



UNIVERSITÀ DEGLI STUDI
DI SALERNO


*Ministero dell'Istruzione dell'Università e Ricerca*

Università degli Studi di Salerno
Dottorato di Ricerca in Informatica
X Ciclo

Ph.D. Thesis
# Fuzzy Concept Analysis for Semantic Knowledge Extraction

Carmen De Maio

November 2011

Coordinator:
Prof. Giuseppe Persiano

Supervisor:
Prof. Vincenzo Loia
Ph.D. Giuseppe Fenza

The majority of this thesis is based on certain parts of the following publications. As a coauthors, I was involved actively in the research, planning and writing these papers.

International Conferences:

- C. De Maio, G. Fenza, V. Loia, S. Senatore, “Towards an automatic Fuzzy Ontology generation” FUZZ-IEEE 2009, ICC Jeju, Jeju Island, Korea, 20-24 August, 2009.
- C. De Maio, G. Fenza, V. Loia, S. Senatore, “Ontology-based knowledge structuring: an application on RSS Feeds”, 2nd International Conference on Human System Interaction, Catania, Italy, May 21-23, 2009.
- C. De Maio, G. Fenza, V. Loia, S. Senatore, "A multi facet representation of a fuzzy ontology population" in Computational Intelligence Approaches for Ontology-based Knowledge Discovery (CIAO) Workshop in The 2009 IEEE / WIC / ACM International Conferences on Web Intelligence (WI'09) and Intelligent Agent Technology (IAT'09), pp. 401-404, 15-18 September 2009, Milan, Italy.
- C. De Maio, G. Fenza, M.Gaeta, V. Loia, F. Orciuoli, S. Senatore “RSS-generated contents through personalizing e-learning Agents” in 9th International Conference on Intelligent Systems Design and Applications (ISDA ‘09) pp. 49-54, November 30 – December 2, Pisa, Italy.
- C. De Maio, G. Fenza, V. Loia, M. Gaeta, F.Orciuoli, “Enhancing Context Sensitivity of Geo Web Resources Discovery by means of Fuzzy Cognitive Maps”, 23 th Third International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems, IEA-AIE 2010, 1- 4 June, 2010, Còrdoba, Spain.
- C. De Maio, G. Fenza, V. Loia, S.Senatore, F.Orciuoli, “An enhanced approach to improve enterprise competency management”, 2010 IEEE World Congress on Computational Intelligence, 18 - 23 July, Barcelona, Spain.
- C. De Maio, G. Fenza, V. Loia, S.Senatore, “OWL-FC Ontology Web Language for Fuzzy Control”, 2010 IEEE World Congress on Computational Intelligence, 18 - 23 July, Barcelona, Spain.
- C.De Maio, G.Fenza, M.C.Gallo, V.Loia, R.Linciano, A.Morrone, “Fuzzy Knowledge Approach to Automatic Disease Diagnosis”, 2011 IEEE International Conference on Fuzzy Systems, June 27-30, 2011- Taipei, Taiwan.

Journal

- C. De Maio, G. Fenza, V. Loia, S. Senatore “Knowledge Structuring to support Facet-Based Ontology Visualization”, International Journal of Intelligent Systems, Special Issue: New Trends for Ontology-Based Knowledge Discovery Volume 25, Issue 12, pages 1249–1264, December 2010
- C. De Maio, G. Fenza, M. Gaeta, V. Loia, F. Orciuoli, A knowledge-based framework for emergency DSS, Knowledge-Based Systems, Volume 24, Issue 8,

December 2011, Pages 1372-1379, ISSN 0950-7051, 10.1016/j.knosys.2011.06.011.

- C. De Maio, G. Fenza, D. Furno, V. Loia, S. Senatore, "OWL-FC: an upper ontology for semantic modeling of Fuzzy Control", Soft Computing, 2011, doi: 10.1007/s00500-011-0790-4
- C. De Maio, G. Fenza, M. Gaeta, V. Loia, F. Orciuoli, S. Senatore, RSS-based e-learning recommendations exploiting fuzzy FCA for Knowledge Modeling, Applied Soft Computing, available online 24 September 2011, ISSN 1568-4946, 10.1016/j.asoc.2011.09.004.
- C. De Maio, G. Fenza, V. Loia, S. Senatore, Hierarchical web resources retrieval by exploiting Fuzzy Formal Concept Analysis. Information Processing and Management, available online 26 May 2011, ISSN 0306-4573, doi:10.1016/j.ipm.2011.04.003
- C. De Maio, G. Fenza, M. Gallo, V. Loia, S. Senatore, Fuzzy Concept Analysis for automatic semantic annotation. Submitted to IEEE Computational Intelligence Magazine.

*"There is a driving force more powerful than the steam, the electricity and the atomic energy: the will."*
*(Albert Einstein)*

*"Computers are incredibly fast, accurate and stupid; humans are incredibly slow, inaccurate and brilliant; together they are powerful beyond imagination."*
*(Albert Einstein)*

# Acknowledgements

I would like to express my gratitude to the people that have contributed to the accomplishment of this thesis.

First of all I would like to thank my primary supervisor Professor Vincenzo Loia for his kind guidance, precious hints and the enthusiasm for the research that he has transmit-ted to me, during this time as Ph.D. student. In the same way, I sincerely appreciate doctor Giuseppe Fenza that gave me consistent support and assistance in the research activities.

I would also like to convey my thanks to the other members of CORISA group: Acampora Giovanni, Furno Domenico, Marzullo Giovanna, Maria Cristina Gallo, Veniero Mario, Sabrina Senatore.

Furthermore, I would like to thank my other friends: Lucia, Giovanni, Dora, Virginia, Jessica, Nunzia, Peppe and many other friends that fill my life every day.

I would also like to thank my parents, my sister and everyone believe in me. I hope that this Ph.D. thesis can represent the beginning of a humble and exciting research career and I thank everyone will read these pages with passion and interests.

# Abstract

Availability of controlled vocabularies, ontologies, and so on is enabling feature to provide some added values in terms of knowledge management. Nevertheless, the design, maintenance and construction of domain ontologies are a human intensive and time consuming task. The Knowledge Extraction consists of automatic techniques aimed to identify and to define relevant concepts and relations of the domain of interest by analyzing structured (relational databases, XML) and unstructured (text, documents, images) sources.

Specifically, methodology for knowledge extraction defined in this research work is aimed at enabling automatic ontology/taxonomy construction from existing resources in order to obtain useful information. For instance, the experimental results take into account data produced with Web 2.0 tools (e.g., RSS-Feed, Enterprise Wiki, Corporate Blog, etc.), text documents, and so on. Final results of Knowledge Extraction methodology are taxonomies or ontologies represented in a machine oriented manner by means of semantic web technologies, such as: RDFS, OWL and SKOS.

The resulting knowledge models have been applied to different goals.

On the one hand, the methodology has been applied in order to extract ontologies and taxonomies and to semantically annotate text. On the other hand, the resulting ontologies and taxonomies are exploited in order to enhance information retrieval performance and to categorize incoming data and to provide an easy way to find interesting resources (such as faceted browsing). Specifically, following objectives have been addressed in this research work:

- *Ontology/Taxonomy Extraction*: that concerns to automatic extraction of hierarchical conceptualizations (i.e., taxonomies) and relations expressed by means typical description logic constructs (i.e., ontologies).
- *Information Retrieval*: definition of a technique to perform concept-based the retrieval of information according to the user queries.
- *Faceted Browsing*: in order to automatically provide faceted browsing capabilities according to the categorization of the extracted contents.
- *Semantic Annotation:* definition of a text analysis process, aimed to automatically annotate subjects and predicates identified.

The experimental results have been obtained in some application domains: e-learning, enterprise human resource management, clinical decision support system.

Future challenges go in the following directions: investigate approaches to support ontology alignment and merging applied to knowledge management.



**Keywords**

Knowledge Extraction, Ontology Extraction, Ontology, OWL, Fuzzy Relational Concept Analysis, Fuzzy Formal Concept Analysis, Semantic Web, Clinical Decision Support System, Faceted-Browsing, Information Retrieval, Semantic Annotation.

# Contents


Acknowledgements ..... i
Abstract ..... iii
Contents ..... v
List of Figures ..... ix
List of Tables ..... xiii
What this thesis is all about? ..... 1
1.1 Objectives ..... 1
1.2 State of the art ..... 2
1.3 Thesis Outline ..... 4
Part I: Theoretical Background ..... 5
Fuzzy Theory: Fuzzy Formal Concept Analysis & Fuzzy Relational Concept Analysis ..... 7
2.1 Formal Concept Analysis ..... 7
2.2 Fuzzy Set Theory ..... 9
2.3 Fuzzy Formal Concept Analysis Theory ..... 10
2.4 Fuzzy Relational Concept Analysis ..... 12
2.5 Algorithms for Generating Concept Lattices ..... 16
Semantic Technologies ..... 19
3.1 Semantic Models (Taxonomies and Ontologies) ..... 20
3.2 Semantic Web Wedding Cake ..... 23
*3.2.1 XML, Namespace, XMLSchema ..... 23*
*3.2.2 Resource Description Framework (RDF) & RDF-Schema ..... 24*
*3.2.3 Ontology & Ontology Web Language ..... 24*
3.3 Semantic Web Vocabularies ..... 26
*3.3.1 Dublin Core Metadata Initiative (DCMI). ..... 26*
*3.3.2 Friend Of A Friend (FOAF) ..... 26*
*3.3.3 Semantically-Interlinked Online Communities (SIOC) ..... 27*

*3.3.4 Simple Knowledge Organization System (SKOS) ..... 29*
3.4 Conclusion ..... 29
Part II: Methodologies & Applications ..... 31
Methodology for Knowledge Extraction ..... 33
4.1 Knowledge Extraction: General Framework ..... 34
*4.1.1 Natural Language Processing Pipe ..... 35*
*4.1.2 Content Analyzer ..... 38*
4.1.2.1 Vectorization ..... 38
4.1.2.2 Concept Data Analysis ..... 40
4.1.2.3 Concepts Labeling ..... 41
*4.1.3 Semantic Modeling ..... 42*
4.2 Knowledge Extraction: Research Objectives ..... 42
*4.2.1 Ontology & Taxonomy Extraction ..... 42*
*4.2.2 Semantic Annotation ..... 46*
*4.2.3 Information Retrieval ..... 48*
*4.2.4 Faceted browsing ..... 52*
Part III: Case Studies ..... 55
Automatic Faceted Browsing and Ontology-based Retrieval of web resources ..... 57
5.1 Faceted browsing of web resources ..... 57
5.2 System validation and experimental results ..... 59
*5.2.1 Analysis of the extracted ontology structure ..... 59*
*5.2.2 Analysis of the lattice consistency and retrieval performance ..... 62*
5.3 Conclusions ..... 65
Taxonomy Extraction applied to Enterprise Competency Management ..... 67
6.1 Competency Management: Advantages and Limitations ..... 68
6.2 Workflow ..... 69
*6.2.1 UGC Semantic Modeling ..... 69*
*6.2.2 Taxonomy Extraction ..... 71*
*6.2.3 Competency-Related storage ..... 72*
*6.2.4 Information Retrieval ..... 73*
6.3 Team Building: a sample Scenario ..... 75
6.4 Conclusion ..... 75

Automatic Textual Resources Annotation ..... 77
7.1 A context-dependent application: textual resources annotation ..... 78
7.2 Experimental Evaluation ..... 85
7.3 Ontology structure evaluation ..... 85
7.4 Text categorization performances ..... 87
7.5 Related Work ..... 88
7.6 Conclusion ..... 92
Ontology based information retrieval applied to e-Learning Recommendations ..... 93
8.1 Intelligent Web Teacher ..... 96
8.2 A sample scenario ..... 97
8.3 Experimental Results ..... 99
8.4 Related Works ..... 103
8.5 Conclusions ..... 105
Ontology based information retrieval applied to Disease Diagnosis ..... 107
9.1 Knowledge Layer ..... 108
*9.1.1 Technologies and Standards ..... 108*
*9.1.2 Controlled Vocabularies & Taxonomies ..... 109*
*9.1.3 Medical Disease Ontology ..... 110*
9.2 Medical Diagnosis Methodology ..... 112
*9.2.1 Mathematical model to support preliminary diagnosis ..... 112*
*9.2.2 Identification of candidate disease ..... 113*
9.3 Case Study ..... 113
*9.3.1 Disease Catalogue Browsing. ..... 113*
*9.3.2 Preliminary medical diagnosis ..... 113*
*9.3.3 Faceted search ..... 116*
9.4 Related Works ..... 117
9.5 Conclusion ..... 118
Conclusion and Future Work ..... 119
10.1 Summary ..... 119
10.2 Future Work ..... 119
Bibliography ..... 121

# List of Figures

Figure 1. Formal Concept Analysis ..... 8
Figure 2. Fuzzy Formal Concept Analysis..... 11
Figure 3. The Fuzzy Concept Lattice resulting from context of Bacteria..... 14
Figure 4. The Fuzzy Context and the relative Lattice resulting of Antibiotic ..... 14
Figure 5. Fuzzy Formal Context obtained using RCA..... 15
Figure 6. Final concept lattice obtained using RCA ..... 16
Figure 7. The central-role of semantic technologies. ..... 19
Figure 8. Spectrum of Knowledge Representation and Reasoning Capabilities..... 21
Figure 9. Semantic Web Wedding Cake (From Berners-Lee, XML 2000 Conference)... 23
Figure 10. Logical View of Overall Process..... 34
Figure 11. Knowledge Extraction: NLP Pipe. ..... 35
Figure 12.Knowledge Extraction: Content Analyzer. ..... 38
Figure 13. Filtering of features in the Vectorization phase. ..... 40
Figure 14. Ontology & Taxonomy Extraction Logical View Process ..... 43
Figure 15. Ontology Extraction from Fuzzy Formal Analysis ..... 44
Figure 16. Dictionary Mapping Step..... 44
Figure 17. Relation Mapping Step ..... 45
Figure 18. Class Mapping Step ..... 46
Figure 19. Individual Mapping Step ..... 46
Figure 20. Semantic Annotation Logical View Process ..... 47
Figure 21. Workflow of Training activity. ..... 50
Figure 22. Workflow of Query Processing activity ..... 51
Figure 23. Faceted Browsing Logical View Process ..... 51
Figure 24. Ontology tree generated by OWL-based lattice representation..... 53
Figure 25. Ontology representation through facet-based Web Interface ..... 58
Figure 26. Inheritance Richness tendency by varying the threshold value..... 59

Figure 27. Fuzzy contexts with relative generated lattices (by varying a threshold T) and the computed inheritance richness (ir) values ..... 61

Figure 28. Example of concepts in the generated ontology and the corresponding OpenLearn categories ..... 62

