netherlands), Edibatec (France), and the Norwegian construction industry, which respectively started their own implementations of ISO-based tools, namely the LexiCon, SCD, and BARBI. In other words, the three of them are independent implementations of dictionaries that are compliant with the specification given in ISO 12006-3.

The eConstruct project developed the Building and Construction eXtensible mark-up Language (bcXML), which supports the eBusiness communication process needed between clients, architects and engineers, suppliers and contractors for the e-procurement of products, components, and services. The bcBuildingDefinitions, the taxonomy developed by eConstruct to show the potencial of bcXML, contains nearly 3000 terms specifically related to doors, expressed in six languages. Such taxonomy can be instantiated to create catalogue contents or the actual requirements (queries) and solutions (answers) messages.

The e-COGNOS project developed a KM-oriented software infrastructure enabled by a semantic pillar: an ontology server (and its respective ontology). Such ontology focuses on construction concepts as they were related to e-COGNOS main objective: consistent knowledge representation of construction knowledge items. The e-COGNOS ontology is composed of two taxonomies (concepts and relations).

The CEN/ISSS eConstruction series of workshops worked towards the standardisation (or as close as possible since CEN means European Committee for Standardisation) in which the required semantic themes were also formulated. This initiative recognised that it is not possible to propose standardised Semantic Resources (SRs – i.e., ontologies, taxonomies, dictionaries, thesauri, and the related resources, referred in this work as controlled vocabularies) for the construction sector but that it was possible to recommend what organisations could do after deciding to use SRs to support their business activities. Additionally, this initiative emphasised the need to take into account two key parameters, namely purpose and application areas when considering development and/or use of SRs. For the sake of clarity, it is worth saying that in this report, the terms Controlled Vocabularies (CV) and Semantic Resources (SR) are used seamlessly and somewhat interchangeably to represent classification systems, dictionaries, thesauri, ontologies and the like.

The FUNSIEC project conducted research efforts in order to make possible the establishment semantic links (mappings) between different SRs. Also, it worked towards the development a framework to evaluate how good semantic links were. It was demonstrated by FUNSIEC, that it is possible establish semantic links between different SRs through the project results: the OSIECS Kernel, and both OSIECS meta-model/model. The former is a software tool built to identify and propose semantic mappings between two SRs. OSIECS meta-model/model are the mapping tables produced by the OSIECS Kernel.

The CONNIE project tackled the problem of exploiting multi-lingual content representing norms and regulations for the European Construction sector. It produced a software infrastructure to help organise, index, classify and use (in a pan-European way), the contents (regulation/norms) currently available within the CONNIE environment. This infrastructure strongly relied on the use of CVs in order to index and share the use of multi-lingual contents in an efficient way.

The SEAMLESS project targeted the deployment of a seamless infrastructure to help SMEs to participate more easily in the e-business world (i.e. providing e-services to support business needs, such as procurement, and production follow-up). The SEAMLESS infrastructure has been developed in a sector-independent way, but in order to demonstrate its potential two vertical sectors were selected: Textile and Construction. The knowledge-related side of SEAMLESS was based on a hierarchy of ontologies covering three levels of representation, namely: the global level (the whole SEAMLESS environment), the mediator level (the intermediate level providing a mapping between the global level and the SMEs), and the local levels (the lowest level where the SMEs are placed with their small CVs). In order to support the operation of the SEAMLESS environment, a sector-specific hierarchy of ontologies was developed.

North America developed a classification system within a single, multi-facetted approach called Omniclass, which started under the name of Overall Construction Classification System (OCCS) in 2001 and was renamed to Omniclass in 2002. It is based on ISO12006-2 as a framework and it uses MasterFormat for work results, UniFormat for elements, and Electronic Product Information Cooperation (EPIC) for structuring products. First version of Omniclass 1.0 was officially released by March 2006.

Last but not least, the picture is complete by referencing the ICONDA Terminology (IRB, ICONDA®Bibliographic 1986) and the Canadian thesaurus. The former is the CV supporting the operation of the whole ICONDA@ family of products (e.g. the ICONDA database holding technical information on Construction problems). The latter is a bi-lingual thesaurus specifically created to represent construction terms in English and French. The enrichment of this thesaurus has been re-launched and new developments and improvements are expected in the near future.

Into what this thesis is concerned, the e-COGNOS project and bsDD will be further detailed below, which have been identified as the more relevant. The reason for this is related with the fact the e-COGNOS ontology will be used and extended for proof-of-concept. From the several initiatives described earlier, bsDD seems to be most active. Several developments have conducted lately by providing an API supporting REST web services, enabling the integration of bsDD and external applications.

2.3.3.1 The e-COGNOS Ontology

The e-COGNOS project developed a web-based KM platform targeting the needs of the Construction industry coined by e-CKMI (e-COGNOS Knowledge Management Infrastructure). The e-CKMI was supported by an ontology, focusing on building and construction concepts addressing e-COGNOS main objective: “a consistent knowledge representation of construction knowledge items”. Such ontology was developed taking into account very relevant sources of

inspiration, namely the IFC model, the bcXML MetaSchema, the BS6100 Classification, and the DAML+OIL language.

The e-COGNOS ontology was developed according to the following guidelines:

1. Incorporation of already established and recognised industry standard taxonomies and classification systems (e.g., BS6100, bcXML, and Talo 90) and the IFC product model.
2. Not intended to be the ultimate ontology for the BC sector.
3. It should be user friendly, i.e., easy to browse and understand.
4. It should be developed incrementally involving the end users.
5. It should be flexible and wide enough to accommodate different business scenarios presented by the end users.
6. It should allow for future expansion.

The conceptual model of the e-COGNOS ontology was based on the bcXML meta-schema. In the development of the e-COGNOS ontology, a taxonomy was considered as the cornerstone upon which all the subsequent efforts were based. The e-COGNOS ontology was essentially composed of two taxonomies (Figure 2.5 and Figure 2.6): a taxonomy of concepts and a taxonomy of relations. The taxonomy of concepts is grounded on the IFC concepts, which are used to form its highest levels, and address the e-COGNOS ontology motto:

In the context of a Project, 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).

The major contributors to the e-COGNOS ontology are showed in Table 2.3.

Table 2.3. Contributions to e-COGNOS Ontology

Reference	Definition	Usage in e-COGNOS
BS6100	Glossary of Building and Civil Engineering terms, produced by the British Standards Institution (BSI, the independent national body responsible for preparing British Standards)	Provides a great deal of synonyms that were used to enrich the ontological concepts.
bcXML	The Building and Construction XML-based Metaschema/language supports the eBusiness communication needed between clients, architects and engineers, suppliers and contractors for the procurement of products, components and services.	Provides the conceptual model and is used as the format to import new taxonomies into the ontology.

DAML+OIL	Darpa Markup Language + Oil Inference Language was the format promoted by the semantic web group to represent ontologies.	Represent the e-COGNOS ontology.
IFC Model	A common language used to exchange information between different software used in the AEC domain	IFC Kernel provided 85 concepts to form the highest levels of the ontology.

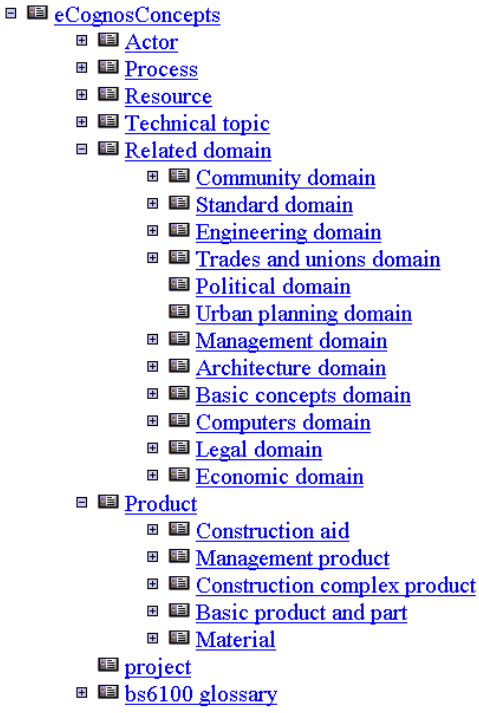


Figure 2.5. The e-COGNOS Taxonomy of concepts

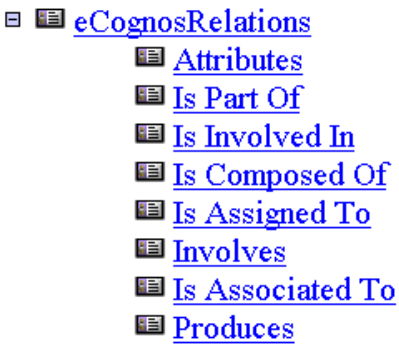


Figure 2.6. Taxonomy of relations

The e-COGNOS ontology was developed following the methodology shown in Figure 2.7 (Wetherill, et al. 2002). It comprises the following steps:

1. Definition of domain and scope using, as an auxiliary mechanism, a list of competency questions. This list has two purposes: provide the guidelines during the development of the ontology and the validation rules, after completion of the ontology. Examples of competency questions are the following:

1.1 Can you provide me with the latest regulations on fire security in buildings open to the public?

1.2 What is the minimum required rate of air flow in educational buildings?

1.3 Please, provide me with a list of all manufacturers of sigma shaped steel beams in France including their contacts.

2. Reuse of ontologies and/or related tools: the e-COGNOS ontology got contributions from the results produced by several international and well-known initiatives, namely the IFC, the DAML+OIL, the CoMMA ontology, and the bcXML Schema and taxonomy.

3. Enumeration the important terms in the ontology: as mentioned in the previous item, the IFC entities, the bcXML taxonomy, and the CoMMA ontology provided the preliminary list of concepts for the e-COGNOS ontology;

4. Definition of concepts and concepts hierarchy based on the relation "is a", i.e., creation of the e-COGNOS taxonomy.

5. Definition of properties of the concepts: properties were taken from IFC entities as much as possible and were completed with the specific properties suggested by the end users

6. Definition of restrictions: each property defined in the entities can have range, restrictions, constraints, etc.; and

7. Population of the ontology: the ontology entities are instantiated. The end users and an ontology manager are responsible for it.

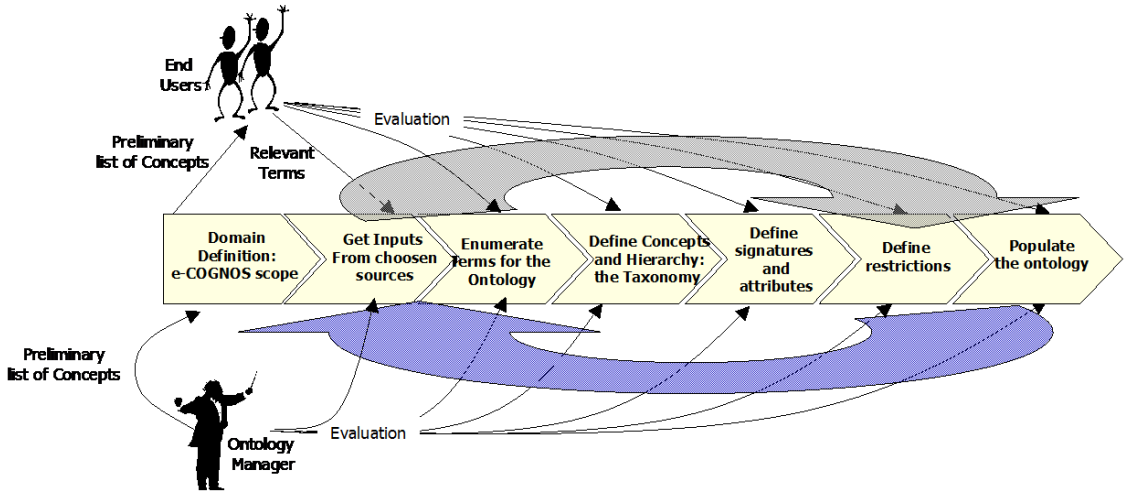


Figure 2.7. Creating the e-COGNOS ontology

The e-COGNOS ontology was created, initially having around 800 concepts, chosen from the sources of inspiration and from a sample of documents provided by the end users. The e-COGNOS goal was to build the biggest possible ontology covering the areas of work identified in the end users business cases. However, in order to grow fast, the manual-based process was not recommended. Therefore, the reuse of existing taxonomies was the solution found to extend the ontology quicker. Together with the DAML+OIL, the bcXML language was adopted as an alternative format to import taxonomies into the e-COGNOS Ontology. This mechanism,

successfully developed in e-Construct, helped to achieve 17 000 concepts very quickly. Some taxonomies were included, such as the bcXML compliant of BS6100, a taxonomy covering the financial aspects related to the construction projects, and a taxonomy for highways.

The end users specific taxonomies were also included into the e-COGNOS ontology. The eConstruct experience again helped to put this process in place. Based on a simple excel spreadsheet, the end users built their taxonomies which were automatically converted to bcXML and imported into the e-COGNOS ontology.

The e-COGNOS ontology was used to support KM practices, such as knowledge acquisition, indexation, and search. The e-COGNOS Infrastructure acquires "Knowledge Items" – KI (e.g. documents, experts, organisations, projects, etc.) and creates the respective "knowledge representation" (KR). This KR is then indexed through keywords and ontological concepts. In the search process, the ontology was used to support a so-called "advanced process" where the user could browse the ontology in order to prepare his/her query in the more precise way.

The e-COGNOS vision over the development of a big ontology was confronted with an unexpected reality. The end users actually showed their preferences to use their very specific, concise, and precise taxonomies. They did not want to handle big ontologies; rather they were perfectly happy if their small resources were in place, providing the results they were expecting. This fact has changed the concept of the e-COGNOS ontology: the big ontology is in place, but it is totally customisable in the sense that a small taxonomy with 100 concepts can replace the full one.

2.3.3.2 BuildingSMART Data Dictionary (bSDD)

The bsDD formally known has IFD which stand for "International Framework for Dictionaries". The name is used both for the IFD library and for the organization running and maintaining it. While the two first words; "International and Framework" pretty well describes IFD, the last word "Dictionaries" is somehow misleading. Yes, IFD is a dictionary, or rather a lot of dictionaries, but it is also much more.

The bsDD standard (ISO 12006-3) is an EXPRESS model with a short explanation of its purpose and use. The model itself is pretty simple seen from an implementers view but it is proven to be very flexible and can therefore result in several different implementations. The two first implementations of the standard were the Norwegian BARBi library and the Dutch LexiCon from STABU (mentioned earlier in this section). There were also other implementations done by e.g. EDIBATEC in France. The structure of bsDD is given in ISO 12006-3 and is a result of many years of standardization work by the ISO TC59/SC 13/WG 6 work group. The standardization work started in October 1999 and has slowly evolved into the official ISO 12006-3 standard that was formally published at April 14. 2007.

In its simplest form bsDD is a mechanism that allows for creation of multilingual dictionaries or ontologies. There is nothing in the ISO 12006-3 standard that limits it to building and construction, and the model as such can be used to describe most things.

bsDD separates the names and languages from the concepts itself. It is not a mapping of words in one language to words in another language, but a mechanism where the concept itself is a separate thing, only connected to the words describing it through relationships Figure 2.8.

E.g. the word “dør” in Norwegian is in a normal dictionary translated to “door” in English. By studying the concept it can be concluded that in Norway it refers to the door with its frame, while door in English only referrers to the door-leaf. The Norwegian dør should be translated to door set. In bsDD this is achieved by separating the concepts itself from the names and descriptions used to name and describe it.

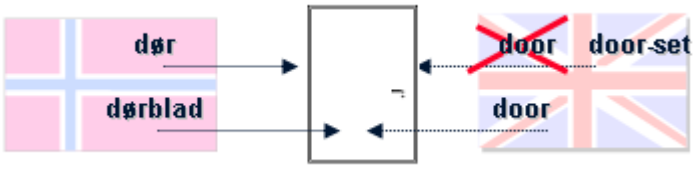


Figure 2.8. Naming Concepts

One concept often has multiple names in the same language (Figure 2.9). Examples are beam and truss but also long and short form names as millimetre and mm. bsDD allows for multiple names, descriptions, short names (e.g. acronyms) and lexemes to be attached to the same concept. As is the case for descriptions, definitions and comments which also is supported in bsDD.

A beam is not just a beam. A name is often used to describe more than one concept even in the same language. As the image shows a 'beam' is not just a 'beam' but a word used to describe multiple concepts. In any computer related exchange of information it is essential to capture the different concepts hidden behind all those different versions of the word 'beam'.

Because of the internal relationships between concepts in bsDD it is possible to translate a specialized concept into a more generalized one. For instance, all of the fifteen different snow types in Greenlandic (East Inuit) might translate into only one word in Bantu. While each of the specialized concept for snow do not have any equivalent in Bantu, they at least share the same supertype being 'snow'. This will also be the case for more specialized building terms e.g. fire-resistance classes in different countries.

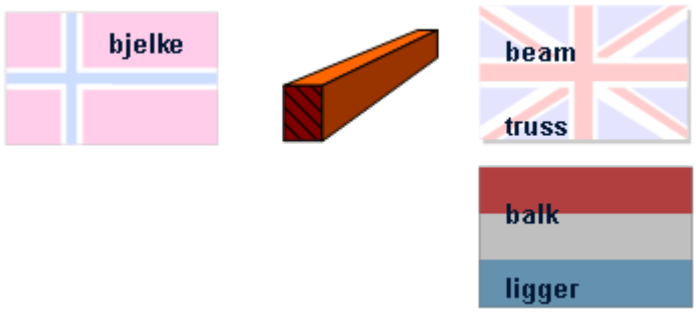


Figure 2.9. bsDD multiple names in the same language

In bsDD a concept is described both by a set of names and definitions in multiple languages but also by relating a concept to other concepts. As stated earlier, bsDD can hold multiple ontologies in the same library. This is possible because of the support for contexts. A context in bsDD is a grouping of relationships that exists between concepts (Figure 2.10). There might be (or rather it is known that there are) multiple ways of viewing a concept. A door will be described differently depending on the phase of the building project, the actor describing it or the standard used. Each of this views can be seen as a set of relationships between the concept being described and the other concepts describing it. Another name for such a view is a 'context'.

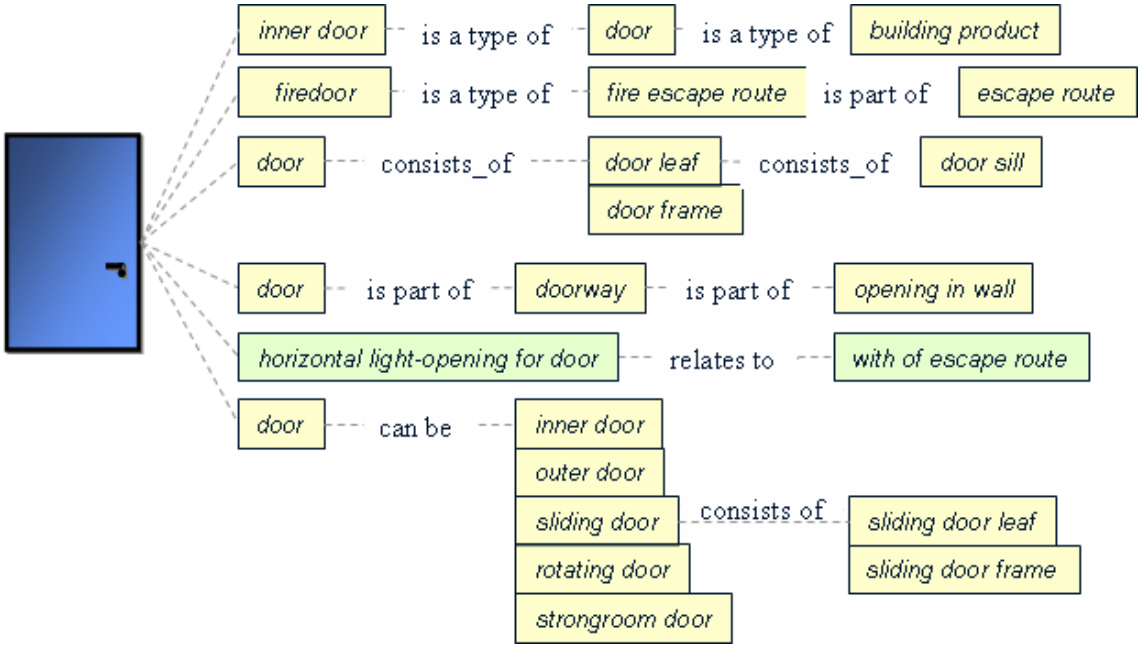


Figure 2.10. Concepts and Relationships

There is no start and end on a bsDD library. bsDD is a web of concepts linked to each other by the use of relationships. When contributing to or extracting information from the bsDD library, a user may start at any point in the structure. Another essential rule of bsDD is: A concept can only exist once, there are no duplicates.

Even the measure concept 'millimetre' only exists once and any measure using millimetre is doing so by creating a relationship between the measure concept and the unit concept. Relationships are also two ways. Looking from 'millimetre' it is possible see all measures using that unit.

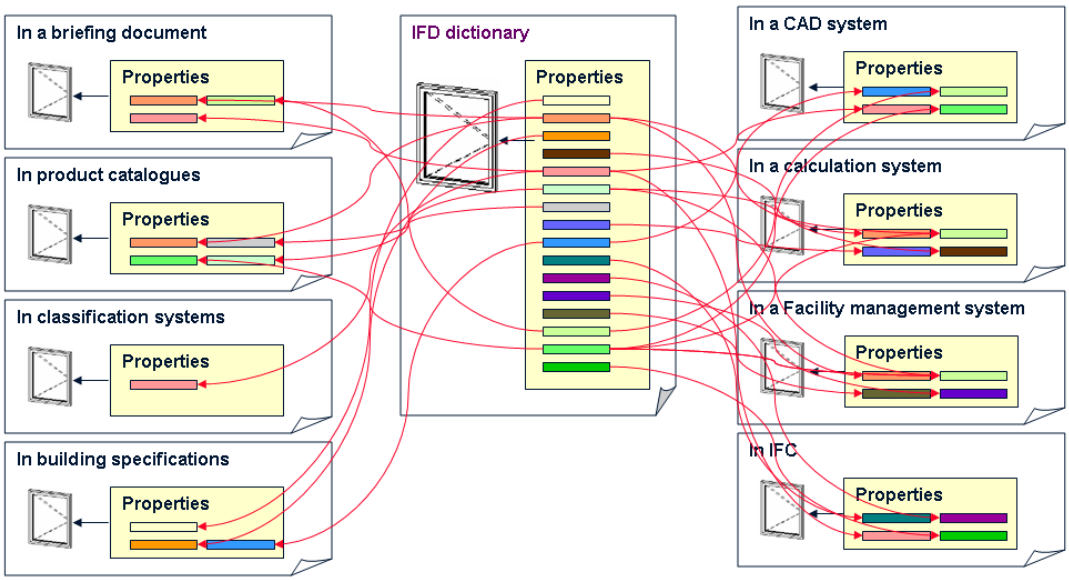


Figure 2.11. bsDD as a Mapping Mechanism

A concept in bsDD is described independent of time and use and each relationship is of the type “can have”. Taking a window as example, by studying the concept of a window in different sources of information, each source will talk about a particular set of e.g. properties for that window (Figure 2.11).

The properties are here represented by coloured boxes. Some properties are 'shared' between the different information sources where a window can appear (in common), but some are different. An information source is often one particular 'view' of the concept for one particular use or in one particular phase. By going through different sources and adding each property to the generic concept of a window in bsDD, you will eventually end up with a concept having the sum of all properties for a window. By keeping the link back to the source where the information originated from, bsDD ultimately becomes a mapping structure as well.

2.4 Remarks and Future Trends

This section aims to wrap-up what has been described during this chapter, also to reinforce the importance of the adoption of approaches based on controlled vocabularies. This section concludes by identifying future trends regarding the adoption of controlled vocabularies within industry.

Communication is about exchanging signs. Humans are able to use words, body gestures, images, and other similar signs. Jargon is often used inside a given community and those not belonging to this community will have problems to communicate. If someone wants to be clear and unambiguous, it must 'control' the vocabulary that using in communication. Only parties knowing the words and their meanings are equipped to engage in communication free from misunderstanding. When it comes to computer-based communication, this is even more crucial since computers cannot establish dialogues in order to know elucidate precisely 'what is meant by that'. The conceptual approach to handle this situation often relies on the adoption of formal CVs (as much as possible) which help define the universe of discourse (the working context) of those involved in the communication process.

Several examples can be found around the world, coming from very different initiatives ranging from industrial support to feasibility projects funded by research programs. Results are emerging; education is gaining a new status in the European scene for several reasons, including European policies, businesses profit, and natural evolution of the area. LexiCon and BARBI (two implementations of ISO 12006) have joined forces; IFCs are becoming the standard supporting the inevitable BIM concept; bsDD is attracting worldwide attention, and governments have published policies that directly or indirectly enforce the adoption of shared CVs and semantic-related resources.

Recalling McGuiness (and adapting her sayings to this context), an ontology (or CV) is required when there is a need to communicate/exchange (transfer and/or share) various sorts of information where the meaning is fundamental. Ontology is also useful when the reuse of existing knowledge is required. From a non-exhaustive list of uses, an ontology can be used for simple kinds of consistency checking, interoperability support, validation and verification testing, configuration support, help to perform structured, comparative and customised search as well as to exploit generalisation/specialisation of information (McGuinness 2003). This means whenever someone must communicate precisely, the vocabulary must be controlled, the jargon must be shared and meaningful, and the semantics must be refined for the sake of the communication process. This is the mission behind the development and use of CVs in the Construction sector. This is the justification for proposing, developing, and assessing CVs.

The work and initiatives described in the previous sections allow saying that good results have been achieved, but additional efforts are still needed in order to adequately capitalise on the

results provided by the research world and standardisation bodies. However, good work has been produced, and solid results are now available. Education is the key word behind what needs to be done in order to push things forward. Research is a key factor in this quest and this thesis is part of it.

The assessment of the results produced by the FUNSIEC project emphasised the importance of education (in the large sense) of the practitioners from Construction regarding the use of semantic resources. CEN/ISSS eConstruction workshop suggested the same approach. Education here means providing good practice examples to the final users showing how they can benefit from the use of CVs in their daily business, how they can expand their capabilities and potentialities in terms of market, what are the tangible benefits/improvements CVs can offer to them.

From a more technical and operational perspective, there is still a long way to go. As already mentioned before, the idea of the 'ultimate classification system' is a utopian one. However, computer-based information management introduces new possibilities and puts some new requirements on information and classification systems. The way forward is further research into a global unified classification system that could be used by all. One of the best examples is the bsDD classification, but further work is needed on its strengths, weaknesses and exclusivity to enable it to be globally used and accepted.

The demand of manual engineering using such technologies stays relatively high. Interfaces between any two applications have to be documented for human software developers using natural languages to communicate the semantics of the information to be exchanged as well as the intended use of its operations. As an example, electronic classifications such as STABU LexiCon, and BARBi tend to be very complex to use and complicated to implement, but nevertheless those were considered the pioneering effort and contributed as a first step to the vision of CVs, and they have been incorporated into bsDD.

The need for automation of information exchange and integration is a must in domains that have a particularly broad spectrum of heterogeneity of information, and whose interdependency concerning business processes and information exchange is high.

Interoperability *per se* is in this case, a common denominator that addresses syntactic and semantic sub-levels. Here, mechanisms must be developed that allow the access of information not only on a document syntactic basis but also to semantically identify and annotate specific parts of the large and complex information models. This allows for a more precise reference, monitoring and downstream use of specific aspects and elements within a building. Instead of indirectly and informally referring to a certain aspect ("reinforcement mesh used in slab 31 in the attached drawing"), future artefacts and enabling processes can be directly referenced and measured.

There is a need to automate (as much possible) the concept mappings between heterogeneous information models using reference domain ontologies. One of the main reasons identified by the slow take-up of such initiatives, is directly related with the amount of manual work that needs to be applied for performing such mappings. Also another fundamental aspect, relates to the amount of work to maintain and keep updated such domain ontologies.

It is believed that in the relatively near future, it will be possible to reduce the human intervention in supervising domain ontologies and at the same time to make the process of mapping between heterogeneous data sources and domain ontologies more autonomous.

This thesis, advocates that the web semantic initiative plays a key role in this process. Methods used for web mining, and more specifically information retrieval, natural language processing and deep web reasoning, must be applied to automate as much possible the integration of heterogeneous data sources to domain ontologies. Such methods must be applied also in order to have a “learning” ontology, which dynamically adapts whenever new knowledge is created. It is expected that web mining methods will increasingly treat content, structure, and usage in an integrated fashion in iterative cycles of extracting and utilizing semantics, to be able to understand and (re)shape the Web.

Further research is needed, and will give rise to new research questions and stimulate further research in the Semantic Web communities, towards the ultimate goal of Semantic Web Mining: “a better Web” for all of its users, a “better usable Web”. One important focus is to enable search engines and other programs to “better understand the content of the Web”. This thesis provides a contribution in this direction, by enriching the content of non-structured information like documents, with semantic information available in external domain ontologies. As a result, a more precise representation of knowledge sources is achieved, providing a better understanding of its context to search engines supporting them to produce improved results.

3

Overview of Information Retrieval

If we knew what it was we were doing, it would not be called research, would it?

- Albert Einstein (1879 – 1955), Nobel Prize in Physics

People have the ability to understand abstract meanings that are conveyed by natural language. This is why reference librarians are useful; they can talk to a library patron about their information needs and then find the documents that are relevant. The challenge of information retrieval is to mimic this interaction, replacing the librarian by an automated system. This task is difficult because the machine understanding of natural language is, in the general case, still an open research problem.

There are generic principles related to information retrieval and knowledge management that can be incorporated into an approach that supports consistency across large knowledge repositories maintained in a heterogeneous and distributed collaborative business environment. This chapter aims to pinpoint those principles in order to develop such an approach based on a solid theoretical foundation (chapter 4), and after to deploy it using knowledge sources (chapter 5).

More formally, the field of Information Retrieval (IR) is concerned with the retrieval of information content that is relevant to user's information needs (Frakes and Baeza-Yates 1992). Information Retrieval is often regarded as synonymous with document retrieval and text retrieval, though many IR systems also retrieve pictures, audio, or other types of non-textual information. The word "document" is used here to include not just text documents, but any source of information. However, according to (Manning, Raghavan e Schütze 2009), as an academic field of study, *information retrieval* might be defined thus:

Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections of material (usually stored on computers).

Following this definition, information retrieval used to be an activity that only a few specialists were engaged in, such as reference librarians, paralegals, and similar professional information researchers. Now the world has changed and hundreds of millions of people engage in information retrieval every day when they use a web search engine or search their email[1]. Information retrieval is fast becoming the dominant form of information access, overtaking traditional structured database searching (the sort that is going on when a clerk says to you: "I'm sorry, I can only look up your order if you can give me your Order ID").

IR can also cover other kinds of data and information problems beyond that specified in the core definition above. The term "unstructured data" refers to data which does not have clear, semantically overt, computer-friendly structure. It is the opposite of structured data, the canonical example of which is a relational database, of the sort companies probably use to maintain product inventories and personnel records. In reality, almost no data are truly "unstructured". This is definitely true of text data given the latent linguistic structure of human languages. Moreover, most blocks of text have structure, such as headings, paragraphs and footnotes, commonly represented in electronic documents by explicit mark-up (such as the coding underlying web pages). IR is also used to facilitate "semi-structured" search such as finding a document where the title contains Java and the body contains threading.

[1] In modern parlance, the word "search" has tended to replace "(information) retrieval"; the term "search" is quite ambiguous, but in the context of this thesis the two are synonymous.

IR also supports users in browsing or filtering document collections or further processing a set of retrieved documents. Clustering is the task of coming up with a good grouping of the retrieved documents based on their contents. It is similar to arranging books on a bookshelf according to their topic. Given a set of topics, standing information needs or other selection categories (such as suitability of texts for different age groups), classification is the task deciding which class(es), if any, each of a set of documents belongs to. It is often approached by first manually classifying some documents and then hoping to be able to classify new documents automatically.

Nowadays, a huge amount of information is available in online documents, e-books, journal articles, technical reports, and digital libraries. The Internet has led to an exponential increase in the volume of electronic documents. Therefore the need for effective automatic classification of documents is now imperative.

Automatic text classification is the task of assigning predefined categories to unclassified text documents. When an unknown document is given to the system it automatically assigns it the most appropriate category. The classification of textual data has practical significance in effective document management. In particular, as the amount of available online information increases, managing and retrieving these documents is difficult without proper classification.

Strongly related with the IR field, document retrieval considers two related activities: *indexing* and *searching* (Jones and Willett 1997). *Indexing* refers to the way documents (i.e. information) are retrieved, and queries (i.e. statements of a user's information needs) are represented for retrieval purposes. *Searching* refers to the process whereby queries are used to produce a set of documents that are relevant to the query. Relevance here means simply that the documents are about the same topic as the query, as would be determined by a human judge. Relevance is an inherently fuzzy concept, and documents can be more or less relevant to a given query. This fuzziness puts IR in opposition to Data Retrieval, which uses deductive and Boolean logic to find documents that completely match a query (Rijsbergen 1979).

Information retrieval systems can also be distinguished by the scale at which they operate, and it is useful to distinguish three prominent scales, namely web search, enterprise search and personal information retrieval. In web search, the system has to provide search over billions of documents stored on millions of computers. Distinctive issues need to gather documents for indexing, being able to build systems that work efficiently at this enormous scale, and handling particular aspects of the web, such as the exploitation of hypertext and not being fooled by site providers manipulating page content in an attempt to boost their search engine rankings, given the commercial importance of the web.

At the other extreme is personal information retrieval. In the last few years, consumer operating systems have integrated information retrieval (such as Apple's Mac OS X Spotlight or Windows8 Instant Search). E-mail programs usually not only provide search but also text

classification: they at least provide a spam (junk mail) filter, and commonly also provide either manual or automatic means for classifying mail so that it can be placed directly into particular folders. Distinctive issues here include handling the broad range of document types on a typical personal computer, and making the search system maintenance free and sufficiently lightweight in terms of start-up, processing, and disk space usage so that it can run on one machine without annoying its owner. In between is the space of enterprise, institutional, and domain-specific search, where retrieval might be provided for collections such as a corporation's internal documents (where this thesis is focusing on), a database of patents, or research articles on biochemistry. In this case, the documents will typically be stored on centralized file systems and one or a handful of dedicated machines will provide search over the collection.

3.1 Models for Documents' Semantics Characterization

Index terms are traditionally used to characterize and describe the semantics of a document. This approach attempts to summarize a whole document with a set of terms that are relevant in the context of the document. While this approach has given some satisfactory results in the area of IR, it still has some limitations as it proceeds by oversimplifying the summarization process through relying on a subset of relevant terms that occur in a document, using them as a means to convey the semantics of the document.

This section discusses the existing IR models, a taxonomy of which is given in Figure 3.1 (Baeza-Yates e Ribeiro-Neto 1999). It describes the three classical models of IR, namely Boolean, Vector, and Probabilistic. In the Boolean model documents are represented as a set of index terms. This model is said to be **set theoretic** (Gudivada, et al. 1997). In the Vector model documents are represented as vectors in a t-dimensional space. The model is therefore said to be **algebraic**. In the probabilistic model, the modelling of documents is based on probability theory. The model is therefore said to be **probabilistic**. Alternative models that extend some of these classical models have been developed recently. The Fuzzy and the Extended Boolean models have been proposed as alternatives to the set theoretic model. The Vector Space Model, the Latent Semantic Indexing, and the Neural Network models have been proposed as alternatives to the algebraic model. The Inference Network, and the Belief Network models have been proposed as alternatives to the Probabilistic Model. It is also worth mentioning that models which reference the structure, as opposed to the text of a document, do exist. Two models have emerged in this area: the Non-Overlapping Lists and the Proximal Node.

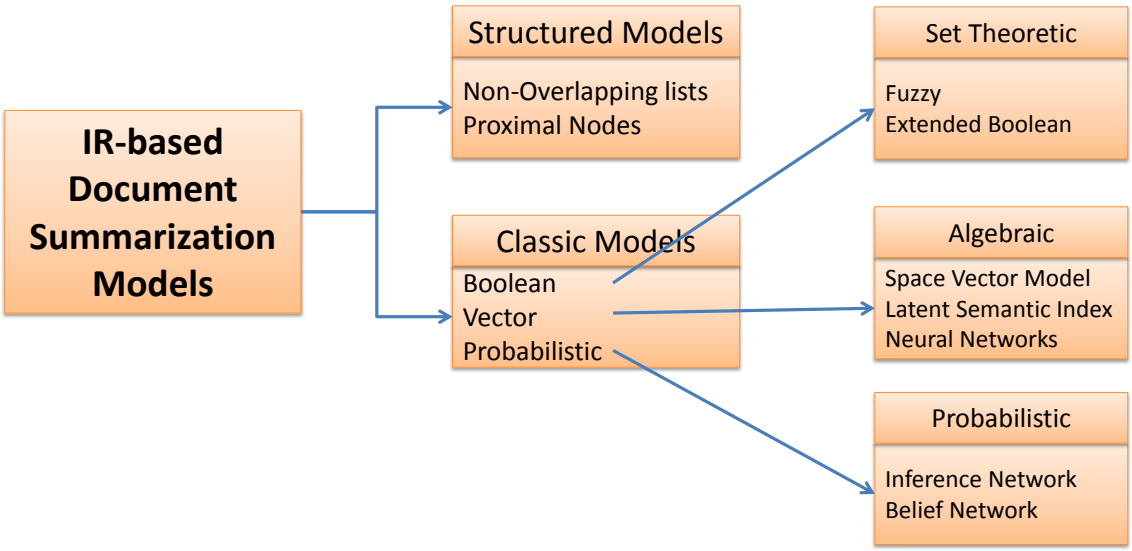


Figure 3.1. Information Retrieval Models (Baeza-Yates e Ribeiro-Neto 1999)

3.1.1 Boolean Model

It is based on the set theory and Boolean algebra. Query expressions are expressed as a combination of Boolean expressions, including Boolean operators which have a clear semantics. It was adopted and had great success in bibliographic and library information systems. The main criticism of the Boolean model (Verhoeff, Goffman and Belzer 1961) lies in its binary evaluation system. A document can be either relevant or not to a given query. There is not inherent ability to rank the document in relation to its relevance to a given query. In other words, there is no notion of partial match to the query conditions. It is commonly acknowledged today that index term weighting provides more satisfactory results in retrieval performance.

3.1.2 Vector Model

It addresses the limitations of the Boolean model by providing an approach that supports document partial matching to a given query. This is achieved by assigning non-binary weights to index terms in documents and queries. These term key word weights are then used in a second stage to sort documents by their level of relevance to the initial query. More details and further description of the Vector model, which is today considered as the most popular IR model, can be found in (Salton, Wong and Yang 1975) and (Salton and Buckley 1988).

3.1.3 Probabilistic Model

This was introduced initially by Robertson and Sparck Jones (Robertson and Jones 1976) as a means to address the Information Retrieval problem within a probabilistic context. It proceeds by refining recursively a guessed initial set of documents matching a user query by involving user feedback to evaluate the relevance of the retained set. For each iteration, the user retains the documents that best match the query. The system then uses this information to refine the description of the ideal response set. As highlighted in (Baeza-Yates e Ribeiro-Neto 1999), the main advantage of the probabilistic model is that documents are ranked in decreasing order of their probability of being relevant. The disadvantages include: (1) needing to guess the initial allocation of documents into relevant and non-relevant sets; and (2) the method does not take into account the frequency of which an index term appears within a document.

3.1.4 Alternative Set Theoretic Models

Several models that make use of the Fuzzy Set theory have been proposed. Fuzzy set theory can be described as a framework for representing classes whose boundaries are not well defined. The key idea is to introduce the notion of a degree of membership associated with the elements of a set, where this degree of membership varies from 0 to 1 and allows modelling the notion of marginal membership. Thus, membership is now a gradual notion, contrary to the ‘crispy’ notion enforced by classic Boolean logic. In the fuzzy set model, queries and docs are

represented by sets of index terms, where a matching function computes the degree to which document d_i satisfies a query q_j. This matching can be modelled using a fuzzy framework, as follows: (i) with each term is associated a fuzzy set; (ii) each document has a degree of membership in this fuzzy set. Such assumptions provide the foundation for many models for IR based on fuzzy theory. The model from Ogawa, Morita and Kobayashi (Ogawa, Morita and Kobayashi 1991) deserves particular attention in that a thesaurus is being used in conjunction with the Fuzzy Set theory to expand the set of index terms in a query and extend the retrieved document set.

Another alternative approach is coined by the Extended Boolean model. The principle behind the extended Boolean model is to overcome the binary limitations of the Boolean model by extending and enhancing it with partial matching and term weighting adopted from the vector model. The Boolean model provides a simple approach when applied into the IR domain, but doesn't provide a ranking mechanism. As with the fuzzy model, a ranking can be obtained by combining the characteristics of the vector model with properties of Boolean algebra. This approach extends the classical Boolean model with the notions of partial matching and term weighting. This model has been introduced by Salton, Fox and Wu (Salton, Fox and Wu 1983).

3.1.5 Alternative Algebraic Models

Alternative Algebraic models include the generalized Vector Space, the Latent Semantic Indexing, and the Neural Network models.

The Vector Space model assumes that two index term vectors might be non-orthogonal which means that there is a possibility for two index terms to be correlated. This term correlation is used as a basis for improving retrieval performance (Salton, Wong and Yang 1975).

The principle behind the Latent Semantic indexing model is that ideas in a text are more related to the concepts that are conveyed within it as opposed to the index terms. By using this approach, a document may be retrieved only by the virtue that it shares concepts with another document that is relevant to a given query. As indicated in (Furnas, et al. 1988), the intent behind the latent semantic indexing model is to map each document and query vector into a lower dimensional space which is associated with concepts. This is achieved by mapping the index term vector into this lower dimensional space (Baeza-Yates e Ribeiro-Neto 1999).

The Neural Network model is based on research carried out in the area of Neural Networks. The principle behind ranking documents that are retrieved against a given query is to match the query index terms against the document index terms. Since Neural Networks have been extensively used for pattern matching purposes, they have quite naturally been used as an alternative model for information retrieval (Wilkinson and Hingston 1991).

3.1.6 Alternative Probabilistic Models

The use of probability theory for quantifying document relevance has always been a field of research in Information Retrieval sciences. Two examples of IR models based on probability theory are the Inference Network and the Belief Network models. Both models are based on Bayesian Belief Networks that provide a formalism combining distinct sources of evidence, including past queries and past feedback cycles. This combination is used to improve retrieval performance of documents (Turtle and Croft 1991).

The Inference Network model takes an epistemological as opposed to a frequency view of the information retrieval problem (Turtle and Croft 1990). It proceeds, as described in (Baeza-Yates e Ribeiro-Neto 1999) by associating random variables with the index terms, the documents, and the user queries. A random variable associated with a user document denotes the event of observing that document. This document observation asserts a belief upon the random variables associated with its index terms. Both index terms and documents are represented as nodes in the network. Edges are drawn from a node describing a document to its term nodes to indicate that the observation of the document yields improved belief on its term nodes. The random number associated with the user query models the fact that the information request specified in the query has been met. This random number is also represented by a node in the network. The belief in the query node is then expressed as a function of the beliefs of the nodes associated with the query terms.

The Belief Network, introduced by (Ribeiro and Muntz 1996), generalizes the inference network model. It is also based on an epistemological interpretation of probabilities. It differs from the inference network model in that it adopts a clearly defined sample space. It therefore provides a separation between the document and query portions of the network. This has the advantage of facilitating the modelling of additional evidential sources, including past queries and past relevance information.

3.2 Document Classification

As briefly explained earlier, classification plays a vital role in many information management and retrieval tasks. On the Web, classification of page content is essential to focused crawling, to the assisted development of Web directories, to topic-specific Web link analysis, to contextual advertising, and to analysis of the topical structure of the Web. Web page classification can also help improving the quality of Web search (Qi and Davison 2009).

Document classification, also known as document categorization, is the process of assigning a document to one or more predefined category labels. Classification is traditionally posed as a supervised learning problem (Mitchell 1997) in which a set of labelled data is used to train a classifier which can be applied to label future examples. Nevertheless, taking into consideration the requirements of this thesis and the limitations of supervised learning for document classification, the approach presented in the present research (detailed further in chapter 4), adopted an unsupervised classification. The reasons why unsupervised classification was chosen will be presented later on.

Focusing on the problem of classification from a more general point of view, it can be divided into more specific problems, such as subject, functional, sentiment, and other types of classification. Subject classification is concerned about the subject or topic of a document. For example, judging whether a document is about “arts”, “business” or “sports” is an instance of subject classification. Functional classification cares about the role that the document plays. For example, labelling a document as “technical report”, “scientific paper” or “white paper” is an example of functional classification. Sentiment classification focuses on the opinion that is presented in a document, that is, the author’s attitude about some particular topic. Other types of classification include genre classification, search engine spam classification, and so on. **This thesis focuses on subject classification.**

Based on the number of classes in the problem, classification can be divided into binary and multiclass, where binary classification categorizes instances into exactly one of two classes (as in Figure 3.2), and multiclass classification deals with more than two classes. Based on the number of classes that can be assigned to an instance, classification can be divided into single-label classification and multi-label classification. In single-label classification, one and only one class label is to be assigned to each instance, while in multi-label classification, more than one class can be assigned to an instance. If a problem is multiclass, for example, four-class classification, it means four classes are involved, for example, “Arts”, “Business”, “Computers”, and “Sports”. It can be either single-label, where exactly one class label can be assigned to an instance (as in Figure 3.3), or multi-label, where an instance can belong to any one, two, or all of the classes (as in Figure 3.4). Based on the type of class assignment, classification can be divided into hard classification and soft classification. In hard classification, an instance can

either be or not be in a particular class, without an intermediate state, while in soft classification, an instance can be predicted to be in some class with some likelihood (often a probability distribution across all classes, as in in Figure 3.5).

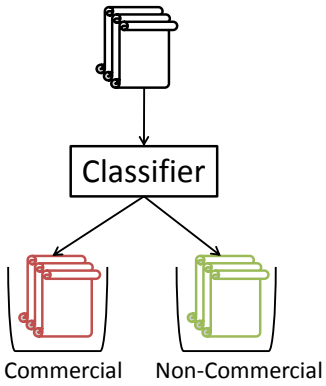


Figure 3.2. Binary Classification

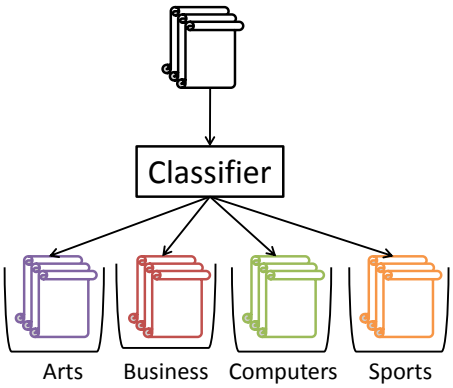


Figure 3.3. Multiclass, single-label, hard classification

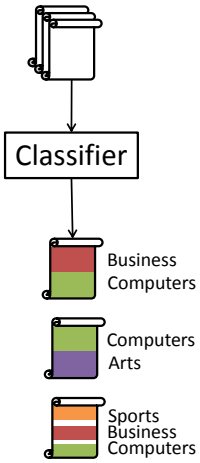


Figure 3.4. Multiclass, multi-label, hard classification

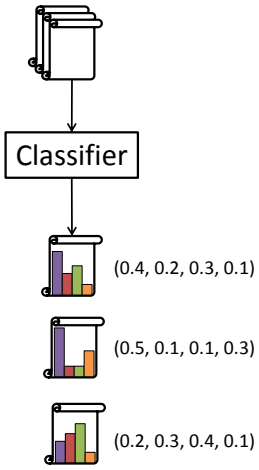


Figure 3.5. Multiclass, soft classification

Based on the organization of categories, document classification can also be divided into flat classification and hierarchical classification. In flat classification, categories are considered parallel, that is, one category does not supersede another, while in hierarchical classification, the categories are organized in a hierarchical tree-like structure, in which each category may have a number of subcategories (Figure 3.6 and Figure 3.7).