Figure 29. Micro-averaging precision/recall by varying the ir values ..... 64

Figure 30. Workflow ..... 70

Figure 31. Uniform semantic envelop associated to the Web 2.0 resources. ..... 71

Figure 32. Concept Matching and a list of employees ranked according to their rating-based competency ..... 73

Figure 33. Example of map semantic annotation ..... 78

Figure 34. Application screenshot ..... 79

Figure 35. Domain Lattice ..... 80

Figure 36. Range Lattice ..... 80

Figure 37 - Relations lattice ..... 81

Figure 38. RCA lattice ..... 82

Figure 39 - Annotation results with Zoom equal to 70% and Hypernym level is equal to 084

Figure 40. Annotation results with Zoom equal to 70% and Hypernym level equal to 3 . 84

Figure 41. Annotation results with Zoom equal to 100% and Hypernym level equal to 385

Figure 42 Ontology quality evaluation ..... 87

Figure 43 - M-Ontomat Annotizer ..... 89

Figure 44. Classification of semantic annotations approaches ..... 90

Figure 45 - KIM tool ..... 91

Figure 46. . A sample of RSS channel ..... 94

Figure 47. The IWT e-Learning Experiences definition process ..... 97

Figure 48. Preparing an RSS-based learning experience in IWT ..... 98

Figure 49. Executing an RSS-based learning experience in IWT ..... 99

Figure 50. Precision/recall evaluation, given the queries ..... 102

Figure 51. Comparative query/answering performance with an incremental query ..... 103

Figure 52. A sketch of Medical Disease Ontology: Disease, Complication and Symptom/Sign. ..... 111

Figure 53. A sketch of some classes of *Medical Disease Ontology*: *Disease* and *Clinical test*. ..... 111

Figure 54. Selection of clinical manifestation by mean textual search (with autocomplete) or categories exploration ..... 114

Figure 55. Preliminary diagnosis retrieved results: (a) correlation degree, (b) links to motivations of results ..... 115

Figure 56. Example of result motivation interface ..... 115

Figure 57. Performance evaluation on Precision and AUP ..... 116

Figure 58. An example of faceted search of diseases ..... 117

# List of Tables

Table 1. The Fuzzy Concept Lattice resulting from context of Bacteria ..... 13
Table 2. The Fuzzy Relation “Resist To” between bacteria and antibiotics ..... 15
Table 3. Properties of algorithms constructing concept lattices ..... 17
Table 4. Input/Output of Knowledge Extraction process..... 33
Table 5. Input/Output of NLP Pipe phase..... 35
Table 6. Input/Output of Content Analyzer phase. ..... 38
Table 7. Input/Output of Vectorization phase..... 38
Table 8: Input/Output of Concept Data Analysis micro-phase..... 40
Table 9: Input/Output of Concepts Labeling micro-phase..... 41
Table 10: Input/Output of Semantic Modeling phase..... 42
Table 11. Summary of training datasets and queries. ..... 63
Table 12. Domain context..... 79
Table 13. Range context ..... 80
Table 14. Relation context ..... 81
Table 15. RCA context ..... 82
Table 16 – Comparison of the proposed system with AlchemyGrid annotation ..... 88
Table 17. System query/answering response time ..... 100
Table 18. OpenLearn categories vs. some FFCA-based specializations..... 101

# Chapter 1

## What this thesis is all about?

The main objective of this research work is to define a methodology for automatic Knowledge Extraction (i.e., Ontologies, Taxonomies, Semantic Annotation) by taking into account content of collection of web resources. This methodology has been applied to support activities of ontologies and taxonomies extraction, semantic annotation, information retrieval and faceted browsing.

From the scientific point of view, a distinctive feature of this research work is related the definition and the application of a fuzzy mathematical model to hierarchical conceptualize text contents. Specifically, *Formal Concept Analysis* theory and variants (i.e., *Fuzzy Formal Concept Analysis* and *Fuzzy Relational Concept Analysis*) are exploited for structuring the elicited knowledge, viz. concepts and relations embedded in the content of the resources.

The contributions of this thesis work are: definition of general framework to extract knowledge models; application of the general framework to support different research objectives.

### 1.1 Objectives

The general framework defined in this research work is composed of following main phases:

- *Natural Language Processing Pipe,* that performs bag of words extraction from textual input*;*
- Content Analyzer, that performs
    - *Vectorization* that carries out feature set and term weighting of input resources;
    - *Concept Data Analysis* aimed at extracting concepts hierarchies represented by means of mathematical model;
    - *Concepts Labeling* which assigns significant labels to each concept in the extracted hierarchy.
- *Semantic Modeling,* that accomplishes activities of machine oriented representation of the extracted concepts schema.

The framework described has been applied to specific research objectives evaluated in different case studies. We distinguish the following research objectives achieved:

- *Ontology/Taxonomy Extraction*: the general framework has been applied to automatically extract hierarchical conceptualizations (i.e., taxonomies) and relations

expressed by means typical description logic constructs (i.e., ontologies). The main case studies examined were aimed at the conceptualization of content included in RSS feeds in the context of e-learning available from the web directory (e.g., OpenLearn[1], Merlot[2], etc.).

- *Information Retrieval*: definition of a technique, based on results taken from the general framework (i.e., Fuzzy Formal Concept Analysis), that addresses the retrieval of information in relation to a user query. In the experiments conducted the approach has produced good results in terms of Precision and Recall (metrics typical of the context of Information Retrieval). In particular, case studies are: a search engine of RSS Feed in the context of e-learning, and a medical diagnosis system that supports disease discovery starting from symptoms and signs.
- *Facet Browsing*: the general framework has been exploited in order to automatically provide faceted browsing capabilities of the extracted resource categories. The main case studies examined in this context are: faceted browsing of web resources and User Generated Content (UGC) categorization to support the enterprise competency management.
- *Semantic Annotation:* the general framework has been extended to introduce Fuzzy Relational Concept Analysis technique (Fuzzy RCA) aimed to automatic annotation of subjects and predicates identified in it. The case studies examined are related to the annotation plain text, but the framework is applicable to other types of multimedia resources.

## 1.2 State of the art

In the past years, the World Wide Web has represented the era in which digital content explodes via Internet. The terms web knowledge discovery and mining greatly describe the old generation of Web activities and leave the scene to the ”Web 2.0” and the successive versioning. Semantic technologies enable the concept representation, the knowledge access and sharing and then, the inter-communication among heterogeneous web applications.

Nowadays, ontologies have acquired a strategic value in many fields, such as knowledge management, information retrieval, information integration, etc. Specifically, ontologies play an important role in supporting automated processes to access information and are at the core of new strategies for the development of knowledge-based systems. Nevertheless, ontology engineering is human-driven, labor-intensive and time-consuming tasks.

Last trend emphasizes the role of approaches aimed at automatic learning ontology from documents. Many automatic and semi-automatic methods apply text mining and machine learning techniques in order to allow ontology extraction. Particularly, many tools have been designed for knowledge structuring in specific domains, for instance, through automatic discovery of taxonomic and non-taxonomic relationships from domain data. OntoGen [1], OntoLT [2], OntoLearn [3] and OntoEdit [4] represent well-known tools for knowledge struc-

[1] http://openlearn.open.ac.uk/

[2] http://www.merlot.org/merlot/index.htm

turing and ontology engineering activities; ASIUM [5] supports the semantic acquisition of knowledge from textual resources and the automatic ontology building. Again, Text2Onto [6] is a framework for ontology learning from textual resources, able to automatically create ontologies from a corpus of documents within a certain domain.

Many methodologies and techniques have been reported in the literature regarding ontology building [8], [9], [10]; in particular the methodologies presented by Grüninger and Fox [10] deal with ontologies using first-order-logic languages, while the method in [9] starts from the identification of the ontology goals and the needs for the domain knowledge acquisition.

Some approaches instead, exploit techniques based on Formal Concept Analysis (FCA) theory for knowledge structuring and ontology building: in [11] a method has been adapted to maintain a concept map for the reuse of knowledge. In [12], instead, FCA is used in Information Retrieval applications to improve the search engine capabilities in the query/answering activities.

Although the ontologies widely contribute to design the domain knowledge modeling, sometimes their expressiveness is not satisfactory to provide a coherent representation of imprecise information of the real world. In the real word, customers or end users prefer to get messages in linguistic expressions rather than in numeric values. Many research works are aimed at defining tolerance to imprecision, uncertainty and vagueness in the process of ontology generation. A possible solution to deal with this issue is to use fuzzy techniques to relax rigid constraints, managing uncertainty in the relationship and conceptual information.

Quite a few approaches integrate Formal Concept Analysis and Fuzzy Logic theory for ontology-based approaches [13], [14]. Pollandt [15] introduces the L-Fuzzy Context, as an attempt to combine fuzzy logic with FCA, where linguistic variables are used to represent ambiguities in the context. Because a human support is required to define the linguistic variables, the approach seems to be not practicable for dealing with large document sets. Moreover, the fuzzy concept lattice generated from the L-fuzzy context tends to generate a larger number of concepts than the traditional one.

In [16], a framework named FOGA achieves the automatic generation of a fuzzy ontology from data. Like the proposed approach, FOGA exploits the FCA theory for building a hierarchy structure of ontology classes, but it uses fuzzy conceptual clustering to cluster formal concepts into conceptual clusters. The proposed approach instead, besides building an OWL ontology, through the hierarchical structure derived from the FCA theory, achieves a classification system of the collected resources. Then, the exploration tree allows users to navigate across the web resources and retrieve specific data derived from the conceptualization of the extracted knowledge.

An interesting proposal is presented in [17], where the representation of uncertainty is defined by a possibility theory-inspired view of FCA. New operators have been introduced for the description of the different possible relations between a set of objects and a set of properties, leading to consider new Galois connections that give birth to the notion of concept [17].

This thesis work exploits fuzzy FCA in order to consider a degree of interrelations (i.e. an approximate subsumption) between linked concepts. The fuzzy lattice reveals knowledge-based, hierarchical dependences among concepts, emphasizing the taxonomic nature of the structure. Besides, it naturally provides a classification of collected data.

## 1.3 Thesis Outline

The thesis work is described according to the following structure.

***Part I: Theoretical Background***

***Chapter 2*** *"Fuzzy Theory: Fuzzy Formal Concept Analysis & Fuzzy Relational Concept Analysis"* – introduces the mathematical model of Formal Concept Analysis. Furthermore, this chapter defines fuzzy extension of Formal Concept Analysis and Relational Concept Analysis that have been applied in this thesis work in order to manage data uncertainty and conceptualization.

***Chapter 3*** *"Semantic Technologies"* – describes the basics of the semantic technologies and the semantic web standard exploited in this research work.

***Part II: Methodologies & Applications.***

***Chapter 4*** *"Methodology for Knowledge Extraction"* – defines the general framework for knowledge extraction. Furthermore, the general framework has been extended and applied in order to support specific research objectives.

***Part III: Case Studies.*** The achieved research objectives have been evaluated in different application domains and case studies**.**

***Chapter 5*** *"Automatic Faceted Browsing and Ontology-based Retrieval of web resources"* – presents the core methodology applied to ontology extraction and automatically faceted browsing that supports data organization and visualization and provides a friendly navigation model.

***Chapter 6 "****Taxonomy Extraction applied to Enterprise Competency Management"* – describes a framework for the dynamic updating of employees' competencies profiles by analyzing and monitoring collaborative activities executed through Enterprise 2.0 tools.

***Chapter 7*** *"Automatic Textual Resources Annotation"* – provides a high level design of a framework aimed at generating automatic semantic annotation of web resources. Specifically, the methodology has been applied to the text annotation.

***Chapter 8*** *"Ontology based information retrieval applied to e-Learning Recommendations"* – presents an approach to enrich personalized e-learning experiences with user generated content, through a contextualized RSS feeds fruition.

***Chapter 9*** *"Ontology based information retrieval applied to Disease Diagnosis"* – describes a system aimed at supporting clinical decision making that exploits semantic based modeling of medical knowledge base.

***Chapter 10*** *"Conclusion and Future Work".*

# Part I: Theoretical Background

# Chapter 2

# Fuzzy Theory: Fuzzy Formal Concept Analysis & Fuzzy Relational Concept Analysis

## 2.1 Formal Concept Analysis

*Formal Concept analysis* (FCA) [18] is a mathematical formalism allowing to derive a concept lattice from a formal context $K = (G, M, I)$. FCA has been used for a number of purposes among which knowledge modeling, acquisition, and processing lattice and ontology design, information retrieval and data mining.

In $K$, $G$ denotes a set of objects, $M$ a set of attributes, and $I$ a binary relation defined on the Cartesian product $G \times M$. In the binary table representing $I \subseteq G \times M$, the rows correspond to objects and the columns to attributes. The use of "*object"* and "*attribute"* is indicative because in many applications it may be useful to choose object-like items as formal objects and then choose their features as formal attributes. For instance in the Information Retrieval domain, documents could be considered object-like and terms considered attribute-like.

The context is often represented as a "cross table" (see Figure 1): the rows represent the formal objects and the columns are formal attributes; the relations between them are represented by the crosses. In the proposed approach, incoming resources (e.g., documents, web pages, etc.) play the role of objects and features (e.g., keywords, categories, etc.) play the role of attributes, in the matrix describing the formal context. For instance, Figure 1 shows formal context in terms of keywords and web resources, attributes and objects respectively.

**Definition 1.** *Formal Concept.* Given a context (G, M, I), for $A \subseteq G$, applying a derivation operator, $A' = \{m \in M | \forall g \in A: (g, m) \in I\}$ and for $B \subseteq M$, $B' = \{g \in G | \forall m \in B: (g, m) \in I\}$. A formal concept is identified with a pair (A, B), where $A \subseteq G$, $B \subseteq M$ such that A' = B and B' = A. A is called the extent and B is called the intent of the concept (A, B).

**Definition 2.** Given two concepts $c_1 = (A_1, B_1)$ and $c_2 = (A_2, B_2)$, then $c_1$ is a subconcept of $c_2$ (equivalently, $c_2$ is a superconcept of $c_1$) $(A_1, B_1) \leq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2$ $(\Leftrightarrow B_2 \subseteq B_1)$. The set of all concepts of a particular context, ordered in this way, forms a complete lattice.