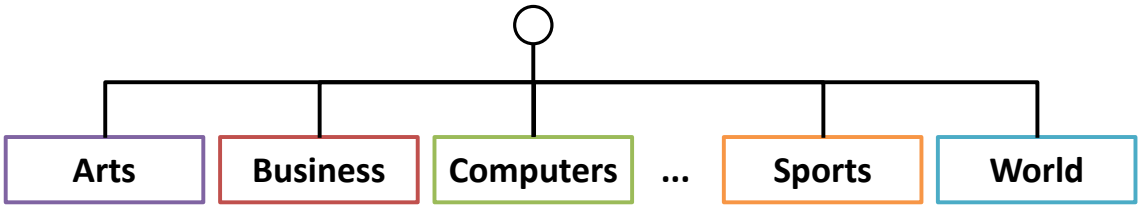


Figure 3.6. Flat Classification

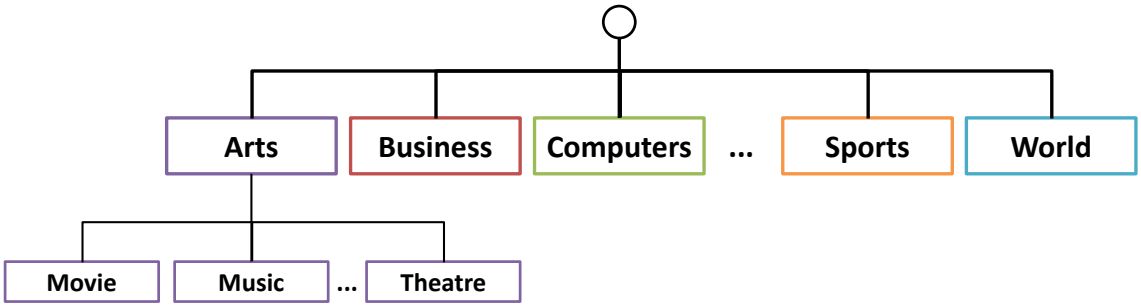


Figure 3.7. Hierarchical Classification

3.2.1 Web classification

In Web classification, query ambiguity is one of the problems that undermine the quality of search results. For example, the query term “bank” could mean the border of a body of water or a financial establishment or a ground slope. Various approaches have been proposed to improve retrieval quality by removing ambiguity in query terms. The work presented by Chekuri (Chekuri, et al. 1997) studied automatic Web page classification in order to increase the precision of Web search. A statistical classifier, trained on existing Web directories, is applied to new Web pages and produces an ordered list of categories in which each Web page can be placed. At query time the user is asked to specify one or more desired categories so that only the results in those categories are returned, or the search engine returns a list of categories under which the pages would fall. This approach works when the user is looking for a known item. In such a case, it is not difficult to specify the preferred categories. However, there are situations in which the user is less certain about what documents will match, for which this approach does not help much.

Search results are usually presented in a ranked list. However, presenting categorized, or clustered, results can be more useful to users. An approach proposed by Chen and Dumais (Dumais and Chen 2000) classifies search results into a predefined hierarchical structure and presents the categorized view of the results to the user. Their user study demonstrated that the category interface is liked by the users better than the result list interface, and is more efficient for users to find the desired information. Compared to the approach suggested by (Chekuri, et al. 1997), this approach is less efficient at query time because it categorizes Web pages on-the-fly. However, it does not require the user to specify desired categories; therefore, it is more

helpful when the user does not know the query terms well. Similarly, Käki (Käki 2005) also proposed presenting a categorized view of search results to users. Käki experiments showed that the categorized view is beneficial for the users, especially when the traditional ranking of results is not satisfactory.

Also Page (Page, et al. 1998) developed the link-based ranking algorithm called PageRank. PageRank calculates the authoritativeness of Web pages based on a graph constructed by Web pages and their hyperlinks, without considering the topic of each page. Since then, research has been conducted to differentiate authorities of different topics. Haveliwala (Haveliwala 2003) proposed Topic-Sensitive PageRank, which performs multiple PageRank calculations, one for each topic. When computing the PageRank score for each category, the random surfer jumps to a page in that category at random rather than just any Web page. This has the effect of biasing the PageRank to that topic. This approach needs a set of pages that are accurately classified. Nie et al. (Nie, Davison and Qi 2006) proposed another Web-ranking algorithm that considers the topics of Web pages. In that method, the contribution that each category has to the authority of a Web page is distinguished by means of soft classification, in which a probability distribution is given for a Web page being in each category. In order to answer the question “To what granularity of topic does the computation of biased page ranks make sense?” Kohlschutter et al. (Kohlschütter, Chirita and Nejdl 2007) conducted an analysis on ODP categories, and showed that ranking performance increases with the ODP level up to a certain point. It seems further research along this direction is quite promising.

Although the presented works are considered as a step forward on web page classification, they still present some limitations and pitfalls which are worth mentioning, and where this thesis brings a contribution. Some examples of PageRank showed that new pages have a low page ranking and they take considerable time to get listed and gain higher ranking, meaning that PageRank takes into account the popularity of pages rather than the real content. A limitation regarding some of the previous works stated is that they can perpetuate and even amplify inaccuracies. If someone inaccurately quotes a “fact” on an web page and subsequent readers go on to quote the inaccuracy on other web pages, then search engines will index all of the inaccurate pages, with the possibility that fiction is widely broadcast as reality. Another limitation is that search results are based on the literal (keywords, tags, meta data) things but not on content meaning.

3.2.2 Text Categorization

Text categorization or classification can be defined as a content-based assignment of one or more predefined categories to free texts. In the 80’s mainly knowledge-based systems were implemented and used for the text categorization. Nowadays statistical pattern recognition and neural networks are used to construct text classifiers. The goal of a classifier is to assign

category(ies) to given documents. Logically each classifier needs some set of data as an input for each single computation. The most frequently used input is the vector of weighted characteristics. Often very complicated and sophisticated methods are used to construct these inputs, also called feature vectors.

Text categorization or the process of learning to classify texts can be divided into two main tasks: Feature Construction and Feature Selection, and the Learning Phase. Whilst the former serves to prepare data for learning machines, the latter task is responsible for training the classification machine on features (feature vectors) obtained from documents from the training data set. Usually each text categorization system is then tested according to texts from outside of the training set. The best would be to make various measurements of performance and compare them to the results of other systems.

3.2.2.1 Feature Construction and Feature Selection

As already mentioned, learning and classification techniques work with vectors or sets of features. The reason is that classifiers use computational algorithms that need to work with some measurable characteristics and not the plain text. Therefore, there is a need to extract some features from a document's plain text, which will characterize the document. Some currently used features are simple words (or strings separated by blanks equivalent to repetitive phrases) with the number of occurrences in documents as the "value" of the feature. Other possibilities are the context of word *w* - set of words, that must co occur in the document with *w* or spare phrases, which is a sequence of nearby, not necessarily consecutive, words (used in Sleeping Experts (Cohen and Singer 1996)). Also word N-grams (ordered sequences of words) (Peng, Schuurmans and Wang 2003) were used. Last but not least it is worth mentioning character N-grams that are very efficient and easy to use in various text categorization tasks (Cavnar and Trenkle 1994). It is worth noting that all these features were parts of the given text. However, the features of the document can be any characteristics which can be observed and described (e.g. the length of the text).

All these features have a common disadvantage. The dimension of the features is very high, which can lead to an overfitting of classification machine. For this reason, the feature selection plays an important role in the text categorization. Feature selection methods are designed to reduce the dimension of the feature, with the smallest possible influence on the information represented by feature vector. There are many methods for dimensionality reduction in statistical literature. Usually the simple threshold cut off method is used. In this case, there are some possibilities of term-suitability criteria, examined in more details in (Yang and Pedersen 1997). This thesis presents another approach to reduce the vector dimensionality using external domain ontologies (Costa, Figueiras e Maló, et al. 2013), which will be presented in chapter 5.

3.2.2.2 Learning Phase

As already mentioned, the classifiers can be divided into two main types. These are binary classifiers and m-ary (m > 2) classifiers. The difference between these two types is that while binary classifiers assign yes/no for a given category and document, independently from its

decisions on other categories, m-ary classifiers use the same classifier for all categories and produce a ranked list of candidate categories for each document, with a confidence score for each candidate. Per-category binary decisions can be obtained by thresholding on ranks or scores of candidate categories. There are several algorithms for converting m-ary classification output into per-category binary decisions.

Usually, algorithms of classifiers are closely related to machine learning algorithms, used for their training. Table 3.1 shows some classifiers and machine learning frameworks, which are commonly used in the text categorization. Some experiments and more details about most of the following can be found in (Yang 1999).

Table 3.1. Document classifiers

Classifier	Description
Distances of vectors	These are the simplest classifiers. For each category and document representative vectors are produced. Some measure of the distance must be defined. This classifier starts by counting all these distances between vectors of categories and a document's vector and the category closest to the document is chosen. Often the dot product cosine value of vectors is used instead of the distance.
Decision Tree	Algorithms that are used to select informative words based on an information gain criterion and predict categories of document according to occurrences of word combinations.
Naive Bayes	Probabilistic classifiers using joint probabilities of words and categories to calculate the category of a given document. Naive Bayes approach is far more efficient than many other approaches with the exponential complexity. Naive Bayes based systems are probably the most frequently used systems in text categorization.
kNN	It is the k-nearest neighbours classification. This system ranks k-nearest documents from the training set and use categories of these documents to predict a category of a given document. kNN belongs to m-ary classifiers
Rocchio algorithm	Uses vector-space model for document classification. The basic idea uses summing vectors of document and categories with positive or negative weights, which depends on belonging of a document to a given category. The weakness of this method is the assumption of one centroid (a group of vectors) per category. This brings problems, when documents with very different vectors belong to the same category.
RIPPER	Is a nonlinear rule learning algorithm. It uses a statistical data to create simple rules for each category and then uses conjunctions of these rules do determine whether the given document belongs to the given category (Cohen and Singer 1996).
Sleeping	Based on the idea of combining predictions of several classifiers. It is

experts	important to obtain a good classifier, the master algorithm, which combines results of these classifiers. Empirical evidence shows, that multiplicative updates of weights of classifiers are the most efficient (Cohen and Singer 1996).
Support Vector Machines	A learning method introduced by Cortes and Vapnik (Cortes and Vapnik 1995). This method is based on the Structural Risk Minimization principle and mapping of input vectors in high-dimensional feature space. Experimental results show, that SVM is good method for text categorization. It reaches very good results in the high-dimensional feature space, avoids overfitting and does not need a feature selection. SVM method has also one of best results in efficiency of text categorization (Joachims 1998).

All these are well-known machine learning techniques that can be used to solve many other problems. However, the essential part of the text categorization is the “similarity measure” of two documents or a document and a category. To count this measure the feature sets or vectors are used. There are many possibilities for how to define such a measure. Often the probability measure is used. Some systems work also with the dot product of some feature vectors. The goal of the machine learning is then to train all parameters and thresholds of the algorithm to obtain best similarities for documents that belong to the same category.

3.3 Text Cleansing Operations

This section describes basic text operations that aim at cleansing documents by eliminating non-discriminating words, and thus reducing drastically text complexity, and therefore allowing better performance in document retrieval and processing. Such operations are part of the feature construction process, referred previously.

3.3.1 Tokenization

Given a character sequence and a defined document unit, tokenization is the task of chopping it up into pieces, called *tokens*, perhaps at the same time throwing away certain characters, such as punctuation. Here is an example of tokenization:

Input: "The construction sector is characterised by collaboration"

Output:

the	construction	sector	is	characterised	by	collaboration

These tokens are often loosely referred to as terms or words, but it is sometimes important to make a type/token distinction. A *token* is an instance of a sequence of characters in some particular document that are grouped together as a useful semantic unit for processing. A *type* is the class of all tokens containing the same character sequence. A *term* is a (perhaps normalized) type that is included in the IR system's dictionary. The set of index terms could be entirely distinct from the tokens, for instance, they could be semantic identifiers in a taxonomy, but in practice in modern IR systems they are strongly related to the tokens in the document. However, rather than being exactly the tokens that appear in the document, they are usually derived from them by various normalization processes.

3.3.2 n-Grams

Other languages make the problem harder in new ways. German has *compound nouns* without spaces (e.g., *Computerlinguistik* 'computational linguistics'; *Lebensversicherungsgesellschaftsangestellter* 'life insurance company employee'). Retrieval systems for German greatly benefit from the use of a *compound-splitter* module, which is usually implemented by seeing if a word can be subdivided into multiple words that appear in a vocabulary. This phenomenon reaches its limit case with major East Asian Languages (e.g., Chinese, Japanese, Korean, and Thai), where text is written without any spaces between words. One approach here is to perform word segmentation as prior linguistic processing. Methods of word segmentation vary from having a large vocabulary and taking the longest vocabulary match with some heuristics for unknown words to the use of machine learning sequence models, such as hidden Markov models or conditional random fields, trained over hand-segmented words.

3.3.3 Stop Words

Sometimes, some extremely common words which would appear to be of little value in helping select documents matching a user need are excluded from the vocabulary entirely. These

words are called *stop words*. The general strategy for determining a stop list is to sort the terms by collection frequency (the total number of times each term appears in the document collection), and then to take the most frequent terms, often hand-filtered for their semantic content relative to the domain of the documents being indexed, as a stop list, the members of which are then discarded during indexing. An example of a stop list is shown in Table 3.2.

Table 3.2. A stop list of 24 semantically non-selective words

a	in	and	are	as	at	be	by	for	from	has	he
in	is	it	of	on	that	the	to	was	were	will	with

Using a stop list significantly reduces the number of postings that a system has to store. And a lot of the time not indexing stop words does little harm: keyword searches with terms like 'the' and 'by' do not seem very useful. However, this is not true for phrase searches. The phrase query "President of the United States", which contains two stop words, is more precise than 'President' AND 'United States'. The meaning of "flights to London" is likely to be lost if the word to is stopped out. A search for Vannevar Bush's article "As one may think" will be difficult if the first three words are stopped out, and the system searches simply for documents containing the word think. Some special query types are disproportionately affected. Some song titles and well known pieces of verse consist entirely of words that are commonly on stop lists ("To be or not to be", "Let It Be", "I don't want to be"...).

The general trend in IR systems over time has been from standard use of quite large stop lists (200–300 terms) to very small stop lists (7–12 terms) to no stop list whatsoever. Web search engines generally do not use stop lists. Some of the design of modern IR systems has focused precisely on how it can be exploited the statistics of language so as to be able to cope with common words in better ways.

3.3.4 Capitalization

A common strategy is to do case-folding by reducing all letters to lower case. Often this is a good idea: it will allow instances of Automobile at the beginning of a sentence to match with a query of automobile. It will also help on a web search engine when most of your users type in ferrari when they are interested in a Ferrari car. On the other hand, such case folding can equate words that might better be kept apart. Many proper nouns are derived from common nouns and so are distinguished only by case, including companies (General Motors, The Associated Press), government organizations (the Fed vs. fed) and person names (Bush, Black).

For English, an alternative to making every token lowercase is to just make some tokens lowercase. The simplest heuristic is to convert to lowercase words at the beginning of a sentence and all words occurring in a title that is all uppercase or in which most or all words are capitalized. These words are usually ordinary words that have been capitalized. Mid-sentence capitalized words are left as capitalized (which is usually correct). This will mostly avoid case-folding in cases where distinctions should be kept apart. The same task can be done more accurately by a machine learning sequence model which uses more features to make the decision of when to case-fold. This is known as true casing. However, trying to get capitalization

right in this way probably does not help if users usually use lowercase regardless of the correct case of words. Thus, lowercasing everything often remains the most practical solution.

3.3.5 Stemming and lemmatization

For grammatical reasons, documents are going to use different forms of a word, such as ‘organize’, ‘organizes’, and ‘organizing’. Additionally, there are families of derivationally related words with similar meanings, such as ‘democracy’, ‘democratic’, and ‘democratization’. In many situations, it seems as if it would be useful for a search for one of these words to return documents that contain another word in the set. The goal of both stemming and lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form, as shown in Table 3.3.

Table 3.3. Stemmed words example

words	Stemmed word
am, are, is	be
car, cars, car’s, cars’	car

However, the two words differ in their flavour. Stemming usually refers to a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes. Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma. If confronted with the token saw, stemming might return just *s*, whereas lemmatization would attempt to return either *see* or *saw* depending on whether the use of the token was as a verb or a noun. The two may also differ in that stemming most commonly collapses derivationally related words, whereas lemmatization commonly only collapses the different inflectional forms of a lemma.

The most common algorithm for stemming English, and one that has repeatedly been shown to be empirically very effective, is Porter’s algorithm (Porter 1980). The entire algorithm is too long and intricate to present here, but it will be described its general nature. Porter’s algorithm consists of 5 phases of word reductions, applied sequentially. Within each phase there are various conventions to select rules, such as selecting the rule from each rule group that applies to the longest suffix. In the first phase, this convention is used with the following rule group, as described in Table 3.4. For a full comprehension of the Porter stemming algorithm, please refer to Annex E - Porter stemming algorithm.

Table 3.4. Example of stemming algorithm

Rule	Example
SSES → SS	caresses → caress
IES → I	ponies → poni
SS → SS	caress → caress

S →	cats → cat

3.4 The Vector Space Model

As already mentioned, the applied techniques for automatic text classification includes vector space model (VSM), artificial neural networks (ANN), K nearest neighbour (KNN), Naives Bayes (NB) and support vector machine (SVM), and gained popularity among text mining and information retrieval (IR) researchers.

Any text-based system requires some representation of documents, and the appropriate representation depends on the kind of task to be performed (Lewis 1992). Moreover, the ability to accurately perform a classification task depends on the representation of documents to be classified (Quinlan 1983). Different from data mining that handles the well-structured data, text mining deals with a collection of semi-structured, even unstructured documents. This makes that one of the main themes supporting text mining is transforming text into numerical vectors, i.e., text representation.

The performance of the most applied language models is greatly influenced by document vector in the process of document formalization as it represents semantic information of documents. In addition, the primary technique for document formalization, called “Term weighting”, is a useful technique for keyword extraction and document classification. By far, several term weighting algorithms are presented, such as *tf-idf* (term frequency – inverse document frequency), Mutual Information, the weight of evidence for text, Information Gain, Expected Cross Entropy, etc.. Such approaches mainly depend on the frequency of terms, called positive weight (PW) function. Some other algorithms introduce more additional statistical information into consideration, position, distribution, HTML tags, contextual information and length of the term have been collected and used into the algorithm.

According to (Xia and Du 2011), the document title is a useful indication of the document content, which always contains primary information of the document. However, the terms in document title may not have high term frequency or other positive statistical characteristics. **In this thesis, it is advocated that such statement is not always true, as will be demonstrated in the section related with the empirical evidences of the presented work.**

In information retrieval, documents are generally identified by sets of terms or keywords that are collectively used to represent their contents. Vector space model (VSM) (Salton, Wong and Yang, A vector space model for automatic indexing 1975) is one of the mostly used models for representation, because of its conceptual simplicity and the appeal of the underlying metaphor of using spatial proximity for semantic proximity. Generally, there are two kinds of works included in text representation: indexing and term weighting (Lewis 1992). Indexing is the job to assign indexing terms for documents. Term weighting is the job to assign the weight for each term, which measures the importance of a term in a document. For the sake of clarity, this work considers indexing and term weighting as two components of text representation scheme, and will not discuss the effectiveness of indexing and term weighting individually.

Currently, there are many term weighting methods, which are derived from the different assumptions for terms’ characteristics in texts. For instance, *idf* (inverse document frequency) assumes that the importance of a term relative to a document is inversely proportional to the

frequency of occurrence of this term in all the documents, while *ridf* (residual inverse documents frequency) holds the assumption that the importance of a term should be measured by the difference between its actual frequency of occurrence in documents and the predicted frequency of occurrence by Poisson distribution (random occurrence).

Essentially, in the task of text classification, which includes IR and text categorization (TC) (Lewis 1992), this thesis is mainly focused on two kinds of properties of the indexing term: semantic and statistical qualities (Hidalgo 2003). Semantic quality is related to a term's meaning, i.e., to how much extent the index term can describe text content. Statistical quality is related to the discriminative (resolving) power of the index term to identify the category of a document in which the term occurs.

One of the purposes of this doctoral work is to study the effectiveness of different representation methods in text classification. Here, it is worth to reaffirm that text classification includes both information retrieval and text categorization though many researchers regard text categorization is the same as text classification. Although *tf-idf*, Latent Semantic Index (LSI) and multiword have been proposed for a long time, there are no substantial comparative studies on these indexing methods, and no results are reported concerning their classification performances. Despite that some indexing methods are accepted as having superior qualities, such as LSI and multi-word with better semantic quality, there is no clear evidence to show to which extent their preferred quality will produce better performances in text classification. Zhang, Yoshida and Tang (Zhang, Yoshida and Tang 2011) conducted a study on text classification comparing *tf-idf*, LSI and multi-words. One of the major conclusions of this study is that, from the authors' experimental results, it can be seen that the number of dimension is still a decisive factor for indexing when using different indexing methods for classification. It is worth mentioning that *tf-idf* is still considered by far the algorithm with better performance concerning computation complexity.

Up to now it only has been described indexes that support Boolean queries: a document either matches or does not match a query. In the case of large document collections, the resulting number of matching documents can far exceed the number a human user could possibly sift through. Accordingly, it is essential for a search engine to rank-order the documents matching a query. In order to do this, the search engine computes, for each matching document, a score with respect to the query at hand.

Scoring has been related to whether or not a query term is present in a zone within a document. The next logical step should be: a document or zone that mentions a query term more often has more to do with that query and therefore should receive a higher score. To motivate this, the notion of a free text query should be recalled: a query in which the terms of the query are typed freeform into the search interface, without any connecting search operators (such as Boolean operators). This query style, which is extremely popular on the web, views the query as simply a set of words. A plausible scoring mechanism then is to compute a score that is the sum, over the query terms, of the match scores between each query term and the document.

Towards this end, each term in a document should have a weight assigned for it that depends on the number of occurrences of the term in the document. The idea is to compute a score between a query term *t* and a document *d*, based on the weight of *t* in *d*. The simplest approach

is to assign the weight to be equal to the number of occurrences of term *t* in document *d*. This weighting scheme is referred to as term frequency and is denoted $tf_{t,d}$, with the subscripts denoting the term and the document in order.

For a document *d*, the set of weights determined by the *tf* weights above (or indeed any weighting function that maps the number of occurrences of *t* in *d* to a positive real value) may be viewed as a quantitative digest of that document. In this view of a document, known in the literature as the bag of words model, the exact ordering of the terms in a document is ignored but the number of occurrences of each term is material (in contrast to Boolean retrieval). Only the information regarding the number of occurrences of each term is retained. Thus, the document "Mary is quicker than John" is, in this view, identical to the document "John is quicker than Mary". Nevertheless, it seems intuitive that two documents with similar bag of words representations are similar in terms of content.

3.4.1 Inverse document frequency

Raw term frequency suffers from a critical problem: all terms are considered equally important when it comes to assessing relevancy on a query. In fact certain terms have little or no discriminating power when used to determine relevance. For instance, a collection of documents on the building and construction industry is likely to have the term 'building' in almost every document. To this end, a mechanism is introduced for attenuating the effect of terms (that occur too often in the collection) to be meaningful for relevance determination. An immediate idea is to scale down the term weights with high collection frequency, defined to be the total number of occurrences of a term in the collection. The idea would be to reduce the *tf* weight of a term by a factor that grows with its collection frequency.

Instead, it is more common place to use for this purpose the document frequency df_t, defined to be the number of documents in the collection that contain a term *t*. This is because when trying to discriminate between documents for the purpose of scoring it is better to use a document-level statistic (such as the number of documents containing a term) than to use a collection-wide statistic for the term.

How is the document frequency *df* of a term used to scale its weight? Denoting as usual the total number of documents in a collection by N, the inverse document frequency (*idf*) of a term *t* is defined as follows:

$$idf_t = \log\left(\frac{N}{df_t}\right) \tag{3.1}$$

3.1.1 *tf-idf* weighting

It works by combining the definitions of term frequency and inverse document frequency, to produce a composite weight for each term in each document.

The *tf-idf* weighting scheme assigns to term *t* a weight in document *d* given by

$$tfidf_{t,d} = tf_{t,d} \times idf_t \tag{3.2}$$

In other words, $tf\text{-}idf_{t,d}$ assigns to term *t* a weight in document *d* that is:

1. highest when *t* occurs many times within a small number of documents (thus lending high discriminating power to those documents);
2. lower when the term occurs fewer times in a document, or occurs in many documents (thus offering a less pronounced relevance signal);
3. lowest when the term occurs in virtually all documents.

tf-idf has evolved from *idf* which is proposed by Sparck Jones (K. S. Jones 2004) with heuristic intuition that a query term which occurs in many documents is not a good discriminator, and should be given less weight than one which occurs in few documents.

The basic idea of *tf-idf* comes from the language modelling theory, which argues that the terms in a given document can be divided into two categories: those words with eliteness and those words without eliteness (S. Robertson 2004), i.e., whether or not a term is relevant with the topic of a given document. Further, the eliteness of a term for a given document can be evaluated by *tf* and *idf* and in *tf-idf* formulation, it is used to measure the importance of a term in the document collection.

However, there are some criticisms of using *tf-idf* for text representation. The first one is that *tf-idf* is too 'ad hoc' because it is not directly derived from a mathematical model, although usually it is explained by Shannon's information theory (Caropreso, Matwin and Sebastiani 2001). The second criticism comes from that the dimensionality (size of feature set) in *tf-idf* for textual data is the size of the vocabulary across the entire dataset, resulting in that it brings about a huge computation on weighting all these terms (Manning and Schütze 2001). Other authors advocate that *tf-idf* lacks in term of relation between terms and their synonyms. Also, the classical *tf-idf* enables that longer documents are more likely to be retrieved, due to higher term frequencies – the same term appears more often.

Although there might be some controversy about the formalization of the weighting scheme, the objective of this thesis is not is evaluating how formal *tf-idf* is. Rather, it argues that the main advantage of traditional vector space model is its simplicity, which could describe unstructured documents with the form of vectors, making it possible to use various mathematic methods to deal with. The *tf-idf* is an efficient and simple algorithm for matching words in a query to documents that are relevant to that query. From the data collected within this thesis scope, using purely *tf-idf* ranking scheme, only a few documents returned were considered highly relevant to a particular query. The reason why *tf-idf* deserves full attention with respect to IR topic is that encoding *tf-idf* is straightforward, making it ideal for forming the basis for more complicated algorithms and query retrieval systems, which is *per se* one of the objectives to be addressed by this thesis. Therefore, this works considers using ontology-based semantic information management methods to improve traditional vector space model, creating a semantic vector space model. The idea is to enhance *tf-idf* representation scheme using external domain ontologies. **How such limitations related with *tf-idf* where overcome will be better detailed in chapters 4 and 5.**

3.4.2 The vector space model for scoring

In a vector space model a vector derived from a document *d* is denoted by $\vec{V}$(d), with one component in the vector for each dictionary term. Unless otherwise specified, the one may assume that the components are computed using the *tf-idf* weighting scheme, although the particular weighting scheme is irrelevant to the discussion that follows. The set of documents in a collection then may be viewed as a set of vectors in a vector space, in which there is one axis for each term. This representation loses the relative ordering of the terms in each document; recalling the example, where it was pointed out that the documents "Mary is quicker than John" and "John is quicker than Mary" are identical in such a bag of words representation.

How to quantify the similarity between two documents in this vector space? A first attempt might consider the magnitude of the vector difference between two document vectors. This measure suffers from a drawback: two documents with very similar content can have a significant vector difference simply because one is much longer than the other. Thus the relative distributions of terms may be identical in the two documents, but the absolute term frequencies of one may be far larger.

In order to compensate the document length effect, the standard way of quantifying the similarity between two documents *d1* and *d2* is to compute the cosine similarity of their vector representations $\vec{V}(d_1)$ and $\vec{V}(d_2)$, as follows:

$$sim(d_1, d_2) = \frac{\vec{V}(d_1) \cdot \vec{V}(d_2)}{|\vec{V}(d_1)||\vec{V}(d_2)|} \tag{3.3}$$

Where the numerator represents the *dot product* (also known as the *inner product*) of the vectors $\vec{V}(d_1)$ and $\vec{V}(d_2)$, while the denominator is the product of their *Euclidean lengths*. The dot product $\vec{x} \cdot \vec{y}$ of two vectors is defined as $\sum_{i=1}^{M} x_i y_i$. Let $\vec{V}(d)$ denote the document vector for *d*, with *M* component $\overrightarrow{V_1}(d)$s …$\overrightarrow{V_M}(d)$. The Euclidean length of *d* is defined to be $\sqrt{\sum_{i=1}^{M} \vec{V}_i^2(d)}$.The effect of the denominator of Equation is thus to *length-normalize* the vectors $\vec{V}(d_1)$ and $\vec{V}(d_2)$ to unit vectors $\vec{v}(d_1) = \frac{\vec{V}(d_1)}{|\vec{V}(d_1)|}$ and $\vec{v}(d_2) = \frac{\vec{V}(d_2)}{|\vec{V}(d_2)|}$. The equation can be re-written as

$$sim(d_1, d_2) = \vec{v}(d_1) \cdot \vec{v}(d_2) \tag{3.4}$$

Thus, the equation can be viewed as the dot product of the normalized versions of the two document vectors. This measure is the cosine of the angle *θ* between the two vectors, shown in Figure 3.8. What use is the similarity measure $sim(d_1, d_2)$? Given a document *d* (potentially one of the d_i in the collection), consider searching for the documents in the collection most similar to *d*. Such a search is useful in a system where a user may identify a document and seek others like it – a feature available in the results lists of search engines as a more like this feature. The problem is reduced to finding the document(s) most similar to *d* to that of finding the d_i with the highest dot products (*sim* values) $\vec{v}(d) \cdot \vec{v}(d_i)$. This could be done by computing

the dot products between $\vec{v}(d)$ and each of $\vec{v}(d_1)$, . . . , $\vec{v}(d_N)$, then picking off the highest resulting *sim* values.

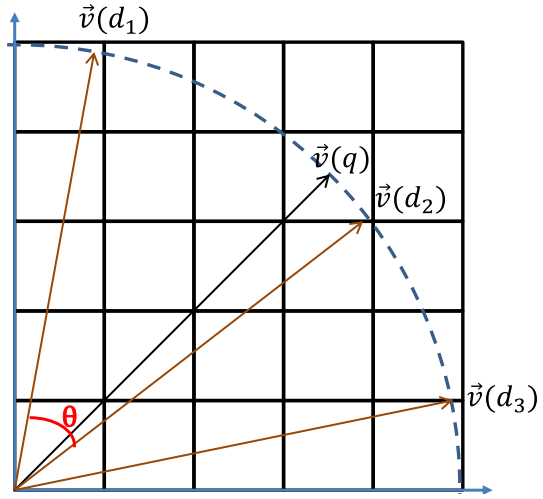


Figure 3.8. Cosine similarity illustrated. sim(d1, d2) = cos θ

Viewing a collection of *N* documents as a collection of vectors leads to a natural view of a collection as a *term-document matrix*: this is an *M*×*N* matrix whose rows represent the *M* terms (dimensions) of the *N* columns, each of which corresponds to a document. As always, the terms being indexed could be stemmed before indexing; for instance, architecture and architect would under stemming be considered as a single dimension. Figure 3.9, represents a *M*×*N* matrix, where documents are represented as rows $D_1 \ldots D_n$ and the terms are represented by columns $T_1 \ldots T_n$. Figure 3.9 depicts an example of a representation of tree different document vectors, each one with different term weights.

	T_1	T_2	...	T_t
D_1	W_{11}	W_{21}	...	W_{t1}
D_2	W_{12}	W_{22}	...	W_{t2}
:	:	:		:
:	:	:		:
D_n	W_{1n}	W_{2n}	...	W_{tn}

Figure 3.9. Term-document matrix

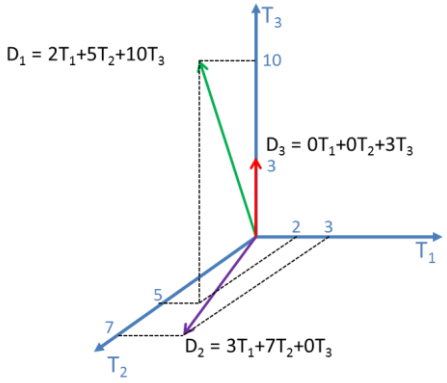


Figure 3.10. Hyper-plan vector representation

There is a far more compelling reason to represent documents as vectors: a query can also be viewed as a vector. The key idea is to assign to each document *d* a score equal to the dot product $\vec{v}(q) \cdot \vec{v}(d)$.

Shortly, by viewing a query as a "bag of words", it is possible to treat it as a small document. As a consequence, it can be adopted the cosine similarity between the query vector and a

document vector as a measure of the score of the document for that query. The resulting scores can then be used to select the top-scoring documents for a query. Thus:

$$score(q,d) = \frac{\vec{V}(q) \cdot \vec{V}(d)}{\left|\vec{V}(q)\right|\left|\vec{V}(d)\right|} \quad (3.5)$$

A document may have a high cosine score for a query even if it does not contain all query terms. Note that the preceding discussion does not focus on any specific weighting of terms in the document vector, although for the present it may be understood as *tf-idf* weights. In fact, a number of weighting schemes are possible for query as well as document vectors.

Computing the cosine similarities between the query vector and each document vector in the collection, sorting the resulting scores and selecting the top K documents can be expensive (in computational resources) — a single similarity computation can entail a dot product in tens of thousands of dimensions, demanding tens of thousands of arithmetic operations.

In a typical setting there is a collection of documents each represented by a vector, a free text query represented by a vector, and a positive integer K. the objective relies on seeking the K documents of the collection with the highest vector space scores on the given query. Typically, such quest uses the K top documents in ordered by decreasing score; for instance many search engines use K = 10 to retrieve and rank-order the first page of the ten best results.

Figure 3.11 gives the basic algorithm for computing vector space scores. The array Length holds the lengths (normalization factors) for each of the N documents, whereas the array Scores holds the scores for each of the documents. When the scores are finally computed in Step 9, all that remains in Step 10 is to pick off the K documents with the highest scores.

The outermost loop beginning Step 3 repeats the updating of Scores, iterating over each query term *t* in turn. In Step 5 it is calculated the weight in the query vector for term t. Steps 6-8 update the score of each document by adding in the contribution from term *t*. This process of adding in contributions one query term at a time is sometimes known as *term-at-a-time* scoring or accumulation, and the *N* elements of the array *Scores* are therefore known as accumulators. For this purpose, it would appear necessary to store, with each postings entry, the weight $wf_{t,d}$ of term *t* in document *d* (using *tf-idf* for this weight). In fact this is wasteful, since storing this weight may require a floating point number. Two ideas help alleviate this space problem. First, the use of inverse document frequency avoids to precompute idf_t; it suffices to store N/df_t at the head of the postings for *t*. Second, the term frequency $tf_{t,d}$ for each postings entry is stored. Finally, Step 12 extracts the top K scores – this requires a priority queue data structure, often implemented using a heap. Such a heap takes no more than 2*N* comparisons to construct, following which each of the K top scores can be extracted from the heap at a cost of O(log *N*) comparisons.

It shall be noticed that the general algorithm of Figure 3.11 does not prescribe a specific implementation of how to traverse the postings lists of the various query terms; it may traverse them one term at a time as in the loop beginning at Step 3, or it could in fact traverse them concurrently. In such a concurrent postings traversal it is computed the scores of one document at a time, so that it is sometimes called document-at-a-time scoring.

CosineScore(*q*)

```
1 float Scores[N] = 0
2 Initialize Length[N]
3 for each query term t
4 do calculate w_{t,q} and fetch postings list for t
5   for each pair(d, tf_{t,d}) in postings list
6   do Scores[d] += wf_{t,d} × w_{t,q}
7 Read the array Length[d]
8 for each d
9 do Scores[d] = Scores[d]/Length[d]
10 return Top K components of Scores[]
```

Figure 3.11. The basic algorithm for computing vector space scores

This section presented the computational aspects of vector space scoring. Luhn (1958) describes some of the earliest reported applications of term weighting. His paper dwells on the importance of medium frequency terms (terms that are neither too commonplace nor too rare) and may be thought of as anticipating *tf-idf* and related weighting schemes. Spärck Jones (1988) builds on this intuition through detailed experiments showing the use of inverse document frequency in term weighting. A series of extensions and theoretical justifications of *idf* are due to Salton and Buckley (Salton and Buckley 1988) Robertson and Jones (Robertson and Jones 1976), Croft and Harper (1988) and Papineni (2001). Singhal et al. (1996) develop pivoted document length normalization. Probabilistic language models develop weighting techniques that are more nuanced than *tf-idf*.

It can be observed that by assigning a weight for each term in a document, a document may be viewed as a vector of term weights, one for each term in the collection. The SMART information retrieval system at Cornell (Salton, Wong and Yang 1975) was perhaps the first to view a document as a vector of weights. The basic computation of cosine scores is due to Zobel and Moffat (2006). The two query evaluation strategies term-at-a-time and document-at-a-time are discussed by Turtle and Flood (1995).

Beyond the notation for *tf-idf* term weighting scheme, Moffat and Zobel (1998) sought to set up a space of feasible weighting functions through which hill climbing approaches could be used to begin with weighting schemes that performed well, then make local improvements to identify the best combinations. However, they report that such hill-climbing methods failed to lead to any conclusions on the best weighting schemes.

3.5 Evaluation in information retrieval

Information retrieval has been developed as a highly empirical discipline, requiring careful and thorough evaluation to demonstrate the superior performance of novel techniques on representative document collections.

Aiming to measure *ad hoc* information retrieval effectiveness in the standard way, a collection consisting of three things needs to be tested:

1. A document collection.
2. A test suite of information needs, expressible as queries.
3. A set of relevance judgments, normally a binary assessment of either relevant or non-relevant for each query-document pair.

The standard approach to information retrieval system evaluation revolves around the notion of *relevant* and *non-relevant* documents. With respect to a user information need, a document in the test collection is given a binary classification as either relevant or non-relevant. This decision is referred to as the *gold standard* or *ground truth* judgment of relevance. The test document collection and suite of information needs have to be of a reason-able size, but considering this point, there are no standard rules referring to the number of documents to be used for testing. Although the performance must be conducted over fairly large test sets, as results are highly variable over different documents and information needs.

Relevance is assessed relatively to an information need, not a query. For example, an information need might be: “Information on whether drinking red wine is more effective at reducing your risk of heart attacks than white wine”. This might be translated into a query such as: wine AND red AND white AND heart AND attack AND effective

A document is relevant if it addresses the stated information need, not because it just happens to contain all the words in the query. This distinction is often misunderstood in practice because the information need is not obvious. Nevertheless, an information need is present. If a user types “python” into a web search engine, he/she might want to know where to purchase a pet python. Or they might want information on the programming language Python. From a one word query, it is very difficult for a system to know what the information need is. But, nevertheless, the user has one need, and can judge the returned results on the basis of their relevance to it. In order to evaluate a system, it is required an evident expression of an information need, which can be used for judging returned documents as relevant or non-relevant. At this point, a simplification can be made: relevance can reasonably be thought of as a scale, with some documents highly relevant and others marginally so. However, for the purpose on illustrating the evaluation metrics on IR, it is assumed just a binary decision of relevance.

Many systems contain various weights (often known as parameters) that can be adjusted to tune system performance. It is wrong to report results on a test collection which was obtained by tuning these parameters to maximize performance on that collection. That is because such tuning overstates the expected performance of the system, because the weights will be set to maximize performance on one particular set of queries rather than for a random sample of

queries. In such cases, the correct procedure is to have one or more development test collections, and to tune the parameters on the development test collection. The tester then runs the system with those weights on the test collection and reports the results on that collection as an unbiased estimation of performance.

Given these ingredients, how is system effectiveness measured? The two most frequent and basic measures for information retrieval effectiveness are precision and recall. These are first defined for the simple case where an IR system returns a set of documents for a query. In chapter 6, such measures will be illustrated in detail to rank retrieval situations.

Precision(*P*) is the fraction of retrieved documents that are relevant, as follows:

$$Precision = \frac{\#(relevant\ items\ retrieved)}{\#(retrieved\ items)} = P(relevant|retrieved) \quad (3.6)$$

Recall(*R*) is the fraction of relevant documents that are retrieved, as follows:

$$Recall = \frac{\#(relevant\ items\ retrieved)}{\#(relevant\ items)} = P(retrieved|relevant) \quad (3.7)$$

These notions can be made clear by examining Table 3.5.

Table 3.5. Typical evaluation measures for IR

	Relevant	**Non-relevant**
Retrieved	true positives (tp)	false positives (fp)
Not Retrieved	false negatives (fn)	true negatives (tn)

$$P = \frac{tp}{tp + fp} \quad (3.8)$$

$$R = \frac{tp}{tp + fn} \quad (3.9)$$

An obvious alternative that may occur to the reader is to judge an information retrieval system by its accuracy, that is, the fraction of its classifications that are correct. In terms of the contingency table above, accuracy = (*tp* + *tn*)/(*tp* + *fp* + *fn* + *tn*). This seems plausible, since there are two actual classes, namely *relevant* and *non-relevant*, and an information retrieval system can be thought of as a two-class classifier which attempts to label them as such (it retrieves the subset of documents which it believes to be relevant). This is precisely the effectiveness measure often used for evaluating machine learning classification problems.

There is a good reason why accuracy is not an appropriate measure for information retrieval problems. In almost all circumstances, the data is extremely skewed: normally over 99.9% of the documents are in the non-relevant category. A system tuned to maximize accuracy can appear to perform well by simply deeming all documents non-relevant to all queries. Even if the system is quite good, trying to label some documents as relevant will almost always lead to a high rate of false positives. However, labelling all documents as non-relevant is completely

unsatisfactory to an information retrieval system user. Users are always going to want to see some documents, and can be assumed to have a certain tolerance for seeing some false positives providing that they get some useful information. The measures of precision and recall concentrate the evaluation on the return of true positives, asking what percentage of the relevant documents has been found and how many false positives have also been returned.

The advantage of having the two numbers for precision and recall is that one is more important than the other in many circumstances. Typical web surfers would like every result on the first page to be relevant (high precision) but have not the slightest interest in knowing let alone looking at every document that is relevant. In contrast, various professional searchers such as paralegals and intelligence analysts are very concerned with trying to get as high recall as possible, and will tolerate fairly low precision results in order to get it. Individuals searching their hard disks are also often interested in high recall searches. Nevertheless, the two quantities clearly trade off against one another: it is always possible to get a recall of 1 (but very low precision) by retrieving all documents for all queries! Recall is a non-decreasing function of the number of documents retrieved. On the other hand, in a good system, precision usually decreases as the number of documents retrieved is increased. In general the objective is to get some amount of recall while tolerating only a certain percentage of false positives.

A single measure that trades off precision versus recall is the *F* measure, which is the weighted harmonic mean of precision and recall and is calculated as follows:

$$F = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P + R} \qquad \text{where } \beta^2 = \frac{1-\alpha}{\alpha} \qquad (3.10)$$

where $\alpha \in [0, 1]$ and thus $\beta^2 \in [0,\infty]$. The default balanced *F* measure equally weights precision and recall, which means making $\alpha = 1/2$ or $\beta = 1$. It is commonly written as F_1, which is short for $F_{\beta=1}$, even though the formulation in terms of α more transparently exhibits the *F* measure as a weighted harmonic mean. When using $\beta = 1$, the formula (11) simplifies to:

$$F_{\beta=1} = \frac{2PR}{P+R} \qquad (3.11)$$

However, using an even weighting is not the only choice. Values of $\beta < 1$ emphasize precision, while values of $\beta > 1$ emphasize recall. For example, a value of $\beta = 3$ or $\beta = 5$ might be used if recall is to be emphasized. Recall, precision, and the F measure are inherently measures between 0 and 1, but they are also very commonly written as percentages, on a scale between 0 and 100.

Why it is used a harmonic mean rather than the simpler average (arithmetic mean)? It was mentioned previously that it is possible to get 100% recall by just returning all documents, and therefore always get a 50% arithmetic mean by the same process. This strongly suggests that the arithmetic mean is an unsuitable measure to use. In contrast, if it is assumed that 1 document in 10,000 is relevant to the query, the harmonic mean score of this strategy is 0.02%. The harmonic mean is always less than or equal to the arithmetic mean and the geometric

mean. When the values of two numbers differ greatly, the harmonic mean is closer to their minimum than to their arithmetic mean.

Definition and implementation of the notion of relevance to a query got off to a rocky start in 1953. Swanson (1988) reported that in an evaluation in that year between two teams, they agreed that 1390 documents were variously relevant to a set of 98 questions, but disagreed on a further 1577 documents, and the disagreements were never resolved.

Rigorous formal testing of IR systems was first completed in the Cranfield experiments, beginning in the late 1950s. A retrospective discussion of the Cranfield test collection and experimentation with it can be found in (Cleverdon 1991). The other seminal series of early IR experiments were those on the system by Gerard Salton and colleagues (Salton 1991). The TREC[2] (Text REtrieval Conference) evaluations are described in detail by Voorhees and Harman (Voorhees e Harman 2005).

The notions of recall and precision were first used by Kent et al. (2007), although the term precision did not appear until later. The *F* measure (or, rather its complement $E = 1 - F$) was introduced by van Rijsbergen (van Rijsbergen 1979). He provides an extensive theoretical discussion, which shows how adopting a principle of decreasing marginal relevance (at some point a user will be unwilling to sacrifice a unit of precision for an added unit of recall) leads to the harmonic mean being the appropriate method for combining precision and recall (and hence to its adoption rather than the minimum or geometric mean).

2 The Text REtrieval Conference (TREC) is an on-going series of workshops focusing on a list of different information retrieval (IR) research areas, or tracks. It is co-sponsored by the National Institute of Standards and Technology (NIST) and the Intelligence Advanced Research Projects Activity (part of the office of the Director of National Intelligence).

3.6 Clustering

Text clustering is one of the fundamental functions in text mining (Fan, et al. 2005). Clustering means to divide a collection of text documents into different category groups so that documents in the same category group describe the same topic such as classic music or Chinese history. There are many uses of clustering in real applications, for example, grouping the Web search results and categorizing digital documents. Unlike clustering structured data, clustering text data faces a number of new challenges. Among others, the volume of text data, dimensionality, sparsity, and complex semantics are the most important ones. These characteristics of text data require clustering techniques to be scalable to large and high dimensional data, and able to handle sparsity and semantics.

Different to the structured data stored in relational databases, text data sources are either semi-structured data (e.g. XML) or unstructured (e.g. free text). However, most existing clustering algorithms were designed for structured data. In order to apply them to text data, the original text formats have to be transformed into structured forms.

One difficulty in clustering large text data is to understand and interpret the clustering results. If the number of text documents was small, a cluster could be understood by looking into the content of all documents in different clusters. If the number of text documents is large, reading the content of all documents becomes infeasible. Instead of looking into the document content, a few keywords can be extracted from each cluster that can best represent the semantic topic of the cluster.

With the abundance of text documents through World Wide Web and corporate document management systems, the dynamic partitioning of texts into previously unseen categories ranks top on the priority list for all business intelligence systems. However, current text clustering approaches still suffer from major problems that greatly limit their practical applicability.

First, text clustering is mostly seen as an objective method, which delivers one clearly defined result, which needs to be “optimal” in some way. This, however, runs contrary to the fact that different people have quite different needs with regard to clustering of texts, because they may view the same documents from completely different perspectives (e.g., a business view vs. a technical view; also cf. (Macskassy, et al. 1998)). Thus, there is a clear need of subjective criteria allowing for a diversity of views from which to look at the clustering task.

Second, text clustering typically is a clustering task working in a high-dimensional space where each word is seen as a potential attribute for a text. Empirical and mathematical analysis, however, has shown that — in addition to computational inefficiencies— clustering in high-dimensional spaces is very difficult, because every data point tends to have the same distance from all other data points (Beyer, et al. 1999).

Third, text clustering *per se* is often rather useless, unless it is combined with an explanation of why particular texts were categorized into a particular cluster, i.e., one output desired from clustering in practical settings is the explanation of why a particular cluster result was produced rather than the result itself. A common method for producing explanations is the learning of

rules based on the cluster results. Again, however, this approach suffers from the high number of features chosen for computing clusters.

Though there are of course different approaches for clustering, simple ones like K-Means or sophisticated ones, based on the consideration virtually all algorithms working on large feature vectors will eventually face the same principal problems without really approaching the matters of subjectivity and explainability (Hotho, Staab and Maedche 2001).

Clustering algorithms group a set of documents into subsets or clusters. The algorithms' goal is to create clusters that are coherent internally, but clearly different from each other. In other words, documents within a cluster should be as similar as possible; and documents in one cluster should be as dissimilar as possible from documents in other clusters.

Clustering is the most common form of unsupervised learning. No supervision means that there is no human expert who has assigned documents to classes. In clustering, it is the distribution and makeup of the data that will determine cluster membership. A simple example is Figure 3.12. It is visually clear that there are four distinct clusters of points.

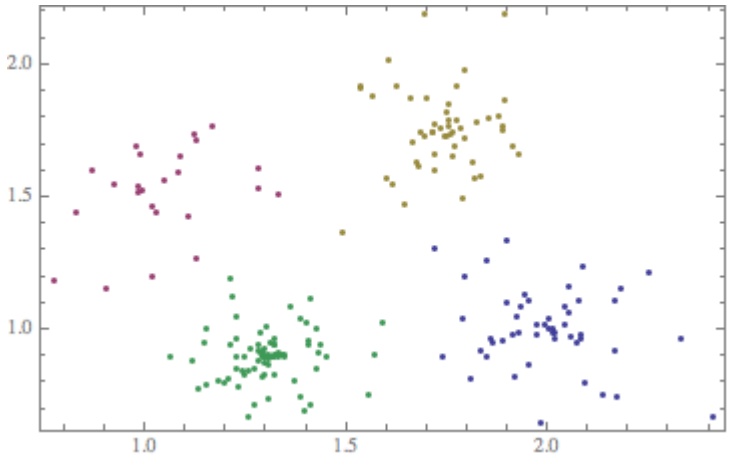


Figure 3.12. An example of a data set with a clear cluster structure

The difference between clustering and classification may not seem great at first. After all, in both cases there is a partition of a set of documents into groups. But as it will be described, the two problems are fundamentally different. Classification is a form of supervised learning: our goal is to replicate a categorical distinction that a human supervisor imposes on the data. In unsupervised learning, of which clustering is the most important example, there is no "teacher" to guide the process.