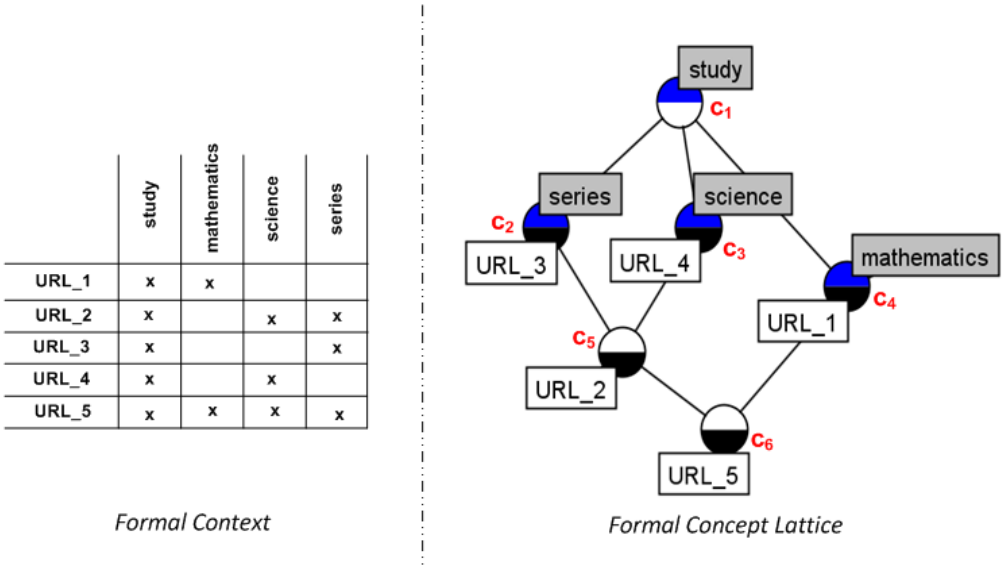



Figure 1. Formal Concept Analysis

Figure 1 shows a so-called line diagram of a concept lattice corresponding to the formal context in the left side of the figure. A concept lattice is composed by the set of concepts of a formal context and the subconcept - superconcept relation between the concepts [18]. The nodes represent formal concepts. Formal objects are noted below and formal attributes above the nodes which they label (see Figure 1).

In the Figure 1, each node can be colored in different way, according to its characteristics: a half-blue colored node represents a concept with own attributes; a half- black colored node instead, outlines the presence of own objects in the concept; finally, a half-white colored node can represent a concept with no own objects (if the white colored portion is the half below of the circle) or attributes (if the white half is up on the circle).

Let us note the names $c_1$, $c_2$, etc. represent the identifier of the concepts. We have introduced them for simplify their reference, but they are not part of the lattice representation. For instance, the node labeled with the formal attribute ‘‘series’’ and formal object ‘‘URL_3’’ shall be referred to as $c_2$. To retrieve the extension of a formal concept one needs to trace all paths which lead down from the node to collect the formal objects. In the example of Figure 1, the formal objects of $c_2$ are URL_3, URL_2 and URL_5. To retrieve the intension of a formal concept one needs to trace all paths which lead up in order to collect all the formal attributes. In the example, there is a node above $c_2$ with ‘‘study’’ as formal attributes attached. Thus $c_2$ represents the formal concept with the extension ‘‘URL_3, URL_2, URL_5’’ and the intension ‘‘study, series’’. Finally, $c_2$ is a sub-concept of $c_1$.

The subconcept – superconcept relation is transitive: a concept is subconcept of all the concepts which can be reached by traveling upwards from it. If a formal concept has a formal attribute then its attributes are inherited by all its subconcepts. This corresponds to the notion of ‘‘inheritance’’. In fact, the lattice can also support multiple inheritances.

## 2.2 Fuzzy Set Theory

In this section, we review some fundamental knowledge of fuzzy theory [19].

In order to support the definition of Fuzzy Set, let us consider ordinary Crisp Set. It is represented by the characteristic function that assumes value into two-element set $\{0,1\}$. Formally, the traditional deterministic set in a universe $U$ can be represented by the characteristic function $\varphi_A$ mapping $U$ into two-element set $\{0,1\}$, namely for $x \in U$

$$\varphi_A(x) = 0 \text{ if } x \notin A$$
$$\varphi_A(x) = 1 \text{ if } x \in A$$

On the other hand, the Fuzzy Set is characterized by a membership function which assigns to each object value in the range [0, 1]. More formally:

**Definition 3.** *Fuzzy Set.* A fuzzy set $A$ on a domain $U$, is defined by a membership function $\mu_A$, mapping $U$ into a closed unit interval [0.1], where for $x \in U$

$$\mu_A(x) \in [0,1].$$

more nearer to 1 the value $\mu_A(x)$ is, the higher is the possibility that $x \in A$.

In particular, Fuzzy Set assigns to each object a grade of membership that is more useful to represent ambiguous or imprecise situations. In fact, the most classes of objects in real physical world do not have precisely defined criteria of membership. Just to give an example, if we consider the class of "numbers which are much greater than 1", the relations of numbers 10 or 100 respect to it are ambiguously evaluable. Again, imprecisely defined classes, like as "tall men", "beautiful women" and so on, do not constitute sets in the usual mathematical sense. The Fuzzy Set theory is suitable to model these ambiguous classes.

**Definition 4.** *Fuzzy Relation* represents a degree of presence or absence of association, interaction or interconnectedness between the elements of two or more sets. Let $X$ and $Y$ be two universes of discourse. A fuzzy relation $R(X,Y)$ is a fuzzy set in the product space $X \times Y$, i.e., it is a fuzzy subset of $X \times Y$, and is characterized by the membership function $\mu_R(x,y)$, i.e., $R(X,Y) = \{((x,y), \mu_R(x,y)) \mid (x,y) \in X \times Y\}$.

**Definition 5.** *Fuzzy Sets Intersection.* The intersection of two fuzzy sets $A$ and $B$ (denoted as $A \cap B$) with membership functions $\mu_A$ and $\mu_B$ respectively is defined as the *minimum* of the two individual membership functions. Formally:

$$\mu_{A \cap B}(x) = min(\mu_A(x), \mu_B(x)).$$

The Intersection operation in Fuzzy set theory is the equivalent of the AND operation in Boolean algebra.

**Definition 6.** *Fuzzy Sets Union* The union of two fuzzy sets $A$ and B (denoted as $A \cup B$) with membership functions $\mu_A$ and $\mu_B$ respectively is defined as the *maximum* of the two individual membership functions. Formally:

$$\mu_{A \cup B}(x) = max(\mu_A(x), \mu_B(x))$$

The Union operation in Fuzzy set theory is the equivalent of the *OR* operation in Boolean algebra.

**Definition 7.** *Fuzzy Set Cardinality*. Let $S_f$ be a fuzzy set on the domain $U$. The cardinality of $S_f$ is defined as

$$|S_f| = \sum_{x \in U} \mu(x)$$

where $\mu(x)$ is the membership of $x$ in $S_f$.

**Definition 8.** *Fuzzy Sets Similarity*. The similarity between two fuzzy sets $A$ and $B$ is defined as

$$E(A,B) = \left|\frac{A \cap B}{A \cup B}\right|$$

**Definition 9.** *Fuzzy Set Max-min Composition*. Let *P(X, Y)* be a fuzzy relation on *X, Y* and *Q(Y, Z)* be a fuzzy relation on *Y, Z.* The max-min composition of *P(X, Y)* and *Q(Y, Z),* denoted as $P \cdot Q$, is defined by:

$$P \cdot Q = max_{y \in Y} min\left(\mu_P(x,y), \mu_q(y,z)\right), \forall x \in X, y \in Y.$$

The max-min composition indicates the strength of relation between the element of *X* and *Z*.

## 2.3 Fuzzy Formal Concept Analysis Theory

Recently, FCA has been exploited in many applications in which uncertain and vague information occur in the representation of the domain. Pioneer studies to exploit the fuzziness into FCA were a generalized Wille's model [20] of FCA to fuzzy formal contexts or, mainly, the extension of the original formal concept analysis by setting truth degree for the propositions "object x has property y" in fuzzy formal contexts by employing a resituated lattice [21], [22], [23]. Degrees are taken from an appropriate scale L of truth degrees. Usually, L is valued with real values in [0, 1]. Thus the entries of a table describing objects and attributes become degrees from L instead of values from {0, 1} as is the case of the basic setting of FCA. This extension is known as Fuzzy Formal Concept Analysis (FFCA). Some definitions which incorporate fuzzy logic into Formal Concept Analysis are given [16].

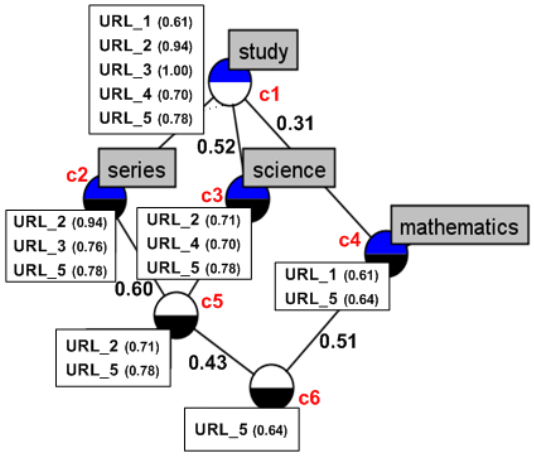
**Definition 10.** A *Fuzzy Formal Context* is a triple $K = (G,M, I = \varphi(G \times M))$, where $G$ is a set of objects, $M$ is a set of attributes, and $I$ is a fuzzy set on domain $G \times M$. Each pair $(g, m) \in I$ has a membership value $\mu_I(g, m)$ in $[0, 1]$.

The set $I = \varphi(G \times M) = \{((g, m), \mu_I (g, m)) \mid \forall g \in G, m \in M\ \mu_I: G \times M \rightarrow [0, 1] \}$ *is a fuzzy relation* $G \times M$. In literature, a similar definition considers a *multi-valued* context [18] which emphasizes the weight (rather than a membership value) associated to each pair (object, attribute).

**Definition 11.** Fuzzy Representation of Object. Each object g in a fuzzy formal context K can be represented by a fuzzy set $\Phi(g)$ as $\Phi(g) = \{(m_1, \mu_I(m_1)), (m_2, \mu_I(m_2)), \dots, (m_m, \mu_I(m_m))\}$ where $\{m_1, m_2, \dots, m_m\}$

is the set of attributes in K and $\mu_I(m_i)$ is the membership associated to attribute $m_i$ . $\Phi(g)$ is called the fuzzy representation of g.

| | study | mathematics | science | series |
|---|---|---|---|---|
| URL_1 | 0.61 | 0.89 | | 0.43 |
| URL_2 | 0.94 | | 0.71 | 1.00 |
| URL_3 | 1.00 | 0.56 | 0.2 | 0.76 |
| URL_4 | 0.70 | | 0.97 | |
| URL_5 | 0.78 | 0.64 | 1.00 | 1.00 |




*Fuzzy Formal Context* *Fuzzy Formal Concept Lattice*

Figure 2. Fuzzy Formal Concept Analysis

Figure 2 shows a fuzzy version of the formal context by means of a cross-table. According to the fuzzy theory, the definition of *Fuzzy Formal Concept* is given as follows.

**Definition 12.** Fuzzy Formal Concept. Given a fuzzy formal context K =(G, M, I) and a confidence threshold T , we define A*= {m ∈ M |∀ g ∈ A: $\mu_I$(g, m) ≥ T } for A ⊆ G and B*= {g ∈ G|∀ m∈ B: $\mu_I$(g, m) ≥ T } for B ⊆ M . A fuzzy formal concept (or fuzzy concept) $A_f$ , of a fuzzy formal context K with a confidence threshold T, is a pair (φ(A), B), where A ⊆ G, $\varphi(A) = \{g, \mu_{\varphi(A)}(g) | \forall g \in A\}$, B ⊆ M , A*=B and B*=A. Each object g has a membership $\mu_{\varphi(A)}(g)$ defined as

$$\mu_{\varphi(A)}(g) = min_{m \in B} \mu_I(g, m)$$

where $\mu_I(g, m)$ is the membership value between object *g* and attribute *m,* which is defined in *I;* Note that if $B=\{\}$ then $\mu_g = 1$ for every *g*. *A* and *B* are the extent and intent of the formal concept *(φ(A),B)* respectively.

In Figure 2 the fuzzy formal context has a confidence threshold *T=0.6* (as said, all the relations between objects and attributes with membership values less than 0.6 are not shown).

**Definition 13.**Let (φ($A_1$), $B_1$) and (φ($A_2$), $B_2$) be two fuzzy concepts of a fuzzy formal context (G, M, I). (φ($A_1$), $B_1$) is the sub-concept of (φ($A_2$),$B_2$), denoted as (φ($A_1$), $B_1$) ≤ (φ($A_2$),$B_2$), if and only if φ($A_1$) ⊆ φ($A_2$) (⇔ $B_2 \subseteq B_1$). Equivalently, (φ($A_2$), $B_2$) is the Super-concept of (φ($A_1$), $B_1$).

For instance, let us observe in Figure 2, the concept $c_5$ is a sub-concept of the concepts $c_2$ and $c_3$. Equivalently the concepts $c_2$ and $c_3$ are super-concepts of the concept $c_5$.

**Definition 14.** A Fuzzy Concept Lattice of a fuzzy formal context *K* with a confidence threshold *T* is a set *F(K)* of all fuzzy concepts of *K* with the partial order ≤ with the confidence threshold *T*.

The FCA theory proposes a hierarchical model, where the concepts (objects and their attributes) are arranged in a subsumption relations (known as "hyponym-hypernym" or "is-a" relationship too).

The fuzzy formal lattice evidences the membership associated to the objects and the class-subclass relationship. More formally:

**Definition 15.** The Fuzzy Formal Concept Similarity between concept $K_1=(\varphi(A_1), B_1)$ and its subconcept $K_2=(\varphi(A_2),B_2)$ is defined as

$$E(K_1, K_2) = \frac{|\varphi(A_1) \cap \varphi(A_2)|}{|\varphi(A_1) \cup \varphi(A_2)|}$$

where ∩ and U refer intersection and union operators [3]; on fuzzy sets, respectively.

As an example, the Fuzzy Formal Concept Similarity computed between the concept $c_2$ = {(URL_2, URL_3, URL_5), (series, study)}, and the concept $c_5$ = {(URL_2, URL_5), (science, study, series)}, shown in Figure 2, is given as follows:

$$E(c_1, c_2) = \frac{|(\min\{0.71,0.94\}) + (min\{0.78,0.78\})|}{|(\max\{0.71,0.94\}) + (max\{0.78,0.78\}) + (max\{0.76\})|} = 0.60$$

Comparing two figures, Figure 1 (relative to FCA theory) and Figure 2 (relative to FFCA theory), the differences in the modeling of two methods, considering the same sample of objects (i.e., web resources) is evidenced. In the classical FCA, the matrix representing the formal context contains binary values indicating the existence or not of the relation between objects and attributes. In the corresponding "fuzzy" table, a cell contains a value in the range [0, 1] says if there is a relation and in addition, gives a valuation about the strength of such relation.