The key input to a clustering algorithm is the distance measure. In Figure 3.12, the distance measure is distance in the 2D plane. This measure suggests four different clusters in the figure. In document clustering, the distance measure is often also Euclidean distance. Different distance measures give rise to different clusters. Thus, the distance measure is an important means by which the outcome of clustering can be influenced. There are many methods to measure this distance, such as cosine similarity and Minkowski distance, including Euclidean, Manhattan, and Maximum distances (Anderberg 1973).

Flat clustering creates a flat set of clusters without any explicit structure that would relate clusters to each other. A second important distinction can be made between hard and soft clustering algorithms. Hard clustering computes a hard assignment – each document is a member of exactly one cluster. The assignment of soft clustering algorithms is soft – a document's assignment is a distribution over all clusters. In a soft assignment, a document has fractional membership in several clusters. Latent semantic indexing, a form of dimensionality reduction, is a soft clustering algorithm.

In text clustering, a set of documents are represented in a matrix where each row vector $<t_1,t_2,\ldots,t_n>$ represents a document and each column represents a term or word in the vocabulary of the document set. Clustering algorithms, such as the Standard K-Means (MacQueen 1967) and its variations (Dhillon, Fan and Guan 2001), (Steinbach, Karypis and Kumar 2000), as well as the hierarchical clustering methods (Duda, Hart and Stork 2000), (Zhao and Karypis 2002), are used to cluster the matrix data. In many real applications, the matrix can be very large because of the large vocabulary and the number of documents. If the set of documents to be clustered contains many different categories of documents, the matrix can be very sparse. Most existing clustering algorithms are not effective in clustering high dimensional sparse data because these algorithms cluster data on the full space while clusters in sparse data often exist in subspaces. This situation makes scalable subspace clustering methods (Parsons, Haque and Liu 2004), (Jain, Murty and Flynn 1999) good candidates for text clustering.

The cluster hypothesis states the fundamental assumption made when using clustering in information retrieval.

Cluster hypothesis. Documents in the same cluster behave similarly with respect to relevance to information needs.

The hypothesis states that if there is a document from a cluster that is relevant to a search request, then it is likely that other documents from the same cluster are also relevant. This is because clustering puts together documents that share many terms.

3.6.1 Problem statement

The goal in hard flat clustering can be defined as follows. Given (i) a set of documents $D = \{d_1, \ldots, d_N\}$, (ii) a desired number of clusters K, and (iii) an *objective function* that evaluates the quality of a clustering, the idea is to compute an assignment $\gamma : D \rightarrow \{1, \ldots, K\}$ that minimizes (or, in other cases, maximizes) the objective function. In most cases, a demand is that γ is surjective, i.e., that none of the K clusters is empty.

The objective function is often defined in terms of similarity or distance between documents. Below, it will be described that the objective in K-means clustering is to minimize the average distance between documents and their centroids or, equivalently, to maximize the similarity between documents and their centroids. The discussion of similarity measures and distance metrics in the previous section also applies to this section. As in previous sections, both similarity and distance are used to talk about relatedness between documents.

For documents, the type of similarity wanted is usually topic similarity or high values on the same dimensions in the vector space model. For example, documents about China have high values on dimensions like Chinese, Beijing, and Mao, whereas documents about the UK tend to have high values for London, Britain, and Buckingham. The approximation of the topic similarity is done, with cosine similarity or Euclidean distance in vector space. If it is intend to capture similarity of a type other than topic, for example, similarity of language, then a different representation may be appropriate. When computing topic similarity, stop words can be safely ignored, but they are important clues for separating clusters of English (in which "the" occurs frequently and "la" infrequently) and French documents (in which "the" occurs infrequently and "la" frequently).

An alternative definition of hard clustering is that a document can be a full member of more than one cluster. Partitional clustering always refers to a clustering where each document belongs to exactly one cluster (but in a partitional hierarchical clustering all members of a cluster are logically also members of its parent). On the definition of hard clustering that permits multiple membership, the difference between soft clustering and hard clustering is that membership values in hard clustering are either 0 or 1, whereas they can take on any non-negative value in soft clustering.

Some researchers distinguish between exhaustive clusterings that assign each document to a cluster and non-exhaustive clusterings, in which some documents will be assigned to no cluster. Non-exhaustive clusterings in which each document is a member of either no cluster or one cluster are called exclusive.

A difficult issue in clustering is determining the number of clusters or *cardinality* of a clustering, which is denoted by *K*. Often *K* is nothing more than a good guess based on experience or domain knowledge.

Since the goal is to optimize an objective function, clustering is essentially a search problem. The brute force solution would be to enumerate all possible clusters and pick the best. However, there are exponentially many partitions, therefore this approach is not feasible. For this reason, most flat clustering algorithms refine an initial partitioning iteratively. If the search starts at an unfavourable initial point, the global optimum can be missed. Finding a good starting point is therefore another important problem that has to be solved in flat clustering.

3.6.2 K-Means

K-means is the most important flat clustering algorithm. Its objective is to minimize the average squared Euclidean distance of documents from their cluster centres where a cluster centre is defined as the mean or *centroid* $\vec{\mu}$ of the documents centroid in a cluster ω:

$$\vec{\mu}(\omega) = \frac{1}{|\omega|} \sum_{\vec{x} \in \omega} \vec{x} \tag{3.12}$$

The definition assumes that documents are represented as length-normalized vectors in a real-valued space in the familiar way. The ideal cluster in *K*-means is a sphere with the centroid as its center of gravity. Ideally, the clusters should not overlap.

A measure of how well the centroids represent the members of their clusters is the Residual Sum of Squares (RSS), the squared distance of each vector from its centroid summed over all vectors:

$$RSS_k = \sum_{\vec{x} \in \omega} |\vec{x} - \vec{\mu}(\omega_k)|^2 \tag{3.13}$$

$$RSS = \sum_{K=1}^{K} RSS_k \tag{3.14}$$

RSS is the objective function in *K*-means and the goal is to minimize it. Since *N* is fixed, minimizing RSS is equivalent to minimizing the average squared distance, a measure of how well centroids represent their documents. The algorithm is illustrated in Figure 3.13.

K-Means ($\{\vec{x}_1, ..., \vec{x}_N\}$, *K*)

($\vec{s}_1, \vec{s}_2, ..., \vec{s}_K$)←SelectRandomseeds($\{\vec{x}_1, ..., \vec{x}_N\}$, *K*)

for k←1 **to** K

do $\vec{\mu}_k \leftarrow \vec{s}_k$

while stopping criterion has not been met

do for k←1 **to** K

  do $\vec{\omega}_k \leftarrow \{\}$

  for n←1 **to** N

  do j←$\arg\min_{j'} |\vec{\mu}_{j'} - \vec{x}_n|$

    $\omega_j \leftarrow \omega_j \cup \{\vec{x}_n\}$ (reassignment of vectors)

for k←1 **to** K

do $\vec{\mu}_k \leftarrow \frac{1}{|\omega_k|} \sum_{\vec{x} \in \omega} \vec{x}$

return $\{\vec{\mu}_1, ..., \vec{\mu}_k\}$

Figure 3.13. The K-means algorithm

The first step of *K*-means is to select as initial cluster centres K randomly selected documents, the seeds. The algorithm then moves the cluster centres around in space in order to minimize RSS. As shown in Figure 3.13, this is done iteratively by repeating two steps until a stopping criterion is met: reassigning documents to the cluster with the closest centroid; and re-computing each centroid based on the current members of its cluster. One of the following termination conditions can be applied:

- A fixed number of iterations *I* has been completed. This condition limits the runtime of the clustering algorithm, but in some cases the quality of the clustering will be poor because of an insufficient number of iterations.
- Assignment of documents to clusters (the partitioning function γ) does not change between iterations. Except for cases with a bad local minimum, this produces a good clustering, but runtimes may be unacceptably long.

- Centroids $\vec{\mu}_k$ do not change between iterations. This is equivalent to γ not changing.
- Terminate when RSS falls below a threshold. This criterion ensures that the clustering is of a desired quality after termination. In practice, it is needed to combine it with a bound on the number of iterations to guarantee termination.
- Terminate when the decrease in RSS falls below a threshold θ. For small θ, this indicates that the convergence is near. Again, there is a need to combine it with a bound on the number of iterations to prevent very long runtimes.

Since there is only a finite set of possible clusters, a monotonically decreasing algorithm will eventually arrive at a (local) minimum. It is important, however, to break ties consistently, e.g., by assigning a document to the cluster with the lowest index if there are several equidistant centroids. Otherwise, the algorithm can cycle forever in a loop of clusters that have the same cost.

While this proves the convergence of *K*-means, there is unfortunately no guarantee that a global minimum in the objective function will be reached. This is a particular problem if a document set contains many outliers, documents that are far from any other documents and therefore do not fit well into any cluster. Frequently, if an outlier is chosen as an initial seed, then no other vector is assigned to it during subsequent iterations. Thus, it ends up with a singleton cluster (a cluster with only one document) even though there is probably a clustering with lower RSS.

It was stated previously that the number of clusters *K* is an input to most flat clustering algorithms. What to do if it is not possible to come up with a plausible guess for *K*?

A naive approach would be to select the optimal value of *K* according to the objective function, namely the value of *K* that minimizes RSS. Defining RSSmin(*K*) as the minimal RSS of all clusterings with *K* clusters, it can be observed that RSSmin(*K*) is a monotonically decreasing function in *K*, which reaches its minimum 0 for *K* = N where N is the number of documents. It would end up with each document being in its own cluster. Clearly, this is not an optimal clustering.

A heuristic method that gets around this problem is to estimate RSSmin(*K*) as follows. First it performs *i* (e.g., *i* = 10) clusterings with *K* clusters (each with a different initialization) and compute the RSS of each. Then it takes the minimum of the *i* RSS values. This minimum is denoted by dRSSmin(*K*). Now it is possible to inspect the values dRSSmin(*K*) as *K* increases and find the “elbow” in the curve – the point where successive decreases in dRSSmin become noticeably smaller.

A second type of criterion for cluster cardinality imposes a penalty for each new cluster – where conceptually starts with a single cluster containing all documents and then search for the optimal number of clusters *K* by successively incrementing *K* by one. To determine the cluster cardinality in this way, it is created a generalized objective function that combines two elements: distortion, a measure of how much documents deviate from the prototype of their clusters (e.g., RSS for *K*-means); and a measure of model complexity. The clustering is interpreted here as a model of the data. Model complexity in clustering is usually the number of clusters or a function thereof. For *K*-means, the selection criterion for *K* is given by:

$$K = \underset{K}{\operatorname{argmin}}[RSS_{\min}(K) + \lambda K] \quad (3.15)$$

where λ is a weighting factor. A large value of λ favours solutions with few clusters. For λ = 0, there is no penalty for more clusters and K = N is the best solution.

In the clustering process, the documents are grouped with bigger similarity into the same category, otherwise, assign them into different groups. Because initially it is assumed that terms in documents are not related, in the other words, semantics are not considered, these measures only count the term frequency in two documents. In addition to Euclidean distance and cosine similarity, Kullback- Leibler divergence is often used in clustering as a measure of how (dis)similar documents and clusters are (Xu e Croft 1999), (Muresan e Harper 2004) and (Kurland e Lee 2004).

However, in the eye of the human beholder, text documents exhibit the rich linguistic and conceptual structures that may let him discover patterns that are not explicit. Based on these considerations it is plausible to assume that, in order to improve the effectiveness and utility of text mining, there is a need to improve the conceptual background knowledge available to text mining algorithms which must be exploited. Therefore, there is a need to investigate clustering algorithms which takes advantage of conceptual background knowledge (Jing, et al. 2006). Considering this thesis target, the addition of conceptual background knowledge available in external domain ontologies play a key role as it is expected to improve the recall and precision measures of the *K*-means clustering algorithm applied in this work.

This section only focused on classical unsupervised clustering. An important current topic of research is how to use prior knowledge to guide clustering (e.g., (Ji e Xu 2006)) and how to incorporate interactive feedback during clustering (e.g., (Huang e Mitchell 2006)). Fayyad et al. (1998) propose an initialization for EM (Expectation-Maximization) clustering.

As a concluding remark to this section, it is worth to mention that this work builds upon the classical vector space model and brings a strong contribution by adding a semantic dimension into the classical approach.

The usage of statistical approaches only take into account the occurrence of words alone in documents, the purpose of this work is also to quantify the meaning of those words within the text, and more important, how those meanings are related with each other. Being able to quantify the relatedness of meanings of terms in the text can bring substantial improvements in terms of document classification (this will be discussed in chapter 6). Such improvements are intended to be measured, using performance metrics (precision and recall) adopting an unsupervised classification algorithm (K-Means).

It is also very important to mention here the adoption of domain ontologies (described in chapter 2) within the whole process. The semantic level can only be achieved if documents can be contextualized against something that brings meaning and is shared by a community of users.

This chapter presented an overview on information retrieval concepts, and the techniques used for the task of knowledge representation and classification. For the task of representing documents contents, the classical vector space model approach was presented in detail mainly due to: (i) its simplicity; (ii) targeting multiclass, soft classification approach; (iii) allows to

calculate a degree of similarity between queries and documents; and (iv) allows to rank documents according to their possible relevance. For the task of clustering, K-Means algorithm was presented and it is considered a relevant approach for evaluating the semantic enrichment due to its fastness, robustness and easier to understand. It is worth to restate that, the objective of this work is not focused in turning IR algorithms more efficient, instead, is to measure the effect on existing classifiers.

4

The Semantic Enrichment Model

"*Some men see things as they are and say why. I dream things that never were and say why not.*"

- Robert Francis Kennedy (1925 – 1968), American politician, New York Senator

The conceptual foundations of the work presented here are grounded on Collaboration, Knowledge, and Semantics (Figure 4.1). Collaboration is related to the work performed by a group of actors in the context of development of engineering projects. Knowledge is the 'currency' exchanged among actors collaborating within a project. Semantics is represented by the use of text mining techniques with the support of a domain ontology, which ensures that knowledge generated during each project is captured, transformed, and mined in order to

support actors in having a common understanding of the knowledge sources that are exchanged.

The overall aim of semantic enrichment process, addressed by this thesis, is to specify, develop, and evaluate, with a support of knowledge experts, a set of capabilities that promote effective and consistent knowledge representations (including capturing, indexing and classify) across corporate domain knowledge, expressed by a domain ontology, within collaborative construction environments.

Distributed knowledge workers and teams lack proactive system support for seamless and natural collaboration on applications like problem solving, conflict resolution, knowledge sharing and receiving expert advice on-demand. The ambition is to have innovative solutions to establish effective partnerships that enable collaboration, that drive creativity, improve productivity, and provide a holistic approach to implementing project phases. Such collaborative working environments of the future will be based on enhanced communication, advanced simulation services, improved visualisation, natural interaction and especially knowledge.

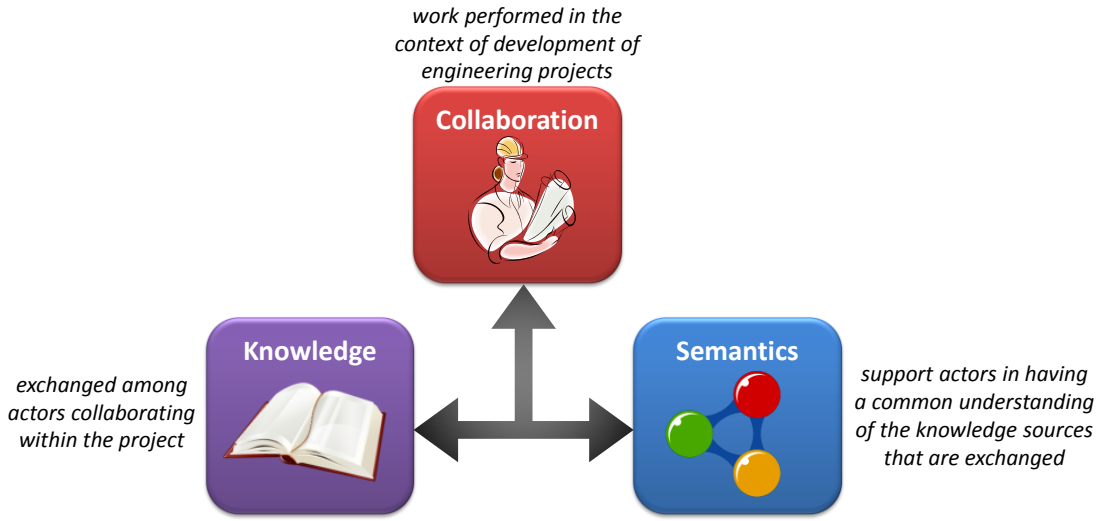


Figure 4.1. Conceptual Foundations of the work

It is vitally important for actors involved in engineering projects to obtain knowledge about the specific domain and to solve any problems that may arise. To achieve this knowledge, actors must learn from the experience of others. Domain experience transfer involves using knowledge gained during the completion of previous projects to maximize the achievement of current project objectives (Reuss and Tatum 1993). In order to share knowledge between similar projects, professionals have traditionally used techniques ranging from annual reviews to face-to-face interviews. In addition to experts' memory, domain experience can be recorded in various media, such as documents, databases, and intranets.

From this point of view, collaboration between teams of professionals is essential so that knowledge can be properly capture and reused. Such collaboration among project members can only be established when win-win situations are created, i.e., when professionals benefit from each other's experience and knowledge is contributed towards the achievement of a common goal. Within the scope of this thesis, it is advocated that opportunities for mutual benefit can be greatly improved if the knowledge required to perform a project is properly managed by appropriate semantic mechanisms which are applied in order to identify/discover and prepare it, thus enabling publishing, sharing, using, and producing new knowledge. The semantic mechanisms can be supported by services which are essential to reduce the complexity of shared multi-users engineering environments. Such mechanisms enable users to concentrate on their tasks by providing specific information in the actual individual, group or process context and by filtering the noise of unrelated status and activity information produced in distributed collaborative working sessions.

The conceptual approach proposed can be applied to any engineering domain which involves the need to enable knowledge sharable and understandable by teams of professionals. Meaning that, the semantic backbone which drives the overall semantic enrichment process can be set to any specific terminology for each engineering domain. The approach is also product and process agnostic, in the sense that it can be instantiated to any kind of engineering process or represent a specific engineering product. The area of application is manifold. It can support organizations' learning strategies, capture corporate knowledge in a common shared repository, keep track of previous projects. It can be focused on a problem-solution representation, enabling users to keep track of problems that have occurred and the decisions made to solve them, which knowledge can be reused whenever necessary to solve new problems.

Although the semantic enrichment conceptual model proposed here is independent of the domain sector addressed, and the principles can be applied to any technical based projects, the Building and Construction sector was considered a particularly suitable test bed to drive the developments of this work since it is essentially ruled by a project-based delivery paradigm (which is intrinsically knowledge intensive) to produce unique products and services. Also, the evaluation activities conducted in the semantic enrichment process took into account B&C related knowledge sources.

Knowledge experts must be aided with mechanisms capable of providing them the information they need, making joint problem solving activities more fruitful. A computational framework driving such needs must take into account: (i) what the experts' requirements are, and (ii) matching those requirements with an historical database of existing knowledge sources. Such a computational framework drives the "knowledge representation enrichment process" and provides a set of capabilities implemented by services and supported by external knowledge

modelled in a domain ontology. The knowledge sources represent the baseline (existing knowledge), which experts rely on to support them to carry out their problem solving activities.

Although B&C is considered as a project-driven engineering environment, its nature makes it different from most other engineering domains, which makes problem solving more challenging. Though two adjacent buildings may look the same, each has unique characteristics when it comes to construction. (Kazi 2005). For example, foundations could be different because of changing ground conditions. B&C projects are characterized by several phases conducted by different teams with different scopes of expertise and skills (e.g. architects, engineers, local authorities, etc.). Such teams of professionals change from phase to phase and have different interest streams regarding the overall project goals. For example, a request for information related to a particular issue can produce different results if raised by a structural engineer or an architect, since different actors often have different requirements. Summing up, the context within B&C projects can be characterized by several key features, such as actor, project type, and phase.

An instantiation of the conceptual model for the B&C domain is depicted in Figure 4.2. It aims to describe the project life cycle in the building and construction and how a semantic enrichment process may happen there in order to drive that life cycle. The building and construction project life cycle is composed of several stages, and in every stage there is a network of check-points called decisional gates (DG) where issues related to design optimization and risk analysis are taken into account. Each DG is a point where all parties in the collaboration process agree on approaches to problem solving, supported by experts from the various disciplines involved. DGs occur within planned meetings (distributed or co-located) chaired or moderated by project managers, and whose recorded outcomes are critical for the project because they aim at solving or avoiding problems. DGs intent is to limit delays in the project progress by the identification of optimum ways to progress.

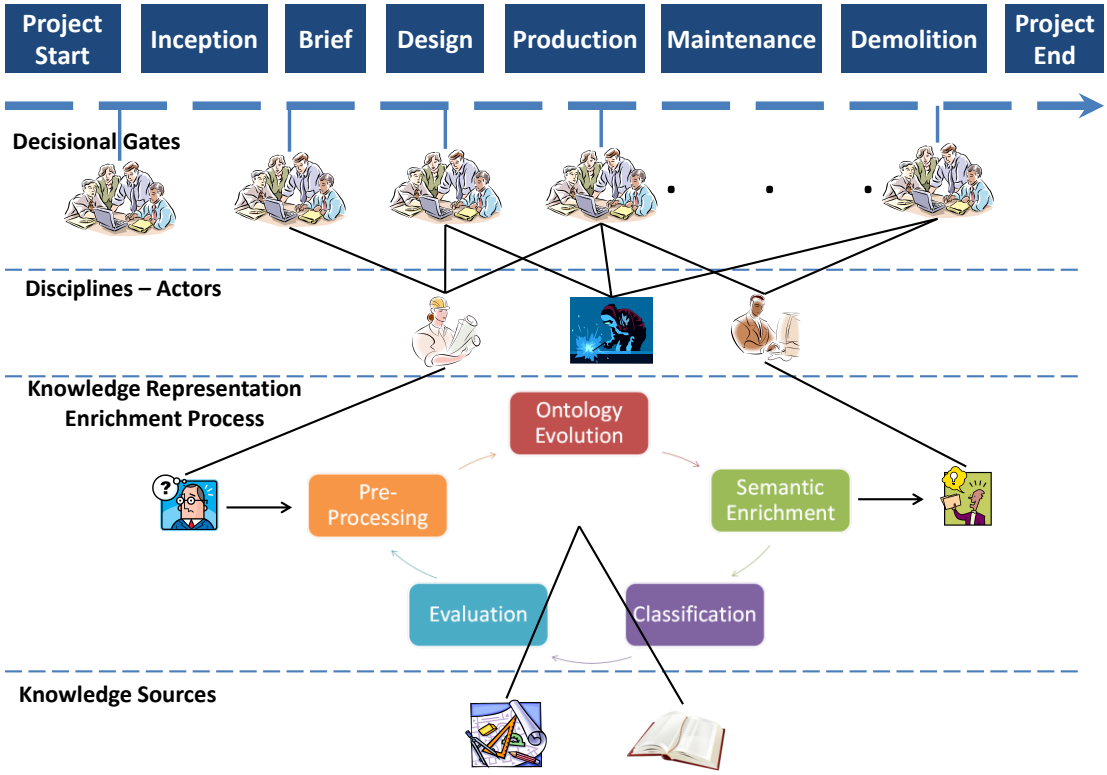


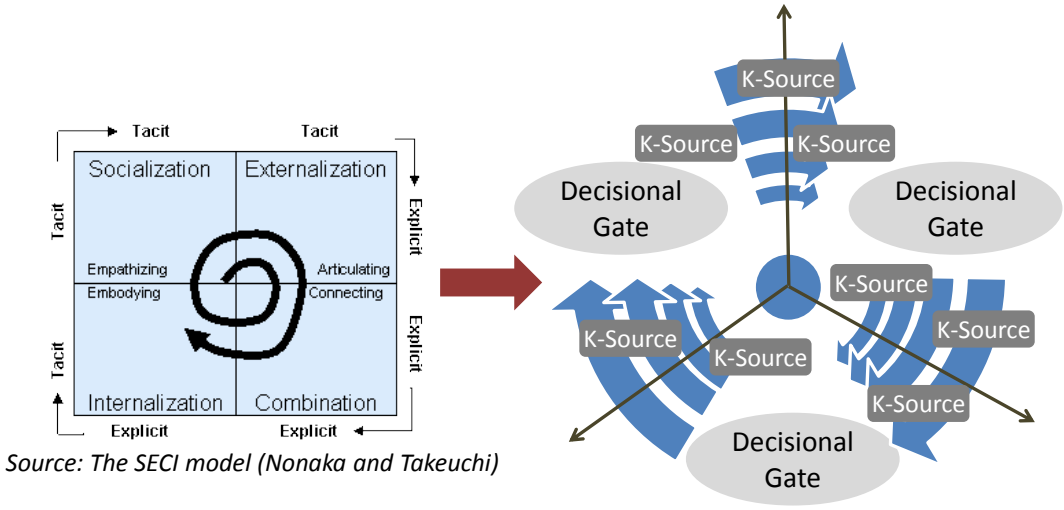
Figure 4.2. Instantiation of the Conceptual model in the B&C sector

4.1 Knowledge Sources

The success of collaboration considering an engineering project, where project teams are working together targeting a shared goal, essentially relies on capitalising on the existing knowledge as well as being capable to find innovative solutions to problems faced. Therefore, it is possible to imagine an instantiation of the SECI model proposed by Nonaka and Takeuchi (Nonaka e Takeuchi, The Knowledge Creating Company 1995), within the collaborative engineering environment towards agile decision making process, where knowledge is: (i) transformed in an evolving way along the time; (ii) managed around problems and solutions in order to be proper capitalised (Costa, Lima, et al. 2010); (iii) better capitalised with the appropriate support of reasoning mechanisms; and (iv) supported by a set of ontology-enabled services to improve semantic accuracy.

As mentioned before, collaborative engineering environments usually rely on a series of meetings and every meeting is considered a Decisional Gate (DG), a convergence point where decisions are made, problems are raised, solutions are considered, and tasks are assigned to project participants. Pre-existing knowledge serves as input to the DG, the project is judged against a set of criteria, and the outputs include decisions (go/kill/hold/recycle) and paths forward (schedule, tasks, to-do list, and deliverables for next DG).

Knowledge needs to be shared in order to be properly capitalised during decision making processes. On one hand knowledge sharing is heavily dependent on technical capabilities and, on the other hand, since the social dimension is very strong during collaboration, there is also an increased need to take into account how to support the culture and practice of knowledge sharing. For instance, issues of trust are critical in collaborative engineering projects, since the distribution of knowledge and expertise means that it becomes increasingly difficult to understand the context in which the knowledge was created, to identify who knows something about the issue at hand, and so forth. This is why decisional gates take a fundamental role in enabling knowledge evolution, as presented by the SECI model (Figure 4.3). Relying on the conceptual basis supporting this thesis presented in Figure 4.2, it is advocated that within collaborative engineering projects, knowledge cannot evolve if not handled inside decisional gates by knowledge experts supported by mechanisms that augment knowledge sources at a semantic level.



Source: The SECI model (Nonaka and Takeuchi)

Figure 4.3. Knowledge Evolution model based on decisional gates

If decisional gates can be seen as a driver to support knowledge evolution, then knowledge instantiated here by knowledge sources (documents) needs to be captured, indexed and added to the accessible knowledge pool.

Every piece of knowledge is held in the form of a Knowledge Representation (KR), both representations of 'external' knowledge sources (e.g. Actors, Documents, and Projects) and also 'internal' knowledge structures (e.g. domain ontologies, as well as more technical items used only by the enrichment process and not seen by the user), as depicted in Figure 4.4. A KR describes in system specific terms information about any given knowledge source, allowing knowledge sources to be indexed, queried, and retrieved. Such representations rely on the so called Semantic Vectors (SVs), which are constructed using external knowledge available in domain ontologies. SV holds information that allows to 'know' what a given knowledge representation is concerned with. This information is given in terms of ontological concepts (and keywords) that are deemed relevant to a given KR, and a respective semantic weight providing a value of how relevant it is. SVs are used in searches and other matching algorithms to determine how similar two knowledge sources are.

Figure 4.4. Some Knowledge Sources

KR takes the form of a vector, designed to encapsulate essential meta-information about knowledge sources. The design of the system is based around the principle that every entity represented within the system is encapsulated in the form of a KR (e.g. Actors, Projects, Organisations, or Documents). The KR 'meaning' is captured in the form of SV, which holds an array of weighted ontological concepts, relevant to the respective knowledge source.

Knowledge representations, being descriptions of knowledge sources adopted by the conceptual model, need to be instantiated for 'real' items of knowledge. The process of instantiation is triggered by the construction of a SV, which will create the knowledge representation in terms of ontological concepts. The processes is automated as far as possible, but in almost all cases a degree of user interaction is required in order to complete the process and produce the most meaningful KR.

SVs are created by an automated process that parses the content of the knowledge source (or, in the case of a text document, its entire content) to produce a list of weighted keyword terms. These are then matched to ontological concepts to produce a weighted semantic vector. Any highly weighted keyword terms not producing a match in the ontology can then be flagged as possible 'Concepts to be added to the Ontology', thus ensuring that the ontology is constantly enriched and developed.

4.2 Actors

The division of labour in modern companies and projects leads to a distribution of expertise, problem solving capabilities, and responsibilities. While specialization is certainly a main driver for productivity, its consequence is that both generation and use of knowledge are not evenly spread across a given organization. This leads to high demands of human interaction in knowledge management practices.

Departments, groups, and individual experts develop their particular views on given subjects. These views are motivated and fully justified by the specifics of the actual work, goals, and situation. Creating a single, globally agreed vocabulary with a level of detail sufficient for all types of participant, may incur high costs (e.g., for negotiation). An ontology-based system should therefore allow balancing between (a) global knowledge which might constitute a shared context, but may also be relatively expensive; and (b) localised expertise which might represent knowledge that is not easily shareable or simply not worth sharing.

In order to create a formal ontology, typically some variation of the following ontology engineering process is followed:

i. The domain to be represented is defined, i.e., what shall be represented and what shall be left out of the representation;

ii. Knowledge to be expressed formally is then acquired from different sources such as domain experts, documents, and databases. This knowledge needs to be organised, and different world views with maybe semantic conflicts need to be sorted out;

iii. Finally, the acquired structured knowledge needs to be encoded in a given knowledge representation formalism.

Attempts have been made to automatically create ontologies, for instance by learning ontologies from natural language text through applying machine learning techniques and prior knowledge about natural language. However, such ontology learning approaches provide only partial support since existing state-of-the-art ontology learning techniques are at the level of extraction of terms and relations when learning from natural language text. Additionally, they currently usefully serve as part of knowledge acquisition activities. Furthermore, even if ontology learning was completely successful, ontology engineering cannot always be seen as merely “re-writing” already known knowledge in a formal language. Sometimes, it is precisely this act of formal specification in which implicit knowledge is made explicit or new knowledge is generated. Eventually, there may not actually be natural language texts or other prior documentation of the knowledge to be formalised at hand. For this reason this thesis argues that some manual intervention with the ontology engineering process will always be necessary where a certain level of expressiveness is expected from these formal models. From this perspective, this work

considers two types of actors involved in ontology engineering, namely ontology experts and domain experts.

An ontology expert is a person carrying out formalisation tasks. Note that in the ontology engineering process there may be: (i) multiple ontology experts at work, and (ii) ontology experts may have different fields of expertise, i.e. ranging from knowledge acquisition to specific knowledge representation formalisms. In most ontology engineering methodologies, the ontology expert is assumed to be a person specifically trained for the above ontology engineering activities and usually trained in knowledge representation formalisms.

For instance, it is considered whether an ontology contains all or only relevant information, whether it is modular enough, whether it is agreed upon by the relevant people (i.e. domain experts, participants in a community).

First, people with a variety of skills must be considered as users of the ontology. This addresses not only the contrast of domain experts vs. ontology experts but a much finer granularity of skills in people participating in ontology engineering. Users may be experts in different domains, and regarding different parts of ontology engineering; for instance some ontology experts may have more experience with knowledge acquisition and others with a particular knowledge representation formalism.

Consider that in a traditional ontology engineering process, domain experts would express their knowledge and structure it informally, not making use of any knowledge representation formalism. Domain experts could then go through a validation procedure and finalise their informal version of the ontology. Ontology experts, experienced in some chosen formalism such as description logics in the form of OWL, would then start with this informal yet “final” ontology, and formalise and implement it.

The user perspective also has a major role in support for contextualization of the domain ontology. Every user has different needs, assumptions, views, and rules on the basis of his/her domain work/expertise and/or the evolving nature of knowledge in their domain.

Within the scope of this thesis, domain experts in B&C were consulted, not only to aid the formalization of their views of the domain in a form of an ontology, but also to help in selecting relevant knowledge sources that should be used for the proof of concept evaluation. As it will be described during this chapter, there is a strong involvement of domain experts throughout the process of semantically enriching knowledge representations. One of the main conclusions that can be already stated is that there are no “pure” automatic mechanisms for knowledge conceptualization and elicitation. Domain experts are key in the entire process, as it will be shown. There are pre-processing stages, where ontologies need to be fine-tuned to address a very specific problem. There are also evaluation indicators that show the performance of the proposed approach and these should be confronted with domain experts. These are some

examples, where domain experts' involvement is essential to the whole process. Figure 4.5, illustrates the main roles covered by domain experts and ontology expert in the scope of this thesis.

Domain Experts

- Informal conceptualization of domain ontology (ontology concepts and relations).
- Gather and label relevant knowledge sources to be used for semantic enrichment process.
- Enrich the informal ontological model with new concepts, relations and equivalent terms to support gathered knowledge sources.
- Draw conclusions about evaluation process.

Ontology Experts

- Formalization and implementation of domain ontology.
- Enrichment of the ontological model with new concepts, relations and equivalent terms in formal language.
- Running semantic enrichment process.
- Conduct the evaluation of the enrichment process.

Figure 4.5. Actors' Roles

4.3 The Ontological Model and Methodology development

This subsection describes the ontological dimension developed, which is the backbone for the semantic enrichment process. The structure and the entities that compose the model are presented here, with also an instantiation to the B&C domain. The methodology used for ontology conceptualization and instantiation is also presented at the end of this subchapter.

4.3.1 The Model

The ontological model used in this work was entirely developed using Protégé Ontology editor (Stanford Center for Biomedical Informatics Research 2013), and was written in OWL-DL language (W3C 2012). The Ontology comprehends two major pillars, namely concepts and their relations. The former relates to specific aspects (named classes) of the domain such as the type of project, project phase, geographical location and similar data. The latter specifies how the ontology concepts are related to each other. This work adopted the following definitions regarding the ontological "elements":

- Entity – an entity may represent a concept, a relation, a signature, or an axiom.
- Concept/Class – this is an entity that represents something which has a semantic value. The concepts can be linked together or they can be part of a specialisation within a given classification. For instance, a *room* is a concept, which can be specialised into *meeting room*, *office*, *bathroom*, and so on.
- Relation – a relation is an entity that connects two concepts. It carries information valid between the related concepts. As with a concept, a relation can have specialisation. For instance, is-in-town defines a relation that may be applied to the concepts "Building" and "City".
- Property/Attribute – a relation that connects a given concept to a simple type (e.g. integer, and string) is called a property. For instance, the relation *has-name* related to a string creates a property.
- Signature – a signature is an entity that connects two concepts via a relation, i.e., a signature is a triplet comprising concept + relation + concept. For instance, building has-room office-manager is a signature (as illustrated in Table 4.1).
- Individuals – this a "concrete" instantiation of a concept carrying instance(s) of relations. For example, "Higher Education Facility" is an instance of the concept "Learning Facility". The concept "Learning Facility" is defined by the attribute *has-name*. This attribute is instantiate by "Higher Education Facility" *has-name* "Higher Education Facility".

- Equivalent Terms – each concept contains a list of equivalent terms. Equivalent terms are used to index knowledge with the corresponding concepts. An equivalent term is also called a *Lexical Entry*. For instance, a lexical entry for the concept “Actor” is “User”.

Table 4.1. Signatures examples

Signatures
An Actor <is assigned to> a Project
A Project <is decomposed in> Tasks
A Product <is produced by> a (set of) Process(es)
A Resource <is allocated to> a Project
A Resource <is used by> an Actor
A Resource <is involved in> a (production) Process

Figure 4.6 depicts the top-level ontology concepts and how they relate to each other.

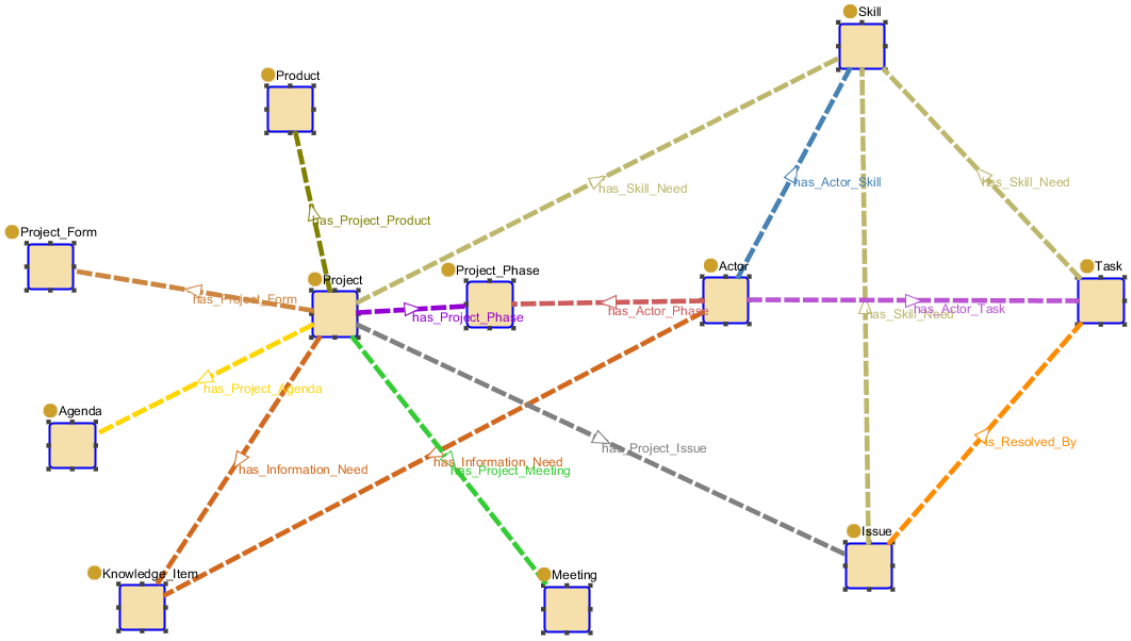


Figure 4.6. High-Level Ontology concepts and their relations

Figure 4.7 depicts the sub-concepts related to the top-level concept ‘Product’. It is worth mentioning that hierarchical relations between concepts and sub-concepts are defined by “is_a” notation.

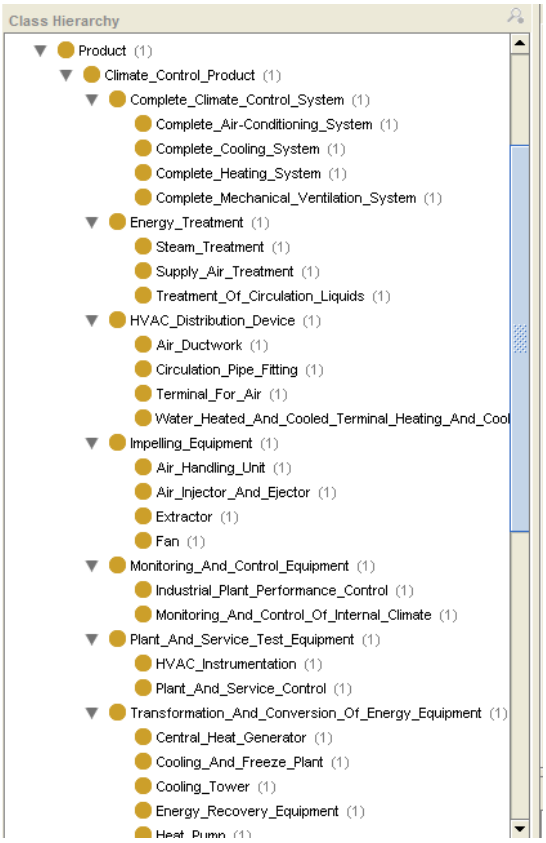


Figure 4.7. Product Concept Hierarchy

As already mentioned, the adopted domain ontology is expressed in OWL-DL language. DL stands for description logic, a field of research that has studied the logics that form the formal foundation of OWL. Under the description logic used by OWL-DL there are many varieties and there is an informal naming convention, roughly describing the operators allowed. The DL expressiveness of the domain ontology is in the form of $\mathcal{ALC}(\mathcal{D})$, where $\mathcal{AL}$ stands for attributive language, which allows: atomic negation (negation of concept names that do not appear on the left hand side of axioms), concept intersection, universal restrictions, and limited existential quantification, and ($\mathcal{D}$) extends it to use of datatype properties and data values. $\mathcal{ALC}$ is simply $\mathcal{AL}$ with complement of any concept allowed, not just atomic concepts. Table 4.2, depicts other relevant metrics that qualify the domain ontology.

Table 4.2. Ontology Metrics

Description	Value
Number of Classes	834
Number of Instances	833
Number of Properties	13
Max Depth	6

Mean Siblings	8
Max Siblings	19

Several levels of specificity are given for all families of concepts, as described for the 'Actor' concept. These specificity levels represent concept hierarchies and, ultimately, taxonomic relations such as 'Architect' <is_a> 'Design Actor' and 'Design Actor' <is_a> 'Actor'. All classes, or concepts, have an instance (individual), which corresponds to the class, and comprises the keywords or expressions gathered and related to each concept, through an ontological data-type property designated 'has Keyword'.

Concepts have a set of terms named 'equivalent terms' which are terms or expressions relevant for capturing different semantic aspects of such concepts. For instance, the 'Learning_Facility' concept has a 'Higher_Education_Facility' individual, and this individual has several equivalent terms such as 'university', 'science college', and 'professional college'. Thus each equivalent term belongs to some higher concept, as shown in Figure 4.8. Moreover, concepts are connected by ontological object properties called 'ontological relations'. Ontological relations relate concepts among themselves and are described by a label (property) and the relevance (weight) of such relation in the context of the domain Ontology.

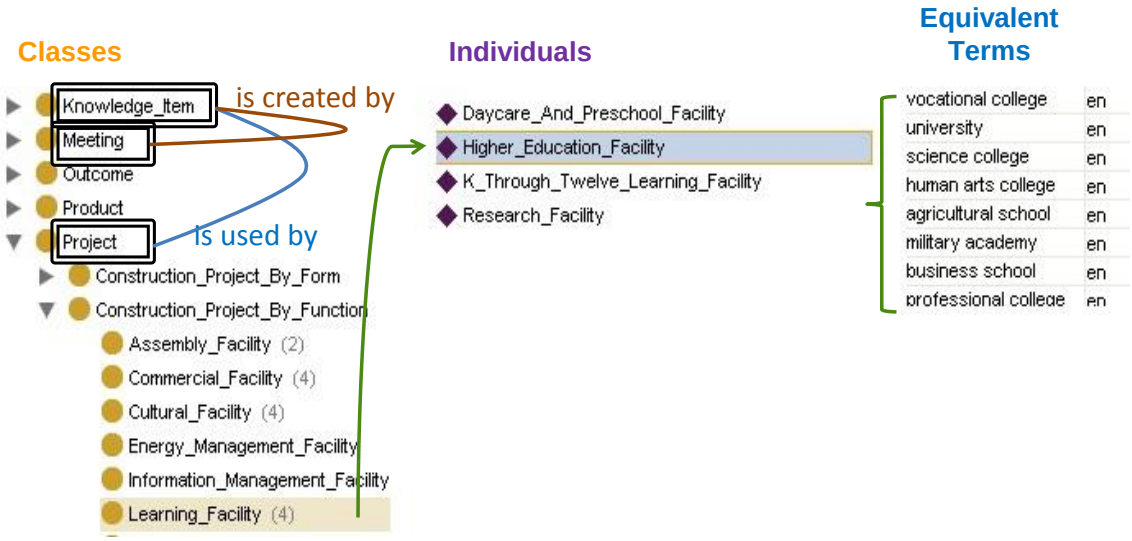


Figure 4.8. Ontological Elements

4.3.2 The Methodology

One of the main difficulties in defining the ontological model is to identify the relevant sources of knowledge, and motivate domain experts to share their knowledge and to invest time to do so. The work targeted at this stage is mainly concerned with knowledge elicitation from domain experts, with ontology learning from text or integration of structured information from

heterogeneous sources. The method adopted here uses an iterative approach (Figure 4.9), which is split into several phases, with each phase containing a set of related tasks.

It is worth recalling, that the proposed method for ontology development was inspired within the approach used by the e-COGNOS project. Also the main concepts that serve as the backbone of the ontology, were also inspired from the e-COGNOS ontology. The objective was not to develop "yet another domain ontology", which is also not the objective of this thesis, but rather to use what was available for deriving the semantic enrichment of knowledge sources. However, for the specific purpose of this thesis, some adaptations and refinements of the ontological model had to be made (highlighted in this section).

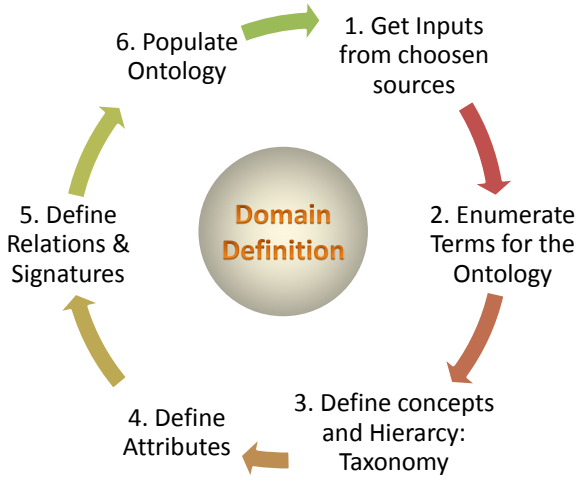


Figure 4.9. Ontology definition process

From a high level point of view, the basic ontological model of the domain ontology was inspired by the e-COGNOS ontology (Lima, El-Diraby and Stephens 2005) and it can be described as follows: 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). As such, the proposed taxonomy includes seven major domains to classify these major concepts: Project, Actor, Resource, Product, Process, Technical Topics (Conditions), and Related Domains (work environment).

All entities (including Process) have three ontological dimensions: state, stage and situation. State concept captures the status of entity development: dormant, executing, stopped, re-executing, completed. Stage concept defines various development stages: conceptualization, planning, implementation and utilization. Situation concept refers to planned entities and unplanned entities.

A Project is a collection of processes. It has two types: Brown field projects and Green field projects. It has a project delivery system, a contract, a schedule, a budget, and resource requirements. It also has a set of related aspects that include: start time, a finish time, duration,

a quality standard, productivity level, a life cycle and a life cycle cost—all of which are defined in the Technical Topics domain.

A Process has an input requirements that include: the completion of all proceeding processes, the availability of required approvals, the availability of required knowledge items (documents, software, etc.), the availability of required Resources (materials, equipment, subcontractors), the availability of required Actors, and the availability of required budget. A Process has three major sub concepts: Phase, Activity and Task. It also has two major types: engineering process and administrative process. A Process has an output that include: update to a product time-line, an update to the project schedule, and update to the project budget, satisfaction/update to the legal conditions/status of Actors, may result in creating some project incidents (e.g. an accident, damage to an equipment).

A Product (also Actors, Processes and Resources) has attributes, parameters and elements, which are defined in Technical Topics. This domain includes the following subdomains: Basic products, Construction complex, Materials, Construction aids, and Management products (e.g. reports and budget).

Technical Topics domain defines the concepts of productivity, quality standard and duration. The following subsections describe the major elements of these seven domains.