Compared to formal lattice, the fuzzy lattice introduces further information about the knowledge structuring and relationship, such as the fuzziness enclosed in each object/resource and the similarities between fuzzy formal concepts.

## 2.4 Fuzzy Relational Concept Analysis

*Relational Concept Analysis* (RCA) [24] was introduced as an extended FCA framework for extracting formal concepts from sets of individuals described by '*local* ' properties and links. In this way, a concept is described with standard binary attributes but also with relational attributes. A relational attribute *r* link objects from a concept $c_i$, the *domain* of *r*, to

[3] The fuzzy intersection and union are calculated using t-norm and t-conorm, respectively. The most commonly adopted t-norm is the minimum, while the most common t-conorm is the maximum. That is, given two fuzzy sets *A* and *B* with membership functions $\mu_A(x)$ and $\mu_B(x)$ $\mu_{A\cap B}(x) = \min(\mu_A(x), \mu_B(x))$ and $\mu_{AUB}(x) = \max(\mu_A(x), \mu_B(x))$.

those of $c_j$, its *range*. RCA has already been used in a previous work in text mining and ontology design [25].

In RCA data are organized within a structure called '*relational context family*' (*RCF*). RCF comprises a set of contexts $K_i = (G_i, M_i, I_i)$ and a set of binary relations $r_k \subseteq G_i * G_j$ where $G_i$ and $G_j$ are the object sets of the contexts $K_i$ and $K_j$ , called respectively the *domain* and *range* of the relation $r_k$.

RCA uses the mechanism of *'relational scaling'* which translates domain structures (concept lattices) into binary predicates describing individual subsets. Thus, for a given relation $r$ which links formal objects from $K_i = (G_i, M_i, I_i)$ to those from $K_j = (G_j, M_j, I_j)$, new kind of attributes, called *'relational attributes'* are created and denoted by $r{:}c$, where $c$ is concept in $K_j$ . For a given object $g \in G_i$, relational attribute $r : c$ characterizes the correlation of $r(g)$ and the extent of $c = (X, Y)$.

Let us consider the relation between bacteria and antibiotics, where the first context is given by fuzzy context apposition in Table 1 and the respective lattice is given in Figure 3, and the second fuzzy context $K_3 = (G_3, M_3, I_3)$ with relative fuzzy lattice is given in Figure 4. The relation *ResistTo* between bacteria and antibiotics is given in Table 2. The application of the RCA process based on the concept lattices of Figure 3 and Figure 4 produces the final concept lattice shown in Figure 6, where the relations explicitly computed by the RCA process are emphasized.

Table 1. The Fuzzy Concept Lattice resulting from context of Bacteria

| | Proteobacteria | J_Proteobacteria | Actinobacteria | Bacilli | Spherical | Sticks | NegativeGram | positiveGram | Aerobic | Anaerobic |
|---|---|---|---|---|---|---|---|---|---|---|
| Helicobacter_P. | 0.8 | | | | 0.78 | | 1.0 | | 0.87 | |
| Klebsiella_P. | 1.0 | 0.98 | | | | 0.8 | 1.0 | | | 0.7 |
| Mycobacterium_S | | | 0.86 | | | 0.97 | | 1.0 | 0.75 | |
| Streptococcus_P. | | | | 1.0 | | 0.95 | | 1.0 | 0.85 | |
| Klebsiella_O | 1.0 | 1.0 | | | | 0.68 | 1.0 | | | 0.7 |

In more details, in Table 2, "Mycobacterium-S." is related through *ResistTo* to Cefotaxim. Examining the fuzzy lattice of antibiotics on Figure 4, it can be seen that Cefotaxim is in the extension of concepts 0, 1, 3, 4 and 6. The relational attributes ResistTo:0, ResistTo:1, ResistTo:3, ResistTo:4, and ResistTo:6, are associated to the object Mycobacterium-S. Then, a new concept lattice is built according to the extended context. At this point, as new concepts have been built, the lattice construction process is iterated and new relational attributes are associated to the bacteria objects whenever possible. If this is the case, the RCA process is iterated again. If this is not the case, this means that the fix-point of the RCA process has

been reached and that the final concept lattice has been obtained. This final lattice is given on Figure 6 (lattice on the left) and the corresponding fuzzy context is given in Figure 5.

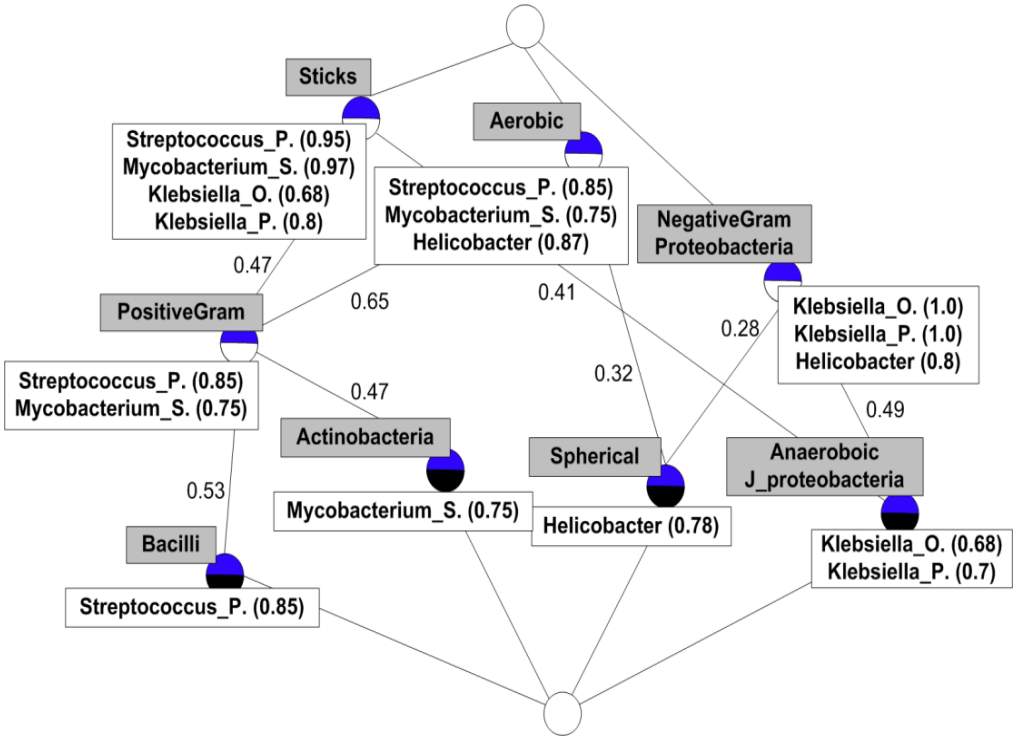



Figure 3. The Fuzzy Concept Lattice resulting from context of Bacteria

| | FRB1 | FRB3 | ARB1 | ARB2 | HBA5 | HBA10 |
|---|---|---|---|---|---|---|
| Clarithromycin | | 1.0 | | | | 1.0 |
| Ciprofloxacin | 1.0 | | | 1.0 | 1.0 | |
| Cefotaxim | | 1.0 | 1.0 | | | 1.0 |
| Macrolide | | | 1.0 | | | 1.0 |

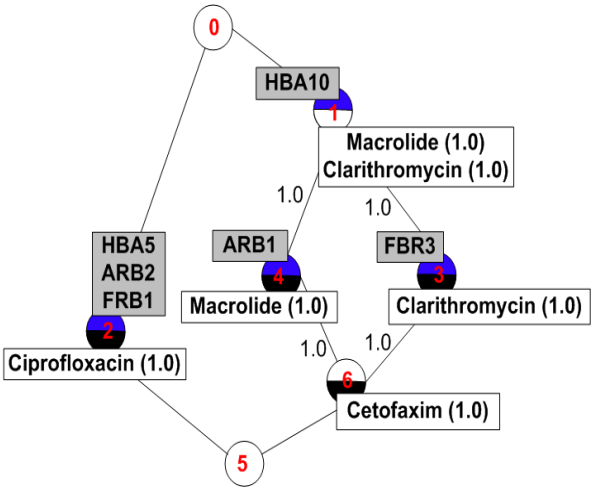



Figure 4. The Fuzzy Context and the relative Lattice resulting of Antibiotic

Table 2. The Fuzzy Relation "Resist To" between bacteria and antibiotics

| ResistTo | | | | |
|---|---|---|---|---|
| | Clarithromycin | Ciprofloxacin | Cefotaxim | Macrolide |
| Helicobacter_P. | 0.68 | 1.0 | | |
| Klebsiella_P. | | 0.53 | | 1.0 |
| Mycobacterium_S. | 0.7 | | 1.0 | |
| Streptococcus_P. | | 1.0 | 0.8 | |
| Klebsiella_O. | 1.0 | | | 0.54 |


| | Fuzzy Context of Bacteria | | | ResistTo:C0 | ResistTo:C1 | ResistTo:C2 | ResistTo:C3 | ResistTo:C6 | ResistTo:C7 | ResistTo:C8 |
|---|---|---|---|---|---|---|---|---|---|---|
| Helicobacter_P. | ... | | ... | 0.8 | | | 1.0 | 0.6 | | 0.6 |
| Klebsiella_P. | | ... | | 0.7 | | | 0.5 | | 1.0 | 1.0 |
| Mycobacterium_S. | | | | 0.8 | | 1.0 | | 0.8 | 1.0 | 0.8 |
| Streptococcus_P. | ... | | | 0.9 | | 0.8 | 1.0 | 0.8 | 0.8 | 0.8 |
| Klebsiella_O. | | | ... | 0.7 | | | | 1.0 | 0.5 | 0.7 |

Figure 5. Fuzzy Formal Context obtained using RCA

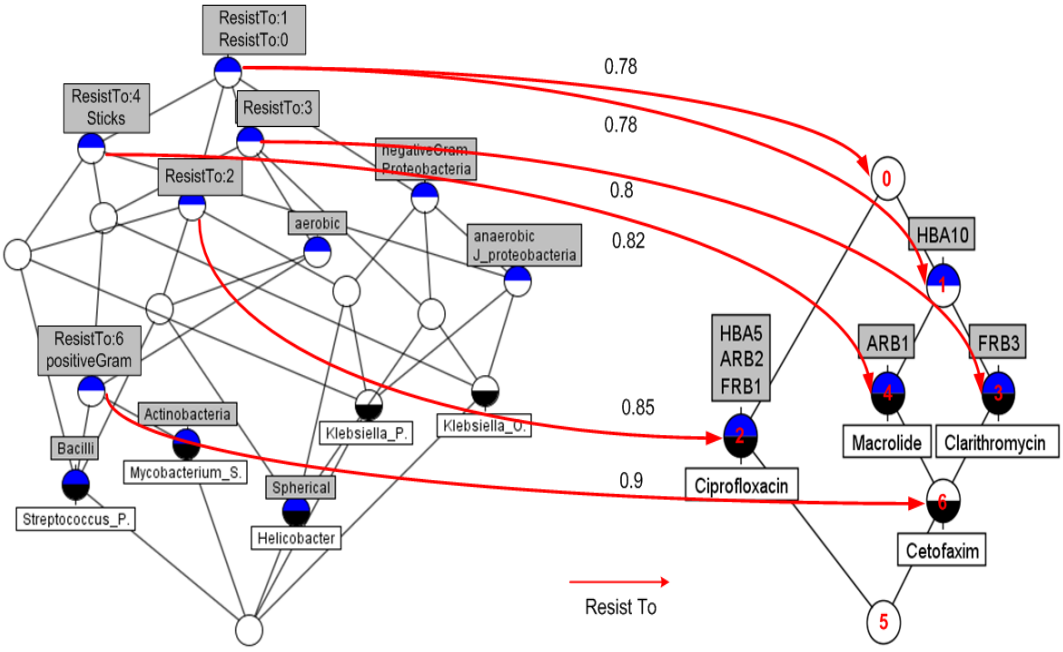



Figure 6. Final concept lattice obtained using RCA

## 2.5 Algorithms for Generating Concept Lattices

In literature several algorithms have been proposed to build a concept Lattice. In [26] a survey on possible algorithm is presented.

In particular, some top-down algorithms that we have analyzed are cited as follow. The algorithm of *Bordat* [27] uses a tree for fast storing and retrieval of concepts. The time complexity of Bordat is $O(|G||M|^2|L|)$, where $|L|$ is the size of the concept lattice, $|G|$ is the size of the object lattice and $|M|$ = is the size of the attribute lattice. Moreover, this algorithm has a polynomial delay $O(|G||M|^2)$. The algorithm proposed by *Ganter* [28] computes closures for only some of subsets of $G$ and uses an efficient canonicity test, which does not address the list of generated concepts. It produces the set of all concepts in time $O(|G|^2|M||L|)$ and has polynomial delay $O(|G|^2|M|)$. The *Close by One (CbO)* [29] algorithm uses a similar notion of canonicity, a similar method for selecting subsets, and an intermediate structure that helps to compute closures more efficiently using the generated concepts. Its time complexity is $O(|G|^2|M||L|)$, and its polynomial delay is $O(|G|^3|M|)$.

Regarding to a bottom-up algorithm (i.e., to generate the bottom concept and then, for each concept that is generated for the first time, generate all its upper neighbors), *Lindig* [30] uses a tree of concepts that allows one to check whether some concept was generated earlier. The time complexity of the algorithm is $O(|G|^2|M||L|)$. Its polynomial delay is $O(|G|^2|M|)$.

Due to their incremental nature, the algorithms considered below do not have polynomial delay. Nevertheless, they all have cumulative polynomial delay.

*Nourine* proposes an $O((|G| + |M|)|G||L|)$ algorithm for the construction of the lattice using a lexicographic tree [31] with edges labeled by attributes and nodes labeled by concepts. Note that this algorithm is only half-incremental. First, this algorithm incrementally constructs the concept set outputting a tree of concepts; next, it uses this tree to construct the diagram graph.

The algorithm proposed by *Norris* [32] is essentially an incremental version of the CbO algorithm. The original version of the Norris algorithm from [20] does not construct the diagram graph. The time complexity of the algorithm is $O(|G|^2|M||L|)$.

The algorithm proposed by *Godin* [33] has the worst-case time complexity quadratic in the number of concepts. This algorithm is based on the use of an efficiently computable hash function $f$ (which is actually the cardinality of an intent) defined on the set of concepts.

Another incremental algorithm is *AddIntent* [34]; in experimental comparison, AddIntent outperformed a selection of other published algorithms for most types of contexts and was close to the most efficient algorithm in other cases. The current best estimate for the algorithm's upper bound complexity to construct a concept lattice $L$ is $O(|G|^3|M|)$.

In this research work we have exploit the *Linding* algorithm and the incremental *AddIntent* algorithm to generate the concept lattice. In particular we have defined and developed a fuzzy extension of these to support the evaluation of vagueness and uncertainty of data. Table 3 show the properties of each algorithm on quoted.

Table 3. Properties of algorithms constructing concept lattices

| | **F1** | **F2** | **F3** | **F4** | **F5** |
|---|---|---|---|---|---|
| Lindig | | | x | x | |
| Lindig* | | x | x | x | x |
| AddIntent | x | | | x | |
| AddIntent* | x | x | | x | x |
| Nourine | x | | | x | |
| Norris | x | | | | |
| Godin | x | | | | |
| Bordat | | | x | x | |
| CbO | | | x | x | |
| Ganter | | | x | x | |

*Legend:*

F1 – incrementally;
F2 – fuzzy;
F3 – sequentially;
F4 – define a tree structure;
F5 – population of Semantic DataBase.
*Our implementation

# Chapter 3

## Semantic Technologies

Semantic technologies are a new paradigm — an approach that deals with the challenges of net-centric infrastructure, knowledge work automation, and building systems that know what they're doing. Semantic technologies are functional capabilities that enable both people and computers to create, discover, represent, organize, process, manage, reason with, present, share and utilize meanings and knowledge to accomplish business, personal, and societal purposes. Semantic technologies are tools that represent meanings, associations, theories, and know-how about the uses of things separately from data and program code. This knowledge representation is called ontology, a run-time semantic model of information, defined using constructs for:

- **Concepts** – classes, things;
- **Relationships** – properties (object and data);
- **Rules** – axioms and constraints;
- **Instances of concepts** – individuals (data, facts);

As described in [35], Semantic technologies have emerged as a central theme across a broad array of ICT research and development initiatives. Figure 7 visualizes the intersections of four major development themes in the semantic wave: networking, content, services, and cognition. Content and Cognition are the two theme emphasized in this research thesis. Specifically, R&D themes include:

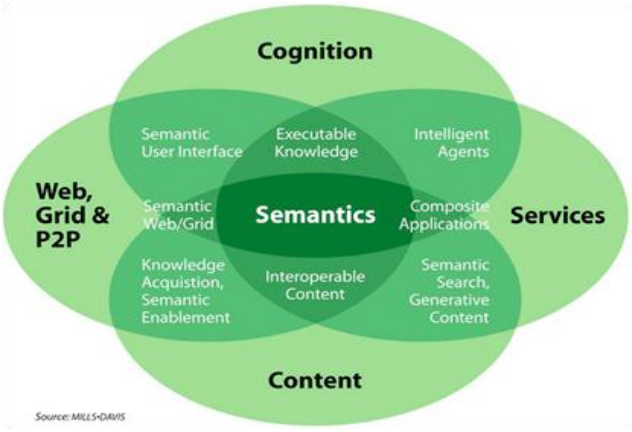



Figure 7. The central-role of semantic technologies.

- ***Content*** — Semantics to make information interoperable, improve search, enable content discovery, access, and understanding across organization and system boundaries, and improve information lifecycle economics;
- ***Cognition*** — Semantics to make knowledge executable by computer; augment capabilities of knowledge workers; enable robust adaptive, autonomic, autonomous behaviors;
- ***Services*** — Semantics to enable computers to discover, compose, orchestrate, and manage services, and link information and applications in composite applications;
- ***Networking*** — Semantics to enable computers to configure and manage dynamic, persistent, virtual systems-of-systems across web, grid & P2P.

In order to connect systems, integrate information and make processes interoperable, the first step is to integrate the knowledge about these systems, content sources, and process flows. Today, people do this offline, manually. Instead, in the Semantic Web vision both people and applications will connect knowledge in real time using automated and semi-automated methods. Semantically modeled, machine executable knowledge lets us connect information about people, events, locations, and times across different content sources and application processes. Instead of disparate data and applications on the Web, we get a Web of interrelated data and interoperable applications. Recombinant knowledge is represented as concepts, relationships and theories that are sharable and language neutral. Semantic technologies provide the means to unlock knowledge from localized environments, data stores, and proprietary formats so that resources can be readily accessed, shared, and combined across the Web. Actual limitations of the systems are spurring development of semantic platforms to provide meaning-based, concept-level search, navigation, and integration across varied content sources and applications found on PCs and other devices.

## 3.1 Semantic Models (Taxonomies and Ontologies)

The pursuit of data models that can adequately and accurately describe the vast array of relationships within an organization, body of information, or other knowledge domain space is an ongoing one. The challenge is heightened when trying to arrive at approaches that are machine computational, meaning that the models can be used by computers in a deterministic and largely autonomous way. Numerous knowledge representation technologies have been devised, some successfully and some not. As a result of these efforts, computer scientists have made significant progress toward finding out the most appropriate manner in which to express highly descriptive relationships and logical concepts existing within business environments, organizational interactions, and, to a larger extent, everyday society.

Overcoming the communication gaps resulting from reliance on numerous vocabularies remains a challenge. Technical challenges have until recently had to do with overlapping and redundant terminological inconsistencies. Without knowing it, business units, individuals, and others have expended scarce resources referring to identical elements using different terminologies and different relationship models, causing confusion and limiting communication possibilities. Identifying and reconciling these semantic distinctions is a fundamental

reason for using semantic models. Figure 8 displays a spectrum of commonly used semantic models.

This diagram shows a range of models, from models on the lower left with less expressive or "weak" semantics to models on the upper right with increasingly more expressive or "strong" semantics. In general, the progression from the lower left to the upper right also indicates an increase in the amount of structure that a model exhibits. Included in the diagram are models and languages such as the relational database model and XML on the lower left. These models are followed by XML Schema, Entity-Relation models, XTM (the XML Topic Map standard), RDF/S (Resource Description Framework/Schema), UML (Unified Modeling Language), OWL (Web Ontology Language), and up to First Order Logic (the Predicate Calculus), and higher. In truth, the spectrum extends beyond modal logic but any such discussion is still largely theoretical as well as outside the scope of this document.

One of the simplest forms of semantic model is a taxonomy. A taxonomy might be thought of as a way of categorizing or classifying information within a reasonably well-defined associative structure. The form of association between two items is inherent in the structure and in the connections between items. A taxonomy captures the fact that connections between terms exist but does not define their nature. All the relationships become hierarchical "parent-child" links. Sometimes this hierarchical structure is called a "tree," with the root at the top and branching downward. In hierarchies, there is an ordered connection between an item and the item or items below it.

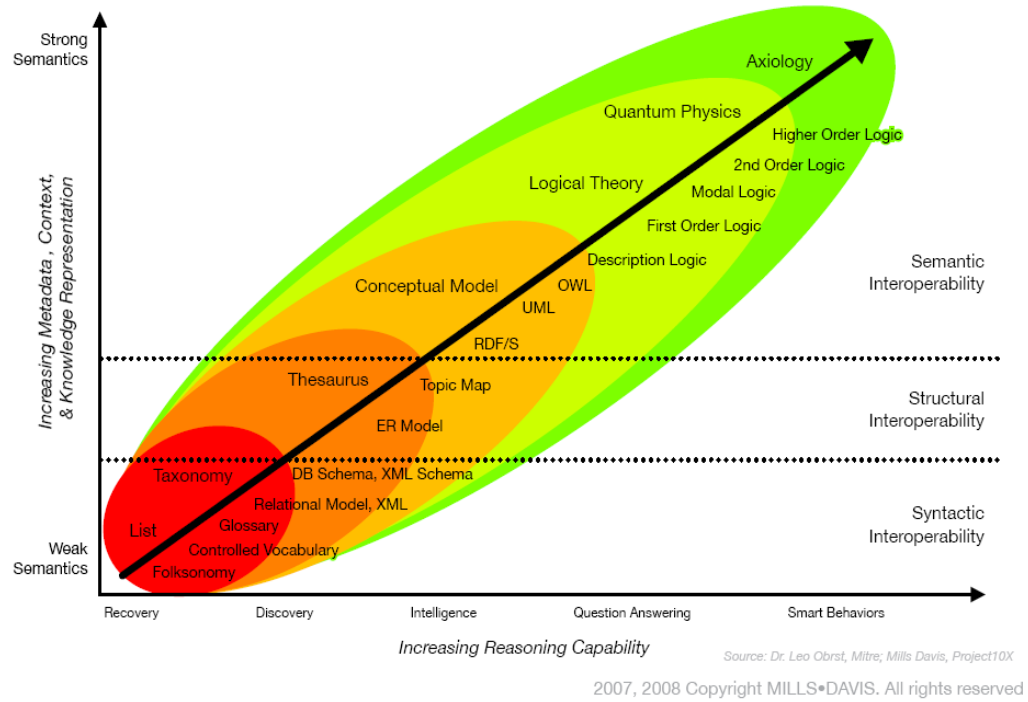



Figure 8. Spectrum of Knowledge Representation and Reasoning Capabilities.

A thesaurus is a higher order form of semantic model than a taxonomy because its associations contain additional inherent meaning. In other words, a thesaurus is a taxonomy with some additional semantic relations in the form of a controlled vocabulary. The nodes in a thesaurus are "terms," meaning they are words or phrases. These terms have "narrower than" or "broader than" relationships to each other. A thesaurus also includes other semantic relationships between terms, such as synonyms.

Taxonomies and thesauri are limited in their semantic expressiveness because they offer only one dimensional axis on which to define relationships. As such, they are typically used to create a classification system, but they fall flat when trying to represent multidimensional and/or varied conceptual domains.

Concepts are the bearers of meaning as opposed to the agents of meaning. They are largely abstract and therefore more complex to model. Concepts and their relationships to other concepts, their properties, attributes, and the rules among them cannot be modeled using taxonomy. Other more sophisticated forms of models, however, can represent these elements. A semantic model in which relationships (associations between items) are explicitly named and differentiated is called an ontology. (In Figure 8, both conceptual models and logical theories can be considered ontologies, the former a weaker ontology and the latter a stronger ontology). Because the relationships are specified, there is no longer a need for a strict structure that encompasses or defines the relationships. The model essentially becomes a network of connections with each connection having an association independent of any other connection. Unlike a taxonomy, which is commonly shown as a "tree," ontology typically takes the form of a "graph," i.e., a network with branches across nodes (representing other relationships) and with some child nodes having links from multiple parents. This connective variability provides too much flexibility in dealing with concepts, because many conceptual domains cannot be expressed adequately with either a taxonomy or a thesaurus. Too many anomalies and contradictions occur, thereby forcing unsustainable compromises. Moreover, moving between unlike concepts often requires brittle connective mechanisms that are difficult to maintain or expand.

Simple ontologies are mere networks of connections; richer ontologies can include, for example, rules and constraints governing these connections. Just as improvements in languages and approaches to model-based programming increased the ability to move from conceptual models to programmatic models without the need for human coding steps, similar advancements have taken place within ontological development. Whereas once ontologies were created primarily for human consumption, the development of robust protocols for expressing ontologies along with a growing infrastructure that support such models, provides increased capabilities for models to deduce the underlying context and draw logical conclusions based on these associations and rules.

The current state of the art on representing and using ontologies has grown out of several efforts that started in the 1980s. Early semantic systems initially suffered from a lack of standards for knowledge representation along with the absence of ubiquitous network infrastructures. With the advent of the World Wide Web and the acceptance of XML as a de facto standard for exchange of information on the Web, ontology efforts have started to converge and solidify. RDF, OWL, and Topic Maps (an ISO standard for representing networks of concepts to be superimposed on content resources) all use XML for serialization. This results in strongly typed representations (with public properties and fields contained in a serial format), making it easy to store and transport these models over the Web as well as integrate

them with other web standards such as Web services. A cautionary note expressed by some in the knowledge management community is that there may be a proliferation of competing ontologies, which may in turn mean continued friction in achieving seamless sharing of structure and meaning across systems. Whereas different ontologies can be aligned for automated transformation from one model to another, it typically requires a good deal of human modeling to get to that point (aligning ontologies of any significant size can be similar to aligning large databases, a task that often requires significant planning and effort). These knowledge management professionals stress that significant benefits can result from using a widely shared foundational ontology, a subject that will be addressed in a later section.

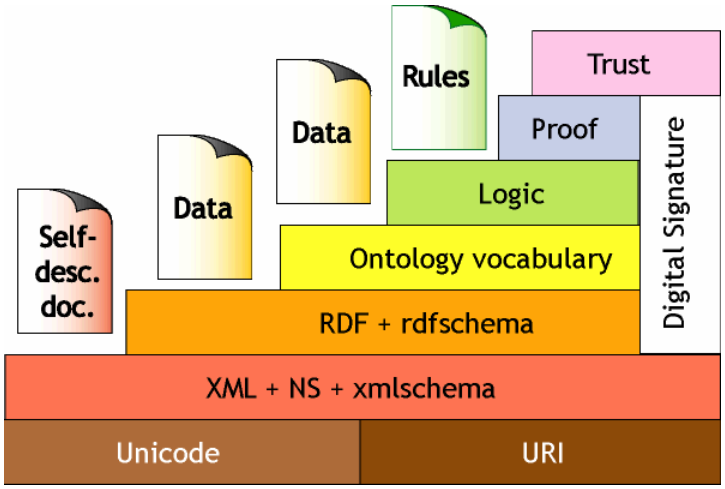



Figure 9. Semantic Web Wedding Cake (From Berners-Lee, XML 2000 Conference).

## 3.2 Semantic Web Wedding Cake

Figure 9 is the Semantic Web layered architecture (Wedding Cake) presented by Tim Berners-Lee in 2000. Languages have increasing expressive power are layered on top of the other. The bottom layer is Unicode and URI, which form the basis of the architecture. Unicode makes people with different languages specify data in the same format. URI helps us indicate resources on the Web.

### *3.2.1 XML, Namespace, XMLSchema*

The second layer from the bottom in Figure 9 consists of XML [36], namespace, and XML Schema. XML is the abbreviation of eXtensible Markup Language. It is a well-defined and flexible text format for electronic data publishing. The structure of an XML document has to follow the defined standard so that it can be processed by computer automatically. Because of those characteristics, XML is suitable to be the underlying format for data exchange.

Namespace is used for resolving resources with the same name but in different URLs. With namespace, we can use the same name for resources in different documents. In XML Schema, there are some built-in datatypes definitions such as string, boolean, decimal, datetime, and etc. XML Schema helps us define data in those datatypes instead of just using

strings. Therefore, all applications who know XML Schema can understand what we mean in a document written in that format. Conventionally, the namespace of XML Schema is named xsd, so when we want to represent an integer, we can write it as <xsd:integer>100</xsd:integer>. XML Schema provides the basic datatype system of the Semantic Web.