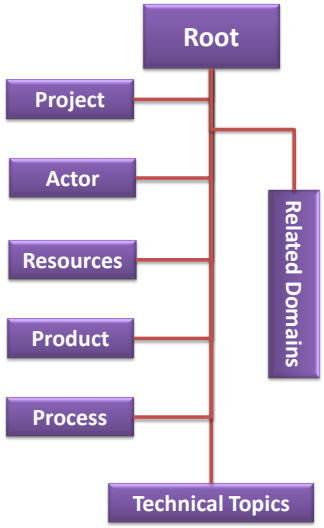


Figure 4.10. Upper-level ontological concepts

4.4 Knowledge Representation Enrichment Process

The enrichment process of KRs is the core contribution of this work, essentially arguing that domain knowledge represented in a given ontology can be used to semantically enrich representations of knowledge sources. The whole process ranges from the pre-processing stage till the final assessment of results achieved after the enrichment process, in a cyclical way since the assessment will provide inputs to improve the quality of whole process, such as the refinement of the domain ontology.

The overall approach comprises 5 stages (Figure 4.11), namely: (i) ***pre-processing*** (preparation of the operational environment and input sources); (ii) ***ontology evolution*** (augmenting semantic coverage of the ontology considering the inclusion of new knowledge sources in the KB repository); (iii) ***semantic enrichment*** (the enrichment process itself); (iv) ***classification*** (application of an unsupervised classification algorithm); and (v) ***evaluation*** (measuring accuracy of the overall approach).

The *pre-processing* stage holds the preparation of both operational environment and input sources. Input sources are: domain ontology and relevant knowledge sources. As previously explained in section 2, the domain ontology was formed based on previous European research initiatives. Knowledge sources represent the relevant and appropriate elements that will be used to support the semantic enrichment process as well as to assess the quality of such enrichment. Logically, experts play a key role to help inspecting and pre-labelling those relevant knowledge sources, in order to provide an initial reference that will be validated against the results produced by the enrichment process. All relevant knowledge sources are selected and stored in a knowledge base repository, to help deal with the management of all sources to be indexed. In the case study reported here, they were selected from the ICONDA[3] database, from Fraunhofer IRB, which is a rich database of technical documents (e.g. reports and papers) related to B&C matters.

3 ICONDA®Bibliographic began life in the mid 1980's as the database of the International Council for Research and Innovation in Building and Construction (within CIB). It holds records and associated full text files of CIB related publications: monographic (conference proceedings, books, reports) as well as serial (journals).

The *ontology evolution* happens when new knowledge sources are included in the knowledge repository. Evolution, in this sense, means that the semantic coverage of the ontology must evolve, considering the new knowledge sources. For instance, new concepts must be added, new relations may be identified and equivalent terms can be extended. This stage is vital since the quality of results achieved strongly relies on the semantic accuracy and relevance of the domain ontology. Moreover, this stage also allows the assessment of the relevance of the domain ontology used regarding the current knowledge base repository.

The *semantic enrichment* is the very heart of this work. Indeed, it tackles the enrichment of knowledge representations (in this work called semantic vectors), extending the classical vector space model approach by including two additional steps in the process: (i) use of taxonomical relations to improve semantic relevance of neighbours concepts; and (ii) use of ontological relations with the same purpose of point (i).

The *classification* stage relies on the application of unsupervised classification algorithm (K-Means clustering), in order to group knowledge sources into various categories, called clusters.

Evaluation, the last stages assess the overall approach using classical precision and recall metrics to measure performance. These two last stages are further detailed in chapter 6.

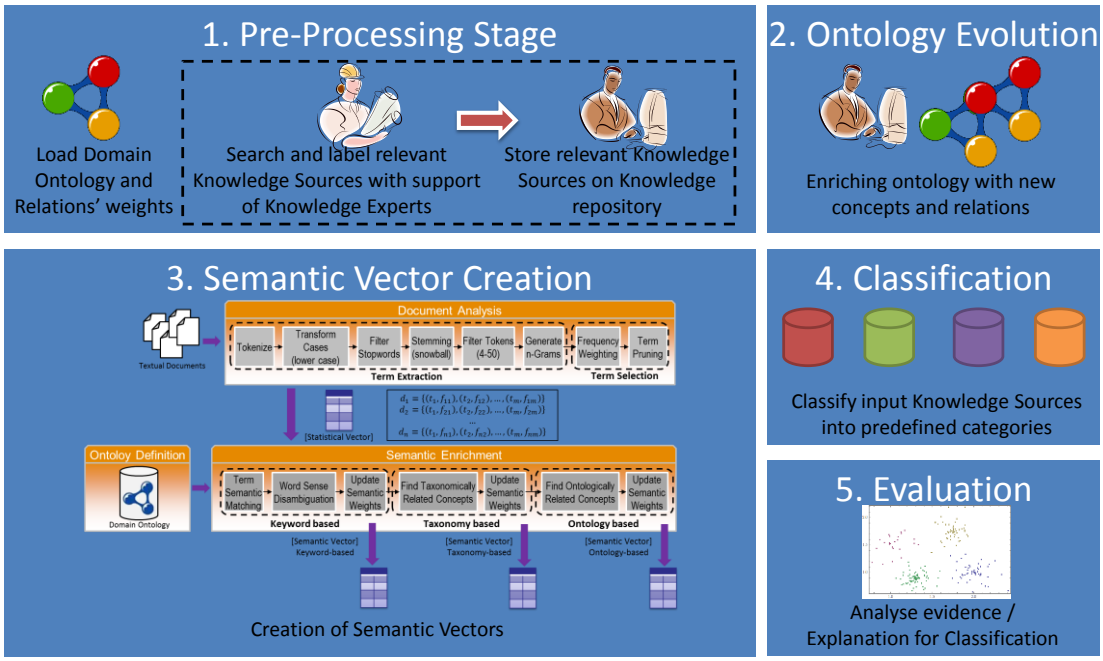


Figure 4.11. Knowledge Representation Enrichment Process

It is worth taking a closer look at the technologies and techniques (both adopted and developed) supporting the whole process (Figure 4.11), according to each phase, namely:

- *Pre-processing*: Protégé is the ontology editor supporting the ontology creation process, expressed in OWL, which is loaded and stored using services provided by the pair

JENA/MySQL. Both *Search for relevant KS* and *Label KS into categories* are technical operations performed by domain experts. Liferay is the tool used to manage the knowledge repository, offering the classical functionalities for the purpose. The assignment of weights to ontological relations is performed automatically in the "Ontology Evolution" step, nevertheless the domain expert must analyse such results and update it when he/she considers necessary.

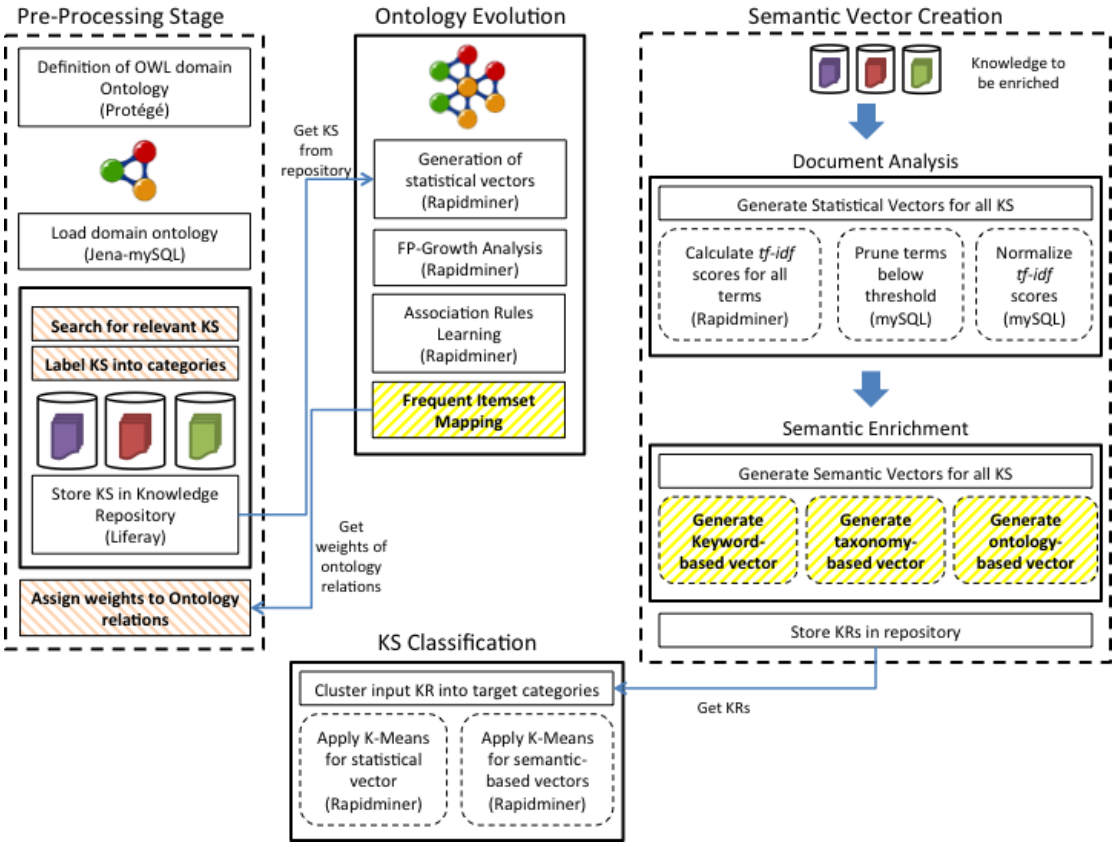


Figure 4.12. Technologies inside the process

- *Ontology Evolution*: *Generation of statistical vectors* (which uses KSs from the knowledge repository), *FP-growth analysis*, and *Association rules learning* are performed by Rapidminer. It is worth noticing that association rules are automatically discovered based on the semantic liaisons connecting ontological concepts. *Frequent Itemservice Mapping,* implemented as part of this work as a Java-based service, create a map of ontological concepts based on the co-occurrence of equivalent terms.

- *Semantic Vector Creation*: in the Document Analysis step, Rapidminer calculates the *tf-idf* scores for all terms, a stored procedure developed on MySQL reduces the size of the statistical vector according to a certain relevance degree defined by the knowledge expert (*Prune terms below threshold*), and another stored procedure normalize the statistical vector after pruning the terms. Next, the semantic enrichment is performed by

three Java services responsible for the generation of the keyword, taxonomy and ontology-based vectors, respectively.

- *KS Classification*: Rapidminer is used to wrap-up the process applying clustering algorithms based on both statistical and semantic vectors, allowing a comparison of results produced. Needless to say that our expectation is to have better results, semantically speaking, in the clusters generated based on the semantic vectors.

It is also worth analysing in detail the following steps within the enrichment process (Figure 4.13): (i) *Document Analysis*: extracts terms from knowledge sources, creates the key term set, and produces a term occurrence statistical vector; and (ii) *Semantic Enrichment*: alters the statistical vector using taxonomical and ontological elements (such as relations, concepts weights) in order to produce a semantically richer KR, called the Semantic Vector.

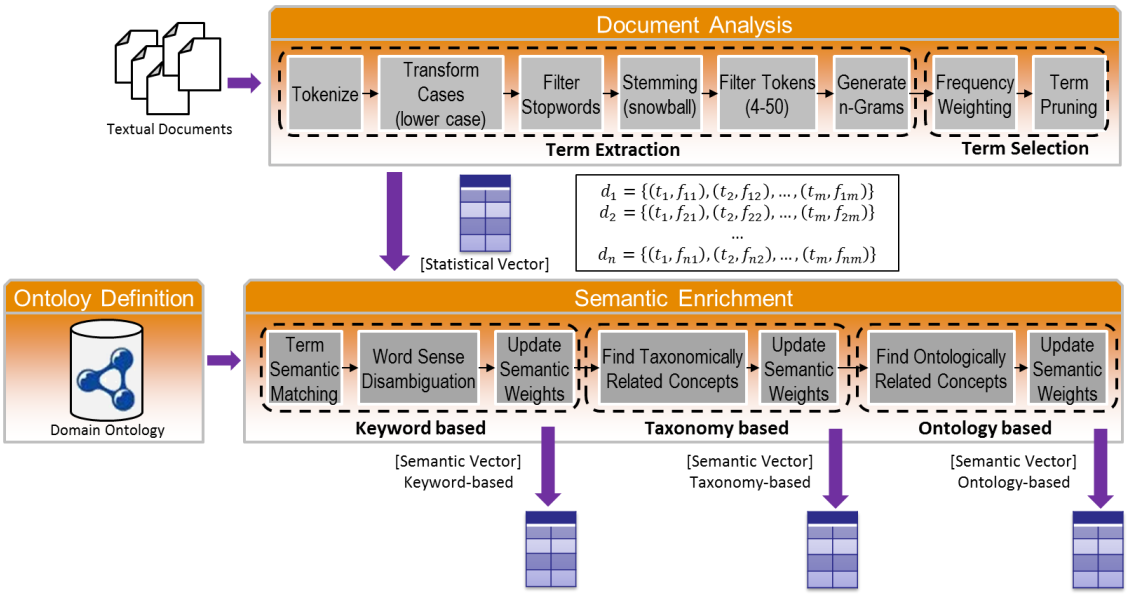


Figure 4.13. The Semantic Vector creation process

4.4.1 Document Analysis phase

The generation of statistical vectors classifying terms from documents by relevance (here, a relevant term means that it characterises best a given document) in a given document as well as in the entire document corpus, is performed by applying a normalized *tf-idf* score. There are two process running in this phase, namely *Term Extraction* and *Term Selection*, which reduce the statistical vector dimension, taking out less relevant terms.

- ***Term Extraction Process***

The extraction process happens in the following way:

1. First, each document is split into sentences. Then, terms in each sentence are extracted as tokens (so called *tokenization*).
2. All tokens found in the document are transformed to lower case font.
3. Terms belonging to a predefined stop word list[4] are removed.
4. The remaining terms are converted to their base forms by a process called stemming, using the snowball[5] method. Terms with the same stem are then combined for frequency counting. In this paper, a term is regarded as the stem of a single word.
5. Tokens whose length is "< 4" or "> 50" characters are discarded.
6. The n-Grams generation is the creation of strings of 1 to N words. For this case we are considering the generation of unigrams (e.g. Energy), bigrams (e.g. Waste Management) and trigrams (e.g. Electric Power Product).

- ***Term Selection Process***

We consider that less relevant terms are most likely to be noise sources and of no use, so we apply the *tf–idf* (term frequency - inverse document frequency) method to select the key terms for the document set. Equation 1 is used for the measurement of $tfidf_{ij}$ for the importance of a term t_j within a document d_i. The main drawback of the *tf-idf* method is that long documents tend to have higher weights than short ones. The method considers only the weighted frequency of the terms in a document but ignores the length of the document. In order to prevent this, in Equation 2, tf_{ij} is the frequency of t_i in d_j, and the total number of occurrences in d_j is the maximum frequency of all terms in d_j that is used for normalization to prevent bias for long documents.

$$tfidf_{ij} = tf_{ij} * idf_i \quad (4.1)$$

$$tf_{ij} = \frac{number\ of\ occurrences\ of\ t_i\ in\ d_j}{total\ number\ of\ occurences\ in\ d_j} \quad (4.2)$$

4 Sometimes, some extremely common words which would appear to be of little value in helping select documents matching a user need are excluded from the vocabulary entirely. These words are called stop words.
5 Stemming is the process for reducing inflected (or sometimes derived) words to their stem, base or root form - generally a written word form. Snowball is a framework for writing stemming algorithms.

$$idf_i = \log \frac{number\ of\ documents\ in\ D}{number\ of\ documents\ in\ D\ that\ contain\ t_i} \quad (4.3)$$

After calculating the weight of each term in each document, those which satisfy a pre-specified minimum *tf–idf* threshold γ are retained. For this work, we consider only terms where the *tf-idf* score is ≥ *0.001* in order to reduce the size of the generated vectors and also the computational power required to process them. Analysis carried out by experts concluded that terms which *tf-idf* score was less than 0.001 where not considered relevant enough. Subsequently, the retained terms form a set of key terms for the document set *D*.

A document, d_i, is a logical unit of text, characterised by a set of key terms t_j together with their corresponding frequency f_{ij}, and can be described in vector form by $d_i = \{(t_1, f_{i1}), (t_2, f_{i2}), \dots, (t_j, f_{ij}), \dots, (t_m, f_{im})\}$, the statistical vector. Thus for each document in the document corpus *D* there is a resultant statistical vector.

4.4.2 Semantic Enrichment Phase

In this phase Semantic Vectors (SVs) for all documents in corpus *D* are built. Semantic vector creation is the basis for the approach in our work. It represents the extraction of knowledge and meaning from KS and the agglomeration of this information in a matrix form, better suited to mathematical handling than the raw text form of documents. A SV is a statistical vector semantically richer with the use of the following ontological elements: concepts, relations, equivalent terms, and weights. Each SV is represented by two columns: the first column contains the concepts that populate the knowledge representation of the KS, i.e., the most relevant concepts for contextualizing the information within the KS; the second column keeps the degree of relevance, or weight, that each term has on the knowledge description of the KS.

Our approach takes into account three complementary procedures to create SVs, where each procedure successively adds semantic richness to the KR. The first step creates a SV keyword-based (SVKB), the second step creates a SV taxonomy-based (SVTB), and the final step creates a SV Ontology-based (SVOB). Each step is described in the following sections.

- ***Semantic vector keyword-based***

SVKB takes into consideration only the association between terms from the statistical vector and the concepts in the domain ontology. This step matches the statistical vector keywords with equivalent terms linked to each ontological concept in the domain Ontology (Figure 4.14).

This process starts by first identifying the statistical vector keywords associated to a particular document and then finding similarities between each keyword and the equivalent terms within the ontology. The calculation of the similarities is done using the cosine similarity. The reason

for choosing the cosine algorithm is that cosine measure can be applied when comparing n-grams similarities of different magnitudes.

Cosine similarity algorithm measures the similarity between two vectors. In this case, we have to compare two n-grams. If we consider each one has a vector, we can use the cosine of the angle θ between x and y, represented in equation (4.4).

$$cos(x, y) = \frac{x}{\|x\|} \cdot \frac{y}{\|y\|} \tag{4.4}$$

Equation (4.4) can be applied to our process in the following manner:

$$\frac{(Shared\ Keyword\ Terms) * (Shared\ Equivalent\ Terms)}{(Keyword\ Total\ Terms) * (Equivalent\ Terms\ Total\ Terms)} \tag{4.5}$$

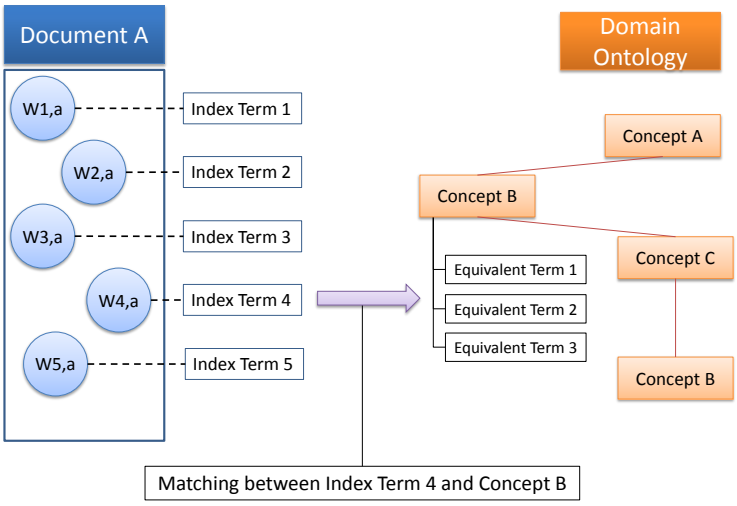


Figure 4.14. Vector terms mapping against the Ontology concepts

Next SVKB is stored in the database in the form $[\sum_{i=1}^{n} x_i ; \sum_{i=1}^{n} w_{x_i}]$, where *n* is the number of concepts in the vector, x_i is the statistical representation of the concept and w_{x_i} is the semantic weight corresponding to the concept. Table 4.3 depicts the weight of every ontological concept associated to each key term within the statistical vector, where the first column corresponds to the concepts that were matched to describe the most relevant terms extracted from the statistical vector shown in column 2, and the third column shows the semantic weight for each concept matched.

Table 4.3. Keyword-based semantic vector

Concept	Key Term	Weight
Sanitary_Disposal_Unit	toilet, urin, water_closet	0,149514
Sanitary_Laundry_and_Cleaning_Equipment_Product	sanitari	0,132629
Team	person, personnel	0,104497

Commitee	subcommitte	0,067880

- ***Semantic vector taxonomy-based***

SVTB is the next level in the semantic evolution of KRs. It is created by adjusting the weights of concepts according to the taxonomic relation among them, i.e., those concepts that are related by the 'is_a' type relation. If two or more concepts taxonomically related appear in a SVKB, then the relation can boost the relevance of the expressions within the KR and therefore enhance weightings. SVTB is created based on kin relations between concepts within the ontological tree. Specifically, the kin relations can be expressed through the notion of homologous/non-homologous concepts (Figure 4.15) as follows (Sheng 2009).

Definition 1: In the hierarchical tree structure of the Ontology, concept A and concept B are homologous concepts if the node of concept A is an ancestor node of concept B. Hence, A is considered the nearest root concept of B, R(A,B). The taxonomical distance between A and B is given by:

$$d(A,B) = |depth(B) - depth(A)| = |depth(A) - depth(B)| \quad (4.6)$$

In Equation 6, depth (X) is the depth of node X in the hierarchical tree structure, with the ontological root concept depth being zero (0).

Definition 2: In the hierarchical tree structure of the Ontology, concept A and concept B are non-homologous concepts if concept A is neither the ancestor node nor the descendant node of concept B, even though both concepts are related by kin; If R is the nearest ancestor of both A and B, then R is considered the nearest ancestor concept for both A and B concepts, R(A,B). The taxonomical distance between A and B is expressed as:

$$d(A,B) = d(R,A) + d(R,B) \quad (4.7)$$

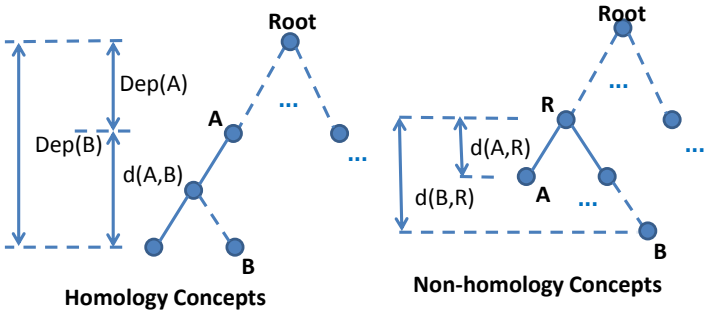


Figure 4.15. Homologous and non-homologous concepts (Sheng 2009)

SVTB is calculated using the keyword-based vector as input, where taxonomical relations are used to boost the relevance of the concepts already present within the vector or to add new concepts. The weight of the concepts is boosted when two concepts found in the keyword-based vector are highly relevant, with the degree of relevance being defined by a given

threshold. If the relevance of the taxonomical relation between two concepts is higher than the predefined threshold, then the semantic weight of such concepts is boosted in the taxonomy-based vector. If a concept already present in the keyword-based vector is taxonomically related to a concept that is not present in the vector, then the related concept is added into the taxonomy-based vector.

One of the major differences between the present work and the work presented by (Sheng 2009) is that, in our approach, new concepts are only added into the taxonomy-based vector if the $d(A,B)$ = 1 for homologous concepts and $d(A,B)$ = 2 for non-homologous. The reason for such limitation is to avoid obtaining a sparse vector and to only add concepts that are highly related to already existing ones.

The intuition behind this work is to alter term vectors by strengthening the discriminative terms in a document in proportion to how related they are to other terms in the document (where relatedness includes all possible relationships modelled in an Ontology). A side effect of this process is the weeding out of the less important terms. Since ontologies model domain knowledge independently of any particular corpus, there is also the possibility of introducing terms in the term vector that are highly related to the document but are not explicitly present in it. The approach used for enhancing term vectors is therefore based on a combination of statistical information and semantic domain knowledge.

The taxonomical similarity is calculated differently for both homologous and non-homologous taxonomical relations defined previously:

If $d(A,B) \neq 0$ and A and B are homologous.

$$Sim(A,B) = \left(1 - \frac{\alpha}{depth(A)+1}\right)\frac{\beta}{d(A,B)}\frac{son(B)}{son(A)} \tag{4.8}$$

If $d(A,B) \neq 0$ and A and B are non-homologous.

$$Sim(A,B) = \left(1 - \frac{\alpha}{depth(R)+1}\right)\frac{\beta}{d(A,B)}\frac{son(A)+son(B)}{son(R)} \tag{4.9}$$

If $d(A,B) = 0$

$$Sim(A,B) = 1 \tag{4.10}$$

- ***Semantic vector ontology-based***

SVOB is the final level in the semantic evolution of KRs, which is based on the ontological relations. We apply association rule theory to construct ontological concept relations and evaluate the relevance of such relations for supporting the enrichment process of a domain ontology. The objective is to analyse the co-occurrences of concepts in unstructured sources of information in order to provide interesting relationships for enriching ontological structures

(Paiva, et al. 2013). The construction of ontological relations between concepts is explained further in section “4.4.3 Creation of Ontological Relations”.

The ranking of such semantic association is also complemented by input from experts in the building and construction domain to establish the final numerical weights on each ontological relationship. Experts’ intervention is an attempt to guarantee that relevancies of relationships reflect a proper knowledge representation requirement.

The creation of the SVOB is a two-stage process using the taxonomy-based SV as input: the first stage boosts weights of concepts already present in the taxonomy-based vector, depending on the ontological relations among them; the second stage adds new concepts that are not present in the input vector, according to ontological relations they might have with concepts belonging to the taxonomy-based vector (Costa, Figueiras e Paiva, et al. 2012).

Analogous to the creation of a SVTB, the new concept is added to the vector only if the importance of an ontological relation exceeds a pre-defined threshold, for the same constraint reasons. The ontological relation’s relevance is not automatically computed; rather, it is retrieved from an ontological relation vector comprising pairs of concepts and the weight associated to their relation, as shown in Table 4.4.

Table 4.4. Ontological Relations

Property	Subject	Object	Weight
is_part_of	Complete_Sanitary_Suite	Sanitary_Laundry_and_Cleaning_Equipment_Product	0,07
is_operated_by	Sanitary_Disposal_Unit	Sanitary_Laundry_and_Cleaning_Equipment_Product	0,07

Equation 11 describes the process of boosting of concepts or the addition of new ones, here Ow_{C_y}, is the new weight of the ontological concept, and Tw_{C_y} is the input taxonomy weight of the concept to be boosted. If the concept is added then Tw_{C_y} should be zero. Tw_{C_x} is the taxonomical weight of the concept related to C_{y} and $TI_{C_xC_y}$ is the weight of the relation between C_{y} and C_{x}.

$$Ow_{C_y} = Tw_{C_y} + \sum (all\ related\ C_xs) \left[Tw_{C_x} * \left(TI_{C_xC_y} \right) \right] \tag{4.11}$$

An example of SVOB is depicted in Table 4.5.

Table 4.5. Part of a Semantic vector ontology-based

Concept	Weight
Sanitary_Disposal_Unit	**0,111718**

Sanitary_Laundry_and_Cleaning_Equipment_Product	**0,099504**
Team	0,074115
Plumbing_Fixture_and_Sanitary_Washing_Unit	0,056649

In this example, the concepts 'Sanitary_Disposal_Unit' and 'Sanitary_Laundry_and_Cleaning_Equipment_Product' were boosted because they are already present in the SVTB and are related by the ontological relation '<is_operated_by>'. On the other hand, concepts 'Team' and 'Plumbing_Fixture_and_Sanitary_Washing_Unit', were not boosted, meaning that their respective weights were decreased after vector normalization.

4.4.3 Creation of Ontological relations phase

When using domain ontologies for identifying semantically related entities, it is possible that the number of relations between entities in a knowledge base be much larger than the number of entities themselves. Using ontological relations to find related entities can result in a large number of results, therefore there is a need for adopting an appropriate ranking scheme, where only the most relevant relations between two entities are provided. This work on creation and ranking ontological relations proposes the adoption of machine learning techniques to determine the relevance of semantic relations in an ontology.

The method proposed here adopts an association rules learning technique in order to discover relevant relations among key terms in a document corpus, and additional human input to perform the mappings between terms (frequent "itemsets") and ontological concepts and the establishment of the final scores on each relation. Simply, frequent "itemsets" are groups of items that often appear together in the data.

Figure 4.16 depicts the process of discovering relevant relations from a document corpus, and how such relations can be ranked in order to define the level of relatedness between ontological entities. The objective is focused in mapping frequent itemsets with ontological concepts of four main modules, namely 'Document Analysis', 'FP-Growth', 'Association Rules' and 'Frequent Itemset Mapping'. As previously explained, 'Document Analysis' module creates a statistical vector for each document analysed. From that point on, the FP-Growth module discovers frequent items that appear in the statistical vector. The 'Association Rules Module' discovers, then, relevant patterns within the statistical vector. The last module maps each frequent itemset within the association rule module, with ontological concepts available in the domain ontology.

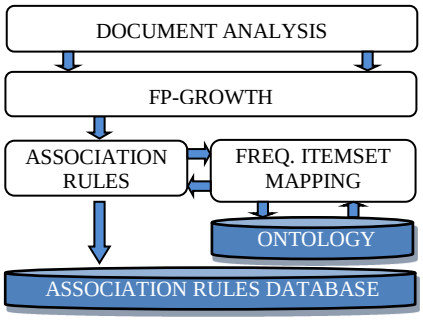


Figure 4.16. Ontological Relations creation process

The frequent-itemsets problem is that of finding sets of terms (items) that appear together in at least a threshold ratio of statistical vectors (also named *transactions* in FP-Growth related literature). This threshold is defined by the 'minimum support' criteria. The support of an itemset is the number of times that itemset appears in the document corpus (example set) divided by the total number of documents.

Association rules are derived from the frequent itemsets. The FP-Growth module finds the frequent itemsets and Association Rules module uses these frequent itemsets for calculating the association rules.

For this work this is interesting to calculate, if concept A appears in a document corpus, what is the probability of concept B also appears in the same corpus? The outputs of the Association Rules module and their definitions are described below:

Confidence is an estimation of the probability of observing Concept B given Concept A, in a valid range between 0 to 1.

Support (also called frequency) is a statistical measure defined as the support of a set of items, represents the percentage of transactions from a database that contains such items. Valid values are also between [0..1]. Higher value means more frequent concepts.

Lift (or interest) is a measure to indicate the independence of Concept A from Concept B. The values are within [0, +∞[and it is given by: $lift = \frac{confidence}{expected\ confidence}$. The expected confidence is identical to the support of the Concept B *S(B)*. It is assumed in the definition of the expected confidence that there is no statistical relation between the Concept A and Concept B. This means that the occurrence of the Concept A does not influence the probability for the occurrence of the Concept B and vice versa. The lift is a value between 0 and infinity:

- A lift value greater than 1 indicates that the Concept A and the Concept B appear more often together than expected, this means that the occurrence of the Concept A has a positive effect on the occurrence of the Concept B.

- A lift smaller than 1 indicates that the Concept A and the Concept B appear less often together than expected, this means that the occurrence of the Concept A has a negative effect on the occurrence of the Concept B.
- A lift value near 1 indicates that the Concept A and the Concept B appear almost as often together as expected, this means that the occurrence of the Concept A has almost no effect on the occurrence of the Concept B.

Conviction is a measure to help of the use of confidence and lift. This measure is used for implication, it matters the distance that it happens (A=>B ≠ B=>A). It can be interpreted as the ratio of the expected frequency that A occurs without B. The value represents the level of implication, as the higher the value, the higher the value of the relationship between both concepts. Like Lift, if the value is 1 the concepts are independent. It has values from [0, +∞[.

PS (also known as Rule Interest, Novelty or leverage) is obtained with the difference between the real support and expected support. Valid values are between [-0.25..0.25]. If a rule equals 0, then the Concepts are independent. A higher value of PS means a higher significant and interesting rule.

Laplace is a "confidence estimator" and also a statistical measure, which indicates that if the support of a given Concept A decreases, its relevance also decreases. This metric had to be parameterized, due to lack of computational power. It was defined a confidence parameter with a value of 0.01. This value was almost zero, because we were interested in all the rules with support above 0.2. This is possible, as we will see, because the corpus of the analysis is small. As we increase the number of documents to be processed, we would help to better parameterize this value. One can start from 0.8 and then start decreasing its value.

- ***Frequent Itemset Mapping***

In this step, frequent itemsets appearing on the relevant association rules will be mapped into ontology equivalent terms, aiming to annotate each frequent itemset with concepts from the domain ontology.

The mapping process is done by calculating the level of similarity between Frequent Itemsets (FI) and Ontology Equivalent Terms (OET). For instance, considering the following FI "waste", several OETs will be considered as possible candidates for mapping, "waste management" or "waste management facility" or even "waste management equipment", with different degrees of similarity. The level of similarity between FIs and OETs is calculated by using the cosine similarity, which is one of the classical information retrieval approaches for similarity ranking.

In this case, two n-grams need to be compared. The cosine similarity can be applied as follows:

$$\frac{(FI\ shared\ Terms) * (OET\ shared\ Terms)}{(FI\ Total\ Terms) * (OET\ Total\ Terms)} - \delta \qquad (4.12)$$

In this equation, “shared terms” are the terms that are in both FI Total Terms and OET Total Terms. The level of similarity is represented between [0, 1], where 0 specifies no similarity between both n-grams, and 1 represents similar n-grams. The variable δ was added in order to deal with n-grams that are similar but the order of terms co-occurring in both n-grams is different. We subtract “0.01” for each equal word in a different place inside the vector, $\delta \in [0{,}02;0{,}03]$. Some examples of several possible cases are illustrated in Table 4.6.

Table 4.6. Examples of Similarities between FI and OET

Case 1: Similarity=1	Case 2: Similarity=0	Case 3: 0 < Similarity < 1
FI={Waste Management Facility} **OET={Facility Waste Management}** **FI Total Terms=3** **OET Total Terms=3** **Shared terms=3** **δ = [All equal terms in different positions] = 0,01*3**	FI ={Waste Management Facility} OET={Complete Chimney System} FI Total Terms=3 OET Total Terms=3 Shared terms=0 δ = [Not applicable]	FI ={Waste} OET={Waste Management Facility} FI Total Terms=1 OET Total Terms=3 Shared terms=1 δ = [Not applicable]
$Similarity = \frac{3^2}{3*3} - 0{,}01*3 = 0{,}97$	$\boldsymbol{Similarity = \frac{0^2}{3*3} = 0}$	$\boldsymbol{Similarity = \frac{1^2}{1*3} = 0{,}33}$

This procedure, aims to reduce the level of uncertainly regarding similarity between FI and OET. Figure 4.17 depicts an example of a frequent itemset mapping, where the similarity of all ontological concept candidates is represented by a colour code to support users within the mapping process.

After defining the ontological rules through the frequent itemset mapping module, they must be stored. Such rules will be stored under a relational database schema, selected by the user. He/she will choose the ones that are considered to be more relevant, based on the similarity measures and in the metrics provided by the association rules method. Table 4.7 contains two columns for the concepts “premise” and “conclusion”, and additional columns for the association rules metrics, each of them with a value corresponding to “confidence”, “conviction”, “lift”, “total support”, “laplace”, “gain”, “Ps”.

Table 4.7. Representation of Association Rules

#	Premise	Conclusion	Confidence	Conviction	Gain	Laplace	Lift	PS	Total Support

1	**Concept A**	**Concept B**	Value A	Value B	Value C	Value D	Value E	Value F	Value G

Each row represents the relevant association rules retrieved from the document corpus. For each “premise” and “conclusion”, the user can select the best matching ontology concept that relates to each particular frequent itemset. The level of relevance is given by the application of cosine similarity (previously presented) and a colour code which identifies the best matching candidates (Figure 4.18).

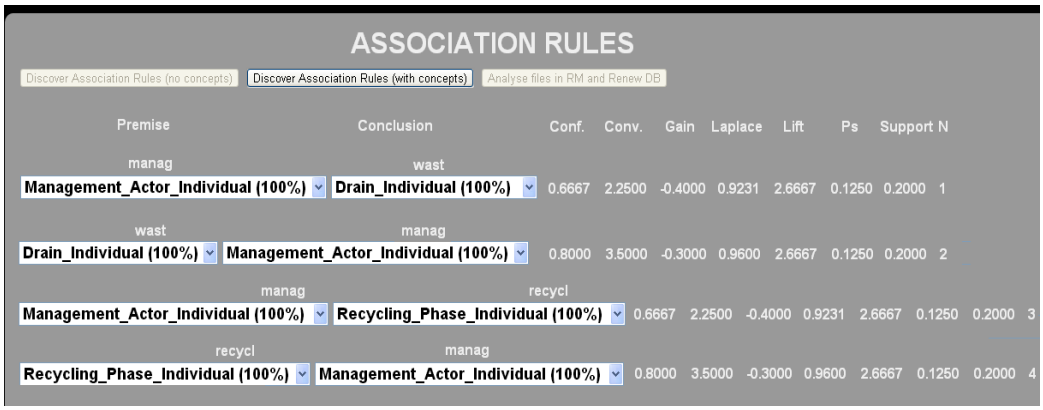


Figure 4.17. Association Rules results

If there are no 100% matches the process will try to get any match. In these cases it will only be showed the red colours with each corresponding similarity measure. When no match at all is found, the process will state that no concept was matched, and asks the user to add new concept to the ontology.

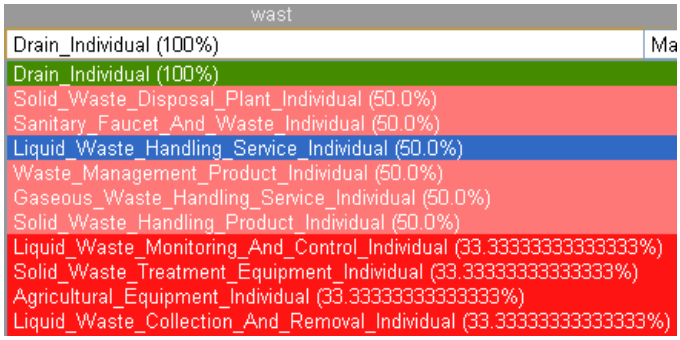


Figure 4.18. List of candidate ontology concepts

For this particular example, we started with a set of scientific articles published, related with building and construction domain. Each article evaluated was composed by an average of 3.500 terms, which is considered a relevant number to test the scalability of the proposed solution.

The support value used was 0.2, meaning that only association rules whose support value has equal or greater than 0.2 were retrieved. This requirement does not only makes the process of

association rules discovery more efficient in terms of computational requirements for processing this job, but on the other hand only the more significant rules are extracted.

Figure 4.19 depicts an example of a relevant association rule retrieved and the matching ontology concepts. This example shows that the frequent itemsets “wast” and “manag” are related. These itemsets were mapped into the corresponding ontology concepts, namely “Drain” and “Management Actor”. Such assertion leads to conclude that, regarding the documents analysed, there is a relevant relation between waste management projects and management actors involved within such projects.

The metrics calculated for the currently example are described below:

Premise: Drain_Individual

Conclusion: Management_Actor_Individual

Confidence:0.67; Conviction: 2.25; Gain:-0.40;

Laplace:0.92; Lift:2.67; PS:0.13; Support:0.20;

Figure 4.19. Association Rule example

The relation between the ontological concepts “Drain” and “Management Actor” are then stored into the association rules database with the weight, which specifies how those tow concepts are related. At this stage, it is assumed an ontological weight similar to the level of confidence provided by the association rule module. It is to be considered as future work, that such metric for determining the degree of relation between concepts needs to be improved.

As a concluding remark to this section, it is worth to mention that the conceptualization of the semantic enrichment process, takes into account several dimensions as an essential requirement for the overall approach. As major dimensions, this work comprehends: (i) the nature of the knowledge sources and how those must be collected and represented; (ii) the role of actors within the whole process, acting as holders of knowledge and specialists in its formulisation; (iii) the ontological model as the backing for the entire process; (iv) and the methods used for the discovery of ontological relations from knowledge sources and also for its semantic enrichment.

The next chapter, presents the aspects dealing with the proof of concept design and implementation.

5

Proof of Concept – Design and Implementation

"Computers themselves, and software yet to be developed, will revolutionize the way we learn."

- Steve Jobs (1955 – 2011), co-founder, chairman, and CEO of Apple Inc

Chapter 5 describes both the design and implementation of the proof of concept (from now on referred to as the SENSE software platform, where SENSE means *Semantic Enrichment kNowledge SourcEs*) providing the computational functionalities required to assess the hypothesis guiding this work and, ultimately, to provide an answer for the research question

underlying this thesis scope. The proof of concept covers the semantic enrichment process, (partially) ontology evolution, and classification & searching of knowledge sources.

This chapter is organized as follows. Section 5.1 introduces the notation used to formally support the design of SENSE platform. Section 5.2 describes SENSE design, covering functional, architectural, and behavioural views. Finally, Section 5.3 describes the implementation details of all components forming SENSE.

5.1 Notation

Unified Modelling Language (UML) is one of the most popular notations for visualizing, specifying, constructing, and documenting the components of software and non-software systems. Efficient and appropriate use of notations is very important for making a complete and meaningful model. Any given model is useless unless its purpose is depicted properly.

Latest UML version has 14 types of diagram divided into two categories: Structure and Behaviour. Structure diagrams define the static architecture of a model. They are used to model the 'things' that make up a model, namely classes, objects, interfaces, and physical components. Additionally, they are used to model the relationships and dependencies between elements. Behaviour diagrams capture the varieties of interaction and instantaneous states within a model as it 'executes' over time, tracking how the system will act in a real-world environment, and observing the effects of an operation or event, including its results.

5.2 Project Design

SENSE design was specified adopting 3 different views (functional, architectural, and behavioural). The functional view shows the interactions between external entities and SENSE. The architectural view presents, in a 3-tier model (presentation, control and data layers), the software structure supporting SENSE. The behavioural view describes the interaction between the components of the software structure.

5.2.1 Functional View

Use Case (UC) diagrams are used to model user/system interactions. They show the interaction between a given system and external entities (external entities are referred to as actors). Actors represent roles that may include human users and other systems. Each use case is a single scenario of meaningful work, providing a high-level view of behaviour observable from outside the system.

SENSE functional view is presented in its major dimensions (pre-processing, ontology evolution, semantic vector creation, classification, and evaluation) including also the usage point of view (search KS).

Pre-processing UC diagram (Figure 5.1) involves two different actors, namely domain expert and ontology expert. In this case, SENSE enables the domain expert to perform several activities, presented as UCs which are described as:

- Informal Conceptualization of Domain Ontology: involves the domain expert actor, which is responsible for informally conceptualizing the domain ontology;
- Collect Relevant KS: is performed by the domain expert, which uses the ICONDA library searching capabilities for collecting the relevant knowledge sources that will be used for evaluating the proof of concept;
- Label Relevant KS: takes into account the labelling each individual KS;
- Store Relevant KS: by using the SENSE platform, the domain expert is able to store KS's in the document repository.

From the ontology expert point of view, SENSE enables the following activities:

- Formalization of Domain Ontology: where the ontology expert is able to specify the domain ontology in a formal way in OWL format
- Load Domain Ontology: into the knowledge repository.

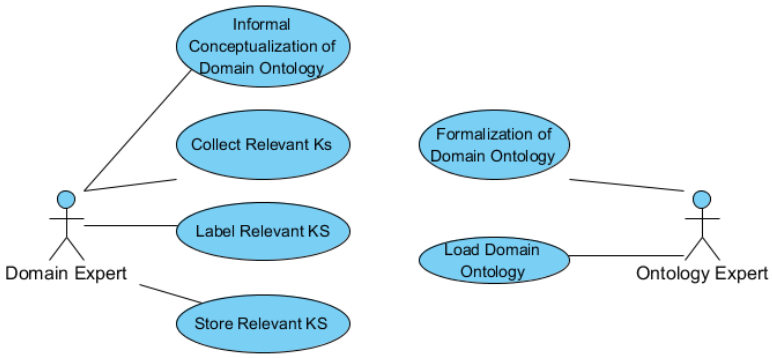


Figure 5.1. Pre-Processing Diagram

The ontology evolution is mainly driven by its usage whenever a new KS is introduced into the knowledge repository; nevertheless the domain expert is responsible for managing such evolution in the following ways:

- Add Ontological Rule: that is discovered whenever a new KS is introduced in the knowledge repository. In this case the domain expert is able to add a new rule into the domain ontology;
- Update Ontological Rule: relates to updating the weight of an ontological relation already existing within the domain ontology. This is triggered when new KSs are introduced in the knowledge repository, which causes existing ontological rules update their weights accordingly;
- Delete Ontological Rules: can be triggered by the domain expert when obsolete ontological relations are identified, the domain expert is able to select the rules from the ontology that need to be deleted;
- Add Ontological Concept: which is triggered whenever a new KS is introduced in the knowledge repository. In this case the domain expert is able to add a new concept into the domain ontology;
- Update Ontological Concept: relates to updating an already existing concept within the domain ontology, by moving its position in the taxonomy hierarchical structure or updating the equivalent terms related to each particular concept. This is triggered when new KSs are introduced in the knowledge repository, which causes existing ontological concepts to be updated;
- Delete Ontological Concept: can occur when obsolete ontological concepts are identified by the domain expert. If this situation occurs, the domain expert is able to select the concepts from the ontology that need to be deleted;

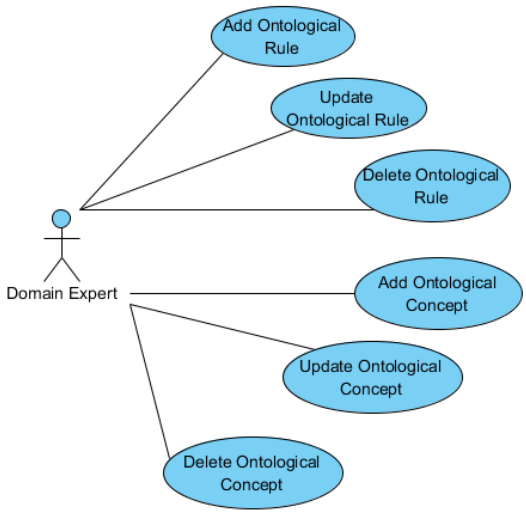


Figure 5.2. Ontology Evolution Diagram

The process of semantic vectors creation is triggered by the domain experts. He/she is able to trigger the creation of KRs in a statistical or semantically enabled form, each time a new KS is stored. This process automatically creates an equivalent KR for each new KS stored in the knowledge repository. The creation of KRs is also automatically triggered by SENSE when a user queries the knowledge repository. The UC diagram presented in Figure 5.3 illustrates the main functionalities in creating a semantic vector, which can be described as follows:

- Create Statistical Vector: generates a statistical representation for each KS taking into account the application of the *tf-idf* algorithm. Such representations are automatically created by SENSE whenever a new KS is introduced or when a new query is launched for searching available KS;
- Create Keyword-based Vector: generates a representation for each KS in a semantic vector keyword-based form. It is worth noticing here that in order to generate a semantic vector keyword-based, a statistical vector must first be generated;
- Create Taxonomy-based Vector: enables the generation of semantic vector taxonomy-based for each KS, which takes into account the hierarchical relations between ontology concepts. The generation of the semantic vector taxonomy-based builds upon the semantic vector keyword-based;
- Create Ontology-based Vector: enables the creation of semantic vector ontology-based. This process takes into account the ontological relations between concepts and their weights. The generation of the semantic vector ontology-based builds upon the semantic vector taxonomy-based.

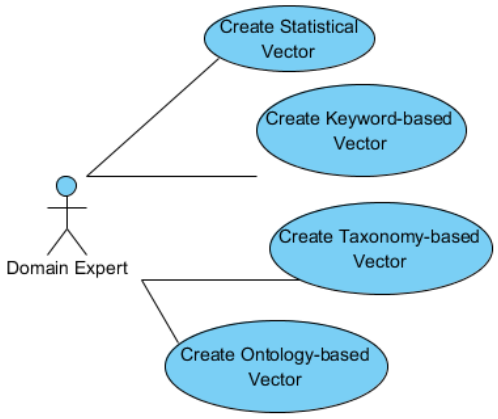


Figure 5.3. Semantic Vector Creation Use Case

The classification process enables the domain expert to run a clustering analysis with KRs associated to each KS and it is illustrated by the UC diagram (Figure 5.4). The clustering is applied separately to each vector set type, in order to assess the true semantic enrichment into the different KRs. The assessment is performed by first applying the clustering method into the statistical vectors, secondly the clustering is applied into the semantic vectors keyword-based, the next stage is to apply the clustering into the semantic vectors taxonomy-based and lastly, to apply the clustering into the semantic vectors ontology-based. The clustering method adopts an implementation of the K-Means algorithm, where metrics of precision and recall are presented to the user after each classification for further evaluation.

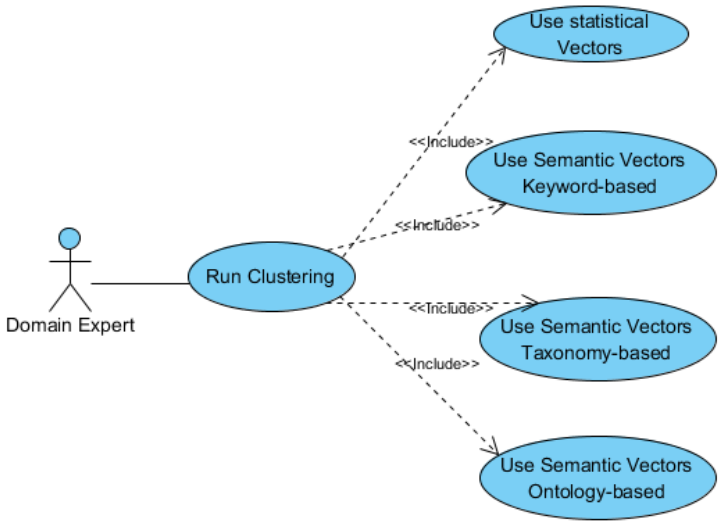


Figure 5.4. Classification Diagram

By the time the clustering is applied into all KRs, the domain expert is able to assess where the semantic enrichment had effect and where it did not. Results can be assessed by analysis of the precision of recall metrics provided by the clustering method (Figure 5.5).

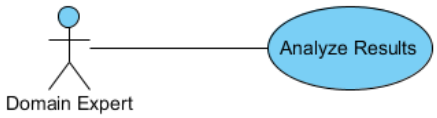


Figure 5.5. Evaluation Diagram

Search for KS UC diagram (Figure 5.6), depicts a scenario where a domain expert is particular interested in searching a KS of a particular domain. The search can be performed in two different ways:

- Search by keywords: where the domain expert can adopt using free text search or;
- Search by ontological concepts: where the query is constructed using concepts from the domain ontology.

The domain expert can choose the method where the search must be performed (against statistical or semantic representations). The idea here is to compare the results provided by each method and check if the use of semantic vectors can bring improved results. The results provided by SENSE are ordered by the relevance of the query to each KR, where the magnitude of relevance is calculated using the cosine similarity.

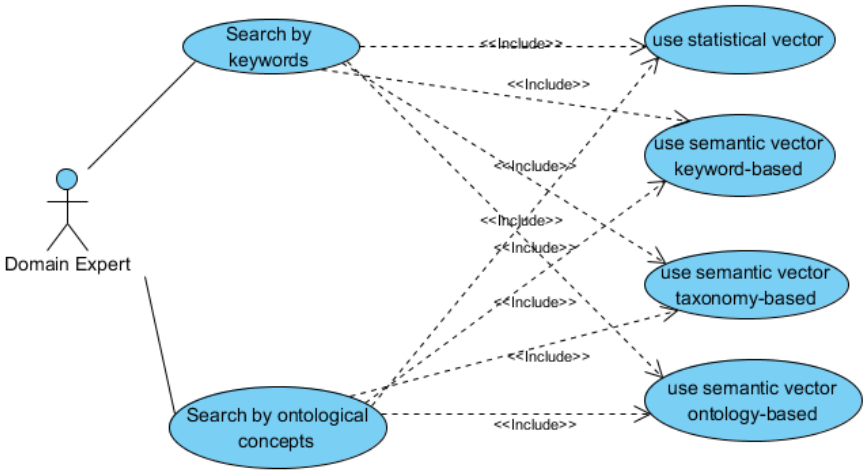


Figure 5.6. Search KS Diagram

5.2.2 Architectural View

SENSE architectural view is described here using UML component diagrams, which illustrate the pieces of software that will make up a system. Components are building blocks that can eventually encompass a large portion of a system.

SENSE Component diagrams (Figure 5.7) cover three different layers, namely Presentation, Control, and Data. Presentation layer holds the components handling the interaction between external agents and system – the interface components. Control layer holds components

responsible for implementing the business logic to perform the operations required for semantic enrichment. Finally, Data layer contains components to handle all data sources used in SENSE.

The interface components are divided in two groups: Web and Desktop interfaces. Web group contains the following components:

- *KS Manager Interface*: provides required functionalities for managing knowledge sources, namely store, update, and delete;
- *KR Creator Interface*: display the functionalities supporting creation of the statistical and semantic vectors for each KS stored in the knowledge repository;
- *Ontology Controller Interface*: supports the user in managing the domain ontology in the following ways: (i) by adding new discovered concepts and (ii) discovery of new ontological rules from the KS in the repository; controller
- *KS Searcher Interface*: enables the user to perform queries into the KS repository. The results of each query are presented by order of relevance.

Desktop interface is based on the *Clustering* component, which enables the user to design the workflow for performing a clustering analysis in order to assess in which cases there was a significant semantic enrichment. It is worth mentioning that this component has not been developed in this thesis but included in the SENSE platform since clustering capabilities were part of the assessment process.

The Control layer holds the following components:

- *KS Manager*: implements the operations regarding storing, retrieve, updating and deleting KS form/to KS repository.
- *Statistical Vector Creator*: creates a statistical representation for each KS available in the knowledge repository.
- *Semantic Vector Keyword-based Creator*: creates a semantic representation keyword-based, for each KS available in the knowledge repository.
- *Semantic Vector Taxonomy-based Creator*: creates a semantic representation taxonomy-based, for each KS available in the knowledge repository.
- *Semantic Vector Ontology-based Creator*: creates a semantic representation ontology-based, for each KS available in the knowledge repository.
- *Rule Manager*: implements the methods required for discovering new ontology rules from KS available in the knowledge repository.
- *KR Creator*: encapsulates the functionalities from the statistical vector and semantic vector components into a higher level of abstraction.

- *Clustering Generator*: implements machine learning libraries for performing a non-supervised classification, taking as inputs the KRs.
- *Query processor*: transforms a given user query into a vector representation.
- *Ontology Controller*: set of methods responsible for discovering new semantic rules from KS and create/update/delete concepts from the domain ontology.
- *KS Searcher*: enables users to search for KS available in the knowledge repository. The search can be performed by two different methods (free text search and search using ontology concepts).
- *Similarity Controller*: provides mechanisms to compare queries and KRs, using cosine similarity algorithm, and presents the results by order of similarity.

Data layer contains the following components:

- *KS Repository*: holds all the KS (documents).
- *KR Repository*: holds all the KRs.
- *Domain Ontology*: holds the domain ontology (concepts, individuals, and equivalent terms) and also the semantic relations among concepts.

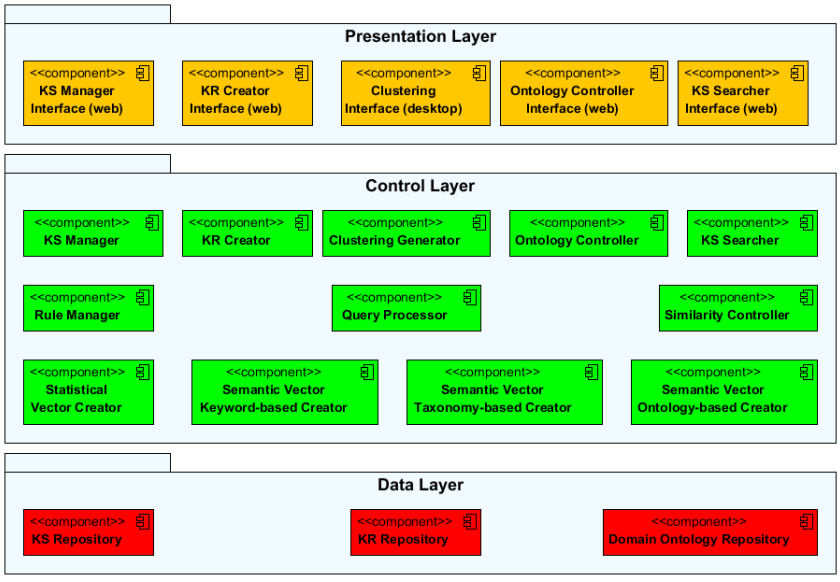


Figure 5.7. SENSE Components Diagram

5.2.3 Behavioural View

Sequence Diagrams (SDs) describe communication among software objects during SENSE execution, and what messages trigger those communications.

The pre-processing SD (Figure 5.8) describes the scenario for setting-up the environment and instantiation of the repositories. This process is triggered by the *domain expert* actor who conceptualizes the domain ontology and delivers the conceptualization to the *ontology expert*, who formalizes the ontology using OWL. After the formalization, the ontology will be loaded into the Ontology Repository through the Ontology Manager component. After loading the domain ontology, the *domain expert* will collect and label the relevant KS, and load them into the KS repository. This is performed by the KS Manager component.

For the sake of clarity, it is important to emphasise that SDs describe the interactions between actors and software objects. In Figure 5.8, presents an interaction between two actors in the pre-processing stage. Such interaction was presented deliberately for proving a more clear understanding about the different roles of these two actors. Although the ontology expert is responsible for formalizing the ontology and loading it in the ontology repository, the domain expert has a more relevant role not only in this stage, but on the entire overall process of semantic enrichment and evaluation, therefore the domain expert is described for providing an overall coverage on this stage in particular.

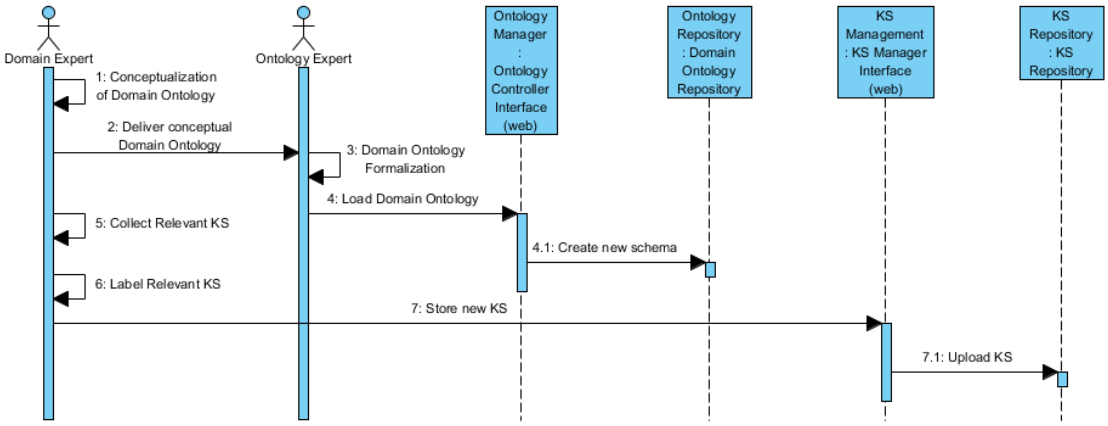


Figure 5.8. Pre-Processing Diagram

The ontology evolution SD (Figure 5.9) describes the scenario for enriching the domain ontology with new rules and concepts discovered from the new KS that were added into the *KS repository*. The process is triggered by the domain expert, which aims to discover new ontological rules from the KS sources that are stored in the *KS repository*. This is performed by the *Ontology Controller* component, which apply the association rule algorithm to KSs and sends the list of the new association rules discovered. The domain expert needs to manually map the frequent itemsets with ontology concepts, creating a new ontology rule. It can happen that a frequent itemset may not have a direct mapping with an ontology concept; in such a case, a new ontology concept must be created. The graphical user interface, provided by the *Ontology Controller Interface (web)* component, enables the user to add new concepts into the

ontology tree. Once all the mappings are performed, the domain expert is able to store the new discovered rules into the *KR repository*.

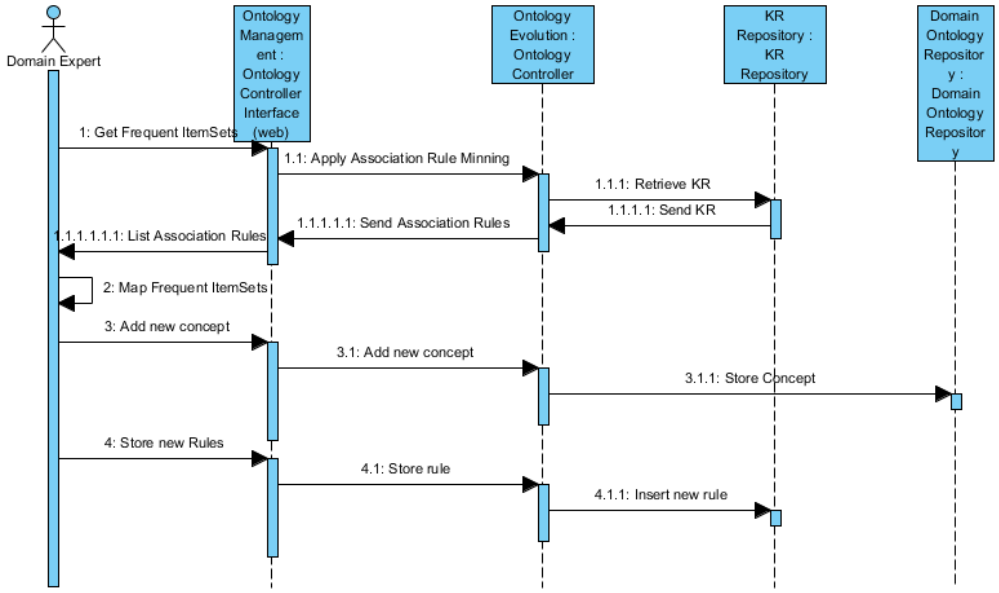


Figure 5.9. Ontology Evolution Diagram

The semantic vector creation diagram (Figure 5.10) describes the creation process of KRs, for each KS available in the *KS repository*. As already mentioned before, this comprehends a four-stage approach. First the domain expert needs to create a statistical representation for each KS. The domain expert accesses the *KR Creator Interface*, which interacts with the *KR Creator* that, in its turn, requests the *Statistical Vector Creator* component to perform a statistical representation and store the representations on the *KR Repository*.

The next phase deals with the creation of the semantic vectors keyword-based. The interactions between components are the same as for the statistical vector creation. Here, the *Semantic Vector Keyword-Based Creator* applies a semantic enrichment into the previous statistical vectors taking into account the ontology concepts. The new semantic vectors created are then stored in the KR repository.

The semantic vector taxonomy-based creation is performed taking into account the already existing semantic vector keyword-based. This operation is accomplished by the *Semantic Vector Taxonomy-Based Creator* component, which reads the KRs from the repository and applies the taxonomy rules to the semantic vector keyword-based, generating new KRs.

Finally, the creation of the semantic vectors ontology-based is conducted by the *Semantic Vector Ontology-Based Creator* component, triggered by a request from the *KR Creator*. It reads the semantic vectors taxonomy-based and applies the ontology rules to them, in order to create a new KR.

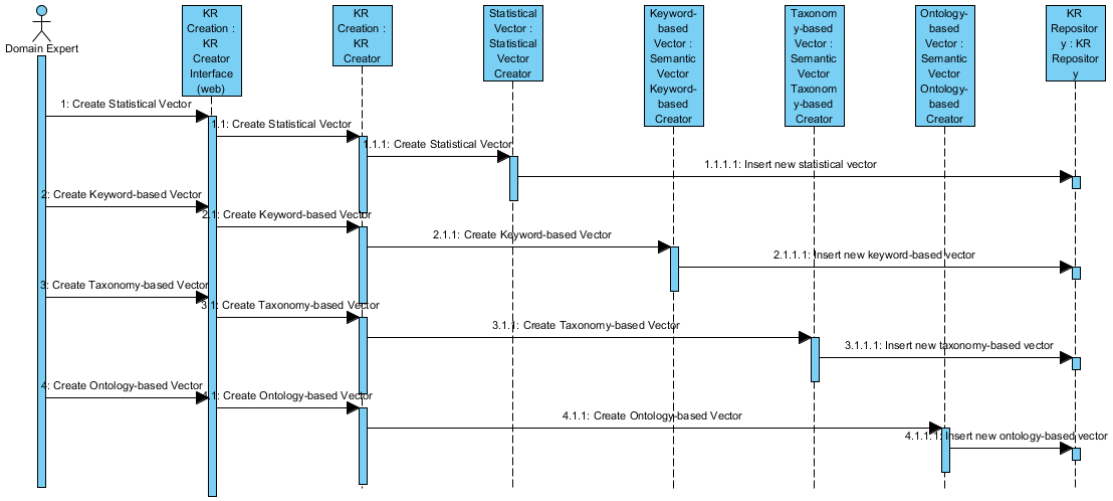


Figure 5.10. Semantic Vector Creation Diagram

The classification process (Figure 5.11) enables the domain expert to perform a cluster analysis of all types of KRs, and to assess where semantic enrichment was evident. This is performed using the *Clustering Interface* component, where the user may choose which type of KRs (statistical or semantic representations) the clustering method should apply. The objective is to apply the clustering algorithm into all types of KRs and measure the effectiveness of the semantic enrichment. *Clustering* component interacts with the KR repository in order to retrieve each individual KR. The K-Means algorithm is used to compute the classification, and the results are sent to the domain expert in a tabular or graphical representation.

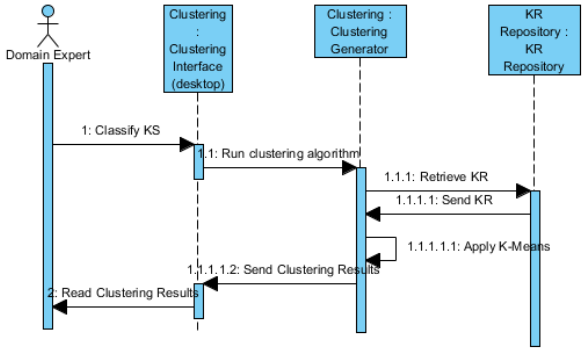


Figure 5.11. Classification Diagram

The search for KS diagram (Figure 5.12) illustrates the process of finding KSs relevant for a given query. This process results from the interaction between *KS Searcher Interface* and *KS Searcher components*. The process starts with the *Query Processor* component, which processes a given query and transforms it into a vector. The next step deals with retrieving the KR from the repository and applies a similarity algorithm in order to compare the query with the KRs. This last step is performed by the *Similarity Ranking Controller* component. The results are presented to the user through the *KS Searcher Interface*.

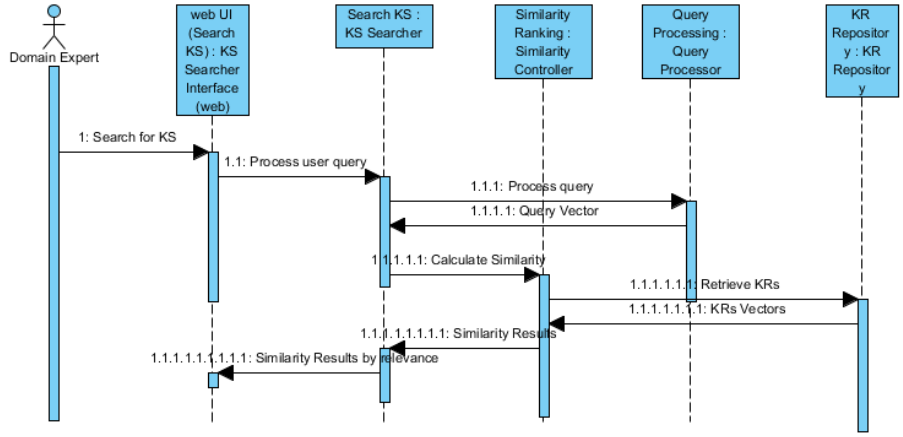


Figure 5.12. Search KS Diagram

5.2.4 Data Model (ERD)

The data model which comprises SENSE, is described using an Entity-Relation diagram notation. For detailed description about SENSE data model, please refer to Annex A – Data Models (Entity-Relation Diagrams).

5.3 Implementation

SENSE implementation adopts a three-tier approach, where presentation, application processing, and data are logically separated (Figure 5.13).

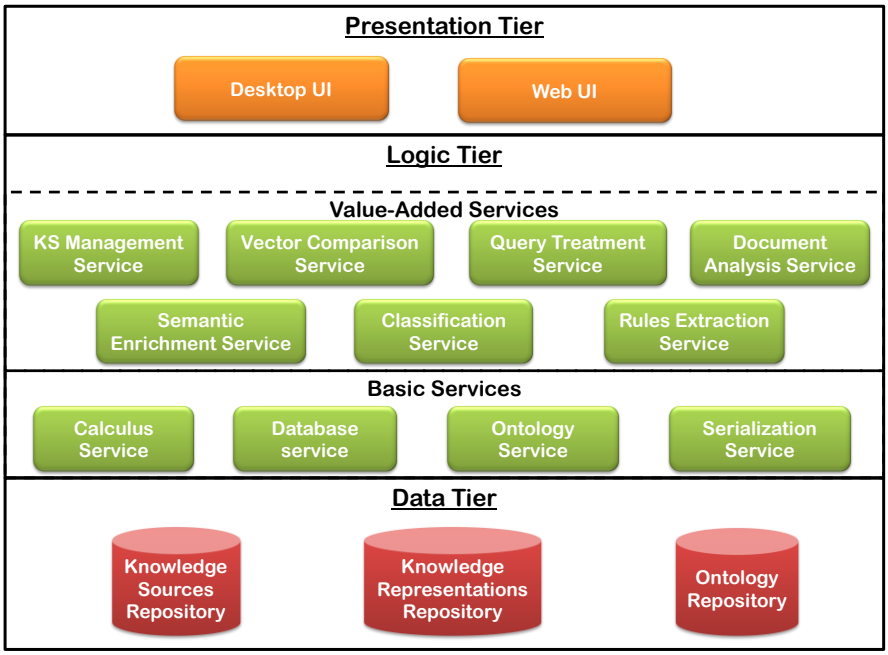


Figure 5.13. Technical Architecture

The presentation tier is the SENSE access window, holding interactions between external agents and SENSE components (and their respective services). It communicates with other tiers by which it puts out the results into the web browser or desktop application (in simple terms it is a layer which users can access directly such as a web page, or an operating systems GUI).

The logical tier is pulled out from the presentation tier and it controls all SENSE's functionalities by performing detailed processing. The SENSE's components described previously are implemented by a set of web-services at logic tier, which is organised in 'basic' and 'value-added' services. Basic services perform low-level functionalities, like direct interaction with databases and ontology, mathematical computation and data serialization for the web-services interface. Value-added services are considered to be high-level functionalities such as semantic vector creation, query treatment and vector comparisons, which are supported by the low-level functionalities provided by the basic services.

Basic services are established by the following: Serialization Services, Calculus Services, Ontology Services, and Database Services. Serialization Services are used by the Web Services Interface to marshal and unmarshal information to and from XML format. Calculus Services are responsible for the needed mathematical computations, as the TF*IDF algorithm and the cosine similarity algorithm. Database Services manage connections and interactions

with the system's database and knowledge source repository. Finally, Ontology Services comprise all methods to persist the system's ontology on a relational database.

Value-added services are a set of se of services established by the following: Document Analysis, Query Treatment, Vector Comparison, KS Management, Semantic Enrichment, Classification and Rules Extraction. Document Analysis, creates a statistical representation for all KS. Semantic Enrichment Services comprises all functions associated with the creation of all three iterations of semantic vectors. Query Treatment Services are responsible for user queries' treatment. Document Comparison Services contain all methods that support the comparison between vectors and ranking the results of this comparison. KS Management are services responsible for storing and retrieving KS from the KS repository. Classification services are responsible for applying data mining techniques for clustering KRs. Rules extraction services are responsible for discovering ontological relations into KRs. For more detailed information about services implementations, please refer to the annex chapter.

Table 5.1, illustrates the mapping between SENSE components and their technical instantiations through web-services implementations.

Table 5.1. Mapping between SENSE components and services

Component	**Service**
KS Manager	KS Management Service
Statistical Vector Creator	Document Analysis Service, Calculus Service
Semantic Vector Keyword-based Creator	Semantic Enrichment Service
Semantic Vector Taxonomy-based Creator	Semantic Enrichment Service
Semantic Vector Ontology-based Creator	Semantic Enrichment Service
KR Creator	Database Service
Rule Manager	Rule Extraction Service
Clustering Generator	Classification Service
Query Processor	Query Treatment Service, Calculus Service
Ontology Controller	Ontology Service
KS Searcher	Database Service
Similarity Controller	Vector Comparison Service

The data tier consists of database servers. Here information is stored and retrieved. This tier keeps data neutral and independent from application servers or business logic. The proposed architecture adopts three distinct repositories: a KS repository, where knowledge sources are stored, an ontology-persisted database map, to provide the system access to the ontology through the Web, and KR database, to store statistical and semantic representations. The knowledge source repository generated by a third-party tool, and the ontology repository is automatically created by the component responsible for ontology persistence and interaction processes. A full description of the Entity-Relation diagrams for each repository is available on Annex A – Data Models (Entity-Relation Diagrams).

The Document table serves as link between knowledge sources in the knowledge source repository and the vectors for those sources. This link is provided by having an identification number for the knowledge source repository, with which knowledge sources can be retrieved according to their semantic vectors. StatisticWeight stores knowledge sources' statistic vectors and KeywordBasedSemanticWeight, TaxonomyBasedSemanticWeight and OntologyBasedSemanticWeight tables store the three iterations of semantic vectors created by SEKS. OntologyRelation and TaxonomyRelation tables are used to keep track of ontological and taxonomical relation occurrences within the knowledge source repository, respectively. RelationImportance table stores the ontological relation importance used on ontology-based semantic vectors creation. The full list of store procedures created in the RDBMS is available in Annex B – Stored Procedures.

5.3.1 Technologies

This section illustrates the main technologies adopted to develop SENSE proof-of-concept. All the technologies used are under open source licencing, and range from DBMSs to CMSs. The objective was to rely on solid, proven technologies which are widely used by the scientific community.

SENSE was designed/modelled with Visual Paradigm for UML. SENSE database was implemented in MySQL, and designed with MySQL Workbench, which is a visual tool for SQL database development. SENSE ontology was coded in OWL-DL with the Protégé Ontology Editor, a visual ontology-editing tool supporting OWL and RDF.

The logic tier was mainly implemented in Java programming language and was developed using Eclipse IDE, which provides a visual integrated environment for several programming languages and paradigms. SENSE is deployed using Apache Tomcat 7 server, which is web application container that supports Java applications. The interaction with the ontology is managed by JENA Semantic Framework.

Communication between View and Controller layer is made with Java Servlets 3.0 technology. In the presentation tier, the user interface was implemented with HTML 5 and CSS 3, and used

jQuery JavaScript Library to perform AJAX requests to the server, event handling and animations, and the Web Services interface is implemented using JAX-WS RI framework, which provides tools and infrastructure to implement Web Services. Table 5.2 shows the core technologies used and a brief introduction to each one.

Table 5.2. Technologies used

	SENSE is strongly relying on Java technology. The web user interface was developed using Java servlets 3.0. Control layer components were developed JAX-WS, a java based API for developing web services. The interface between the control layer and data layer was developed using Java classes.
	OWL, the W3C recommendation, is a knowledge representation language for specifying ontologies. OWL was adopted to formalize the domain ontology.
	Apache Tomcat was the web server chosen to deploy SENSE. It is an open source servlet container that provides server-side capabilities for deploying Java-based web applications.
	Protégé is an open source ontology editor. It was used in the pre-processing stage of the conceptual model, as a mean to formalize the domain ontology in OWL language.
	RapidMiner is a software tool that provides an environment for machine learning, data mining and text mining. The libraries provided by RapidMiner were used in order to create the statistical vectors, perform the association rule mining and clustering algorithms. Such functionalities were modelled in RapidMiner as workflows (RapidMiner process) and then used in a Java project environment through RapidMiner API.
	Jena is an open source Semantic Web framework for Java, it provides an API to extract data from and write to OWL models. Jena was used as an abstraction layer, able to manipulate the domain ontology serialized in a relational database. It enables to access the ontology contents by the implemented java classes in the control layer.
	Liferay is an open source enterprise web portal built in Java. It provides document management functionalities. It was used as a KS repository but also, providing functionalities to manage such

KS.

MySQL is an open-source relational database management system (RDBMS). It was used as KS, KR repository and domain ontology repository.

5.3.2 Database Stored Procedures

For a full description of the business logic, implemented by MySQL stored procedures, please refer to Annex B – Stored Procedures

5.3.3 RapidMiner Workflows

For a full description of the RapidMiner workflows implemented, please refer to Annex D – RapidMiner Workflows.

6

Evaluation and Analysis

"Good tests kill flawed theories; we remain alive to guess again."

- Karl Popper (1902 – 1994), philosopher and professor

This chapter describes the method used for evaluating the SENSE platform, in other words to assess to what extent semantic enrichment can improve representation of knowledge sources when compared to traditional statistical approaches. The chapter starts by introducing the sample data set used to perform the assessment, followed by the evaluation process including the techniques that were used for evaluation. Next, the techniques for data transformation and cleaning are presented which deal with certain inconsistencies in raw data. Finally, the chapter concludes with the results and analysis of the initial hypothesis validation.

6.1 Data Samples

The data samples (in the form of text documents) used for performing the evaluation task were selected from the B&C domain, since the scenario to be evaluated under the scope of this thesis focuses on the this activity domain.

The task of collecting relevant data sources used the ICONDA® library (IRB, ICONDA®Bibliographic 1986). ICONDA®Bibliographic is a comprehensive worldwide database of systems for retrieval of planning and building related scientific publications. It began life in the mid 1980's as the database of the International Council for Research and Innovation in Building and Construction (within CIB[6]). Since then, ICONDA has found a key role in various information products. Fraunhofer IRB, the Information Centre for Planning and Building of the Fraunhofer-Gesellschaft, presently coordinates maintenance of the database and its marketing. Today, ICONDA is a supranational organisation incorporating content provided by 1 supranational and 23 national organizations in 14 countries worldwide. Access to ICONDA based products is facilitated by a multilingual terminology of around 100 000 terms in English, German, French, and Spanish. Principal terminology sources are the INIST Vocabulary, the Canadian Thesauri, and ICONDA own vocabulary.

For the purpose of evaluation the performance of the SENSE platform, a corpus containing 20 relevant 20 data samples focused on the B&C sector were collected. Table 6.1 describes the data samples collected, showing for each data sample: the document identification, the title each data sample refers to, and the number of words existing in each data sample. It is worth emphasising that the total number of relevant words used for the experiment exceeds 70.000, which turned out as an interesting challenge, as presented later in this chapter.

Table 6.1. Data Samples used for evaluation

[6] CIB is the acronym of the abbreviated French (former) name: "Conseil International du Bâtiment" (in English this is: International Council for Building). In the course of 1998, the abbreviation has been kept but the full name changed into: International Council for Research and Innovation in Building and Construction. CIB was established in 1953 as an Association whose objectives were to stimulate and facilitate international cooperation and information exchange between governmental research institutes in the building and construction sector, with an emphasis on those institutes engaged in technical fields of research.

Doc	Title	Nr. of words
1	Evaluation of Deterioration behavior of Surface Coating for RC Buildings by Permeation-Diffusion	2.291
2	A Study on the Carbonation Progress of Concrete Concerning the Influence of Deterioration of the Coating Material for Textured Finish	2.869
3	The Experimental Evaluation of Parameters Contributing to the Durability of Coating Materials for Colouring and Protecting External Plastered Surfaces	3.744
4	Durability Evaluation of Highly Reflective Coating Materials for Roofing	2.819
5	Waste Today Gone Tomorrow Sustainable Waste Management: Malta, a Case Study	6.706
6	Waste Management Strategies during Post Disaster Phase: A Case of Sri Lanka	4.647
7	Sustainable Construction Waste Management in Malaysia: A Contractor's Perspective	4.714
8	Planning for SMEs' Proactive Waste Management in Office Building Retrofit Projects	4.075
9	Integration of sustainability solutions in sanitary installations: the example of the AveiroDOMUS "House of the Future"	1.801
10	Study of sanitary equipments installed on light-weight partitions	2.564
11	Provision scales of sanitary accommodation in public toilets	3.085
12	Present state and future challenge on installation number of the sanitary fixture	3.790
13	Lighting in New Zealand Homes – Lighting Efficiency as a Sustainability Indicator	3.031
14	A Systematic Review on the Therapeutic Lighting Design for the Elderly	5.656
15	ICT for Energy Efficiency: Towards Smart Buildings, Manufacturing, Lighting and Grids	4.279
16	Light Trespass from Exterior Lighting in Urban Residential Areas of Compact Cities	3.491
17	Sustainable Office Building: Should I Focus on HVAC-system Design or	2.057

	Building Envelope	
18	Earth, Wind and Fire Towards New Concepts for Climate Control in Buildings	3.824
19	Hvac Integrated Control For Energy Saving and Comfort Enhancement	3.553
20	Rfid-Based Occupancy Detection Solution for Optimizing HVAC Energy Consumption	3.316
Total		72.312

Taking into account the amount of words used for the experiment, Figure 6.1 illustrates how such terms are scattered within each data sample. In other words, terms that occur in few documents can be considered good discriminators, as opposed to terms that occur in most of the documents within the data samples. For readability purposes, Figure 6.1 only presents terms with occurrence levels above 100. From this figure is possible to identify which terms can be considered as good discriminators and terms which are bad discriminators. For example, terms such as “disaster”, “requirement” and “malta”, can be considered as good discriminators as they only occur in a single document with levels of occurrence above 100, and “waste” which has an occurrence level below 600 and only occurs in 5 documents. As bad discriminators, terms such as “building”, “time” and “base”, tend to have high levels in occurrence in all of the documents in the corpus knowledge base.

Nevertheless, it is possible to imagine that the amount of good discriminators from the total amount of words available in the data samples tends to be quite “high”. Although the notion of what can be considered as good discriminator and a bad discriminator is a subjective matter, and it must prevail a balanced judgment between avoiding overfitted vectors but at the same time keeping the most relevant terms. Overfitting here relates to a statistical model which describes noise instead of the underlying relationship. Overfitting generally occurs when a model is excessively complex, such as having too many parameters relative to the number of observations. In order to avoid overfitted vectors, the domain ontology will work as filter where only good discriminators which in fact are related with the B&C will be considered and all other will be rejected, as it will be further discussed and analysed.

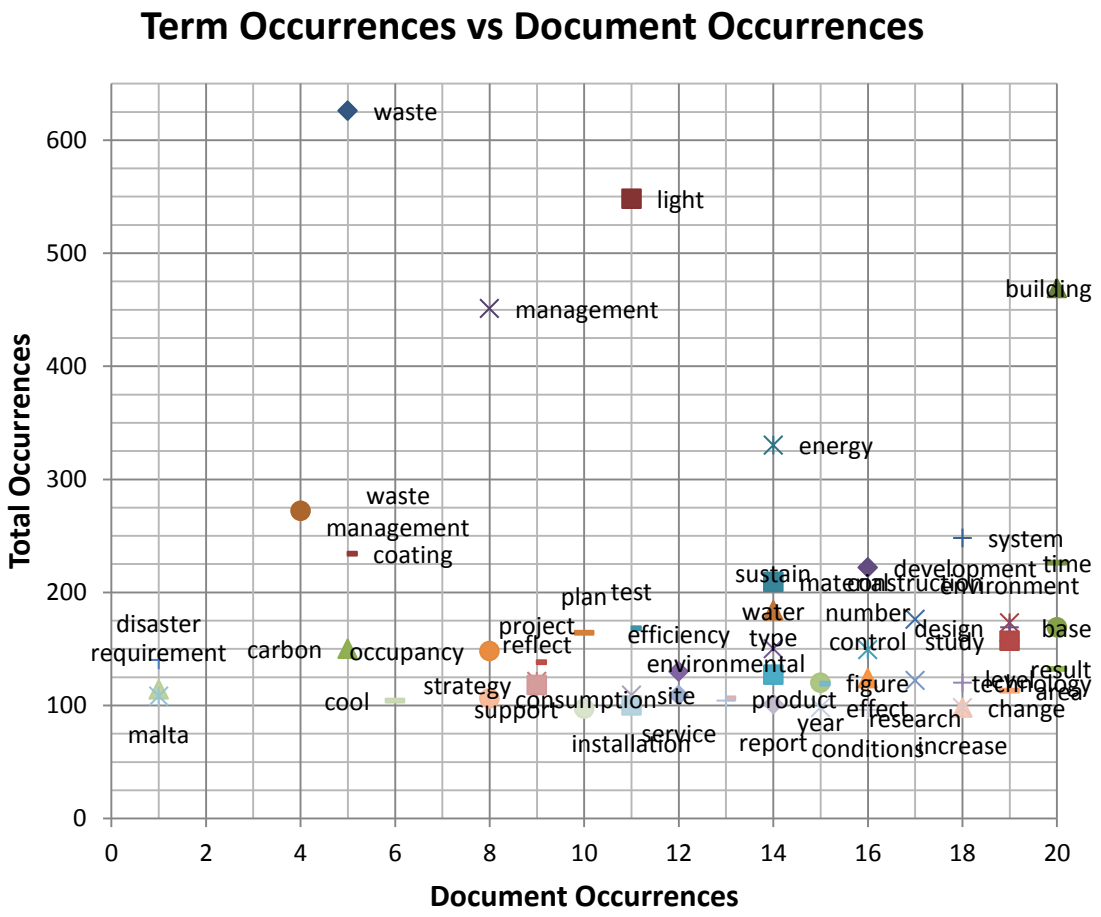


Figure 6.1. Term Dispersion through data samples

Figure 6.2 depicts the best discriminators collected from the data samples, taking into account its *tf-idf* score. For readability purposes, the figure only displays the top 10 best discriminators. As mention previously, terms such as “disaster”, “malta” and “waste” are considered good discriminators, since its *tf-idf* score is above 50% in some data samples, meaning that they are ideal candidates for contextualizing the information in each data sample. The *tf-idf* score indicates how important a term is for a given data sample, as illustrated in Figure 6.2, the term “waste” is relevant for documents 5, 7 and 8, but not so relevant for document 6, since its *tf-idf* score is below 3%.

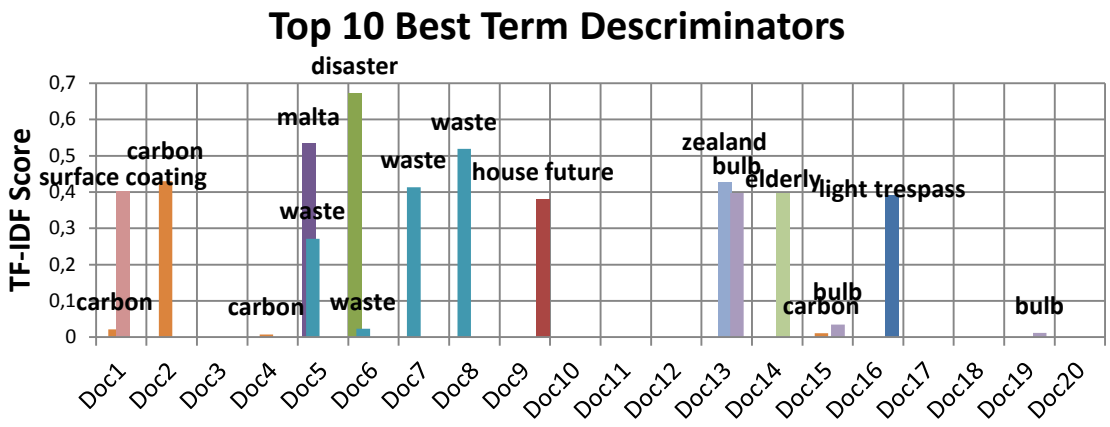


Figure 6.2. Top best term discriminators

As one can imagine and taking into account the dimensionality of each data samples in terms of number of words, also the dimensionality of each statistic vector generated for each data sample is directly proportional to the dimensionality of each data sample itself. For example as depicted in Figure 6.3, Doc5 and Doc14 present high scores in terms of statistic vector dimensionality in accordance with the number of words that represent each data sample. It is worth mentioning that the number of terms in the statistic vector does not match the number of words in each data sample, which is caused by the elimination of stop words and the inclusion of n-grams within each statistic vector.

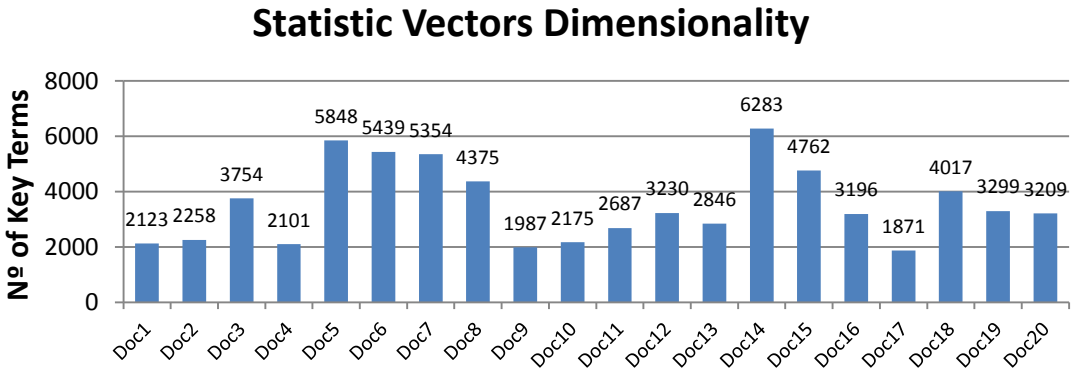


Figure 6.3. Statistic Vectors Dimensionality

The creation of a semantic vector and more specifically the creation of a keyword-based semantic vector, will lead to a significant reduction of the dimensionality of vector. This occurs due to the fact that most of terms within the semantic vector do not have a relation within the domain ontology used. That reduction is mainly influenced by two causes: (i) the adopted ontology was specifically designed to accommodate concepts which are highly related with the B&C, meaning that the relevant terms within the statistical vector which are not highly related with the domain ontology will be discarded; (ii) the adopted ontology suffers from being too

generic with respect to B&C sector, meaning that more specialized vocabulary will also be discarded. Figure 6.4 depicts the dimensionality of the generated keyword-based semantic vector.

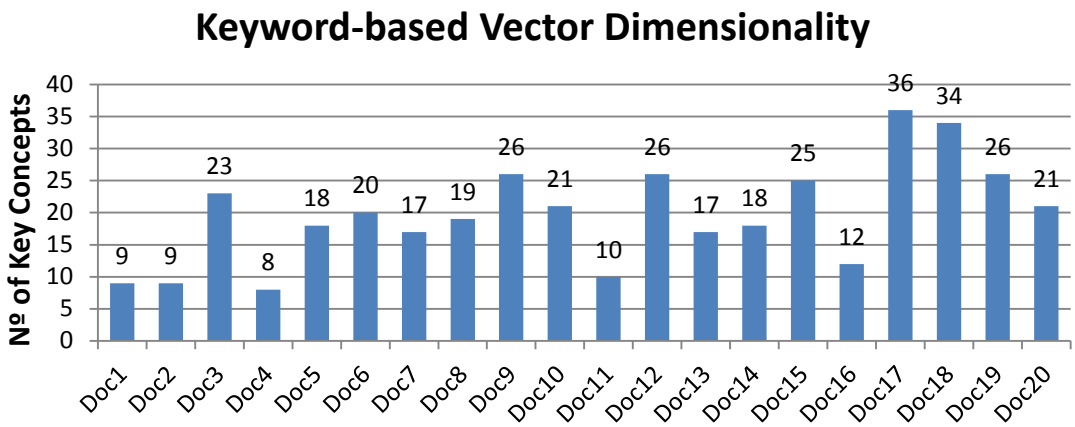


Figure 6.4. Keyword-based Vectors Dimensionality

Figure 6.5 depicts the overall percentage of the information that was kept by each semantic vector. The objective is to present the cases where the ontology lacked to make a relation between relevant terms within the semantic vector and the cases where such loss is not some meaningful. As an example, within Doc4 only 37% of information presented in the statistic vector was covered, in contrast with the 82% from Doc5, where most of the relevant terms from the statistic vector where related with ontology concepts.

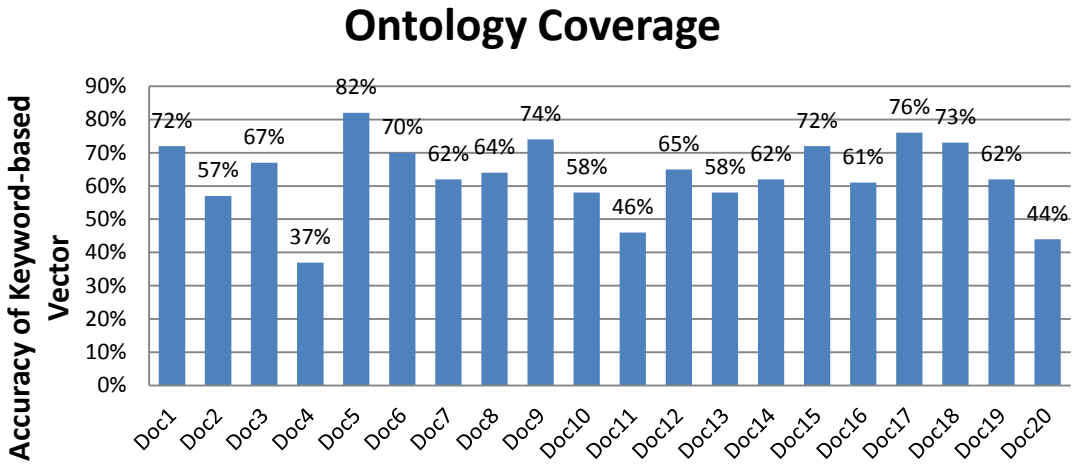


Figure 6.5. Accuracy of Ontology Coverage on data samples

Figure 6.6 shows part of the taxonomy into which the documents were classified. Although the full taxonomy related to products has 16 sub-categories, a smaller subset (5 categories) was selected in order to analyse and explain the results in a clearer fashion. For simplicity, some of

the categories were renamed with shorter labels, e.g. “Covering Cladding and Finish” to “Coating”, “Electric Power and Lighting” to “Lighting”, and “Sanitary Laundry and Cleaning” to “Sanitary”. For convenience, the renamed categories will be used from this point on.

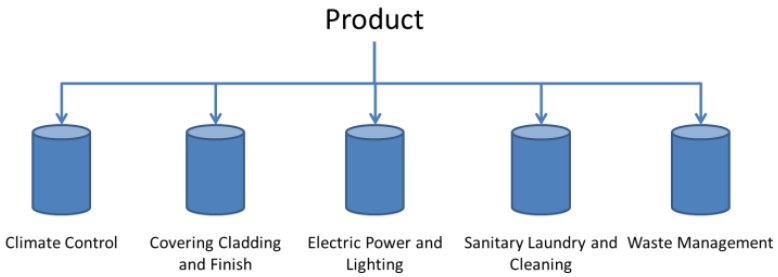


Figure 6.6. Categories used for evaluation

Documents used in the assessment were manually pre-labelled with the support of ICONDA search engine (Figure 6.7). The ICONDA library provides a searching functionality, where users can choose between a free-text based search or use of document metadata (e.g. keywords, abstract, title, publication type, just to name a few).

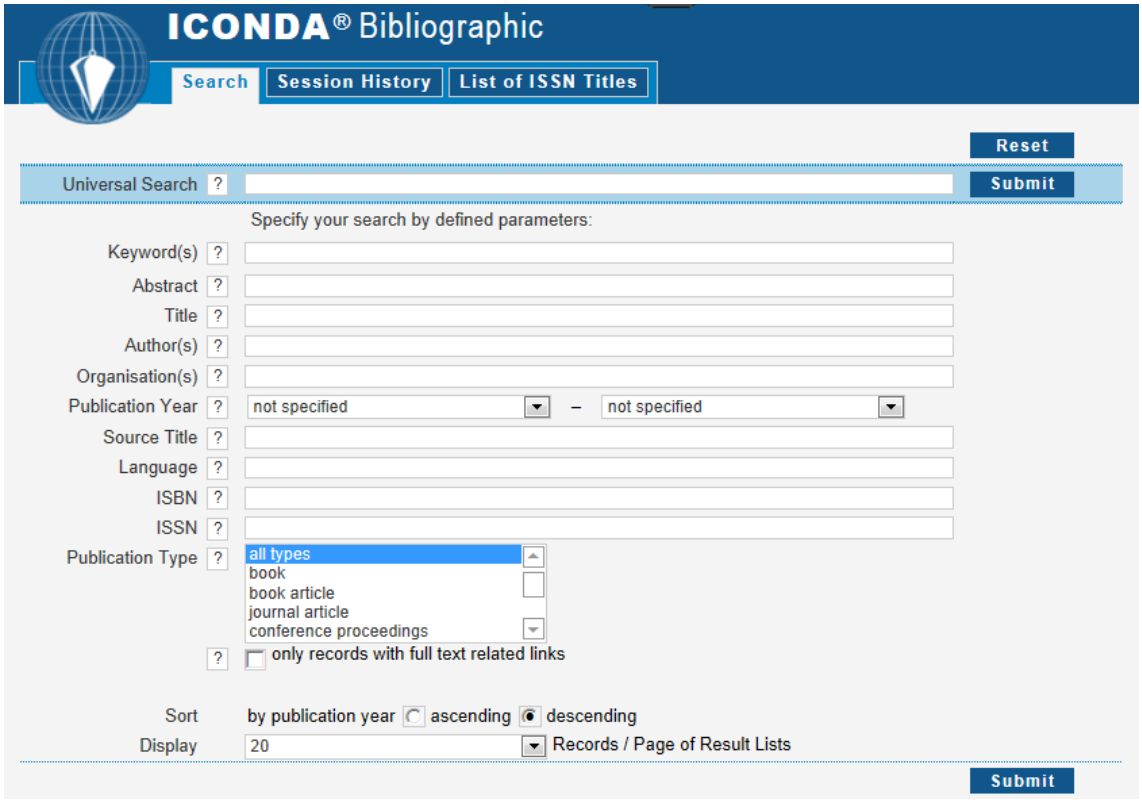


Figure 6.7. ICONDA search engine

Relying on the ICONDA search engine only for the collection process proved not to be a reliable approach. The example depicted in Figure 6.8 describes a document included in the result list provided by the ICONDA search engine, where the term ‘lighting’ was used in the title, abstract

and keywords fields. After a close inspection, it was concluded that the related document had a very small relation with the topic 'Electric Power and Lighting', because ICONDA search engine adopts a purely string match criterion between user query and document terms. Therefore, a close human evaluation was also used in the collection process, to make sure that all documents collected had a sufficiently close relation with the categories of documents used for evaluation. The procedure is that, after each document is retrieved from the ICONDA database, a domain expert was consulted in order to assess the relevance between each document and the categories used, and pre-label such document against one of the 5 categories.