### *3.2.2 Resource Description Framework (RDF) & RDF-Schema*

The next layer is RDF and RDF Schema. RDF [37] stands for Resource Description Framework, which provides a way to describe resources over the Internet. RDF is written in XML so that its underlying structure is based on XML syntax and structure. RDF can be used for knowledge sharing, resources cataloging and searching, etc. There are two major parts in RDF: Resources and Properties. Resources are identified using URIs, which can be a person, a book, or anything else. Properties are attributes used to describe a resource. A property's value can be a XML schema datatype or another resource. RDF expressions allow us to describe some resources by defining its related properties.

From another point of view, if we parse all contents in an RDF file, we can get the so called RDF, which is a set of triples composed of a resource's URI, a property, and a value. They are also called a subject, a predicate, and an object. All information in an RDF file can be represented as RDF triples. By the way, an RDF file can also be represented as a graph. If we set a subject as a node in a graph, and set its predicate as an outgoing edge from it to another node that represents its object, we can construct a graph representing the RDF triples. RDF also defined some data structures defined such as collection and container. RDF supports three type of container. They are rdf:Alt, rdf:Bag, and rdf:Seq. The collection data structure is also defined in RDF. RDF collection adopts the concept of list so that a collection is consist of a set of dummy nodes with first and rest properties to indicate real collection elements.

RDF Schema (abbreviated as RDFS) is a schema language of RDF [38]. It is a semantic extension of RDF. RDF Schema vocabulary descriptions are written in RDF, so it is also an RDF document.

RDF Schema helps us define the relationships between resources and properties and add more semantics to the model described above. RDFS supports defining the class and subclasses relationship with the tags rdfs:Class and rdfs:subClassOf. It also supports defining domain and range of a property with the tags rdfs:domain and rdfs:range. Subproperties in RDF Schema is defined using the rdfs:subPropertyOf tag. RDF Schema can help people construct the relationships in a model.

RDF and RDF Schema provide the basic functionalities for semantic markup. However, their expressive power is not enough. The layer on top of them is ontology vocabulary. It provides more expressive power. Nowadays, some logics are adopted in this layer of the Semantic Web architecture. It is mainly based on Description Logics.

### *3.2.3 Ontology & Ontology Web Language*

Ontology is a very important part in Semantic Web. It enables sharing, exchanging, and reusing knowledge by formalization of concepts of interest in a specific domain. When we want to describe something, we can use the definitions of an ontology to describe it. Therefore, concepts of the ontology can be used in communicating with each other. By referring to

the same ontology, different entities can understand and talk to each other. Ontologies are very important in this work research. In this section, we introduce some current languages for ontology construction. Standardization of ontology languages has been an important issue in W3C for years. OWL (Web Ontology Language) [39] is their result and is going to become the new standard for ontology definition. OWL is a revision of DAML+OIL. There are some major modifications such as removing of synonyms for RDF and RDFS classes and properties, supporting versioning, renaming of some properties and classes, adding new classes and properties, and etc.

OWL has three sublanguages with different expressive power. They are OWL Lite, OWL DL, and OWL Full, respectively. OWL Lite supports classification hierarchy and simple constraints while having lower complexity, OWL DL is for users who need the maximum expressiveness without losing computational completeness, and OWL Full is the sublanguage with the maximum expressiveness and syntax freedom but no computational guarantees. OWL DL is an extension of OWL Lite, and OWL Full is an extension of OWL DL. So a legal OWL Lite ontology is also a legal OWL DL ontology and a legal OWL DL ontology is a legal OWL Full ontology too.

OWL is based on XML and RDF and all data in an OWL file can be represented as a set of RDF triples. An OWL document has four main parts in it. The first part is ontology header. It contains information about namespaces, version, imports, and compatibility with other OWL documents. The second part is class axioms. Class and subclass relationship definitions are in this part. The first letter of a class name should be capital. The third part is property axioms. Property definitions, which are domain and range, are defined here. The first letter of a property name should not be capital. The last part is individual (instance) axioms. Individuals of classes defined in the part of class axioms are declared here. By the way, all the four parts can be written in any order.

In OWL, we define classes using the <owl:Class> tag. A class can be subclass of one or multiple classes using <owl:subClassOf>. OWL has two built-in basic classes, owl:Thing and owl:Nothing, stand for top (everything) and bottom (empty set) respectively. Every user-defined class is implicitly a subclass of owl:Thing and every individual in OWL is in the set of owl:Thing individuals.

We can also define properties of a class. Like RDF, there are two kinds of properties in OWL. The ObjectProperty has points to a class or individual and DatatypeProperty points to a primitive datatypes such as integer, decimal, string, etc. These two kinds of property can be used to define properties or attributes of a class. We can model most of concepts in a rough way by combining the class and property.

Moreover, we can also define sub-properties of properties, transitive properties, and inverse properties of some other properties. For example, we can define ancestor as a transitive property, father as a sub-property of parent, child as an inverse property of parent.

Furthermore, OWL supports sameClassAs and samePropertyAs. They can help us to define classes or properties having the same content but different name. They are useful to define one concept with many different names. OWL also provides intersectionOf, complementOf, and disjointUnionOf, that helps us to form the concepts of models more completely.

In early 2004, W3C has announced that RDF and OWL are their recommendation for exchanging knowledge and representing information on the Web. In the announcement, they describe a infrastructure for sharing data on the Web. In the infrastructure, XML provides

rules and syntax for structured documents, RDF forms a data framework for the Web, and OWL is used to publish and share ontologies. OWL adds more vocabularies for describing classes and properties in order to support advanced Web search, knowledge management, and software agents. In this research, is followed W3C's recommendation to build ontologies and construct knowledge bases.

## 3.3 Semantic Web Vocabularies

This section describes the basic Semantic Web technologies for defining, inference, storing and querying knowledge. With this insight it is possible to use these technologies to represent concepts that are relevant to the domain of an e-Research society.

Dublin Core, SKOS, FOAF, SIOC and RSS all provide conceptualizations for particular facets required to represent a sharing content.

### *3.3.1 Dublin Core Metadata Initiative (DCMI).*

The purpose the Dublin Core Metadata Initiative has been to define standards, vocabularies and practices for metadata [40]. DCMI has an abstract model which builds on the work of RDF and RDF Schema. This abstract model breaks down into three separate sub-models: *Resource Model* - Resources can be described by properties that have either a literal or non-literal value; *Description Set Model* - a resource may have one or more descriptions that make up that resource's description set. Each description may have one or more statements, i.e. property-value pairs. The value of the statement may be a literal or a non-literal that could either be a vocabulary encoding scheme URI or a URI or string described by a vocabulary encoding scheme; *Vocabulary Model* - Vocabularies have terms that may be classes, properties, vocabulary encoding schemes or syntax encoding schemes, i.e. classes of literals. Classes and properties may have sub-classes/sub-properties of each other and properties may have domains and ranges that can also be classes. A resource can then be an instance of a class or a member of a vocabulary encoding scheme.

DCMI provides a schema called DCMI Metadata Terms that can be specified as an RDF schema [41]. DCMI Metadata Terms reuses the legacy properties from the DCMI Elements Set[4] and defines additional properties and classes. The properties cover standard bibliography fields and properties required in the publication process. The classes allow for new instances of non-literal values that some properties require, such as location, license document, le format, media type, linguistic system etc. as well as any agents or classes of agent that may be involved in the publication process.

### *3.3.2 Friend Of A Friend (FOAF)*

The Friend of a Friend (FOAF) project, one of the largest projects in the semantic web [42], is a descriptive vocabulary built based on RDF and OWL, for creating a Web of machine-readable pages for describing people, the links between them and the things they create and do [42]. It is accepted as standard vocabulary for representing social networks, and many large social networking websites use it to produce Semantic Web profiles for their us-

[4] http://dublincore.org/2008/01/14/dcelements.rdf

ers [43]. FOAF has the potential to become an important tool in managing communities [44], and can be very useful to provide assistance to new entrants in a community, to find people with similar interests or to gather in a single place, people's information from several different resources, decentralizing the use of a single social network service for example [43]. The things described in the web are connect by people. People attend meetings, create documents, are depicted in photos, have friends, and so on. Consequently, there are a lot of information that might be said about people and the relations between them and objects (documents, photos, meeting, etc) [45]. FOAF describes the most common information we usually want to know about a person and because it is built upon RDF, it also uses some vocabulary from other resources, such as the Dublin Core (DC).

The base class described in FOAF is the foaf:Agent class. The Agent class describes "the things that do stuff" [43] and have foaf:Group, foaf:Person and foaf:Organization as subclasses. FOAF describes resources such as foaf:Document, foaf:Image or foaf:OnlineAccount and people properties like foaf:name, foaf:title or foaf:mbox (email box).

One important property that should be mentioned is the foaf:knows property. It can be used to link two people together [44]. FOAF identifies other people by stating their properties.

Nowadays there are already many projects fostering the use of FOAF. Some examples can be:

- Google Social Graph API[5]: indexes all the public FOAF data in the Web. Social Graph utilizes public connections their users have already created in other web services.
- Origo[6]: a Web-application that enables users to manage their social community profiles utilizing semantic technologies. It allows to unite their different profiles and to browse through their semantic social network across various platforms, using the FOAF structure and the RELATIONSHIP ontology[7] to specify different kinds of relationships between users.
- Flink: a system for the aggregation and visualization of online social networks, extracted from a electronic information sources such as web-pages, emails, publication archives and FOAF profiles [46].

### *3.3.3 Semantically-Interlinked Online Communities (SIOC)*

The SIOC project (Semantically-Interlinked Online Communities), is an ontology for representing rich metadata from the Social Web in RDF/OWL, accepted by W3C. It aims to enable the integration of online community information (wikis, message boards, weblogs, etc) [47]. SIOC aims to meet the needs of communities and users on the evolving Web, as com-

[5] http://code.google.com/intl/en/apis/socialgraph/

[6] http://code.google.com/p/origo/

[7] http://vocab.org/relationship/.html

munity-centric content sites become more prevalent and finding relevant items from these communities is now more important than ever [48].

As an ontology, SIOC can't incorporate on it everything that might be important to know about communities, about their users and about the contents that users create, otherwise it would be too large [49]. Being built over RDF, we can take advantage of other specific description vocabularies, to complement the domain we want to specify. Being built in a modular design, we can create additional ontology modules for specializing and further extending classes and properties contained within the SIOC core ontology [48]. Currently there are two modules defined:

- *SIOC Types Module*: to extend sub-classes of classes such as Forum, Post, Item or Container [49].
- *SIOC Services Module:* a sioc:Service allows us to indicate that a web service is associated with (located on) a sioc:Site or a part of it [49].

To make the link between SIOC ontology and specific domain ontologies, SIOC Types module uses an rdfs:seeAlso property to point SIOC Types objects to the related vocabularies and classes [48]. There are SIOC exporter tools that can be used to export RDF information about the contents and structure of Web 2.0 platforms (wikis, forums, blogs, message boards, etc) [50]. This allows information from every page of a site to be represented in RDF, making all the information contained there available in a machine readable form and so, ready for reuse [48]. Some examples of those exporters are the Wordpress exporter[8] or the vBulletin exporter[9].

There are many classes and properties in SIOC, the main notion is that a sioc:User (individuals for this class are members of an online community) creates a sioc:Post (individuals for this class are messages or articles) that is contained in a sioc:Forum (individuals of this class are channels or discussion areas) that is hosted on a sioc:Site (individuals of this class are locations of online communities). The sioc:has_creator property relates a post to the user who creates it. Another SIOC property of interest is sioc:title defining a property that a particular discussion post is associated to. The sioc:content and sioc:has_reply (with its inverse property sioc:reply_of). The first one is used to report the real content in order to perform better search operations. Another property that is useful for prefixed aims is the sioc:last_activity_date. This property enables to fix the date and time of the last activity associated to a SIOC concept instance (e.g. creation of a post).

SIOC can be also used in synergy with the SCOT Ontology[10] and the Richard Newman's Ontology given that tags:Tagging instances can be associated with sioc:User instances. So, an instance of sioc:Post can be tagged by using the property scot:hasTag with domain sioc:Item (note that sioc:Post is a subclass of sioc:Item) and range scot:Tag. Furthermore, SIOC does not offer a mechanism to manage rating for published posts. In order to fill the aforemen-

[8] http://sioc-project.org/wordpress

[9] http://wiki.sioc-project.org/index.php/VBSIOC

[10] http://scot-project.org

tioned lack we think to use Review RDF. Review RDF is a domain specific vocabulary used to describe the main properties of a review in RDF. The most important properties are rev:createdOn, rev:hasReview, rev:maxRating, rev:minRating, rev:rating, rev:reviewer and rev:text. Rating is useful to enable to emphasize valued content and improve the reputation of employees within the organizations.

### *3.3.4 Simple Knowledge Organization System (SKOS)*

SKOS is a vocabulary which is intended to represent Knowledge Organisation Systems (KOS) like thesauri, term lists and controlled vocabularies [51].

Using the SKOS data model to translate the domain knowledge, we can define each concept as an individual of the skos:Concept class. In this data model, we don't have "pure" classes hierarchy like in an OWL ontology, but we have skos:narrower and skos:broader properties, to construct the knowledge structure. For non-hierarchic linking we can use the property skos:related. OWL language in its basis already defines the transitive closure of the relations between the classes while in the SKOS data model we have to explicitly define it by using the relational properties skos:narrowerTransitive and skos:broaderTransitive, to say for instance that if <B> is a sub-class of <A> and <C> a sub-class of <B>, <C> is also a sub-class of A. From a SKOS vocabulary, we can't make instances of the concepts because they are already an instance of the class "skos:Concept" and so cannot be further instantiated.

A big advantage of SKOS data model is that it can be connected with SIOC ontology model, by the property sioc:topic. Since this is a platform with social and collaborative characteristics this is an important aspect because it simplifies the process of integrating socio-collaborative information with domain knowledge.

Another interesting aspect is the possibility of defining lexical labels to the concepts: skos:prefLabel, skos:altLabel and skos:hiddenLabel. The properties skos:prefLabel and skos:altLabel allow us to define a clear meaning for each one of the concepts while the property skos:hiddenLabel can be used for example to tag misspelled words of the concept, which users usually search for. In SKOS we also have definition (skos:definition) and notes properties to further describe the concepts. This kind of properties helps in the interoperability issues because they provide a way to better understand each concept.

## 3.4 Conclusion

This chapter synthesizes the semantics technologies to enable computers to discover, compose, orchestrate, and manage knowledge. In particular, Semantic Web shared vocabularies are detailed to introduce the technological background behind the research of work thesis.