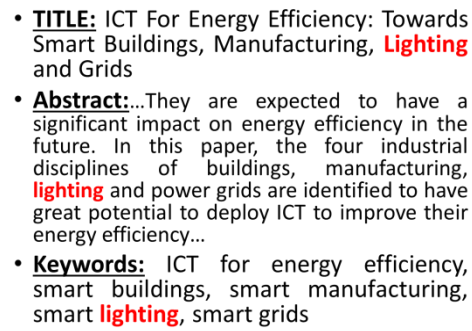

- **TITLE:** ICT For Energy Efficiency: Towards Smart Buildings, Manufacturing, **Lighting** and Grids
- **Abstract:**…They are expected to have a significant impact on energy efficiency in the future. In this paper, the four industrial disciplines of buildings, manufacturing, **lighting** and power grids are identified to have great potential to deploy ICT to improve their energy efficiency…
- **Keywords:** ICT for energy efficiency, smart buildings, smart manufacturing, smart **lighting**, smart grids

Figure 6.8. Pre-labelling mismatch

The SENSE platform was evaluated using 20 scientific publications containing on average 3.500 words each. The reason for choosing scientific publications was the significant number of words in each document, which makes the dispersion of key terms in each document much higher when compared to short webpages or news headlines. It makes the precise classification a greater challenge, not only in terms of document complexity and heterogeneity but also in term of hardware requirements and computation power to process such high volumes of data. Taking as a random example the paper entitled "Evaluation of Deterioration Behavior of Surface Coating for RC Buildings by Permeation-Diffusion", after applying a *tf-idf*, it presented a list of 2123 terms after stemming and n-gram generation (Table 6.2 illustrates a small subset). Such an example shows the complexity and quantity of data to be processed, meaning that any approach proposed must be able to scale up to very large amounts of data. It is worth mentioning that all validation tests were performed on a machine with an Intel quad core 2.4GHz processor, 4GB of RAM and running Windows 7 64-bit OS.

The creation of a *tf-idf* score for each document was performed using RapidMiner libraries, using a TXT format for each document. It took an average of 30min for the entire dataset to be processed, using parallelization. The reason why the TXT format was used instead of the PDF format, is due to the very low resolutions of scanned documents in PDF formats, which resulted in phrases with some blank spaces or non-standard ASCII characters, which sometimes caused the process to halt.

Table 6.2. Representation of "cover cladding and finish" related KS (sample)

Term	Weight	Term	Weight	Term	Weight	Term	Weight
surfac_coat	**0.01569**	*diffus*	**0.00777**	*deterior*	**0.00501**	*coat_materi_build*	**0.00373**
surfac_coat_materi	**0.01456**	*cycl_cycl*	**0.00672**	*diffus_test*	**0.00448**	*permeat_diffus*	**0.00336**
coat	**0.00968**	*carbon*	**0.00570**	*diffus_coeffici*	**0.00402**	*cycl*	**0.00319**
coat_materi	**0.00963**	*surfac*	**0.00562**	*materi_build*	**0.00373**	*permeat*	**0.00315**

6.2 The evaluation process

The core objective of the evaluation is to measure the effectiveness of the altered term vectors. The question we are trying to answer is whether our intuition of adding new terms to a term vector and boosting weights of existing terms does, in practice, meaningfully amplify important terms and weed out less important ones? And at the same time, is it possible to represent knowledge sources with more accuracy with the support of domain ontologies? We believe that having more accurate representations of knowledge sources can improve semantic interoperability among project teams, and consequently facilitate increased knowledge sharing and reuse.

The evaluation process and comparison is performed using the four term vectors, namely statistical, keyword-based, taxonomy-based, and ontology-based.

As mentioned in earlier sections, the focus of this work is not on improving classification algorithms. The evaluation process relies on the altered term vectors as inputs to various classification algorithms - specifically, we used an unsupervised classification algorithm for the evaluations (K-Means clustering). The evaluation process (Figure 6.9) is described in more detail.

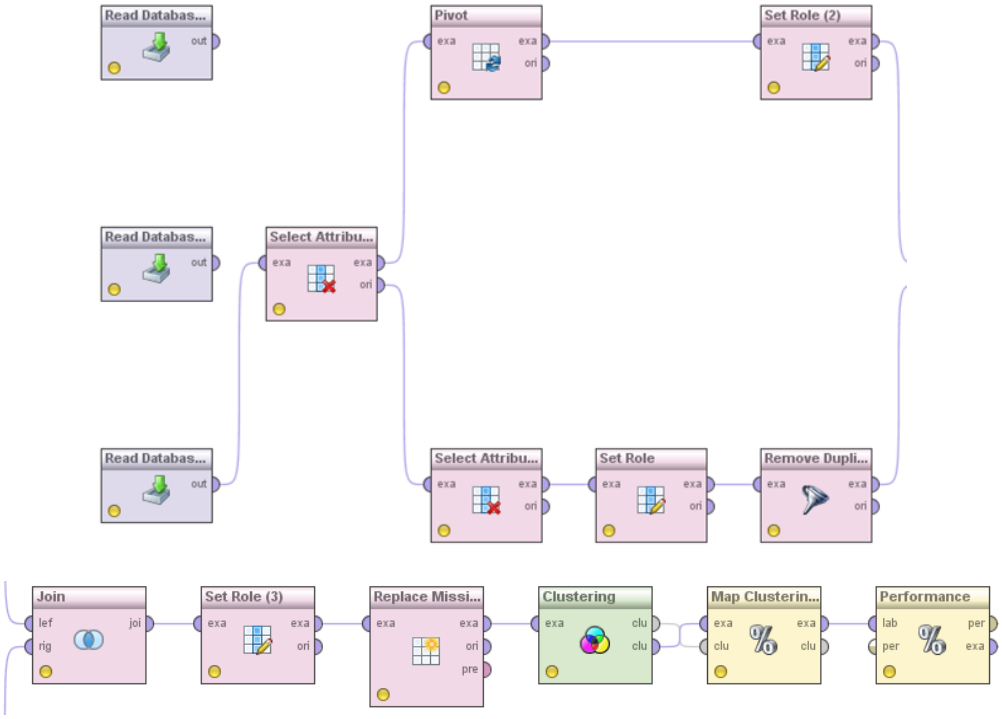
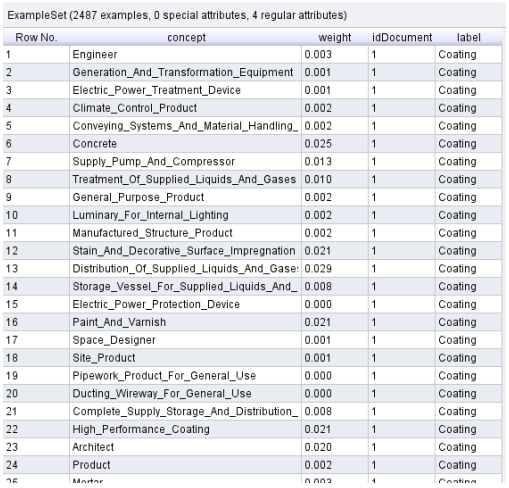


Figure 6.9. Evaluation Process

The unsupervised classification was one of the methods used for evaluation purposes and was modelled and implemented in Rapidminer using several operators. Operators can be described

as building blocks which implement functionalities including data operations, machine learning algorithms and performance measures. Those will be further described further. The “Read Database” operator within the process handles reading the KRs from the knowledge repository; it is used for reading an *ExampleSet*[7] from the specified SQL database. This operator is properly configured in order to extract only the type of representation (statistical, keyword, taxonomy, or ontology-based), which the classification is concerned with. Figure 6.10 illustrates an example of the output generated by the “Read Database” operator related to semantic ontology-based vectors.



ExampleSet (2487 examples, 0 special attributes, 4 regular attributes)

Row No.	concept	weight	idDocument	label
1	Engineer	0.003	1	Coating
2	Generation_And_Transformation_Equipment	0.001	1	Coating
3	Electric_Power_Treatment_Device	0.001	1	Coating
4	Climate_Control_Product	0.002	1	Coating
5	Conveying_Systems_And_Material_Handling_	0.002	1	Coating
6	Concrete	0.025	1	Coating
7	Supply_Pump_And_Compressor	0.013	1	Coating
8	Treatment_Of_Supplied_Liquids_And_Gases	0.010	1	Coating
9	General_Purpose_Product	0.002	1	Coating
10	Luminary_For_Internal_Lighting	0.002	1	Coating
11	Manufactured_Structure_Product	0.002	1	Coating
12	Stain_And_Decorative_Surface_Impregnation	0.021	1	Coating
13	Distribution_Of_Supplied_Liquids_And_Gase	0.029	1	Coating
14	Storage_Vessel_For_Supplied_Liquids_And_	0.008	1	Coating
15	Electric_Power_Protection_Device	0.000	1	Coating
16	Paint_And_Varnish	0.021	1	Coating
17	Space_Designer	0.001	1	Coating
18	Site_Product	0.001	1	Coating
19	Pipework_Product_For_General_Use	0.000	1	Coating
20	Ducting_Wireway_For_General_Use	0.000	1	Coating
21	Complete_Supply_Storage_And_Distribution_	0.008	1	Coating
22	High_Performance_Coating	0.021	1	Coating
23	Architect	0.020	1	Coating
24	Product	0.002	1	Coating
25	Mortar	0.003	1	Coating

Figure 6.10: "Read Database" example

The “Select Attributes” operator selects which attributes of an *ExampleSet* should be kept and which attributes should be discarded. This is used in cases when not all attributes of an *ExampleSet* are required. Often the need arises for selecting attributes before applying some operators. This is especially true for large and complex data sets. Different filter types are provided to assist attribute selection. Only the selected attributes will be delivered from the output port and the rest will be removed from the *ExampleSet*. The objective here is to construct a table in the form of “document id”, “label” and “term-n” where “document id” should

[7] Under the scope of this thesis, ExampleSet refers to a sample of data structured by several data fields(columns)

correspond to the each KS available in the document repository. This operation is performed by “Pivot” and “Set Role” operators.

The “Pivot” operator rotates the given *ExampleSet* by grouping multiple examples of same groups to single examples. The group attribute parameter specifies the grouping attribute (i.e. the attribute which identifies examples belonging to the groups). The resultant *ExampleSet* has ***n*** examples where ***n*** is the number of unique values of the group attribute. The index attribute parameter specifies the attribute whose values are used to identify the examples inside the groups. The values of this attribute are used to name the group attributes, which are created during the pivoting. The resultant *ExampleSet* has ***m*** regular attributes in addition to the group attribute where ***m*** is the number of unique values of the index attribute.

The “Set Role” operator reflects the part played by that attribute in an *ExampleSet*. Changing the role of an attribute may change the part played by that attribute in a process. One attribute can have exactly one role. This operator is used to change the role of one or more attributes of the input *ExampleSet*. Different learning operators require attributes with different roles. This operator is frequently used to set the right roles for attributes before applying the desired operator. The change in role is only for the current process, i.e. the role of the attribute is not changed permanently in the *ExampleSet*. Figure 6.11 illustrates the KR transformation in the form of “document id” and “term-n”.

ExampleSet (20 examples, 1 special attribute, 552 regular attributes) View Filter (20 / 20): all

Row No.	idDocument	weight_Acc...	weight_App...	weight_Arc...	weight_Ceil...	weight_Cli...	weight_Co...	weight_Co...	weight_Con...	weight_Con...	weight_Cov...	weight_Des...	weight_Dist...	weight_Dist...	weight_Duc...	weight_Ele...
1	1	0.002	0.375	0.020	0.025	0.002	0.002	0.008	0.025	0.002	0.033	0.001	0.001	0.029	0.000	0.002
2	2	?	0.220	0.015	0.022	0.006	?	0.004	0.094	0.007	0.149	0.001	?	0.006	?	0.006
3	3	0.001	0.116	?	0.008	0.001	0.001	?	0.002	0.002	0.011	?	?	?	?	0.001
4	4	?	0.172	?	0.011	0.000	?	?	?	0.000	0.014	?	?	?	?	0.000
5	5	?	0	0.057	?	0.005	?	?	?	0.006	0.006	0.003	?	?	?	0.005
6	6	?	0	?	?	0.009	?	?	?	0.010	0.010	0.001	?	?	?	0.008
7	7	?	0	?	?	0.004	?	?	?	0.005	0.005	?	?	?	?	0.004
8	8	?	0	?	?	0.003	?	?	?	0.003	0.003	?	?	?	?	0.002
9	9	?	0	?	?	0.001	?	0.001	?	0.001	0.001	?	?	?	?	0.001
10	10	0.002	0	0.000	?	0.002	0.002	?	0.014	0.002	0.002	0.000	0.003	?	?	0.004
11	11	?	0	?	?	0.001	?	?	?	0.001	0.001	0.001	?	?	?	0.001
12	12	?	0	0.016	?	0.002	?	0.007	?	0.002	0.002	0.002	?	?	?	0.002
13	13	0.028	?	?	0.001	?	0.028	?	?	?	?	?	0.013	?	0.000	0.011
14	14	0.035	?	0.002	0.001	?	0.045	?	?	?	?	0.004	0.030	?	?	0.028
15	15	0.008	0	?	?	0.001	0.008	?	?	0.001	0.001	?	0.116	?	?	0.021
16	16	0.044	?	?	?	?	0.186	?	?	?	?	?	0.019	?	?	0.018
17	17	0.001	0	0.030	?	0.004	0.001	?	?	0.000	0.000	0.006	0.001	?	0.000	0.001
18	18	0.000	0	0.089	?	0.008	0.000	?	?	0.000	0.000	0.013	?	?	0.001	0.000
19	19	0.001	0	?	?	0.005	0.001	?	?	0.000	0.000	0.000	?	?	0.000	0.000
20	20	?	0	?	?	0.010	?	?	?	0.000	0.000	?	?	?	?	0.000

Figure 6.11. ExampleSet transformation

The objective at this stage is to include the label to each row in the example set. This is achieved by using the “Remove Duplicates” operator, which removes duplicate examples from the *ExampleSet* (presented in Figure 6.10) by comparing all examples with each other on the basis of the specified attributes. This operator removes duplicate examples such that only one of all the duplicate examples is kept. Two examples are considered duplicate if the selected attributes have the same values in them. Attributes can be selected from the attribute filter type

parameter and other associated parameters. Figure 6.12 illustrates an example of the “Remove Duplicates” applied to the initial example set.

ExampleSet (20 examples, 1 special attribute, 1 regular attribute)

Row No.	idDocument	label
1	1	Coating
2	2	Coating
3	3	Coating
4	4	Coating
5	5	Waste Mana
6	6	Waste Mana
7	7	Waste Mana
8	8	Waste Mana
9	9	Sanitary
10	10	Sanitary
11	11	Sanitary
12	12	Sanitary
13	13	Lighting
14	14	Lighting
15	15	Lighting
16	16	Lighting
17	17	Climate Con
18	18	Climate Con
19	19	Climate Con
20	20	Climate Con

Figure 6.12. Remove duplicates from example set

The “Join” operator joins the *ExampleSet* illustrated in Figure 6.11 with the *ExampleSet* on Figure 6.12 using one or more attributes of the input *ExampleSet*s as key attributes. Identical values of the key attributes indicate matching examples. The attribute with id role is selected as key by default but an arbitrary set of one or more attributes can be chosen as key. Four types of joins are possible: inner, left, right, and outer join. All these types of joins are explained in the parameters section. Figure 6.13 illustrates the output of the “Join” operator when applied at the two example sets.

ExampleSet (20 examples, 1 special attribute, 553 regular attributes) View Filter (20 / 20): all

Row No.	idDocument	label	weight_Acc...	weight_App...	weight_Arc...	weight_Ceil...	weight_Cli...	weight_Co...	weight_Co...	weight_Con...	weight_Con...	weight_Cov...	weight_Des...	weight_Dist...	weight_Dist...	weight_Duc...
1	1	Coating	0.002	0.375	0.020	0.025	0.002	0.002	0.008	0.025	0.002	0.033	0.001	0.001	0.029	0.000
2	2	Coating	?	0.220	0.015	0.022	0.006	?	0.004	0.094	0.007	0.149	0.001	?	0.006	?
3	3	Coating	0.001	0.116	?	0.008	0.001	0.001	?	0.002	0.002	0.011	?	?	?	?
4	4	Coating	?	0.172	?	0.011	0.000	?	?	?	0.000	0.014	?	?	?	?
5	5	Waste Mana	?	0	0.057	?	0.005	?	?	?	0.006	0.006	0.003	?	?	?
6	6	Waste Mana	?	0	?	?	0.009	?	?	?	0.010	0.010	0.001	?	?	?
7	7	Waste Mana	?	0	?	?	0.004	?	?	?	0.005	0.005	?	?	?	?
8	8	Waste Mana	?	0	?	?	0.003	?	?	?	0.003	0.003	?	?	?	?
9	9	Sanitary	?	0	?	?	0.001	?	0.001	?	0.001	0.001	?	?	?	?
10	10	Sanitary	0.002	0	0.000	?	0.002	0.002	?	0.014	0.002	0.002	0.000	0.003	?	?
11	11	Sanitary	?	0	?	?	0.001	?	?	?	0.001	0.001	0.001	?	?	?
12	12	Sanitary	?	0	0.016	?	0.002	?	0.007	?	0.002	0.002	0.002	?	?	?
13	13	Lighting	0.028	?	?	0.001	?	0.028	?	?	?	?	?	0.013	?	0.000
14	14	Lighting	0.035	?	0.002	0.001	?	0.045	?	?	?	?	0.004	0.030	?	?
15	15	Lighting	0.008	0	?	?	0.001	0.008	?	?	0.001	0.001	?	0.116	?	?
16	16	Lighting	0.044	?	?	?	?	0.186	?	?	?	?	?	0.019	?	?
17	17	Climate Con	0.001	0	0.030	?	0.004	0.001	?	?	0.000	0.000	0.006	0.001	?	0.000
18	18	Climate Con	0.000	0	0.089	?	0.008	0.000	?	?	0.000	0.000	0.013	?	?	0.001
19	19	Climate Con	0.001	0	?	?	0.005	0.001	?	?	0.000	0.000	0.000	?	?	0.000
20	20	Climate Con	?	0	?	?	0.010	?	?	?	0.000	0.000	?	?	?	?

Figure 6.13. Join example

In order to apply the clustering operation, the *ExampleSet* must contain numerical values. For this case there are certain terms/concepts which do not occur in a particular document and, for these situations, all weights must be set to zero. The “Replace Missing Values” operator replaces missing values in examples of selected attributes by a specified replacement. Missing values can be replaced by the minimum, maximum, or average value of that attribute. In this case, Zero will replace missing values. Figure 6.14 illustrates the output of the “Replace Missing Values” operator.

ExampleSet (20 examples, 2 special attributes, 552 regular attributes) View Filter (20 / 20): all

Row No.	idDocument	label	weight_Acc...	weight_App...	weight_Arc...	weight_Ceil...	weight_Cli...	weight_Co...	weight_Co...	weight_Con...	weight_Con...	weight_Cov...	weight_Des...	weight_Dist...	weight_Dist...	weight_Duc...
1	1	Coating	0.002	0.375	0.020	0.025	0.002	0.002	0.008	0.025	0.002	0.033	0.001	0.001	0.029	0.000
2	2	Coating	0	0.220	0.015	0.022	0.006	0	0.004	0.094	0.007	0.149	0.001	0	0.006	0
3	3	Coating	0.001	0.116	0	0.008	0.001	0.001	0	0.002	0.002	0.011	0	0	0	0
4	4	Coating	0	0.172	0	0.011	0.000	0	0	0	0.000	0.014	0	0	0	0
5	5	Waste Mana	0	0	0.057	0	0.005	0	0	0	0.006	0.006	0.003	0	0	0
6	6	Waste Mana	0	0	0	0	0.009	0	0	0	0.010	0.010	0.001	0	0	0
7	7	Waste Mana	0	0	0	0	0.004	0	0	0	0.005	0.005	0	0	0	0
8	8	Waste Mana	0	0	0	0	0.003	0	0	0	0.003	0.003	0	0	0	0
9	9	Sanitary	0	0	0	0	0.001	0	0.001	0	0.001	0.001	0	0	0	0
10	10	Sanitary	0.002	0	0.000	0	0.002	0.002	0	0.014	0.002	0.002	0.000	0.003	0	0
11	11	Sanitary	0	0	0	0	0.001	0	0	0	0.001	0.001	0.001	0	0	0
12	12	Sanitary	0	0	0.016	0	0.002	0	0.007	0	0.002	0.002	0.002	0	0	0
13	13	Lighting	0.028	0	0	0.001	0	0.028	0	0	0	0	0	0.013	0	0.000
14	14	Lighting	0.035	0	0.002	0.001	0	0.045	0	0	0	0	0.004	0.030	0	0
15	15	Lighting	0.008	0	0	0	0.001	0.008	0	0	0.001	0.001	0	0.116	0	0
16	16	Lighting	0.044	0	0	0	0	0.186	0	0	0	0	0	0.019	0	0
17	17	Climate Con	0.001	0	0.030	0	0.004	0.001	0	0	0.000	0.000	0.006	0.001	0	0.000
18	18	Climate Con	0.000	0	0.089	0	0.008	0.000	0	0	0.000	0.000	0.013	0	0	0.001
19	19	Climate Con	0.001	0	0	0	0.005	0.001	0	0	0.000	0.000	0.000	0	0	0.000
20	20	Climate Con	0	0	0	0	0.010	0	0	0	0.000	0.000	0	0	0	0

Figure 6.14. Replace missing values example

The *ExampleSet* is now ready to be clustered. The “Clustering” operator performs clustering using the k-means algorithm. K-Means clustering is an exclusive clustering algorithm i.e. each object is assigned to precisely one of a set of clusters. Objects in one cluster are similar to each other. The similarity between objects is based on a measure of the distance between them. The notion of the centre of a cluster is generally called the centroid. Here the Euclidean distance was used as a measure to define the centroid of a cluster. This is the notional point for which each attribute value is the average of the values of the corresponding attribute for all the points in the cluster. For this example of clustering ***k***=5 was used, where ***k*** represents the number of clusters available in the example set. Figure 6.15 depicts the centroids found for ***k***=5.

Attribute	cluster_0	cluster_1	cluster_2	cluster_3	cluster_4
weight_Accessory_For_Lighting	0	0.029	0	0.001	0.001
weight_Applied_Coating	0	0	0	0	0.220
weight_Architect	0.014	0.000	0.005	0.024	0.009
weight_Ceiling_Covering_Cladding_And_Lining	0	0.000	0	0	0.016
weight_Climate_Control_Product	0.005	0.000	0.001	0.006	0.003
weight_Communication_Lighting_Specialty	0	0.067	0	0.001	0.001
weight_Complete_Supply_Storage_And_Distribution_System	0	0	0.003	0	0.003
weight_Concrete	0	0	0	0.003	0.030
weight_Conveying_Systems_And_Material_Handling_Product	0.006	0.000	0.002	0.001	0.003
weight_Covering_Cladding_And_Finish_Product	0.006	0.000	0.002	0.001	0.052
weight_Design_Actor	0.001	0.001	0.001	0.004	0.000
weight_Distribution_Device	0	0.045	0	0.001	0.000
weight_Distribution_Of_Supplied_Liquids_And_Gases	0	0	0	0	0.009
weight_Ducting_Wireway_For_General_Use	0	0.000	0	0.000	0.000
weight_Electric_Power_And_Lighting_Product	0.005	0.019	0.001	0.001	0.002
weight_Electric_Power_Protection_Device	0	0.013	0	0.001	0.000
weight_Electric_Power_Treatment_Device	0	0.015	0	0.001	0.000
weight_Engineer	0.002	0.003	0.001	0.009	0.002
weight_Equipment_And_Furnishings	0.007	0.000	0.002	0.001	0.003
weight_Floor_Covering	0	0	0	0	0.022
weight_General_Pipework_And_Ductwork_Product	0	0.000	0	0.000	0.000
weight_General_Purpose_Construction_Accessories_And_Surfacing_Product	0.006	0.000	0.001	0.001	0.003
weight_General_Purpose_Product	0.005	0.000	0.001	0.001	0.003
weight_Generation_And_Transformation_Equipment	0	0.019	0	0.001	0.000
weight_Grout	0	0	0	0.002	0.004
weight_Heater_For_Supplied_Liquids	0	0	0	0	0.002

Figure 6.15. Centroid clusters

After having the clusters, the next step deals with mapping each identified cluster to the initial proposed labels. Such mapping is performed by the “Map Clustering on Labels” operator, which expects a clustered *ExampleSet* and a cluster model as input. Using these inputs, the operator estimates a mapping between the given clustering and prediction. It adjusts the given clusters with the given labels and so estimates the best fitting pairs. The resultant *ExampleSet* has a

prediction attribute which is derived from the cluster attribute. Figure 6.16 illustrates the mapping procedure between clusters and initial cluster labels.

ExampleSet (20 examples, 9 special attributes, 552 regular attributes) View Filter (20 / 20): all

Row No.	idDocument	label	cluster	prediction(label)	confidence(Coating)	confidence(Waste Management)	confidence(Sanitary)	confidence(Lighting)	confidence(Climate Control)
1	1	Coating	cluster_4	Coating	1	0	0	0	0
2	2	Coating	cluster_4	Coating	1	0	0	0	0
3	3	Coating	cluster_4	Coating	1	0	0	0	0
4	4	Coating	cluster_4	Coating	1	0	0	0	0
5	5	Waste Management	cluster_0	Waste Management	0	1	0	0	0
6	6	Waste Management	cluster_0	Waste Management	0	1	0	0	0
7	7	Waste Management	cluster_0	Waste Management	0	1	0	0	0
8	8	Waste Management	cluster_0	Waste Management	0	1	0	0	0
9	9	Sanitary	cluster_2	Sanitary	0	0	1	0	0
10	10	Sanitary	cluster_3	Climate Control	0	0	0	0	1
11	11	Sanitary	cluster_2	Sanitary	0	0	1	0	0
12	12	Sanitary	cluster_2	Sanitary	0	0	1	0	0
13	13	Lighting	cluster_1	Lighting	0	0	0	1	0
14	14	Lighting	cluster_1	Lighting	0	0	0	1	0
15	15	Lighting	cluster_1	Lighting	0	0	0	1	0
16	16	Lighting	cluster_1	Lighting	0	0	0	1	0
17	17	Climate Control	cluster_3	Climate Control	0	0	0	0	1
18	18	Climate Control	cluster_3	Climate Control	0	0	0	0	1
19	19	Climate Control	cluster_3	Climate Control	0	0	0	0	1
20	20	Climate Control	cluster_3	Climate Control	0	0	0	0	1

Figure 6.16. Map Clustering on Labels example

Prior to analysing the performance of the clustering method and ultimately to analyse where semantic enrichment was achieved, there is a need to determine first if the number of clusters (K) initially set to 5 is the correct number of clusters in the example set. The centroid based clustering operators like the K-Means produce a centroid cluster model and a clustered set, as previously explained. The centroid cluster model has information regarding the clustering performed. It tells which examples are parts of which cluster. It also has information regarding centroids of each cluster. The “Cluster Distance Performance” operator takes this centroid cluster model and clustered set as input and evaluates the performance of the model based on the cluster centroids. The performance measure supported is the “average within cluster distance”, which is calculated by averaging the distance between the centroid and all examples of a cluster.

Nevertheless, it is worth mentioning that determining the number of clusters in a data set, a quantity often labelled *k* as in the k-means algorithm, is a frequent problem in data clustering, and it is a distinct issue from the process of actually solving the clustering problem.

For a certain class of clustering algorithms (in particular k-means algorithm), there is a parameter commonly referred to as *k* that specifies the number of clusters to detect. The correct choice of *k* is often ambiguous, with interpretations depending on the shape and scale of the distribution of points in a data set and the desired clustering resolution of the user. In addition, increasing *k* without penalty will always reduce the amount of error in the resulting clustering, to the extreme case of zero error if each data point is considered its own cluster (i.e., when *k* equals the number of data points, *n*). Intuitively then, the optimal choice of *k* will strike a balance between maximum compression of the data using a single cluster, and maximum accuracy by assigning each data point to its own cluster. If an appropriate value of *k* is not apparent from prior knowledge of the properties of the data set, it must be chosen somehow. There are several categories of methods for making this decision.

The method adopted here was to look at the percentage of variance explained as a function of the number of clusters. The number of clusters should be chosen in a way that adding another cluster does not give much better modelling of the data. More precisely, if plotting the percentage of variance explained by the clusters against the number of clusters, the first clusters will add much information (explain a lot of variance), but at some point the marginal gain will drop, giving an angle in the graph. The number of clusters is chosen at this point, hence the "elbow criterion" (Ketchen e Shook 1996). Figure 6.17 illustrates the average centroid distances for each cluster against several K values. For K=5 the variance of the average centroids distances tends to decrease.

Figure 6.17. Elbow criterion for determining *K*

The results regarding performance of the clustering will be presented in section 6.4 and analysed in section 6.5.

6.3 Challenges and corrective measures

As described previously, one of the objectives of the ontology evolution stage within the knowledge representation enrichment process is to achieve an efficient domain knowledge representation through a domain ontology that could express the semantics behind the knowledge available in the document corpus, not only in terms of domain concepts but also in terms of concept relations. In other words, in order to produce meaningful results, the domain ontology has to be previously "tuned", in order to address each knowledge representation.

For illustrative purposes, let's consider an example where a set of KRs belonging to "Covering Cladding and Finish Product" category as expressed in Figure 6.6, and the initial domain ontology. Figure 6.18 illustrates that most of the relevant terms presented in the knowledge representation were not expressed within the ontology through equivalent terms. Therefore, new equivalent terms must be added into the domain ontology, in order to guarantee a meaningful characterization of the category to be processed.

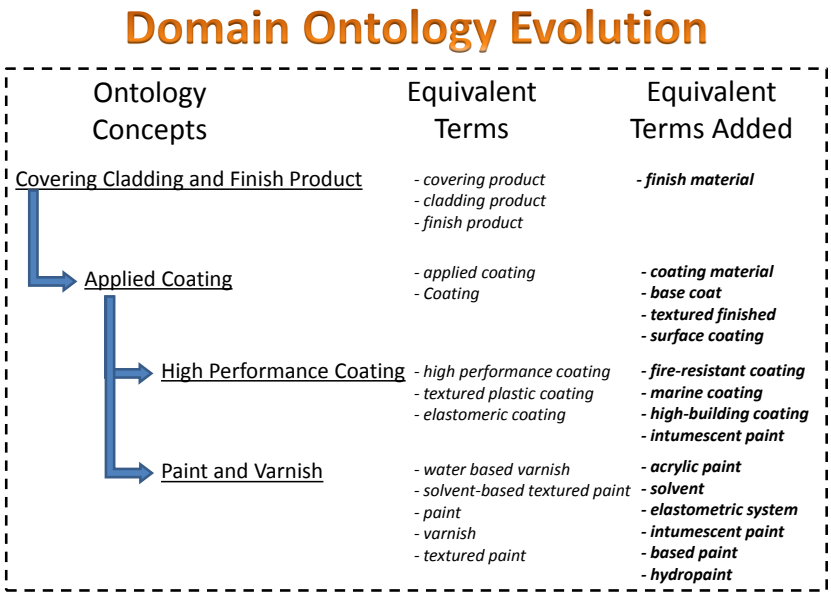


Figure 6.18. Ontology evolution example

Another characteristic to take into account, when evaluating the approach prosed by this thesis, is related to the type of classification to be applied. Most IR classification approaches rely on a supervised classification method, but supervised classification is inherently limited by the information that can be inferred from the training data, as discussed in chapter 1. In other words, the accuracy and the representativeness of the training data, and also the distinctiveness of the classes must be taken into account. This tends to be a problem when dealing with a large document corpora, when no previous in-depth knowledge about the documents is assumed.

Some documents tend to overlap even when belonging to different categories. Such situations are quite common when working with documents with an average of 3.500 words each. In general, text classification is a multi-class problem (more than 2 categories). Training supervised text classifiers requires large amounts of labelled data whose annotation can be expensive. A common drawback of many supervised learning algorithms is that they assume binary classification tasks and thus require the use of sub-optimal (and often computationally expensive) approaches such as one vs. rest to solve multi-class problems, let alone structured domains such as strings and trees.

Documents used in the assessment were manually pre-labelled with the support of ICONDA search engine and human expert evaluation, which sometimes helped in resolving some inconsistencies. For example looking into Figure 6.19, ICONDA search engine considered such document into some extend related with 'lighting' concept, but after close inspection such document was pre-labelled as 'climate control'.

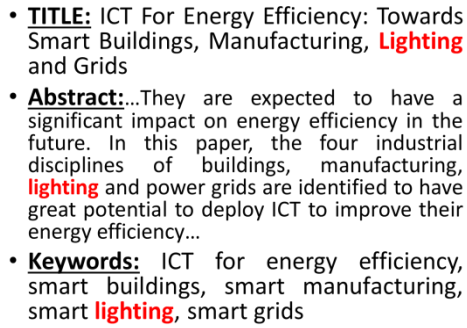


Figure 6.19. pre-labelling using ICONDA search engine

Labelling such documents manually beforehand is not a trivial task and may affect adversely the training set of the classification algorithm. Our intention is to reduce as far as possible human intervention in the classification task and also to scale up our approach to a large set of scientific publications.

The goal of the assessment is to evaluate if the semantic enrichment process improves the similarity level among documents, even when such documents were not considered similar using purely statistical approaches but, indeed, they are in fact similar from a semantic perspective.

Another very important issue is the management of stemmed words. As mentioned before, stemming refers to the process of reducing inflected (or sometimes derived) words to their stem, base, or root form. Although it is a very widely used approach in IR literature, it has some drawbacks when it comes to reverse the stemmed word into the original form. This operation has to be taken into account when matching ontology equivalent terms with relevant terms available in statistical representations. The approach used here to overcome this situation is to

use a cosine similarity between terms in the statistical representation and the equivalent terms for each ontology concept.

6.4 Results

This section presents the results regarding the application of the clustering algorithm into the knowledge representations of the data set. The metrics used for evaluation are the traditional notions of precision and recall, and are computed as follows:

$$\text{Precision} = \frac{\text{nº of documents correctly assigned to the category}}{\text{nº of documents correctly assigned to the category} + \text{nº of documents incorrectly assigned to the category}}$$

$$\text{Recall} = \frac{\text{nº of documents correctly assigned to the category}}{\text{nº of documents correctly assigned to the category} + \text{nº of documents incorrectly rejected from the category}}$$

$$\text{Accuracy} = \frac{\text{nº of documents correctly assigned to the category} + \text{nº of documents correctly rejected from category}}{\text{n}}$$

where ***n*** = nº of documents correctly assigned to the category + nº of documents incorrectly assigned to the category + nº of documents incorrectly rejected from the category + nº of documents correctly rejected from the category.

Although the classical IR metrics have been used for evaluating the classification process, it is worth mentioning that the "correctness" of the classification tends to be a subjective issue. What is a satisfactory classification for an application setting that has weighted ontological semantic relationships a certain way might be unacceptable in other classification settings. The importance of relationships between ontological concepts is therefore an additional independent and tuneable component that affects the precision and recall metrics.

It is presented the overall statistics of the clustering algorithm and after, the analysis regarding success and failure patterns observed during correlation with the results of the classification. Tables 2 to 9, show average recall and precision values for 5 product categories comparing all four vectors. The tables presented here are in the form of "predicted category" vs "true category, where the "predicted category" corresponds to the number of knowledge sources that were predicted by the clustering algorithm as being part of a certain category. The "true category" corresponds to the real number of knowledge sources that are part of a certain category.

Table 6.3 illustrates the results of the clustering algorithm, when applied to the knowledge representations in the form of statistical vectors.

Table 6.3. Performance using Statistical-based Vector

Accurracy:40%						
	True *Coating*	**True** *Waste Management*	**True** *Sanitary*	**True** *Lighting*	**True** *Climate Control*	**Class Precision**
Predicted *Coating*	2	0	0	0	0	**100%**
Predicted *Waste Management*	1	4	3	3	4	**26,67%**
Predicted *Sanitary*	0	0	1	0	0	**100%**
Predicted *Lighting*	0	0	0	1	0	**100%**
Predicted *Climate Control*	1	0	0	0	0	**0%**
Class Recall	**50%**	**100%**	**25%**	**25%**	**0%**	

Table 6.4. Statistical cluster centroids

Waste Management	**Climate Control**	**Lighting**	**Sanitary**	**Coating**
waste	reflective	light trespass	light weight	carbon
waste	coating	trespass	board	surface coating
management	gray	advertisement	light weight	coating
management	coating	light	partition	surface coating
disaster	solvent type	illumination	weight	material
malta	coating solvent	window	partition	coating material
bulb	coating solvent	illumination	partition	diffusion
zealand	type	signboard	bolt	finishing
fixture	gray reflective	advertisement	column	finishing material
light	gray reflective	light	weight	deteoration
occupancy	coating	lamp	sanitary	concrete
cool	solvent	window	equipment	carbon
	polyurethane	roadway	wall	resistance
	solar reflection		hollow	

Table 6.5 illustrates the results of the clustering algorithm, when applied into the knowledge representations in the form of semantic vectors keyword-based.

Table 6.5. Performance using Keyword-based Vector

Accurracy:85%						
	True *Coating*	**True** *Waste Management*	**True** *Sanitary*	**True** *Lighting*	**True** *Climate Control*	**Class Precision**
Predicted *Coating*	4	0	0	0	0	**100%**
Predicted *Waste Management*	0	4	0	0	0	**100%**
Predicted *Sanitary*	0	0	2	0	0	**100%**
Predicted *Lighting*	0	0	0	3	0	**100%**
Predicted *Climate Control*	0	0	2	1	4	**57,14%**
Class Recall	**100%**	**100%**	**50%**	**75%**	**100%**	

Table 6.6. keyword-based cluster centroids

Waste Management	**Climate Control**	**Lighting**	**Sanitary**	**Coating**
Waste management product Contractor Recycling phase Agenda Project Territory Solid waste handling product Committee Board	Presence detection and registration HVAC distribution device Complete cooling system Cooling and freeze plant Structural frame Monitoring and control equipment	Lighting Lamp Communication lighting specialty Luminary for internal lighting Residence Owner Roadway and runway Window	Complete sanitary suite Low-rise linear building Buyer Sanitary disposal unit Sanitary laundry and cleaning equipment Project Electrical energy recording device	Applied coating Paint and varnish Paint for particular applications Supply and distribution of liquids and gases product Covering cladding and finish product Concrete Chemical fo construction

Table 6.7 illustrates the results of the clustering algorithm, when applied into the knowledge representations in the form of semantic vectors taxonomy-based.

Table 6.7. Performance using Taxonomy-based Vector

Accurracy:90%						
	True *Coating*	**True** *Waste Management*	**True** *Sanitary*	**True** *Lighting*	**True** *Climate Control*	**Class Precision**
Predicted *Coating*	4	0	0	0	0	**100%**
Predicted *Waste Management*	0	4	0	0	0	**100%**
Predicted *Sanitary*	0	0	2	0	0	**100%**
Predicted *Lighting*	0	0	0	4	0	**100%**
Predicted *Climate Control*	0	0	2	0	4	**66,67%**
Class Recall	**100%**	**100%**	**50%**	**100%**	**100%**	

Table 6.8. taxonomy-based cluster centroids

Waste Management	**Climate Control**	**Lighting**	**Sanitary**	**Coating**
Waste management product Contractor Recycling phase Solid waste handling product Agenda Liquid waste handling service Project Gaseous waste handling service	Complete cooling system HVAC distribution device Complete mechanical ventilation system Presence detection and registration Cooling and freezing plant Monitoring and control equipment	Lighting Lamp Communication lighting specialty Luminary for internal lighting Distribution device Luminary for external lighting Residence Owner Manufacturer	Complete sanitary suite Low-rise linear building Sanitary disposal unit Buyer Sanitary laundry and cleaning equipment Sanitary equipment Project Plumbing fixture and sanitary washing unit	Applied coating Paint and varnish Paint for particular applications Covering cladding and finish product Supply and distribution of liquids and gases product Concrete Chemical for construction

Table 6.9 illustrates the results of the clustering algorithm, when applied into the knowledge representations in the form of semantic vectors taxonomy-based.

Table 6.9. Performance using Ontology-based Vector

Accurracy:95%						
	True *Coating*	**True** *Waste Management*	**True** *Sanitary*	**True** *Lighting*	**True** *Climate Control*	**Class Precision**
Predicted *Coating*	4	0	0	0	0	**100%**
Predicted *Waste Management*	0	4	0	0	0	**100%**
Predicted *Sanitary*	0	0	3	0	0	**100%**
Predicted *Lighting*	0	0	0	4	0	**100%**
Predicted *Climate Control*	0	0	1	0	4	**80%**
Class Recall	**100%**	**100%**	**75%**	**100%**	**100%**	

Table 6.10. ontology-based cluster centroids

Waste Management	**Climate Control**	**Lighting**	**Sanitary**	**Coating**
Waste management product Contractor Recycling phase Solid waste handling product Agenda Liquid waste handling service Project Gaseous waste handling service	Complete cooling system HVAC distribution device Complete mechanical ventilation system Presence detection and registration Cooling and freezing plant Monitoring and control equipment	Lighting Lamp Communication lighting specialty Luminary for internal lighting Distribution device Luminary for external lighting Residence Owner Manufacturer	Complete sanitary suite Sanitary disposal unit Sanitary laundry and cleaning equipment Low-rise linear building Buyer Plumbing fixture and sanitary washing unit Team Sanitary equipment	Applied coating Paint and varnish Paint for particular applications Covering cladding and finish product Supply and distribution of liquids and gases product Concrete Chemical for construction

6.5 Analysis

The dataset used for evaluation (intentionally) considered a few categories that had minor characteristic differences. Such categories have many similar predictor variables or terms that make classifying and allocating documents to the categories a challenge. Statistical term vectors that rely solely on document contents can rarely reliably classify a document as falling into one category or another, as illustrated in tTable 6.3. The predicted category "Waste Management" although achieving 100% in terms of recall, the precision of this predicted cluster is very low (26,67%) when compared with other clusters. This can be explained by the cluster centroids, where the centroid related to "Waste Management" category is much broader in terms of the nature of terms, when compared to other clusters where terms in the centroid are much more specific in their nature. One explanation for such behaviour is related to the magnitude of knowledge representations. As explained previously, the statistical vector was pruned where weights of relevant terms were below 0,0001, due to computational limitations. If not considering any kind of pruning, the probability of getting better results is higher. Another explanation is related with the initial set of parameters assigned to the clustering algorithm. Parameters such as the maximal number of runs of K-Means with random initialization that are performed and also the maximal number of iterations performed for one run of K-Means, influence the overall accuracy of the algorithm. Nevertheless, the initial set of parameters is not changed during the semantic enrichment process. As already explained, the objective here is not to outperform existing clustering algorithms, but rather to identify where semantic enrichment was measurably achieved.

Looking closely at some categories in order to understand the above results better, it is possible to discover some interesting patterns when the use of this approach added value and also patterns when it did not. Considering the 'Sanitary Laundry and Cleaning' category, it can be concluded that using this approach there was a substantial improvement in terms of recall metric, from 25% using the statistical-based approach to 75% using the Ontology-based approach. In this case, the usage of ontological relations present in the domain Ontology (as shown in Table 6.9), improved the recall metric reliability from 50% to 75%.

This evaluation also indicated that quite a few documents had minimal or no direct matching with Ontology equivalent term instances, mostly because of an incomplete domain ontology model (further investment in extending the Ontology knowledge base can address this issue to some extent) and the lack of a robust method for removing word ambiguity during the matching process (as previously explained in section 6.1). It is quite possible for a domain Ontology to have no influence on the classification. Therefore the goal is to achieve no worse a result than the statistical-based approach whether the Ontology is relevant or wholly irrelevant.

Adding to the evaluation on the clustering results as a means to identify where semantic enrichment was performed, another approach for assessment was conducted in order to guarantee the efficient of the proposed approach against user queries.

SENSE user interface enables users to query the knowledge repository and using the 4 different types of knowledge representations (statistic and semantic keyword-based, taxonomy-based and ontology based).

In order to show some examples of the behaviour of the proposed approach (detailed in chapter 4), two different types of queries were conducted. The first example shows the behaviour using the term “bulb”. The second example shows the behaviour using the term “cleaning product”.

Figure 6.20 shows the example of a statistic search using the query term “bulb”. The result only retrieved 3 knowledge sources, meaning that, only those 3 knowledge sources contained explicitly the word “bulb”. The result is displayed by order of relevance, meaning that document ID 13 is the most relevant for this particular query.

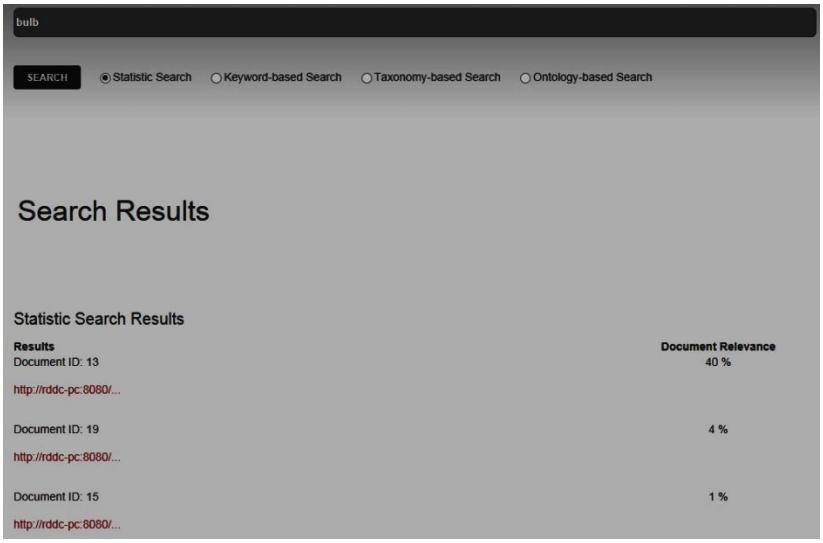


Figure 6.20. "bulb" statistic search

Figure 6.21, shows the example of a keyword-based search for the same term above. The result indicates that more knowledge sources were included in the result, but also the relevance of each particular knowledge source has changed. It is important to mention here that the term “bulb” corresponds to an equivalent term of the ontology concept “Lamp”. The reason why the relevance of document ID 13 has increased is due to the fact that such document contains other words which are considered equivalent terms of the concept “Lamp”. Another important aspect to note is the appearance of document ID 16 in the results list. The document does not contain the word “bulb” at all explicitly, but on the other hand, other words (ex. fluorescent lamp and incandescent lamp) do occur in the document, which are equivalent terms of the same ontology concept “Lamp”.

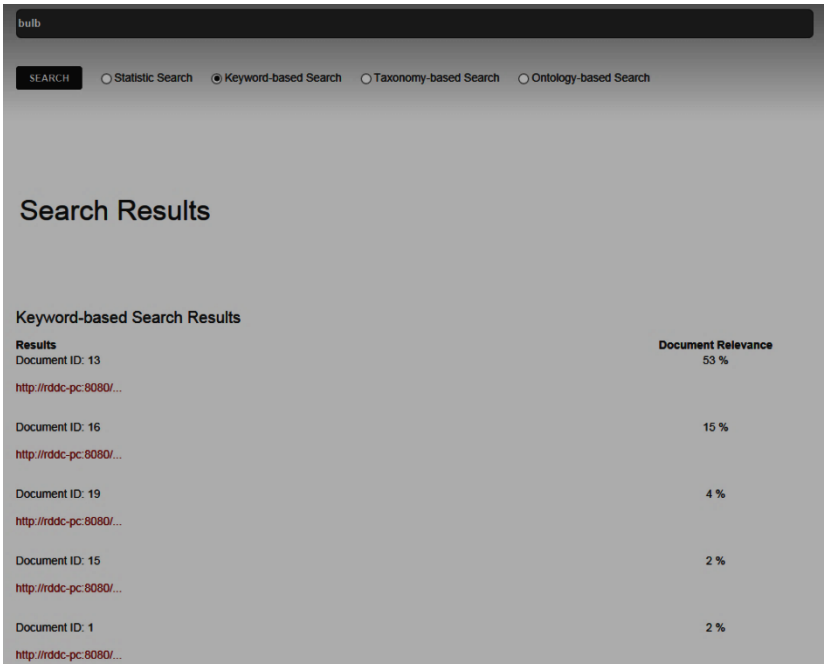


Figure 6.21. "bulb" keyword-based search

Figure 6.22 shows the example of a taxonomy-based search for the same term “bulb”. As occurred previously in the above example, so also here; the result indicates that more knowledge sources were included in the result, but also the relevance of each particular knowledge source has changed. Not the inclusion of document ID 14, due to the fact that its KR includes relevant ontological concepts (such as “Lighting”, “Luminary for Internal Lighting”, and “Communication Lighting Specialty”) which are taxonomically related to the concept “Lamp”. The relevance of other knowledge sources has also changed accordingly, taking into account the number and weight of taxonomically related concepts available in each KR.

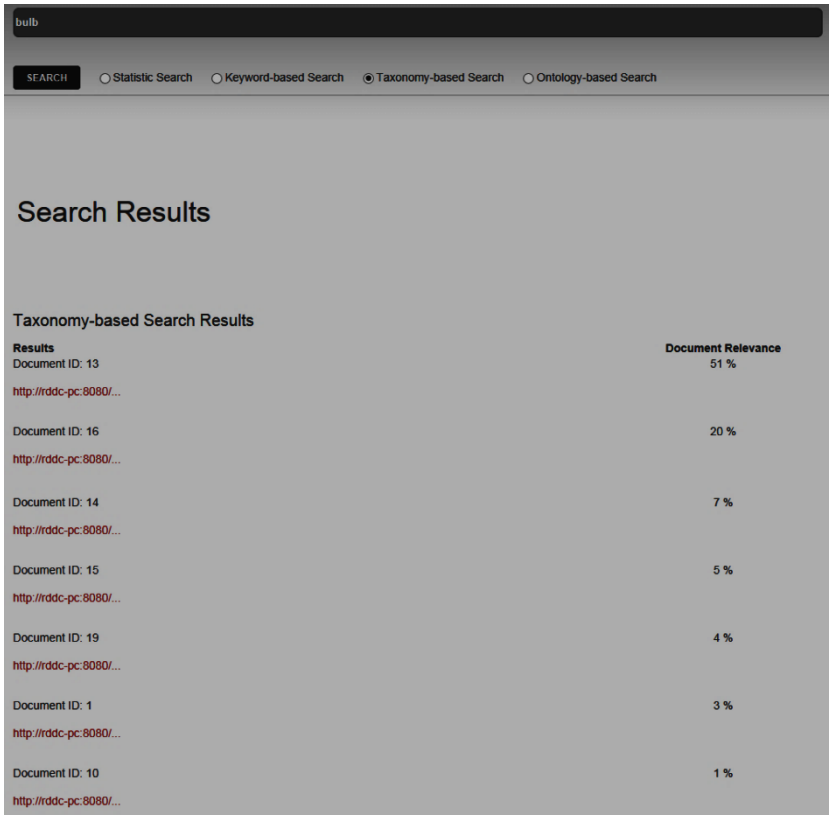


Figure 6.22. "bulb" taxonomy-based search

Figure 6.23 shows another example, where the user is interested in searching for knowledge sources related to the term “cleaning product”. Since the term does not occur in any of the knowledge sources available in the knowledge repository, no results are retrieved using the statistic search.

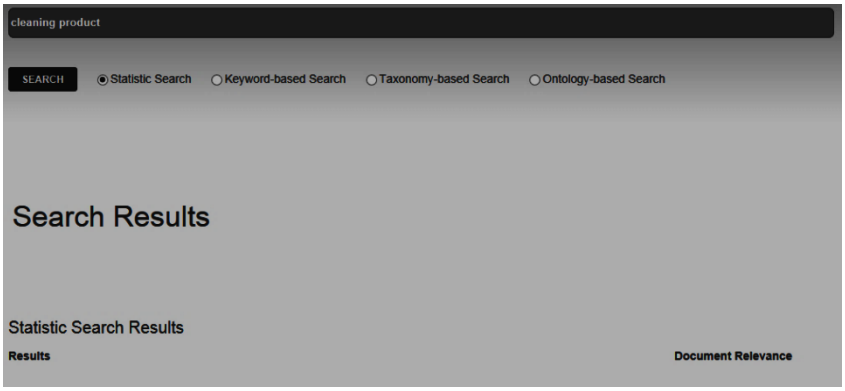


Figure 6.23. "cleaning product" statistic search

However, some results are found using a keyword-based search as illustrated in Figure 6.24, mainly due to the fact that “cleaning product” is an equivalent term related to the ontology concept “Sanitary Laundry and Cleaning Equipment”. This means that the knowledge sources

which were retrieved do not explicitly contain the term “cleaning equipment”, but other equivalent terms related to the same ontological concept.

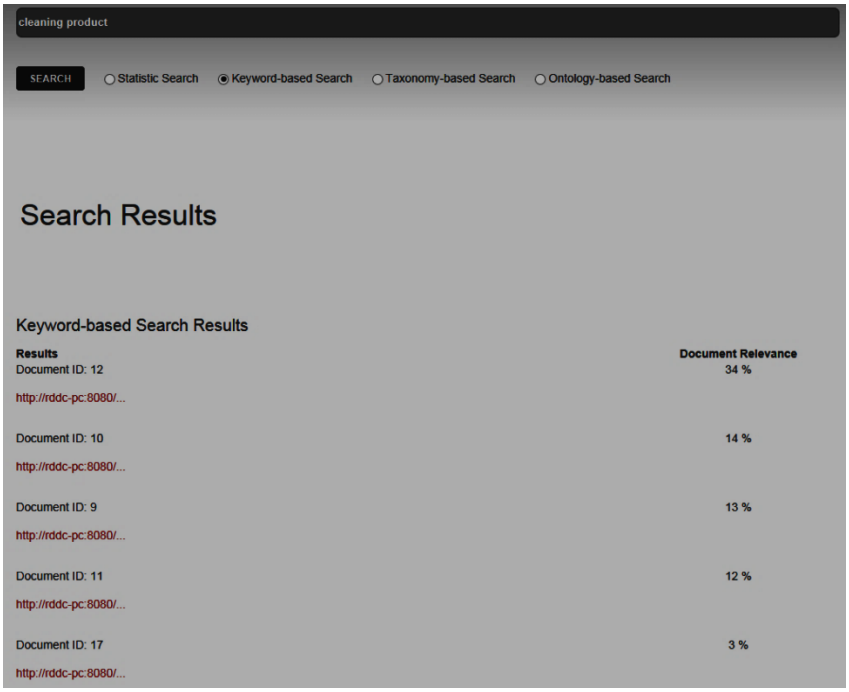


Figure 6.24. "cleaning product" keyword-based search

Figure 6.25 illustrates the result for the same query, but using the taxonomy-based search. Here it is possible to identify that the relevance of the knowledge sources did not suffer from major modifications, mainly due to the fact that taxonomic representations of such knowledge sources have not been significantly enriched by the taxonomic relations. But additional documents have been included, due to the fact there was a significant semantic enrichment by the inclusion of the concept “Sanitary Laundry and Cleaning Equipment” in such representations.

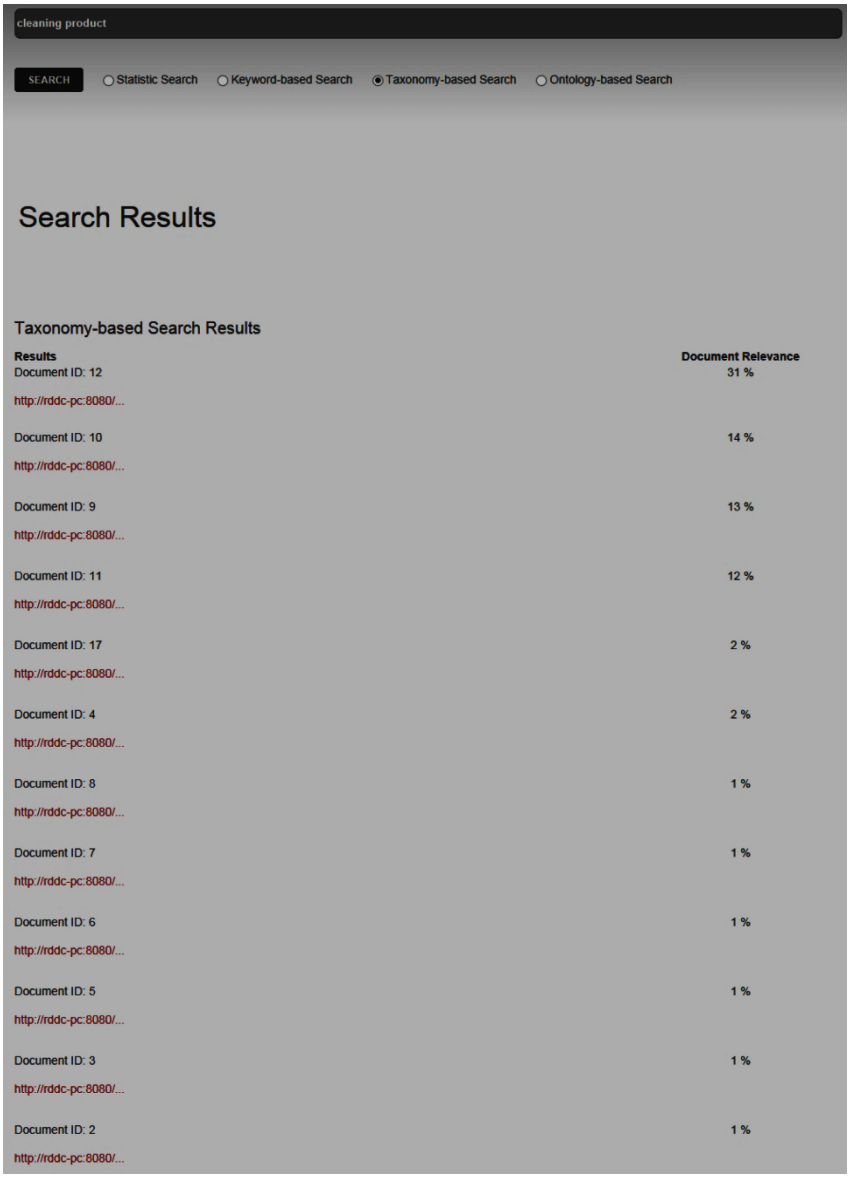


Figure 6.25. "cleaning product" taxonomy-based search

Figure 6.26 illustrates the same example, but this time using the ontology-based search. It is possible to identify that the relevance of some knowledge sources have changed. Such result was due to the fact that ontology concepts available in the knowledge representations will be affected by the boost effect, where strongly related concepts will have their weights increased. As an example, the concepts “Complete Sanitary Suite” and “Sanitary Laundry and Cleaning Equipment Product” are ontologically related.

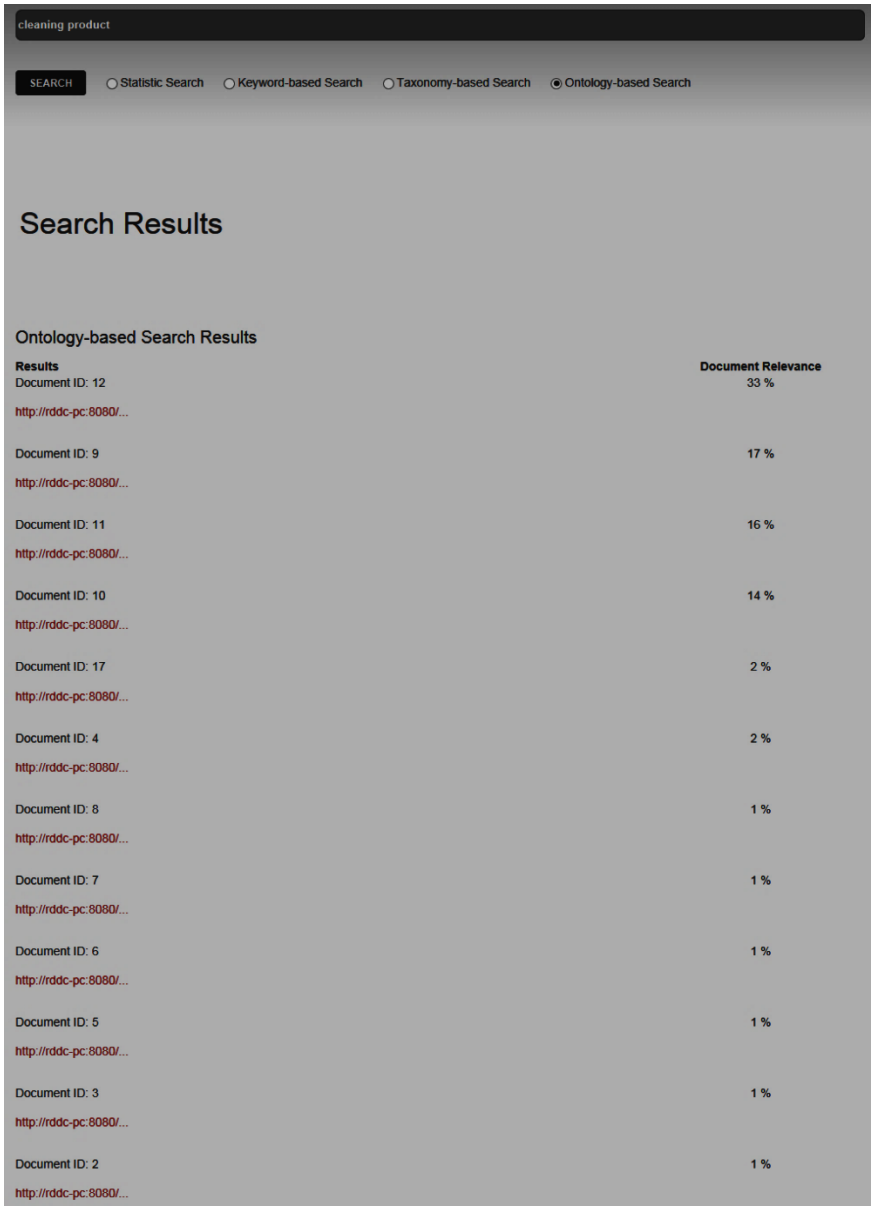


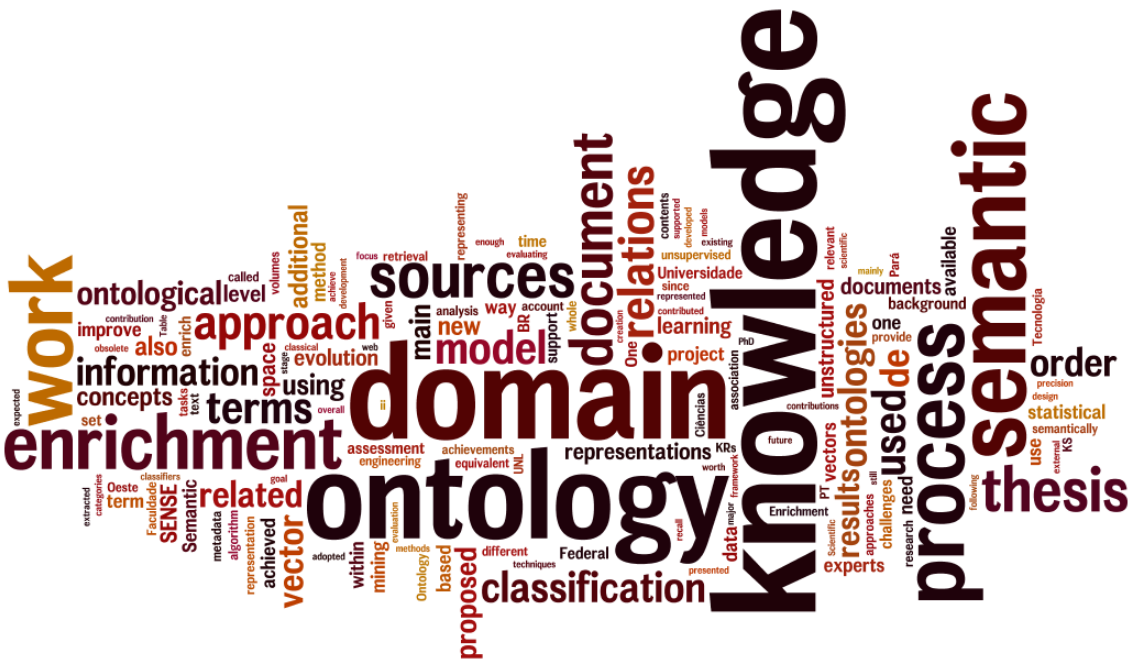
Figure 6.26. "cleaning product" ontology-based search

7

Conclusions

"I think and think for months and years. Ninety-nine times, the conclusion is false. The hundredth time I am right."

- Albert Einstein (1879 – 1955), Nobel Prize in Physics



This chapter summarises the contributions of the current PhD research on the field of semantic enrichment of unstructured information, supported by the external knowledge available in domain ontologies. It gives an overview of the work, presents the thesis outcomes and also discusses the future work.

The research addressed by this thesis is focused on knowledge representations. It targets the development of a computational framework (SENSE - *Semantic Enrichment kNowledge*

SourcEs) to support the creation and use of knowledge representations, using a vector space model (VSM) approach enriched with background knowledge from a domain ontology. The major steps of the work include the analysis of the relations between ontological concepts, and the knowledge sources (KS) they are representing, as well as the enhancement of such relations with semantic associations among concepts.

One of the main challenges addressed by this work relies on the fact that most existing information retrieval techniques are based upon indexing keywords extracted from KS. Regrettably, keywords or index terms alone often cannot adequately capture the document contents, resulting in poor retrieval and indexation performances. Nevertheless, keyword indexing is widely used in commercial systems because it is still the most viable way by far to process large amounts of text.

Such challenges motivate the following **question**: How to intuitively alter and add contents to a document's statistical term vector (a basic knowledge representation) and thereby provide classifiers with more semantically richness than directly found in the document?

Within this thesis, it was proposed to use knowledge available in domain ontologies in order to support the process of representing knowledge sources (e.g. project reports, meeting minutes, description of problems/solutions) picking a case study focused on Building & Construction sector, thus improving the classification of such knowledge sources. Our **hypothesis** is that semantic knowledge from domain ontologies can be used to enrich statistical term vectors. Therefore, one of the main contributions of this work is to affect the document term vectors in a way that we can measure the effect of such semantic enrichment on existing classifiers. Validation of the formulated hypothesis was performed by: (i) collecting results using an unsupervised learning algorithm for document classification, which indicates that the proposed approach does improve the precision and recall of classifications; (ii) collecting results using a query-search mechanism and evaluating the relevance of the retrieved KS according to user's queries.

The CoSpaces project defined the environment to be explored by this thesis, by specifying a collaborative workspace as a composition of a set of checkpoints called decisional gates where issues related to design optimization and risk analysis are taken into account. Each decisional gate is where every party in the collaboration process agrees on an approach to current problem solving, supported by discipline experts. Such an approach acted as an application scenario, which this thesis could build upon, by adding the semantic enrichment dimension to the collaboration process. It could be stated that the innovative ideas pursued by CoSpaces project, established the background for the objectives of this work.

It is also worth mentioning the EU e-Cognos research project worked as one source of inspiration, in order to understand which approaches and methods were deserving of special focus and how they could be improved in order to tackle the domain of work under the scope of

this thesis. In this respect, e-Cognos has contributed in important areas: (i) a method for designing and developing a domain ontology with inputs from knowledge experts, which has an important role within the entire process; (ii) the semantic basis for forming a domain ontology for the B&C sector and (iii) and some initial considerations & thoughts guiding the creation of knowledge representations.

7.1 Results

The main result from this thesis is a *process designed for achieving and evaluating the semantic enrichment of knowledge sources from a given domain*. Furthermore, the main outcome can be sub-divided into several specific scientific and technological achievements, as shown in Table 7.1.

Table 7.1. Scientific and Technological achievements

Scientific
Conceptual model
Ontology formalization & learning
Semantic Enrichment of the VSM
Scientific publications
Academic dissertations/thesis
Technological
Semantic referential
SENSE proof of concept

Regarding scientific achievements, this thesis has contributed the following:

- A domain neutral conceptual model establishing the foundations for enabling semantic enrichment of knowledge sources with the support of knowledge experts, which is intended to be applied to any engineering domain where there is a need for knowledge sharing. The semantic backbone driving the overall semantic enrichment process can be set to any specific terminology for an engineering domain. The model is also product and process neutral, in the sense that it can be instantiated to any kind of engineering process or engineering product. The area of application is manifold; it can support organizations' learning strategies, provide specification for capturing corporate knowledge in a common shared repository, act as a basis for keeping track of previous projects, or focus on a problem-solution representation, enable users to keep track of problems that have occurred and decisions made to solve them, which can be reused whenever necessary to solve new problems.
- A method for ontology formalization and learning, where the discovery of new ontological relations from unstructured knowledge sources is applied. The main difficulties in defining the ontological model were to identify the relevant sources of knowledge which should be

used for ontology enrichment (as explained before; those were extracted from ICONDA digital library). The work performed was mainly concerned with knowledge elicitation from domain experts, with ontology learning from text and integration of unstructured information from heterogeneous sources. The method adopted used an iterative approach split into several phases, each phase containing a set of related tasks. This process turned out to be a time consuming task, in order to achieve a significant expressive model. To note here, the level of expressiveness of the ontology is directly related to the accuracy handled by the semantic enrichment process.

- A method for enriching the traditional vector space model, using background knowledge available in domain ontologies. **The enrichment process of KRs is the core contribution of this work**, demonstrating that knowledge represented in a given ontology can be used to semantically enrich representations of knowledge sources. The whole process ranges from the pre-processing stage till the final assessment of results achieved after the enrichment process, in a cyclical way since the assessment is likely going to provide inputs to improve the quality of the whole process, such as the refinement of the domain ontology. The overall approach comprises 5 stages namely: (i) pre-processing (preparation of the operational environment and input sources); (ii) ontology evolution (augmenting semantic coverage of the ontology considering the inclusion of new knowledge sources in the KB repository); (iii) semantic enrichment (the enrichment process itself); (iv) classification (application of an unsupervised classification algorithm); and (v) evaluation (measure accuracy of the overall approach).
- A set of scientific publications (described previously in chapter 1), where 14 of them were published (1 paper journal, 1 book chapter and 12 conference proceedings) and 2 waiting for decision.
- Support for the execution of academic dissertations/thesis, where three master theses were concluded and another master thesis is about to be finished (Table 7.2). Additionally three PhD studies will extend the achievements provided by this thesis (Table 7.3).

Table 7.2. Concluded MSc dissertations

Title	**Year of conclusion**	**Author**	**Institution**
Discovering Semantic Relations from Unstructured Data for Ontology Enrichment	2014 (expected)	Luis Paiva	Faculdade de Ciências e Tecnologia, UNL, PT

A framework for supporting knowledge representation – an ontological based approach	2012	Paulo Alves Figueiras	Faculdade de Ciências e Tecnologia, UNL, PT
Desenho e implementação de um sistema computacional para apoiar a gestão de projectos utilizando técnicas de data mining	2010	Vitor Miguel Marques Parada	Faculdade de Ciências e Tecnologia, UNL, PT
Design and implementation of an autonomous, proactive, and reactive software infrastructure to help improving the management level of projects	2010	João Pedro Dias Antunes	Faculdade de Ciências e Tecnologia, UNL, PT

Table 7.3. Ongoing PhD thesis

Title	**Starting Year**	**Author**	
Framework conceitual para integração das fontes de conhecimento entre a pós-graduação e os ensinos fundamental e médio	2013	Cássio David Pinheiro	Universidade Federal do Oeste do Pará, BR
Identificação botânica de espécies amazônicas através do reconhecimento de padrões de madeira e óleo essencial: um framework baseado em Ontologia	2013	Márcio José Moutinho da Ponte, Universidade Federal do Oeste do Pará, BR	Universidade Federal do Oeste do Pará, BR
Aprendizado no ensino colaborativo: uma abordagem baseada em Gestão do Conhecimento	2012	Socorro Vânia Lourenço Alves,	Universidade Federal de Pernambuco, BR

From a technological perspective, this thesis contributed through the design and implementation of the SENSE (Semantic Enrichment kNowledge SourcEs) software platform. It is a proof-of-concept offering a query-search engine providing semantic enrichment capabilities enabling

knowledge experts to search for relevant knowledge sources in a semantically richer way. SENSE development covers the semantic enrichment process, (partially) ontology evolution, and classification & searching of knowledge sources. The SENSE platform was developed adopting a notation used to formally support the design, covering functional, architectural, and behavioural views.

7.2 Overview of the work

As presented throughout this thesis, document classification is the process of classifying documents into a pre-defined set of categories, which is one of the most common tasks aimed at grouping and retrieving similar documents. Like many information retrieval tasks, classification techniques rely on using content independent metadata (e.g. author, creation date) or content dependent metadata (i.e., words in the document). One of the challenges that drove this thesis is related to the fact that existing classifiers tend to be inherently limited by the information that is present in the documents. For some of these reasons, the work presented here investigated on efforts towards exploring the use of external semantic metadata available in Ontologies in addition to the metadata central to documents.

Approaches based on extending vector space models are not new and are widely used in information retrieval tasks; they include the Generalized Vector Space Model, Topic-based Vector Space Model, and Latent Semantic Analysis. All have proven to be quite effective for the task of classification. They mainly rely on the explicit co-occurrence of terms and other lexical and morphological normalizations of term vectors. Recently, with the inception of semantic domain models, there were efforts to couple-in the information in vocabularies like WordNet®[8] to enhance the term vectors for text clustering and web document classification. One of the main differences between the approaches above and the one presented in this thesis is that semantic relations between terms expressed by domain ontologies are exploited as a way to semantically enrich KRs.

The classical vector space model has proven to be a quite effective method for representing document contents, but it mainly relies on term occurrences based on a purely statistical approach to creating a vector of terms. This is the reason why this work adds a semantic dimension to the classical vector space model. The semantic enrichment pursued here is evaluated using an unsupervised document classification algorithm.

The intuition behind this work was to alter vectors of terms by strengthening the discriminative terms in a document in proportion to how strongly related they are to other terms in the

[8] WordNet® is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept

document (where relatedness includes all possible relationships modelled in an Ontology). A side effect of the process was the weeding out of the less important terms. Since external knowledge described by ontologies is independent of any document corpus, there is also the possibility of introducing relevant new concepts into the semantic vector that are highly related to the document but not explicit in it.

The results achieved by this thesis show that in general the usage of background knowledge from domain ontologies improves both precision and recall metrics for document classification used for evaluating the semantic enrichment of knowledge sources. Thus it is possible to conclude that the semantic enrichment of KRs can be achieved, with the adoption of background knowledge from domain ontologies.

For the sake of clarity, it is worth mentioning here that domain ontology adopted under this work was not intended to serve as “the one and only one” process model for the domain for two main reasons. First, relevant literature shows that a single ontology could not fully cover a domain of interest. Second, there is no “perfect” ontology and no “optimum” way for modelling a certain domain. Therefore, the main goal was not to build a model that would meet all requirements in a certain domain of interest; rather, the goal was to reuse and extend an ontology which was sufficient (i.e., “good enough”) within a given domain. For this reason, the ontology was evaluated based on contributions coming from domain experts. The ontology used here was developed in a structured, extendable, and flexible format to facilitate its refinement, modification, and extension.

The enrichment process of KRs is the core contribution of this work, essentially arguing that domain knowledge represented in a given ontology can be used to semantically enrich representations of knowledge sources. The whole process stretches from the pre-processing stage to the final assessment of results achieved following the enrichment process. It is cyclical since the assessment is likely to provide inputs that improve the quality of whole process, such as the refinement of the domain ontology.

The semantic enrichment stage is the main focus of this thesis. It tackles the enrichment of knowledge representations (in this work called semantic vectors), extending the classical vector space model approach by including two additional steps in the process: (i) use of taxonomical relations to improve semantic relevance of neighbouring concepts; and (ii) use of ontological relations with the same purpose as point (i).

Regarding ontology evolution, it was proposed to analyse the potential use of a data mining technique called association rule mining for enrichment of domain ontologies. It showed how a domain ontology can be enriched by analysing co-occurrences of related concepts discovered from unstructured data. Firstly proposed to integrate user knowledge to association rule mining using a domain ontology. By applying the proposed approach over huge volumes of unstructured data, it allows the integration of domain expert knowledge into the “Frequent

itemset Mapping" step in order to reduce the number of rules (for additional information regarding "Frequent itemset Mapping", please refer to section 4.4.3 on this document). This step is also supported by the definition of a colour code based schema, which identifies the level of similarity between frequent itemsets and ontology equivalent terms, thus assisting users in defining the ontology relations.

The results achieved by this work indicated that rule mining can be an interesting instrument to explore semantic relations untapped into unstructured information sources. The mined association rules reveal various factors like the skills of actors that are more commonly involved in the different types of construction project, or the nature of relations between different sub-processes of the construction phase of a project.

The assessment of the work was carried out using documents from the ICONDA digital library, containing an average of 3500 terms each, using 5 different categories from the B&C domain. Since the goal of this work was to correctly assess the value of this approach, human inspection was needed in order to validate both precision and recall results, which limited the size of the experiment. For evaluation purposes, the work adopted an unsupervised classification algorithm (K-Means clustering), in order to group knowledge sources into various categories, called clusters.

The functionalities provided by the proof of concept developed under the scope of this work were implemented by means of java web services and mysql stored procedures. Such services were needed to be integrated with third-party tools for performing additional functionalities, such as: RapidMiner for performing the statistical analysis and Liferay for deploying a document management system. It is worth highlighting that documents had to be converted into *txt* format prior to being processed for statistical analysis, since such transformation enables the optimization of processing time within each document. Protégé editor was used for formalizing the domain ontology in OWL-DL specification.

7.3 Future Work – Challenges in the semantic enrichment quest

The results achieved by this thesis indicate that the inclusion of additional information available in domain ontologies in the process of representing knowledge sources can enrich and improve knowledge representations. Nevertheless, in order to reach more formal conclusions, additional evaluation needs to be undertaken including additional metrics for assessing the performance of the proposed method.

As future work, it is expected to conduct similar tests on a larger dataset in order to assess the scalability of the proposed approach. Machine learning technology faces challenges in handling "Big Data" – vast volumes of online data such as web pages, news stories and articles. A dominant solution has been parallelization. The challenges faced when working with a larger dataset are strongly related to the efficient computational capabilities which are needed in order to compute large dataset volumes. As previously discussed by this thesis, most of machine learning algorithms used for document representation and classification are highly resource intensive. For the purpose of handling highly dimensionality vector space models and vast volumes of online data, it would be interesting to analyse how the approach proposed here scales up by integrating it with the MapReduce[9] paradigm.

The need to have a richer ontological model was also identified as a major challenge to be taken into account, where concepts and equivalent terms are in a level of detail enough to cover a high percentage of knowledge sources contents. One of the major drawbacks regarding this work is directly related with the level of expressiveness of ontological model used, which failed to achieve a level of detail enough to get "acceptable" results. Considerable extra effort was spent on enriching the ontology with additional new concepts and equivalent terms in order to achieve more substantial results.

Additional efforts for ontology evolution mechanisms must be taken into consideration. Ontologies are sometimes handled as something that is static, and which does not evolve over

[9] MapReduce is a programming model and an associated implementation for processing and generating large data sets.

time as organizational knowledge does. There is a need to automatically update (improve) the domain ontology every time new knowledge sources are created, reducing at the same time human intervention. This work has provided a valuable contribution in this domain by proposing an approach for discovery and extracting ontological relations from unstructured information, however it still requires human intervention. Such ontological relations can also be represented by a taxonomy of relations, since ontology relations can be viewed by interdependencies between them with different levels of granularity. A taxonomy of relations can contribute to the calculation of more precise weights relations, and not only focusing on association rules mining from information extracted from documents.

Ontology evolution aims to automatically extract ontological concepts and relationships from related text repositories and is expected to be more efficient and scalable than manual ontology development. One of the challenging issues associated with ontology evolution is Word Sense Disambiguation (WSD). As future work, additional methods are required to reduce word ambiguity by taking account of the context where each word occurs. Such methods are required for the process of matching terms in the statistical vector with the equivalent terms from the domain ontology. Some more recent approaches on WSD employ resources such as WordNet, Wikipedia, and social media. This needs further investigation on how to integrate such approaches with NLP techniques such as Part of Speech[10] (POS) tagging, in order to more accurately acquire the meaning of each word within the KS context.

Also related to ontology evolution, there is a need to deal with obsolete entities. Any obsolete entity may still be retained within the ontology even if it is not being used anymore. The obsolete entities can live within the ontology for as long as necessary until a 'refresh' operation cleans up and reorganizes the ontology. It might also live indefinitely if historical views of the semantic referential need to be kept available.

With respect to the adoption of the proposed work to other domain areas, further research must take into account the need to guarantee a level of semantic interoperability between the SENSE and other domain ontologies. This means developing new mechanisms that enable other domain ontologies to be easily integrated with SENSE semantic specification.

[10] Part of Speech is a linguistic category of words (or more precisely lexical items), which is generally defined by the syntactic or morphological behavior of the lexical item in question

8

Bibliography

Baeza-Yates, Ricardo, and Berthier Ribeiro-Neto. *Modern Information Retrieval.* Boston, MA: Addison-Wesley Longman, 1999.

Ackoff, Russell. “From Data to Wisdom.” *Journal of Applied Systems Analysis* 16 (1989): 3-9.

Ahsan, Syed, and Abad Shah. “Data, Information, Knowledge, Wisdom: A Doubly Linked Chain?” *University of Engineering and Technology, Lahore*, 2006.

Alavi, Maryam, and Dorothy E. Leidner. “Knowledge Management Systems: Issues, Challenges, and Benefits.” *Communications of the Association for Information Systems* 1 (February 1999): 1-28.

Anderberg, Michael. *Cluster analysis for applications.* Academic Press, 1973.

Barresi, Simona, Yacine Rezgui, Celson Lima, and Farid Meziane. “Architecture to support semantic resources interoperability.” *First international workshop on Interoperability of heterogeneous information systems.* Bremen: ACM, 2005. 79-82.

Bellinger, Gene, Durval Castro, and Anthony Mills. *Data, Information, Knowledge, and Wisdom.* 2004. http://www.systems-thinking.org/dikw/dikw.htm (accessed September 22, 2009).

Beyer, Kevin, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. “When Is "Nearest Neighbor" Meaningful?” *International Conference on Database Theory.* Jerusalem: Springer, 1999. 217-235.

Böhms, Michel, Celson Lima, Graham Storer, and Jeffrey Wix. “Framework for Future Construction ICT.” *International Journal of Design, Sciences & Technology*, 2004: 153-162.

Brown, John Seely, and Paul Duguid. *The Social Life of Information.* Harvard Business School Press, Boston, MA, 2000.

Caropreso, Maria Fernanda, Stan Matwin, and Fabrizio Sebastiani. “A Learner-Independent Evaluation of the Usefulness of Statistical Phrases for Automated Text Categorization.” In *Text databases and document management: theory and practice*, by Amita G Chin, 78-102. Hershey: Idea Group Publishing, 2001.

Cavnar, William, and John Trenkle. “N-Gram-Based Text Categorization.” *3rd Annual Symposium on Document Analysis and Information Retrieval.* Las Vegas, 1994. 161-175.

Cerovsek, Tomo, Gudni Gudnason, and Celson Lima. “CONNIE - Construction News and Information Electronically.” *Joint International Conference on Computing and Decision Making in Civil and Building Engineering.* Montreal, Canada, 2006. 2437-2446.

Checkland, Peter, and Jim Scholes. *Soft systems methodology in action.* New York: John Wiley & Sons, Inc., 2000.

Chekuri, Chandra, Michael Goldwasser, Prabhakar Raghavan, and Eli Upfal. “Web Search Using Automatic Classification.” *Sixth International World Wide Web Conference.* Santa Clara, 1997.

Cleverdon, Cyril. “The significance of the Cranfield tests on index languages.” *14th annual international ACM SIGIR conference on Research and development in information retrieval.* Chicago: ACM Press, 1991. 3-12.

Cohen, William, and Yoram Singer. “Context-sensitive learning methods for text categorization.” *19th annual international ACM SIGIR conference on Research and development in information retrieval.* Zurich: ACM, 1996. 307-315.

Corcho, Oscar, Mariano Fernandez-Lopez, and Asuncion Gomez-Perez. “Methodologies, tools and languages for building ontologies. Where is their meeting point?” *Data and Knowledge Engineering*, 2003: 41-64.

Cortes, Corinna, and Vladimir Vapnik. “Support-Vector Networks.” *Journal of Machine Learning*, 1995: 273-297.

Costa, Ruben, Celson Lima, João Antunes, Paulo Figueiras, and Vitor Parada. “Knowledge Management Capabilities Supporting Collaborative Working Environments in a Project Oriented Context.” *European Conference on Intellectual Capital.* Lisbon: ACI, 2010. 208-216.

Costa, Ruben, Paulo Figueiras, Luis Paiva, Ricardo Jardim-Gonçalves, and Celson Lima. “Capturing Knowledge Representations Using Semantic Relationships.” *The Sixth International Conference on Advances in Semantic Processing.* Barcelona, Spain: IARIA, 2012.

Costa, Ruben, Paulo Figueiras, Pedro Maló, and Celson Lima. “Classification of Knowledge Representations using an Ontology-based Approach.” *5th International Conference on Knowledge Engineering and Ontology Development.* Vilamoura: SciTePress, 2013.

Croft, William Bruce, and David John Harper. “Using probabilistic models of document retrieval without relevance information.” In *Document retrieval systems*, by Peter Willett, 161-171. London: Taylor Graham Publishing, 1988.

Dhillon, Inderjit, James Fan, and Yuqiang Guan. “Efficient Clustering Of Very Large Document Collections.” *Data mining for scientific and engineering applications*, 2001: 357-381.

Duda, Richard, Peter Hart, and David Stork. *Pattern Classification.* Wiley-Interscience, 2000.

Dumais , Susan, and Hao Chen. “Hierarchical classification of Web content.” *23rd annual international ACM SIGIR conference on Research and development in information retrieval.* Athens: ACM, 2000. 256-263.

El-Diraby, Tamer, Celson Lima, and Bruno Fiès. “Domain Taxonomy for Construction Concepts: Toward a Formal Ontology for Construction Knowledge.” *Journal of Computing in Civil Engineering* 19, no. 4 (2005): 394-406.

Fan, Weiguo, Linda Wallace, Stephanie Rich, and Zhongju Zhang. “Tapping into the Power of Text Mining.” *Communications of ACM*, 2005: 76–82.

Fayyad, Usama, Cory Reina, and Paul Bradley. “Initialization of Iterative Refinement Clustering Algorithms.” *ourth International Conference on Knowledge Discovery and Data Mining.* New York: AAAI Press, 1998. 194-198.

Fitzgerald, Brian, and Debra Howcroft. “Competing dichotomies in IS research and possible strategies for resolution.” *International conference on Information systems.* Helsinki: Association for Information Systems, 1998. 155-164.

Frakes, William, and Ricardo Baeza-Yates. *Information retrieval: data structures and algorithms.* New Jersey: Prentice-Hall, 1992.

Furnas, George, et al. “Information retrieval using a singular value decomposition model of latent semantic structure.” *11th annual international ACM SIGIR conference on Research and development in information retrieval .* Grenoble: ACM, 1988. 465-480.

Gruber, Thomas. “Toward principles for the design of ontologies used for knowledge sharing.” *International Journal of Human-Computer Studies*, 1993: 907-928.

Gudivada, Venkat, Vijay Raghavan, William Grosky, and Rajesh Kasanagottu. “Information Retrieval on the World Wide Web.” *IEEE Internet Computing*, 1997: 58-68.

Haveliwala, Taher. “Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search.” *IEEE Transactions on Knowledge and Data Engineering*, 2003: 784-796.

Hey, Jonathan. *Berkeley Expert Systems Technology.* December 2004. http://best.berkeley.edu/~jhey03/files/reports/IS290_Finalpaper_HEY.pdf (accessed January 4, 2010).

Hidalgo, José María Gómez. “Text Representation for Automatic Text Categorization.” 12 April 2003. http://www.esi.uem.es/~jmgomez/tutorials/eacl03/slides.pdf (accessed May 28, 2013).

Holsapple, Clyde W. *Handbook on Knowledge Management.* Berlin: Springer-Verlag, 2003.

Hotho, Andreas, Steffen Staab, and Alexander Maedche. “Ontology-based Text Clustering.” *IJCAI-2001 Workshop “Text Learning: Beyond Supervision".* Seattle, 2001.

Huang, Yifen, and Tom Mitchell. “Text clustering with extended user feedback.” *29th annual international ACM SIGIR conference on Research and development in information retrieval.* Seattle: ACM Press, 2006. 413-420.

IRB, Fraunhofer. “ICONDA®Bibliographic.” 1986.

—. *ICONDA®Bibliographic.* 1986. http://www.iconda.org/ (accessed 06 29, 2012).

Jain, Anil Kumar, M Narasimha Murty, and Patrick Joseph Flynn. “Data clustering: a review.” *ACM Computing Surveys*, 1999: 264-323.

Jasimuddin, Sajjad M., Jonathan H. Klein, and Con Connell. “The paradox of using tacit and explicit knowledge.” *Management Decision* 43 (2005): 102-112.

Ji, Xiang, and Wei Xu. “Document clustering with prior knowledge.” *29th annual international ACM SIGIR conference on Research and development in information retrieval.* Seattle: ACM Press, 2006. 405-412.

Jing, Liping, Lixin Zhou, Michael Ng, and Joshua Huang. “Ontology-based Distance Measure for Text Clustering.” *SIAM SDM workshop on text mining.* Bethesda, 2006.

Joachims, Thorsten. “Text Categorization with Suport Vector Machines: Learning with Many Relevant Features.” *Proceedings of the 10th European Conference on Machine Learning.* Chemnitz: Springer-Verlag, 1998. 137-142.

Jones, Karen, and Peter Willett. *Readings in information retrieval.* San Francisco: Morgan Kaufmann Publishers Inc., 1997.

Jones, Karen Sparck. “A statistical interpretation of term specificity and its application in retrieval.” In *Document retrieval systems*, by Peter Willett, 132-142. London: Taylor Graham Publishing, 1988.

Jones, Karen Spärck. “Idf term weighting and ir research lessons.” *Journal of Documentation*, 2004: 521–523.

Käki , Mika. “Findex: search result categories help users when document ranking fails.” *SIGCHI Conference on Human Factors in Computing Systems.* Portland: ACM, 2005. 131-140.

Kazi, Abdul Samad. *Knowledge management in the construction industry: A socio-technical perspective.* London: Idea Group Publishing, 2005.

Kent, Allen, Madeline Berry, Fred U, Luehrs Jr, and J Perry. “Machine literature searching VIII. Operational criteria for designing information retrieval systems.” *American Documentation*, 2007: 93-101.

Ketchen, David, and Christopher Shook. “The Application of Cluster Analysis in Strategic Management Research: An Analysis and Crtique.” *Strategic Management Journal*, 1996: 441-458.

Kohlschütter, Christian, Paul-Alexandru Chirita, and Wolfgang Nejdl. “Utility analysis for topically biased PageRank.” *16th international conference on World Wide Web.* Banff: ACM, 2007. 1211-1212.

Kurland, Oren, and Lillian Lee. “Corpus structure, language models, and ad hoc information retrieval.” *27th annual international ACM SIGIR conference on Research and development in information retrieval*, 2004: 194-201.

Lewis, David. “Text representation for intelligent text retrieval: a classification-oriented view.” In *Text-based intelligent systems*, by Paul Jacobs, 179-197. New Jersey: Lawrence Erlbaum Associates, Inc., 1992.

Lima, Celson. “European eConstruction Ontology by eEurope Pilot Project SPICE.” CWA4 proposal, 2004.

Lima, Celson, Alain Zarli, and Graham Storer. “Controlled Vocabularies in the European Construction Sector: Evolution, Current Developments, and Future Trends.” In *Complex Systems Concurrent Engineering*, by Geilson Loureiro and Richard Curran, 565-574. Springer London, 2007.

Lima, Celson, Catarina Silva, Pedro Sousa, and João Pimentão. “Interoperability among Semantic Resources in Construction: Is it Feasible?” *CIB / W78 22nd Conference on Information Technology in Construction.* Dresden, Germany: CIB Publication, 2005. 285-292.

Lima, Celson, Flavio Bonfatti, Silvia Sancho, and Anastasiya Yurchyshyna. “Towards an Ontology-enabled Approach Helping SMEs to Access the Single European Electronic Market.” *13th ISPE International Conference on Concurrent Engineering: Research and Applications.* Nice, France: IOS Press, 2006. 57-68.

Lima, Celson, Jeff Stephens, and Michel Böhms. “The bcXML: supporting eCommerce and knowledge management in the construction industry.” *Itcon Journal*, 2003: 293-308.

Lima, Celson, Tamer El-Diraby, and Jeff Stephens. “Ontology-based optimisation of knowledge management in e-Construction.” *ITcon* 10 (2005): 305-327.

Luhn, Hans Peter. “The automatic creation of literature abstracts.” *IBM Journal of Research and Development*, 1958: 159-165.

MacQueen, J. “Some methods for classification and analysis of multivariate observations.” Berkeley: University of California Press, 1967.

Macskassy, Sofus, Arunava Banerjee, Brian Davison, and Haym Hirsh. “Human Performance on Clustering Web Pages: A Preliminary Study.” *International Conference on Knowledge Discovery and Data Mining.* New York: AAAI Press, 1998. 264-268.

Manning, Chris, and Hinrich Schütze. *Foundations of Statistical Natural Language Processing.* Cambridge: MIT Press, 2001.

Manning, Christopher, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval.* New York, NY: Cambridge University Press, 2009.

McGuinness, Deborah. “Ontologies Come of Age.” In *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*, by Dieter Fensel, J im Hendler, Henry Lieberman and Wolfgang Wahlster. MIT Press, 2003.

Mitchell, Tom. *Machine Learning.* New York: McGraw Hill, 1997.

Muresan, Gheorghe, and David Harper. “Topic modeling for mediated access to very large document collections.” *Journal of the American Society for Information Science and Technology*, 2004: 892-910.

Nagarajan, Meenakshi, Amit Sheth, Marcos Aguilera, Kimberly Keeton, Arif Merchant, and Mustafa Uysal. “Altering Document Term Vectors for Classification - Ontologies as Expectations of Co-occurrence.” *16th international conference on World Wide Web.* Alberta: ACM, 2007. 1225-1226.

Nie, Lan, Brian Davison, and Xiaoguang Qi. “Topical link analysis for web search.” *29th annual international ACM SIGIR conference on Research and development in information retrieval.* Seattle: ACM, 2006. 91-98.

Nonaka, I., R. Toyama, and N. Konno. “SECI, Ba and Leadership: a Unified Model of Dynamic Knowledge Creation.” *Long Range Planning.* 33, no. 1 (February 2000): 5-34.

Nonaka, Ikujiro, and Hirotaka Takeuchi. *The Knowledge Creating Company.* New York, NY: Oxford University Press, 1995.

—. *The Knowledge Creating Company.* New York, NY: Oxford University Press, 1995.

Ogawa, Yasushi, Tetsuya Morita, and Kiyohiko Kobayashi. “A fuzzy document retrieval system using the keyword connection matrix and a learning method.” *Fuzzy Sets and Systems - Special issue on applications of fuzzy systems theory*, 1991: 163-179.

Page, Lawrence, Sergey Brin, Rajeev Motwani, and Terry Winograd. *The PageRank Citation Ranking: Bringing Order to the Web.* Unpublished draft, Stanford: Stanford University, 1998.

Paiva, Luis, Ruben Costa, Paulo Figueiras, and Celson Lima. “Discovering Semantic Relations from Unstructured Data for Ontology Enrichment - Association rules based approach.” *8th Iberian Conference on Information Systems and Technologies.* Lisbon: IEEE, 2013.

Papineni, Kishore. “Why inverse document frequency?” *Second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies.* Pittsburgh: Association for Computational Linguistics, 2001. 1-8.

Parsons, Lance, Ehtesham Haque, and Huan Liu. “Subspace clustering for high dimensional data: a review.” *ACM SIGKDD Explorations*, 2004: 90-105.

Peng, Fuchun, Dale Schuurmans, and Shaojun Wang. “Language and task independent text categorization with simple language models.” *Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology.* Edmonton: Association for Computational Linguistics, 2003. 110-117.

Polanyi, Michael. *The Tacit Dimension.* London, UK: University of Chicago Press, 1966.