# Part II: Methodologies & Applications

# Chapter 4

## Methodology for Knowledge Extraction

Knowledge Extraction concerns with the creation of knowledge from structured (relational databases, XML) and unstructured (text, documents, images) sources. The resulting knowledge needs to be in a machine-readable and machine-interpretable format and must represent knowledge in a manner that facilitates inferencing. Nowadays, semantic technologies play a crucial role in order to enable machine oriented knowledge representation. Nevertheless, design and construction of domain knowledge models is a human intensive process which requires allocation of huge resources in terms of cost and time.

This research work defines general framework mainly aimed to extract concepts and relations by analyzing textual sources. On the other hand, this general framework has been applied in order to face with different problems of knowledge extraction and discovery, such as: ontologies and taxonomy extraction, semantic annotation, information retrieval and faceted browsing.

This chapter introduces the general framework for knowledge extraction defined in this research work (see Section 4.1). Subsequently, application of general framework to specific research objectives will be described (see Section 4.2).

Table 4. Input/Output of Knowledge Extraction process.

| **Knowledge Extraction Processing Pipe** | |
|---|---|
| Input | Digital resources of heterogeneous nature: documents and resources from Web, scientific papers, forums, blogs, wikis, etc. |
| Output | Hierarchical conceptualization of resources' content represented by exploiting semantic technologies, such as: RDF[11] (Resource Description Framework), RDFS[12] (RDF Schema), OWL[13] (Web Ontology Language). |

[11] http://www.w3.org/TR/rdf-schema

[12] http://www.w3.org/TR/rdf-schema/

[13] http://www.w3.org/TR/owl-ref/

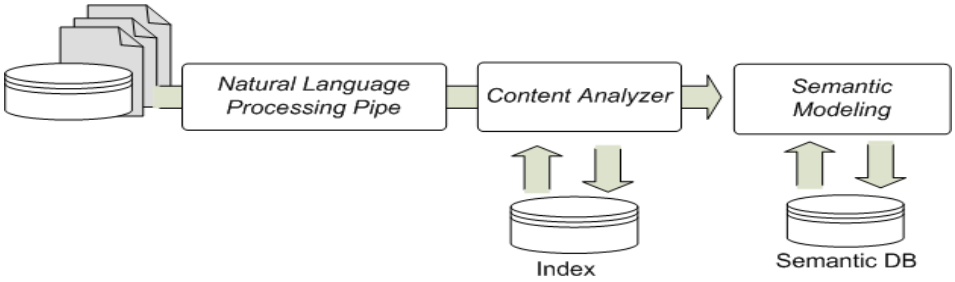



Figure 10. Logical View of Overall Process

## 4.1 Knowledge Extraction: General Framework

As highlighted in Table 4, the process of knowledge extraction takes into account text content of digital resources and extracts a knowledge model represented by means of semantic technologies (e.g., RDF, RDFS, OWL, etc.).

Figure 10 illustrates the overall process of Methodology for Knowledge Extraction. Essentially, there are three main phases:

- *Natural Language Processing Pipe*, aimed at analyzing the content of the input resources to extract keywords that characterize the text content. This phase performs natural language processing (e.g., stop-words removal, stemming, lemmatization, etc.) by using commonsense external knowledge, i.e. WordNet[14]. The output of the NLP phase is the extraction of disambiguated keywords that characterize the content of the resources collection analyzed;
- *Content Analyzer*, that consists of:
  - *Vectorization*, aimed to extract a mathematical representation of the resources examined. The mathematical model extracted is a vector model: each resource analyzed is represented by a keywords vector (i.e., extracted from the pipe NLP) weighted according to the resource content.
  - *Concept Data Analysis*, which implements the technique of data analysis, based on *Fuzzy Formal Concept Analysis* returns as output a resources conceptualization according to shared characteristics. In particular, the mathematical model resulting is an algebraic structure, that is lattice, in which the interconnections between concepts denote subsumption relationships between them. The fuzziness plays a crucial role in determining: the belonging degree of the resources analyzed to extracted concepts; the subsumption degree of related concepts.
  - *Concepts Labeling,* performs an automatic labeling of concepts of lattice extracted by previous phase.

[14] http://wordnet.princeton.edu/

- *Semantic Modeling*, a functional module that translates the mathematical model extracted in the previous phase, in a taxonomy or ontology by formalisms, introduced by the W3C as part of the Semantic Web, such as: RDF, RDFS and OWL.

The following sections provide a detailed view and an example of the execution of each phase illustrated in Figure 10.

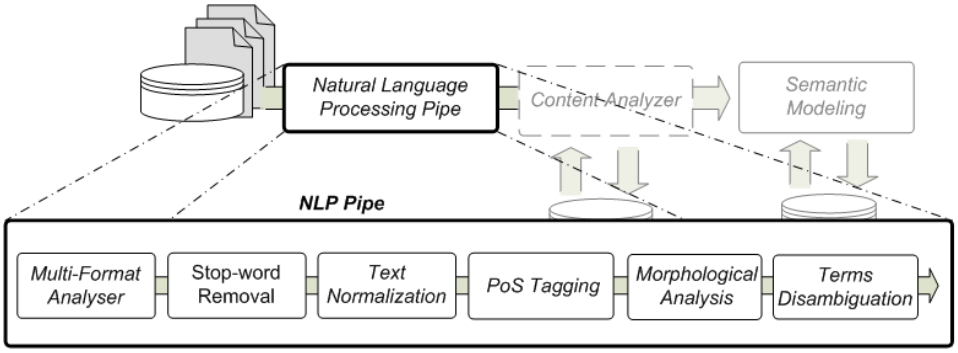



Figure 11. Knowledge Extraction: NLP Pipe.

### *4.1.1 Natural Language Processing Pipe*

The process starts by performing NLP Pipe. It consists of analyzing textual information in natural language in order to extract and disambiguate terms according to the context where them are used. As highlighted in Table 5, final result of NLP Pipe is set of disambiguated terms.

Table 5. Input/Output of NLP Pipe phase.

| **Natural Language Processing Pipe** | |
|---|---|
| Input | Digital resources of heterogeneous nature: documents and resources from Web, scientific papers, forums, blogs, wikis, etc. |
| Output | A set of most relevant terms (i.e., features), their associated senses and digital resources in which they appear. |

Figure 11 shows a detailed view of NLP Pipe. In particular, Figure 11 highlights activities of NLP Pipe (e.g., PoS Tagging and Terms Disambiguation, etc.) that take place during text parsing in order to extract the right set of terms in the resource's content. The text contents of resources are processed and lexical dictionary, such as WordNet[15], are used in order to accomplish the activities. In particular for the task of NLP Pipe we consider the following steps:

[15] http://wordnet.princeton.edu/

- ***Multi Format Analyser***: understanding the format (e.g., PDF, doc, HTML, etc) of the input documents and extracting text from them using existing parser libraries (e.g., POI for Microsoft Words, PDFBox for PDF, etc.);
- ***Stop-word removal*:** together with the non-informative words, such as articles, prepositions, conjunctions and so on, all words not included in the dictionary (i.e., Wordnet) are filtered;
- ***Text normalization***: the content of resources have words in uppercase and lower case which causes difference in handling same word with only difference in case. Then the system, normalize the text to lowercase to make it uniform;
- ***Part of Speech Tagging****:* the classification of words into lexical categories (i.e., noun, verb, etc.). Some examples of algorithms described in literature are classified as:
    - rule-based taggers [52], [53] that try to assign a tag to each word in the text using a set of hand-written rules;
    - probabilistic approaches [54], [55] that use a training text to choice the most probable tag for a word. In this work one of this approach, i.e., Tree Tagger [55] have used.

Example of PoS Tagging output.

| |
|---|
| **NNP** Microsoft **NNP** Corporation **-LRB-** (**NNP** NASDAQ: : **NNP** MSFT**-RRB-** ) **VBZ** is **DT** an **JJ** American **JJ** public **JJ** multinational **NN** corporation **VBN** headquartered **IN** in **NNP** Redmond, , **NNP** Washington, , **NNP** USA **WDT** that **VBZ** develops, , …. |

In this example, tags are assigned to words according to their lexical categories:

| Tag | Description |
|---|---|
| CC | Coordinating conjunction |
| DT | Determiner |
| FW | Foreign word |
| IN | Preposition or subordinating conjunction |
| JJ | Adjective |
| NN | Noun |
| NNP | Proper noun |
| NNPS | Proper noun |
| PDT | Predeterminer |
| PRP | Personal pronoun |
| PRP$ | Possessive pronoun |
| RB | Adverb |
| RBR | Adverb, comparative |
| VB | Verb, base form |
| VBD | Verb, past tense |
| VBZ | Verb, 3rd person singular present |

- ***Morphological Analysis****:* it is the identification, analysis and description of the structure of morphemes and other units of meaning in a language such as words, affixes, parts of speech, intonation/stress, or implied context. The Morphological Analysis is composed of *Stemming* and *Lemmatization*. Stemming is the process for reducing inflected (or sometimes derived) words to their stem, base or root form - generally a written word form. Lemmatization is the algorithmic process of determining the lemma (i.e. dictionary form) for a given word. Some example of algorithms described in literature are:
    - *Porter's algorithm* published in 1980 [56] and developed into a whole stemming framework Snowball [57]. This stemmers apply set of transformation rules to each word, trying to cut off known prefixes or suffixes;
    - *Lancaster Stemming Algorithm* [58] utilises a single table of rules, each of which may specify the removal or replacement of an ending.
    - *WordNet Lemmatizer*[16] uses the WordNet Database to lookup lemmas. In this research work this algorithm have used.

Example of stemming output.

| |
|---|
| Microsoft Corpor~~*ation*~~ NASDAQ MSFT is an American public multin~~*ational*~~ corpor~~*ation*~~ headquart~~*ered*~~ in Redmond Washington USA that develop~~*s*~~ manufacture~~*s*~~ license~~*s*~~ and support*s* a wide rang~~*e*~~ of product~~*s*~~ and service~~*s*~~ predominantly relat~~*ed*~~ to comput~~*ing*~~ through it~~*s*~~ variou*s* product divis~~*ions*~~... |

Example of lemmatization output.

| |
|---|
| Microsoft Corpor*ation* NASDAQ MSFT **be** an American public multinational corporation **headquarter** in Redmond Washington USA that **develop manufacture license** and **support** a wide rang*e* of **product** and **service** predominantly relate to **compute** through **it** various product **division**... |

- ***Terms Disambiguation****:* the activity of automatically assigning the most appropriate sense of a polysemous term by analyzing the context in which it is used. Terms Disambiguation is useful because different senses of a polysemous term can be treated as different terms of the feature set. The *Wordnet::SenseRelate*[17] algorithm has been used in my research thesis to accomplish this step.

Final output of NLP Pipe is the set of disambiguated terms with associated digital resources in which they appear.

[16] http://nltk.googlecode.com/svn/trunk/doc/api/nltk.stem.wordnet.WordNetLemmatizer-class.html

[17] http://www.d.umn.edu/~tpederse/senserelate.html

Table 6. Input/Output of Content Analyzer phase.

| Content Analyzer | |
|---|---|
| Input | A set of most relevant terms (i.e., features), their associated senses and digital resources in which they appear. |
| Output | Labeled concept lattice. |

### *4.1.2 Content Analyzer*

As highlighted in Table 6, output of the Content Analyzer phase is the hierarchical structure of concepts in the content of resources. This phase is divided into three steps *Vectorization, Concept Data Analysis and Concepts Labeling.* These steps are described in the following subsections.

Figure 12 shows a detailed view of Content Analyzer activity.

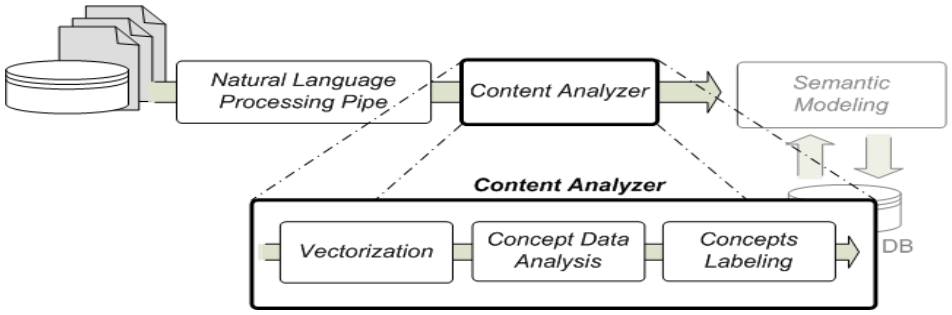



Figure 12.Knowledge Extraction: Content Analyzer.

#### *4.1.2.1 Vectorization*

The Vectorization represents the process of converting incoming resources into a vector-based model according to the terms (i.e., features) extracted during NLP tasks. The resource is represented as a vector of words, where each cell contains the weight associated to a specific word in a specific resource. This weight is calculated with well known technique of TF-IDF [59] in order to extract the set of most recurrent keywords appearing in the textual content of resources.

Table 7. Input/Output of Vectorization phase.

| Vectorization | |
|---|---|
| Input | A set of most relevant terms (i.e., features), their associated senses and digital resources in which they appear. |
| Output | A term-document matrix containing vectors associated to each input resource, in which columns are resources themselves and rows are representative features. |

Specifically, let $W = \{w_1, w_2, \dots, w_m\}$ be the set of terms extracted by means of NLP Pipe. Then, words are enriched by using WordNet synonyms according to the associated sense. In particular, the weight associated to each selected word in the content of web resources is computed constructing the set of the synonym dictionary, $D_h = \mathrm{Dict}_{\mathrm{Syn}}(w_h)$, with $w_h \in W$.

Let us compute the term-frequency $tf_{ij}^h$, as the measure of the importance of a term $w_i$ belonging to a synonym dictionary $D_h = \{w_1, w_2 \dots, w_h, \dots, w_l\}$ in the content of the resource *j*:

$$tf_{i,j}^h = \frac{f_{i,j}^h}{\sum_{k=1}^{l} f_{k,j}^h}$$

where $f_{ij}^h$ is the number of occurrences of some term $w_i \in D_h$ in resource j.

Then, fixed the set $D_h$, let us select the maximum value $tf_{ij}^h$, among all the terms $w_i \in D_h$ (i=1, … , l) in the resource j:

$$tf_{\hat{\imath},j}^h = max_{i=1,\dots,l}\left(tf_{i,j}^h\right)$$

where $\hat{\imath}$ is the index of terms $w_i$ corresponding to maximum value $tf_{ij}^h$.

This value is exploited to compute the final value that characterizes the frequency associated to each synonym dictionary $D_h$:

$$\widehat{tf}_{h,j} = tf_{\hat{\imath},j}^h \times \left(1 + \sum_{\substack{i=1 \\ i \neq \hat{\imath}}}^{l} tf_{i,j}^h\right)$$

In particular, this sum represents the relevance of the dictionary $D_h$ with respect to the given web resource j.