Porter, M. “An algorithm for suffix stripping.” *Program*, 1980: 130-137.

Qi, Xiaoguang, and Brian Davison. “Web page classification: Features and algorithms.” *ACM Computing Surveys*, 2009: 1-31.

Quinlan, John Ross. “Learning efficient classification procedures and their applications to chess and games.” In *Machine learning: An artificial intelligence approach*, by Ryszard Michalski, Jaime Carbonell and Tom Mitchell, 463-482. Los Altos: Springer Berlin Heidelberg, 1983.

Quinn, James Brian, Philip Anderson, and Sydney Finkelstein. “Managing Professional Intellect: Making the Most of the Best.” In *Harvard Business Review on knowledge management*, by Peter F. Drucker and David A. Garvin. Cambridge, MA: Harvard Business Review, 1996.

Reuss, Mark, and C. Bob Tatum. *Requirements and Tools for Transferring Construction Experience Between Projects.* Technical Report, Stanford: Stanford University, 1993.

Ribeiro, Berthier, and Richard Muntz. “A belief network model for IR.” *19th annual international ACM SIGIR conference on Research and development in information retrieval.* Zurich: ACM, 1996. 253-260.

Rijsbergen, Cornelis. *Information Retrieval .* MA: Butterworth-Heinemann Newton, 1979.

Robertson, S, and K Jones. “Relevance weighting of search terms.” *Journal of the American Society for information space*, 1976: 129-146.

Robertson, Stephen. “Understanding inverse document frequency: On theoretical arguments for IDF.” *Journal of Documentation*, 2004: 503-520.

Rowley, J. “The wisdom hierarchy: representations of the DIKW hierarchy.” *Journal of Information Science* 33, no. 2 (2007): 163-180.

Ryle, Gilbert. *The concept of mind.* Chicago, IL: University of Chicago Press, 1984.

Salton, Gerard. “The smart document retrieval project.” *14th annual international ACM SIGIR conference on Research and development in information retrieval.* Chicago: ACM Press, 1991. 356-358.

Salton, Gerard, A Wong, and C Yang. “A vector space model for automatic indexing.” *Communications of the ACM*, 1975: 613-620.

Salton, Gerard, and Christopher Buckley. “Term-weighting approaches in automatic text retrieval.” *Information Processing and Management*, 1988: 513-523.

Salton, Gerard, Edward Fox, and Harry Wu. “Extended Boolean information retrieval.” *Communications of the ACM*, 1983: 1022-1036.

Sheng, Li. “A Semantic Vector Retrieval Model for Desktop Documents.” *Journal of Software Engineering and Applications* 2, no. 1 (2009): 55-59.

Singhal, Amit, Chris Buckley, and Mandar Mitra. “Pivoted document length normalization.” *19th annual international ACM SIGIR conference on Research and development in information retrieval.* Zurich: ACM, 1996. 21-29.

Snowden, D. “The Knowledge You Need, Right when You Need It.” *Knowledge Management Review* 5, no. 6 (2003): 24-27.

Steinbach, Michael, George Karypis, and Vipin Kumar. “A comparison of document clustering techniques.” *KDD Workshop on Text Mining.* 2000.

Studer, Rudi, V Richard Benjamins, and Dieter Fensel. “Knowledge engineering: Principles and methods.” *IEEE Transactions on Data and Knowledge Engineering*, 1998: 161-97.

Swanson, Don. “Historical note: Information retrieval and the future of an illusion.” *Journal of the American Society for Information Science*, 1988: 92–98.

Turtle, Howard Robert, and William Bruce Croft. “Inference networks for document retrieval.” *13th annual international ACM SIGIR conference on Research and development in information retrieval.* Brussels: ACM, 1990. 1-24.

Turtle, Howard, and James Flood. “Query evaluation: strategies and optimizations.” *Information Processing and Management: an International Journal*, 1995: 831-850.

Turtle, Howard, and William Bruce Croft. “Evaluation of an inference network-based retrieval model.” *ACM Transactions on Information Systems - Special issue on research and development in information retrieval*, 1991: 187-222.

van Rijsbergen, Cornelis. *Information Retrieval.* Newton: Butterworth-Heinemann, 1979.

Verhoeff, J, W Goffman, and Jack Belzer. “Inefficiency of the use of Boolean functions for information retrieval systems.” *Communications of the ACM*, 1961: 557-558.

Voorhees, Ellen, and Donna Harman. *TREC: Experiment and Evaluation in Information Retrieval.* MIT Press, 2005.

Wenger, Etienne, and William Snyder. “Communities of practice: The organizational frontier.” *Harvard Business Review*, 2000: 139-145.

Wetherill, Matthew, Yacine Rezgui, Celson Lima, and Alain Zarli. “Knowledge management for the construction industry: the e-cognos project.” *ITcon*, 2002: 183-196.

Wilkinson, Ross, and Philip Hingston. “Using the cosine measure in a neural network for document retrieval.” *14th annual international ACM SIGIR conference on Research and development in information retrieval.* Chicago: ACM, 1991. 202-210.

Wyatt, J.C. “Management of explicit and tacit knowledge.” *Journal of the Royal Soceity of Medicine* 94 (January 2001): 6-9.

Xia, Tian, and Yi Du. “Improve VSM text classification by title vector based document representation method.” *6th International Conference on Computer Science & Education.* Singapore: IEEE, 2011. 210-213.

Xu, Jinxi, and William Bruce Croft. “Cluster-based language models for distributed retrieval.” *22nd annual international ACM SIGIR conference on Research and development in information retrieval.* Berkeley: ACM Press, 1999. 254-261.

Yang, Yiming. “An Evaluation of Statistical Approaches to Text Categorization.” *Journal of Information Retrieval*, 1999: 69-90.

Yang, Yiming, and Jan Pedersen. “A Comparative Study on Feature Selection in Text Categorization.” *Fourteenth International Conference on Machine Learning.* Nashville: Morgan Kaufmann Publishers Inc., 1997. 412-420.

Zeleny, Milan. “Management Support Systems: Towards integrated knowledge management.” *Human Systems Management.* 7 (1987): 59-70.

Zhang, Wen, Taketoshi Yoshida, and Xijin Tang. “A comparative study of TF-IDF, LSI and multi-words for text classification.” *Expert Systems with Applications*, 2011: 2758-2765.

Zhao, Ying, and George Karypis. *Comparison of Agglomerative and Partitional Document Clustering Algorithms.* Technical report, University of Minnesota, 2002.

Zobel, Justin, and Alistair Moffat. “Exploring the similarity space.” *ACM SIGIR Forum*, 1998: 18-34.

Zobel, Justin, and Alistair Moffat. “Inverted files for text search engines.” *ACM Computing Surveys*, 2006.

9

Annexes

9.1 Annex A – Data Models (Entity-Relation Diagrams)

This section illustrates the Entity-Relation diagrams, which were designed in order to implement the knowledge repository, which is composed by the KR representation repository (Figure 9.1), Ontology repository (Figure 9.2).

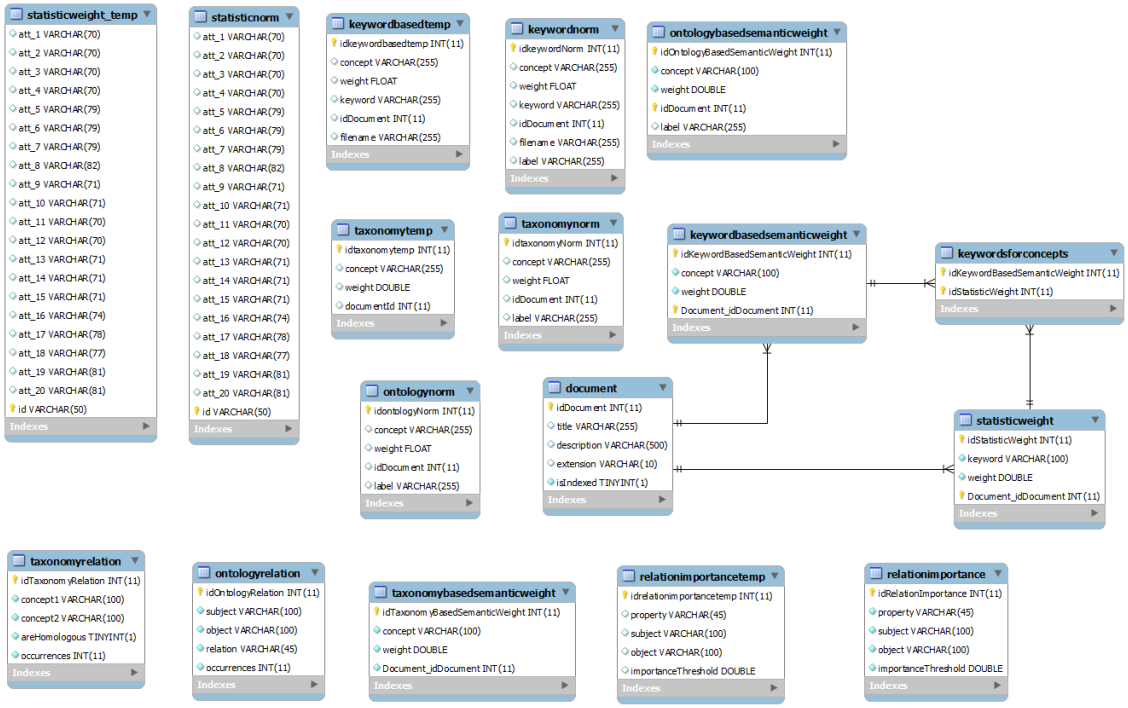


Figure 9.1. KR Repository

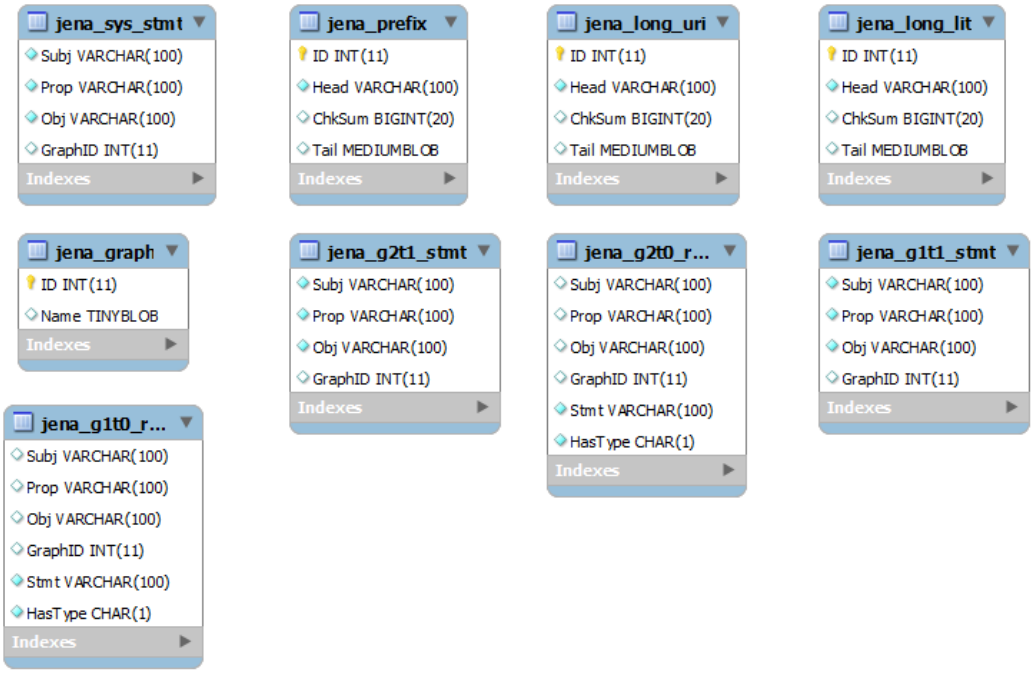


Figure 9.2. Ontology Repository

9.2 Annex B – Stored Procedures

This section describes the operations which are performed by the stored procedures implemented.

Table 9.1. Stored Procedures description

Name	Input : type	Description
Select Procedures		
getAllDocumentIDs		Fetches all primary keys from table *Document*
getDocumentNumWithConcept	*concept* : varchar	Fetches the number of primary keys from instances of table *KeywordBasedSemanticWeight* that have their *concept* fields equal to the input parameter
getKeywordBasedWeightsWithDocID	*documentID* : int	Selects all instances of table *KeywordBasedSemanticWeight* that have their *Document_idDocument* fields equal to the input parameter
getMaxTaxonomyRelationOccurrences		Selects the instance of table *TaxonomyRelation* that has the higher value for field *occurrences*
getNotIndexedDocumentIDs		Selects all instances of table *Document* that have their fields *isIndexed* equal to *false* (0)
getOntologyBasedWeightsWithDocID	*documentID* : int	Selects all instances of table *OntologyBasedSemanticWeight* that have their *Document_idDocument* fields equal to the input parameter
getOntologyRelationOccurrences	*concept1* :varchar *concept2* : varchar	Selects all instances of table *OntologyRelation* where the fields *subject* and *object* are equal to the input parameters
getRelationImportanceWithConcepts	*concept1* :varchar *concept2* : varchar	Selects all instances of table *RelationImportance* where the fields

		subject and *object* are equal to the input parameters
getRelationImportanceWithMinimumThreshold	*threshold* : int	Selects all instances of table *RelationImportance* where the field *importanceThreshold* is equal to the input parameter
getStatisticWeightsWithDocID	*documentID* : int	Selects all instances of table *StatisticWeight* that have their *Document_idDocument* fields equal to the input parameter
getTaxonomyBasedWeightsWithDocID	*documentID* : int	Selects all instances of table *TaxonomyBasedSemanticVector* that have their *Document_idDocument* fields equal to the input parameter
getTaxonomyRelationOccurrences	*concept1* :varchar *concept2* : varchar	Selects all instances of table *TaxonomyRelation* where the fields *subject* and *object* are equal to the input parameters
getTotalDocumentNum		Fetches the total number of instances of table *Document*
Insert Procedures		
insertKeywordBasedWeight	*concept* : varchar *weight* : double *documentID* : int	Inserts a new instance on table *KeywordBasedSemanticWeight*, setting its values with the input parameters
insertOntologyBasedWeight	*concept* : varchar *weight* : double *documentID* : int	Inserts a new instance on table *OntologyBasedSemanticWeight*, setting its values with the input parameters
insertOntologyRelation	*subject* : varchar *object* : varchar *relation* : varchar	Inserts a new instance on table *OntologyRelation*, setting its values with the input parameters
insertRelationImportance	*property* : varchar *subject* : varchar	Inserts a new instance on table *RelationImportance*, setting its values

	object : varchar *threshold* : double	with the input parameters
insertTaxonomyBasedWeight	*concept* : varchar *weight* : double *documentID* : int	Inserts a new instance on table *TaxonomyBasedSemanticWeight*, setting its values with the input parameters
insertTaxonomyRelation	*subject* : varchar *object* : varchar *relation* : varchar	Inserts a new instance on table *TaxonomyRelation*, setting its values with the input parameters
Update Procedures		
updateOntologyRelationOccurrences	*concept1* :varchar *concept2* : varchar *relation* : varchar	Increments the field occurrences of all instances of table *OntologyRelation* that have their fields *subject*, *object* and *relation* equal to the input parameters
updateTaxonomyRelationOccurrences	*concept1* :varchar *concept2* : varchar *relation* : varchar	Increments the field occurrences of all instances of table *TaxonomyRelation* that have their fields *subject*, *object* and *relation* equal to the input parameters

9.3 Annex C – Classes Interfaces

Implemented as a Java Web application, compliant with Java 6, Java Servlets 3 and JAX-WS 2.2.6 and built to run on Apache Tomcat 7. The system was developed using Eclipse Integrated Development Environment (IDE) and its configuration files, Java packages and class structure are shown in Figure 9.3.

The *.owl* file stores the SENSE ontology. It is available to be accessed by the persistent model, whenever there is a need for it, such as the creation of a new version of the ontology, or a swap of domain-specific ontologies. The three *.xml* files are database access configuration files that configure databases' hosts, ports, databases names and MySQL usernames and passwords: *jenaConfig.xml* configures the access to the persistent model of the ontology; *lportalConfig.xml* manages the database connection with Liferay's document repository; and *svdbConfig.xml* is responsible for the access to statistic and semantic vectors from SENSE database.

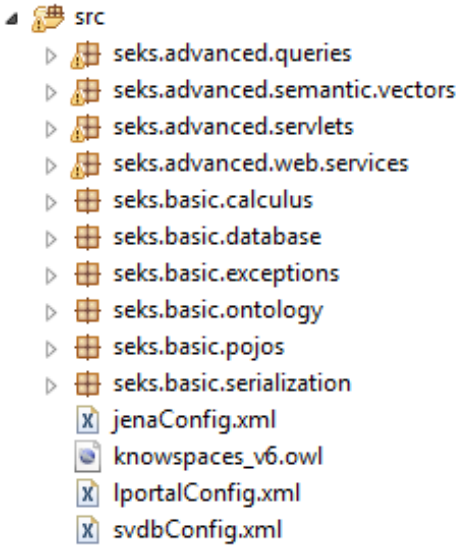


Figure 9.3. SENSE system class structure

SENSE system includes basic and advanced service packages. The implementation classes were designed and modeled using UML Class Diagrams (UCD's).

9.3.1 Basic Services

Basic services contain five class packages, beginning with the package name *seks.basic*, four of which implement its services, as shown in Figure 9.3: calculus, database, *serialization* and *ontology*. The last class package, *seks.basic.pojos*, comprises Java object classes needed in the process for system performance and database data retrieval purposes, and is represented in Figure 9.4.

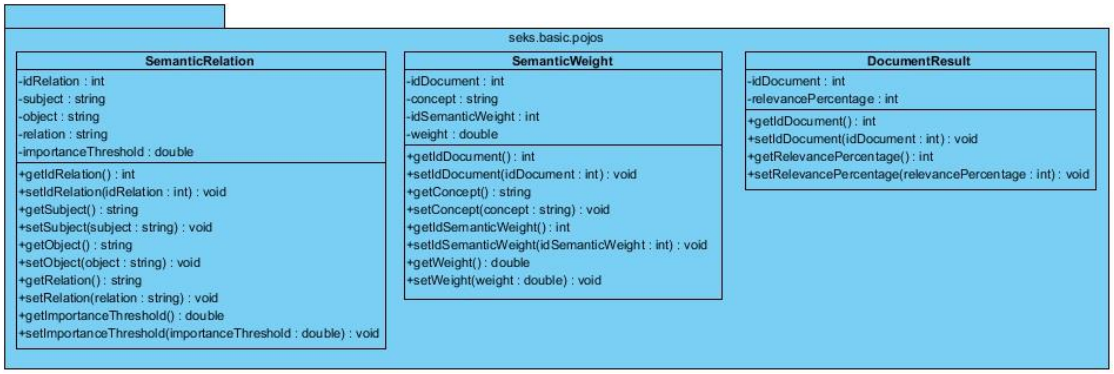


Figure 9.4. Plain Old Java Objects (POJOs) classes

Class package *seks.basic.calculus*, represented in Figure 9.5, contains the *tf-idf* algorithm, a vector normalization function, the homologous and non-homologous factor computation algorithm and the Euclidian distance algorithm.

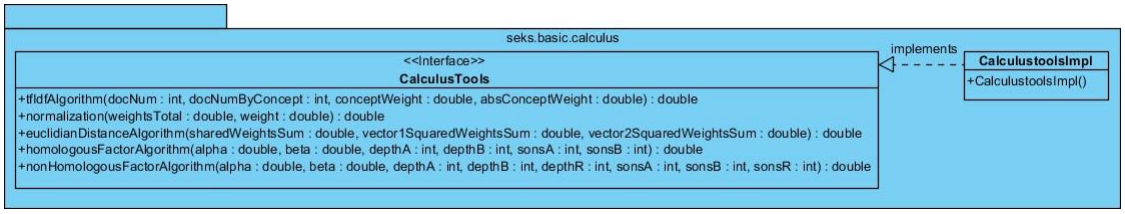


Figure 9.5. Calculus Services classes and interfaces

Class package *seks.basic.database*, shown in the UCD of Figure 9.6, is responsible for opening and closing MySQL connections to interact with the system's databases and repositories, and for calling database routines presented in Figure 9.7.

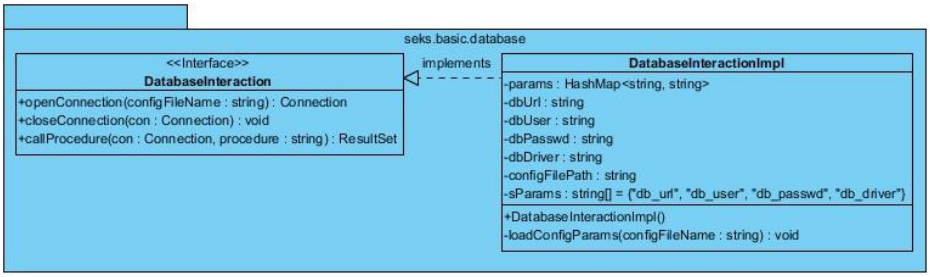


Figure 9.6. Database Services classes and interfaces

Class package *seks.advanced.ontology* (Figure 9.7) has two interfaces with corresponding classes: *OntologyPersistence.java* and *OntologyInteraction.java*. *OntologyPersistence.java* class creates a database map of the ontology for online interaction with SENSE or other systems that use the SENSE Web Services Interface. *OntologyInteraction.java* contains all methods that interact with SENSE ontology. These methods are supported by the Apache Jena Semantic Framework.

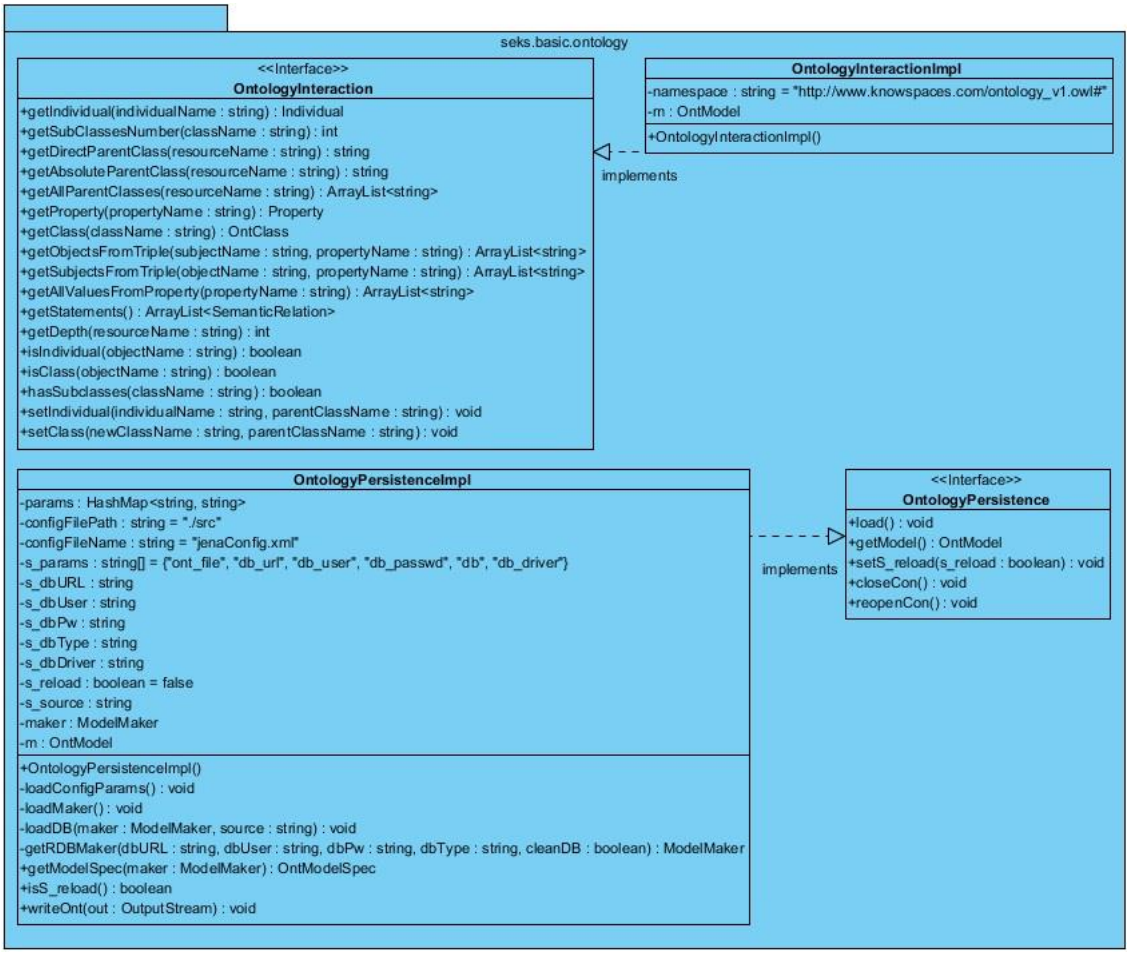


Figure 9.7. Ontology Services classes and interfaces

Finally, class package *seks.basic.serialization* (Figure 9.8) is responsible for the serialization and deserialization methods used by the Web Services Interface to transmit responses to other systems that use SENSE functionalities.

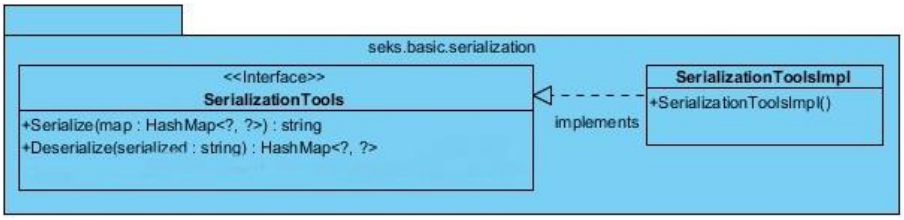


Figure 9.8. Serialization Services classes and interfaces

Such transmission is often made with XML strings, as is the case for the presented work. Serialization mechanisms were needed specifically because semantic vectors are managed by the system as *java.util.HashMap* objects, which are not automatically serialized by JAX-WS framework.

9.3.2 Advanced Services

Advanced Services cover two packages: one for query handling and the other for knowledge source semantic vector creation and vector comparison. Package *seks.advanced.sematic.vectors* (Figure 9.9) manages the creation of all three semantic vector iterations and also handles vector comparison. It comprises one class and one interface for each of the processes mentioned. *KeywordBasedSVCreation.java* class interacts with all Basic Services (except *seks.basic.serialization* class package which only interacts exclusively with the Web Services Interface) to create keyword-based semantic vectors.

The same interaction applies to *TaxonomyBasedSVCreation.java* and *OntologyBasedSVCreation.java*, which responsibility is to create the respective taxonomy- and ontology-based semantic vectors. Finally, SemanticVectorComparison.java handles all methods needed for vector comparison, including statistic and semantic vector union and interacting also with all Basic Services class packages.

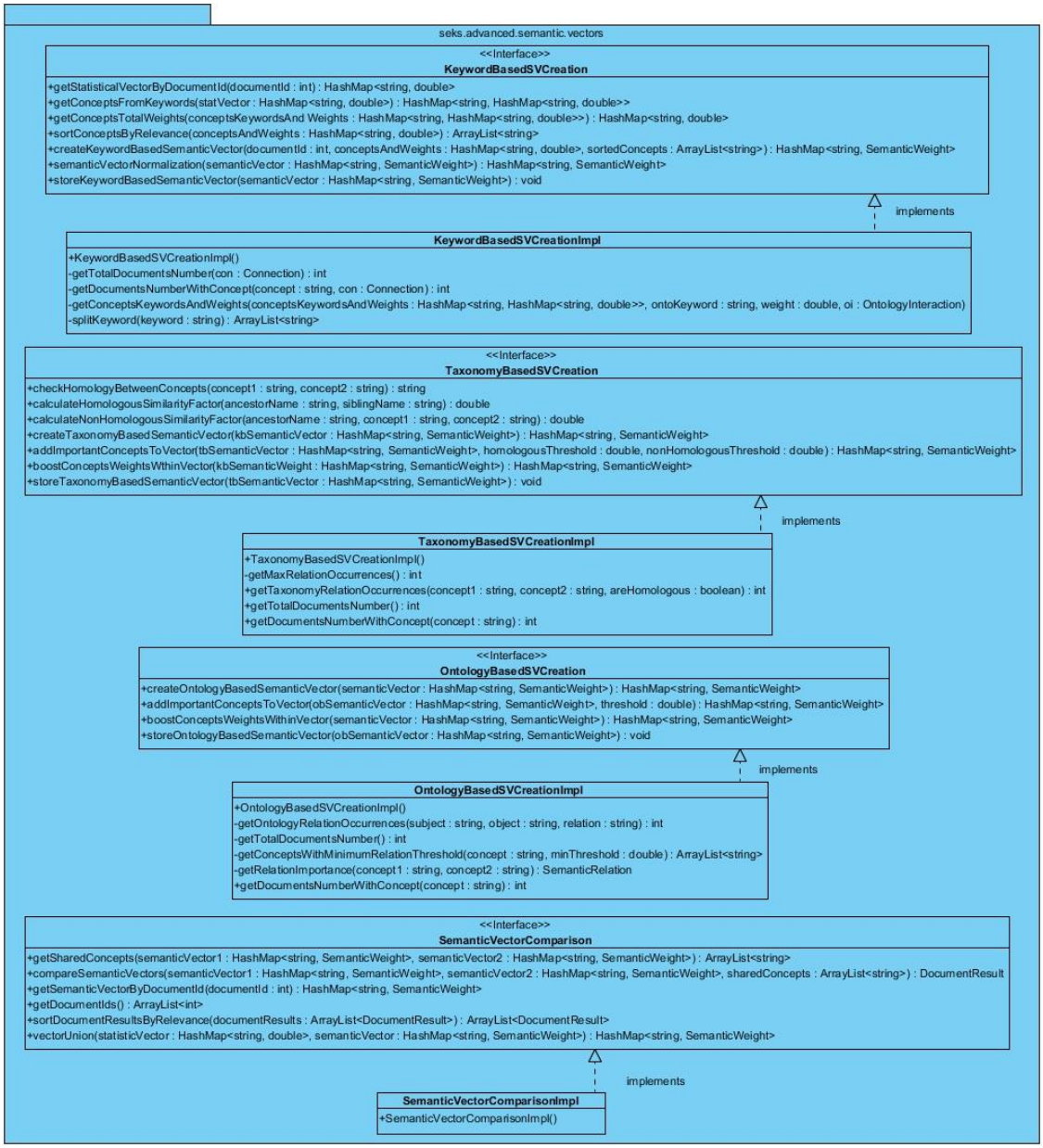


Figure 9.9. Semantic Vector Services and Document Comparison Services classes and interfaces

Package *seks.advanced.queries* (depicted in Figure 9.10) is responsible for splitting query strings into keywords, creating statistic and semantic vectors for queries, and to get all ontology keywords, used by User Interface for autocomplete purposes over the keyword search field. When a user starts typing its query in the User's Interface search field, an autocomplete mechanism is triggered so that ontology keywords that start with the letter or letters inserted by the user are shown below the search field. Users can then select the desired keyword if it exists in the ontology, saving time in query typing. If a user's keyword does not exist in the ontology, the autocomplete is disabled until the next keyword insertion.

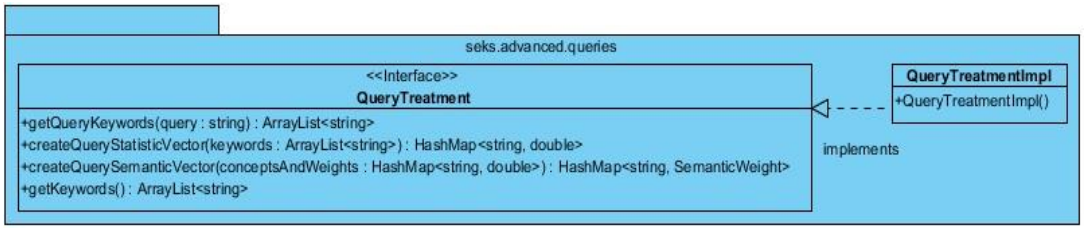


Figure 9.10. Query Treatment Services classes and interfaces

9.3.3 Interfaces

The system presents two different interfaces to the exterior. The User Interface acts like a normal search-engine web site, in order to provide a common and intuitive visual interaction with its users. The Web Services Interface is viewed more as a framework, providing functions that can be used by other systems, if their developers wish to use SENSE functionalities. Interfaces' server-side Java classes and Servlets are also presented using UCD's.

9.3.4 Web Services Interface

Classes provided through the Web Services Interface, represented in Figure 9.11, mirror SENSE Advanced Services' classes. *DocumentSemanticVectorsService.java* and *QuerySemanticVectorsService.java* classes provide all mechanisms for semantic knowledge source and query indexation, respectively. *VectorComparisonService.java* offers access to all knowledge source comparison and result ranking capabilities provided by SENSE. Finally, the *ClientSupportService.java* class provides access to the JavaScript Object Notation (JSON) visual ontology tree, used for supporting drag-and-drop ontology concept search, and to the ontology keywords, for instance, for autocomplete purposes. SENSE Web Services are developed with JAX-WS Framework, which automatically generates the Web Service Definition Language (WSDL) files needed for Web Services operation.

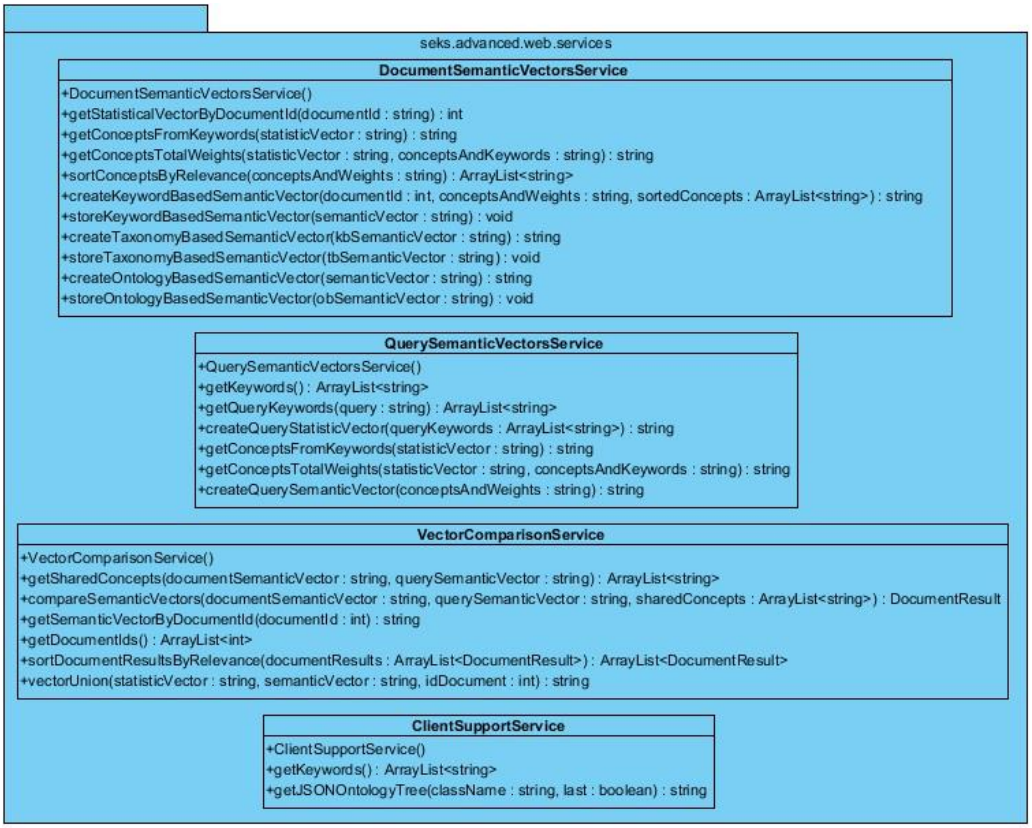


Figure 9.11. Web Services Interface classes

9.3.5 User Interface

The User Interface classes are Java Servlet classes, as shown in Figure 9.12. Such classes only respond to client-side requests, and react to those requests, sending responses. The exception in this case is *InitDocumentIndexationServlet.java* class that is called directly from server-side, to initiate scheduled knowledge source indexation processes. *KeywordSearchServlet.java* manages users' queries and starts the knowledge source search process. The *UploadFileServlet.java* class interacts with the knowledge source repository in order to upload users' documents. *GetKeywordsServlet.java* is used to fetch all ontology keywords, for autocomplete purposes, as previously mentioned. *ConceptsTreeServlet.java* and *ConceptSearchServlet.java* classes provide support for ontology concept-based search by creating the JSON visual ontology tree, and for performing subsequent search based on ontology concepts chosen from the JSON tree, respectively.

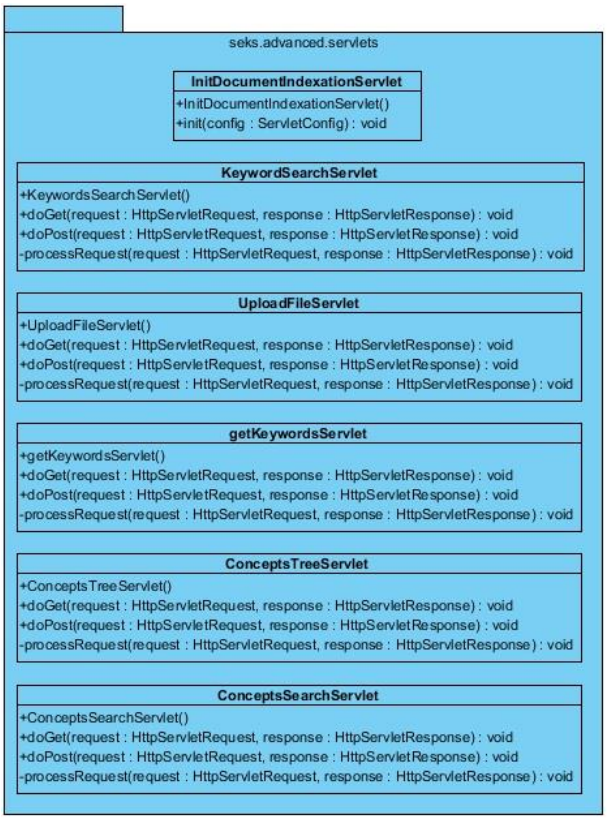


Figure 9.12. User Interface classes

9.4 Annex D – RapidMiner Workflows

This section illustrates the workflows which were designed in RapidMiner application. Those are related with processing documents (Figure 9.13), association rule mining (Figure 9.14) and clustering & evaluation (Figure 9.15).

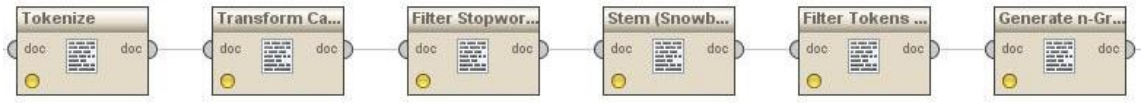


Figure 9.13. Process documents

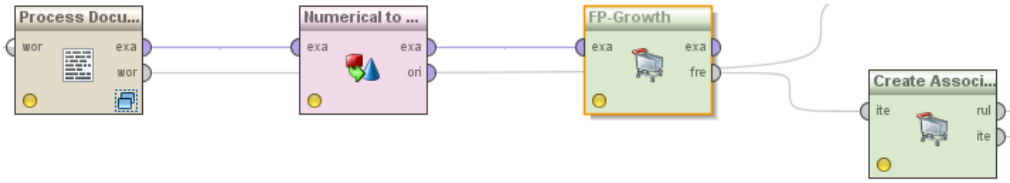


Figure 9.14. Association Rule mining

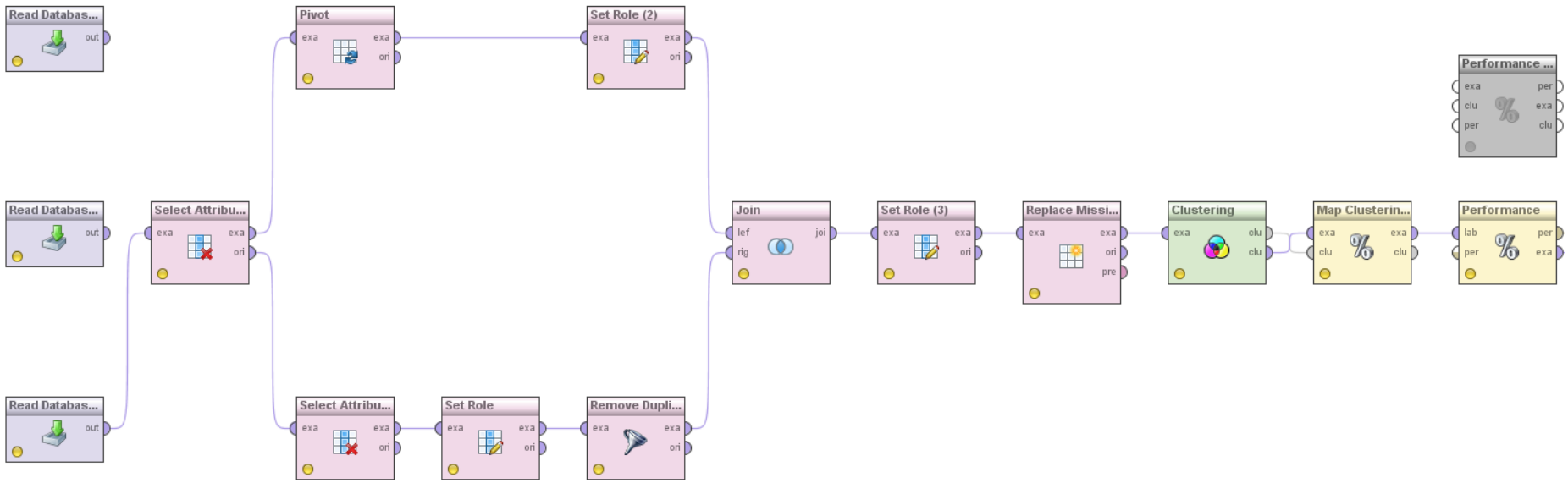


Figure 9.15. Clustering & Evaluation

9.5 Annex E – Porter stemming algorithm

A *consonant* in a word is a letter other than A, E, I, O or U, and other than Y preceded by a consonant. (The fact that the term 'consonant' is defined to some extent in terms of itself does not make it ambiguous.) So in TOY the consonants are T and Y, and in SYZYGY they are S, Z and G. If a letter is not a consonant it is a vowel.

A consonant will be denoted by c, a vowel by v. A list ccc... of length greater than 0 will be denoted by C, and a list vvv... of length greater than 0 will be denoted by V. Any word, or part of a word, therefore has one of the four forms:

CVCV ... C

CVCV ... V

VCVC ... C

VCVC ... V

These may all be represented by the single form

[C]VCVC ... [V]

where the square brackets denote arbitrary presence of their contents. Using $(VC)^m$ to denote VC repeated m times, this may again be written as

$[C](VC)^m[V]$.

m will be called the *measure* of any word or word part when represented in this form. The case m = 0 covers the null word. Here are some examples:

m=0 TR, EE, TREE, Y, BY.

m=1 TROUBLE, OATS, TREES, IVY.

m=2 TROUBLES, PRIVATE, OATEN, ORRERY.

The *rules* for removing a suffix will be given in the form

(condition) S1 -> S2

This means that if a word ends with the suffix S1, and the stem before S1 satisfies the given condition, S1 is replaced by S2. The condition is usually given in terms of m, e.g.

(m > 1) EMENT ->

Here S1 is 'EMENT' and S2 is null. This would map REPLACEMENT to REPLAC, since REPLAC is a word part for which m = 2.

The 'condition' part may also contain the following:

*S - the stem ends with S (and similarly for the other letters).

v - the stem contains a vowel.

*d - the stem ends with a double consonant (e.g. -TT, -SS).

*o - the stem ends cvc, where the second c is not W, X or Y (e.g. -WIL, -HOP).

And the condition part may also contain expressions with and, or and not, so that

(m>1 and (*S or *T))

tests for a stem with m>1 ending in S or T, while

(*d and not (*L or *S or *Z))

tests for a stem ending with a double consonant other than L, S or Z. Elaborate conditions like this are required only rarely.

In a set of rules written beneath each other, only one is obeyed, and this will be the one with the longest matching S1 for the given word. For example, with

SSES -> SS

IES -> I

SS -> SS

S ->

(here the conditions are all null) CARESSES maps to CARESS since SSES is the longest match for S1. Equally CARESS maps to CARESS (S1='SS') and CARES to CARE (S1='S').

In the rules below, examples of their application, successful or otherwise, are given on the right in lower case. The algorithm now follows:

Step 1a

SSES -> SS	caresses -> caress
IES -> I	ponies -> poni
	ties -> ti
SS -> SS	caress -> caress
S ->	cats -> cat

Step 1b

(m>0) EED -> EE	feed -> feed

	agreed -> agree
(*v*) ED ->	plastered -> plaster
	bled -> bled
(*v*) ING ->	motoring -> motor
	sing -> sing

If the second or third of the rules in Step 1b is successful, the following is done:

AT -> ATE	conflat(ed) -> conflate
BL -> BLE	troubl(ed) -> trouble
IZ -> IZE	siz(ed) -> size
(*d and not (*L or *S or *Z)) -> single letter	hopp(ing) -> hop
	tann(ed) -> tan
	fall(ing) -> fall
	hiss(ing) -> hiss
	fizz(ed) -> fizz
(m=1 and *o) -> E	fail(ing) -> fail
	fil(ing) -> file

The rule to map to a single letter causes the removal of one of the double letter pair. The -E is put back on -AT, -BL and -IZ, so that the suffixes -ATE, -BLE and -IZE can be recognised later. This E may be removed in step 4.

Step 1c

(*v*) Y -> I happy -> happi

sky -> sky

Step 1 deals with plurals and past participles. The subsequent steps are much more straightforward.

Step 2

(m>0) ATIONAL -> ATE	relational -> relate
(m>0) TIONAL -> TION	conditional -> condition
	rational -> rational
(m>0) ENCI -> ENCE	valenci -> valence

(m>0) ANCI -> ANCE	hesitanci -> hesitance
(m>0) IZER -> IZE	digitizer -> digitize
(m>0) ABLI -> ABLE	conformabli -> conformable
(m>0) ALLI -> AL	radicalli -> radical
(m>0) ENTLI -> ENT	differentli -> different
(m>0) ELI -> E	vileli -> vile
(m>0) OUSLI -> OUS	analogousli -> analogous
(m>0) IZATION -> IZE	vietnamization -> vietnamize
(m>0) ATION -> ATE	predication -> predicate
(m>0) ATOR -> ATE	operator -> operate
(m>0) ALISM -> AL	feudalism -> feudal
(m>0) IVENESS -> IVE	decisiveness -> decisive
(m>0) FULNESS -> FUL	hopefulness -> hopeful
(m>0) OUSNESS -> OUS	callousness -> callous
(m>0) ALITI -> AL	formaliti -> formal
(m>0) IVITI -> IVE	sensitiviti -> sensitive
(m>0) BILITI -> BLE	sensibiliti -> sensible

The test for the string S1 can be made fast by doing a program switch on the penultimate letter of the word being tested. This gives a fairly even breakdown of the possible values of the string S1. It will be seen in fact that the S1-strings in step 2 are presented here in the alphabetical order of their penultimate letter. Similar techniques may be applied in the other steps.

Step 3

(m>0) ICATE -> IC	triplicate -> triplic
(m>0) ATIVE ->	formative -> form
(m>0) ALIZE -> AL	formalize -> formal
(m>0) ICITI -> IC	electriciti -> electric
(m>0) ICAL -> IC	electrical -> electric
(m>0) FUL ->	hopeful -> hope
(m>0) NESS ->	goodness -> good

Step 4

(m>1) AL ->	revival -> reviv
(m>1) ANCE ->	allowance -> allow
(m>1) ENCE ->	inference -> infer
(m>1) ER ->	airliner -> airlin
(m>1) IC ->	gyroscopic -> gyroscop
(m>1) ABLE ->	adjustable -> adjust
(m>1) IBLE ->	defensible -> defens
(m>1) ANT ->	irritant -> irrit
(m>1) EMENT ->	replacement -> replac
(m>1) MENT ->	adjustment -> adjust
(m>1) ENT ->	dependent -> depend
(m>1 and (*S or *T)) ION ->	adoption -> adopt
(m>1) OU ->	homologou -> homolog
(m>1) ISM ->	communism -> commun
(m>1) ATE ->	activate -> activ
(m>1) ITI ->	angulariti -> angular
(m>1) OUS ->	homologous -> homolog
(m>1) IVE ->	effective -> effect
(m>1) IZE ->	bowdlerize -> bowdler

The suffixes are now removed. All that remains is a little tidying up.

Step 5a

(m>1) E -> probate -> probat

rate -> rate

(m=1 and not *o) E -> cease -> ceas

Step 5b

(m > 1 and *d and *L) -> single letter controll -> control

roll -> roll