Then, given all m dictionaries, let us select the maximum value $\widehat{tf}_{m,j}$, in the resource *j*:

$$\widehat{tf}_{\hat{\imath},j} = max_{i=1,\dots,m}\left(\widehat{tf}_{m,j}\right)$$

Finally, according to the augmented normalized term-frequency [60] the final weight associated to the dictionary $D_h$ of the word $w_h$ for the resource j, is:

$$wtf_{h,j} = 0.5 + \left(0.5 \times \widehat{tf}_{h,j} / \widehat{tf}_{\hat{\imath},j}\right)$$

At this point, for each web resource j, the associated vector is compound of all the weight $wtf_{h,j}$, relative to each synonym dictionary $D_h$ for all the words in the set W (see Figure 13).

All generated vectors form the matrix which represents the Fuzzy Formal Context. It constitutes the input of data analysis techniques such as fuzzy FCA performed in the next step (see Section 4.1.2.2).

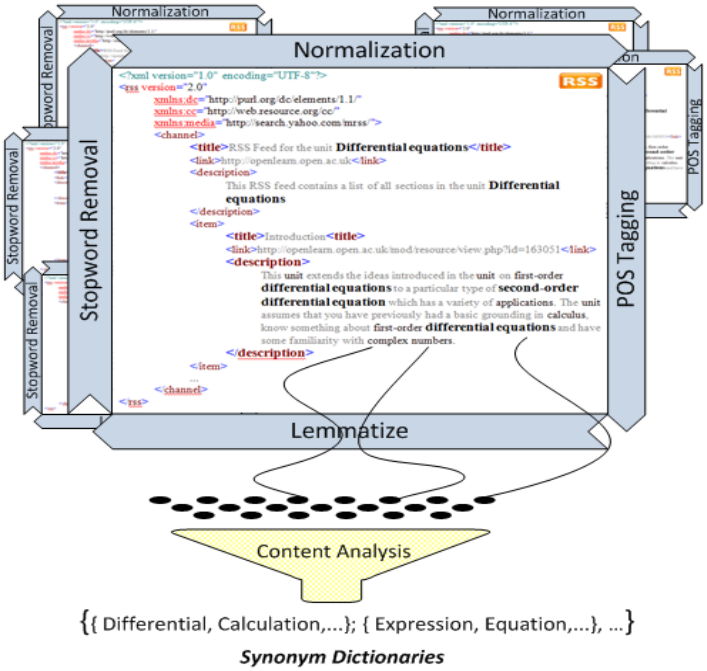



Figure 13. Filtering of features in the Vectorization phase.

### *4.1.2.2 Concept Data Analysis*

Concepts Data Analysis means application of techniques of Data Analysis to find concepts and relationships among them by taking into account digital resources and terms in a given domain. As highlighted in Table 8, main output of the Concept Data Analysis phase is the hierarchical structure of concepts in the content of resources.

Table 8: Input/Output of Concept Data Analysis micro-phase.

| **Concepts Data Analysis** | |
|---|---|
| Input | A term-document matrix containing vectors associated to each input resource, in which columns are resources themselves and rows are representative features. |
| Output | A concept lattice representing a mathematical modeling of the extracted knowledge. |

This step takes in input the term-document matrix (that is fuzzy formal context) created in the previous step, in which each cell of the matrix is the $wtf_{h,j}$ value of the feature in respect to a specific incoming resource. By considering this term-document matrix, Concept Data Analysis (i.e., application of technique of *Fuzzy FCA* described in sections 2.3) is aimed at

arranging digital resources (that are named objects) and features (that are named attributes) according to the shared meaning. Intuitively, we are interested in grouping together the maximum number of objects that share the same set of attributes, and viceversa.

Specifically, through formal contexts, they enable the representation of the relationships between objects and attributes in a given domain. Formal concepts can be interpreted from the concept lattice. The concept lattice represents a mathematical modeling of knowledge which is more informative than traditional treelike conceptual structures [61]. In fact, the lattice represents a structure which, with no human mediation, reveals "hidden" information about the knowledge structuring and the relationships as well as similarities between formal concepts. The resulting lattice is permanently stored in the database after a serialization process, which makes easier to consult the structure through an XML-based description. This process is crucial to enabling the query processing.

#### *4.1.2.3 Concepts Labeling*

The Concept Data Analysis process does not produce meaningful labels for the extracted concepts. So, this step defines a technique for automatic labeling, as highlighted in Table 9.

Table 9: Input/Output of Concepts Labeling micro-phase.

| **Concepts Labeling** | |
|---|---|
| Input | A concept lattice representing a mathematical modeling of the extracted knowledge. |
| Output | Labeled concept lattice. |

The Concepts Labeling is based on the lattice representation. Specifically, automatic labeling of formal concept has been accomplished as follows:

- Each concept takes the name of the most representative attribute, viz. the label of concept is the name of attribute whose membership (in the matrix of the fuzzy formal context) is the highest one. If the concept has more than one attributes with the same membership, the label is composed by the concatenation of all the names of these attributes.
- During this naming procedure, there might be nodes, or concepts with no own attributes[18]. In this case, a possible solution is to require help to domain expert or ontology designer. On the other hand, the process generates automatically a name as a concatenation of the label of the proper parents.

The following pseudo-code gives an idea about how to select the most representative name attribute as a candidate label of a concept. Let a concept lattice *C* be composed of a

[18] We refer to own attributes as the attributes of concept that aren't present in the ancestor concepts.

couple (*G, M*) where *G* is a set of objects (i.e., the analyzed resource) and *M* is a set of attributes (i.e., the features extracted by resource).

**for each concept C = (G, M) do**
  **for each attribute m ∈ M do**
    $\mu_m \leftarrow min_{g \in G} \mu(g, m)$
  **end for**
  **max ← compute maximum among all $\mu_m$**
  **label(C) ← attribute name whose $\mu_m = max$**
**end for**

In other words, according to Definition 12 (see Section 2.3), we compute $\mu_m$ (dually to the definition of $\mu_g$) and then for all the attributes, we assign, as a label, the attribute name, whose membership is the maximum among all $\mu_m$.

### *4.1.3 Semantic Modeling*

By taking into account extracted fuzzy lattice this phase is aimed at representing knowledge in a manner machine-understandable in order to allow inferencing.

Table 10: Input/Output of Semantic Modeling phase.

| **Semantic Modeling** | |
|---|---|
| Input | A concept lattice representing a mathematical modelling of the extracted knowledge |
| Output | Hierarchical conceptualization of resources' content represented by exploiting semantic technologies, such as: RDF, RDFS, OWL, SKOS, etc. |

As highlighted in Table 10, the output of Semantic Modeling phase is the mapping of the concept lattice into an ontology. Since, ontology extraction is the main research objective achieved by the specialization of the general framework for knowledge extraction, details about Semantic Modeling are given in Section 4.2.1where this phase is instantiated in order to carry out OWL ontology.

## 4.2 Knowledge Extraction: Research Objectives

The following subsections describe the objectives that have been addressed in this research work by applying the general framework described above.

### *4.2.1 Ontology & Taxonomy Extraction*

The general framework for knowledge extraction has been applied in order to build ontologies and taxonomies. Specifically, the defined approach extracts:

- *hierarchical conceptualizations* concerning with taxonomy relations among concepts of the lattice;
- *relations* and *constraints* expressed by means typical description logic constructs embedded into definition of the ontology's concepts.

Figure 14 shows the enabling steps to build an ontology.

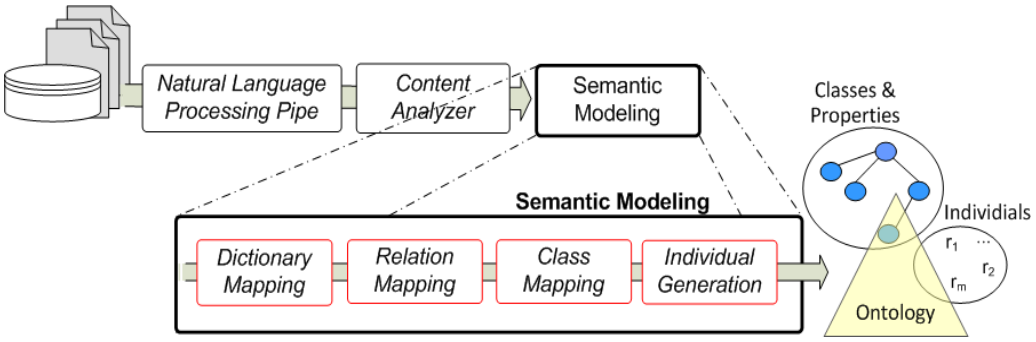



Figure 14. Ontology & Taxonomy Extraction Logical View Process

The mapping process enables us to extract the knowledge embedded in the resources collection, producing the ontology conceptualization and population. In particular, Figure 15 shows a portion of fuzzy formal context and the relative concepts in the lattice (on the left hand); the corresponding generated ontology and population are shown on the right hand.

The mapping of a concept of the fuzzy lattice to ontology class exploits the "owl:Class" construct; instances of these class, i.e. individuals, represent the extent of concepts of the fuzzy lattice; the mapping of the intent (i.e. attributes) to ontology properties exploits "owl:DatatypeProperty" and "owl: ObjectProperty". Moreover, each attribute in a concept of the lattice is mapped as an ontological class that represents the dictionary of its synonyms.

The mapping concerns the formal concepts and the super/sub-concept relations, evinced into the lattice, as described formally in the following definitions.

> **Definition 16.** Let K = (φ(A),B) be a fuzzy concept of the fuzzy formal context (G, M, I). The function of mapping M on K produces a class C, where the objects of the extension A $\subseteq$ G becomes individuals or instances of the concept C while the attributes of the intent B $\subseteq$ M describe the properties relative to the class concept C.

Thus, let $C_1$ and $C_2$ be the classes obtained by the mapping of two fuzzy formal concepts $K_1=(\varphi(A_1), B_1)$ and $K_2=(\varphi(A_2), B_2)$ of the fuzzy formal context $(G, M, I)$, respectively. Let $K_1$ be the subconcept of $K_2$ ( i.e., $K_1 \leq K_2$). The function of mapping M on the subsumption relation $K_1 \leq K_2$ produces a correspondent subclass relation, i.e., $C_1$ is subclass of $C_2$. Equivalently, as $K_2$ is the superconcept of $K_1$, then $C_2$ is a superclass of $C_1$.

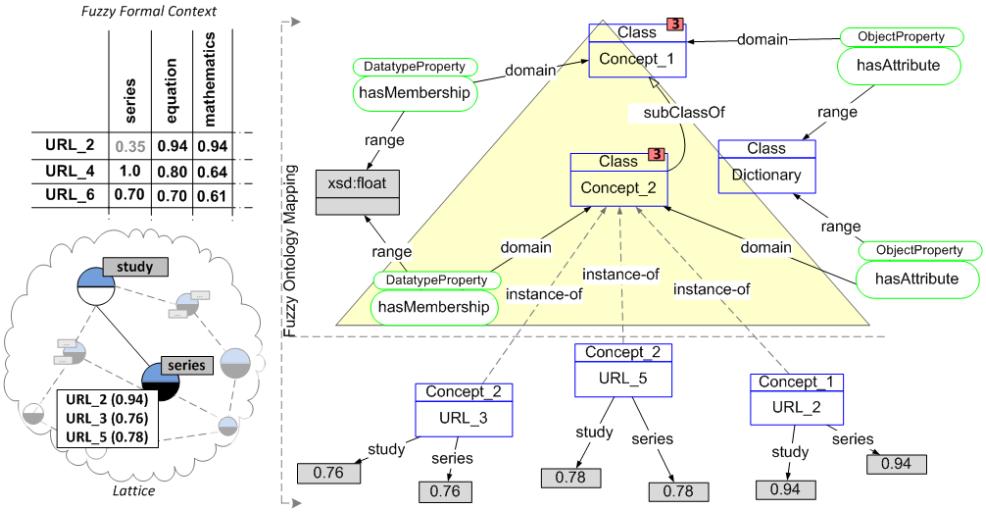



Figure 15. Ontology Extraction from Fuzzy Formal Analysis

The building of the ontology is based on the object-attribute relationships of the formal context. In the proposed approach, the objects of the fuzzy formal context are web resources and attributes are keywords extracted by parsing the content of these resources. According to Definition 16, the translation of a fuzzy concept is described by the following mapping steps. The mapping of a fuzzy concept in Definition 16 requires the characterization of the class and its properties, and then the specification of the related population. Specifically, four steps have been defined as follows:

- *Dictionary Mapping*: each attribute (i.e., keywords/words/terms) belonging to a fuzzy concept, is mapped onto a Synonym Dictionary that represents the set of its synonyms. In other words, for each attribute (i.e., *study*) belonging to a concept, we have defined an ontological class that represents the dictionary of synonyms (i.e., *StudyDictionary*). More precisely, we define an abstract class called *Dictionary*. Its specialization represents an effective class composed of a set of synonyms associated to a specific term. For instance, Figure 16 shows the OWL code of the class *StudyDictionary* associated to the attribute *study* and composed of two other terms "learning" and "work", besides the term itself "study".

**Dictionary Mapping**

| | study | series |
|---|---|---|
| URL_2 | 0.94 | 1.00 |
| URL_3 | 1.00 | 0.76 |
| URL_5 | 0.78 | 1.00 |

```
<owl:Class rdf:about="#StudyDictionary">
    <owl:equivalentClass>
      <owl:Class>
        <owl:oneOf rdf:parseType="Collection">
          <rdf:Description rdf:about="#Learning"/>
          <rdf:Description rdf:about="#Study"/>
          <rdf:Description rdf:about="#Work"/>
        </owl:oneOf>
      </owl:Class>
    </owl:equivalentClass>
    <rdfs:subClassOf rdf:resource="#Dictionary"/>
</owl:Class>
```

Figure 16. Dictionary Mapping Step

- *Property/Relation Mapping:* two kinds of *OWL properti*es have been defined from mapping the attributes in the concepts of the fuzzy lattice:

    - *hasAttribute* is an owl:ObjectProperty that enables us to associate each attribute (i.e., word) of the intent of a concept of the fuzzy lattice to a corresponding object (i.e., resource). The domain of OWL Objectproperty is the top concept (i.e., *Concept_0*) and the range is the class *Dictionary*, as shown in Figure 17.

    - *hasMembership* is an owl:DatatypeProperty that represents the membership values associated to object, according to Definition 12. The domain of *hasMembership* is the top concept (i.e., *Concept_0*) while the range is a float datatype. This property allows us to associate a membership value to an actual individual of a class.

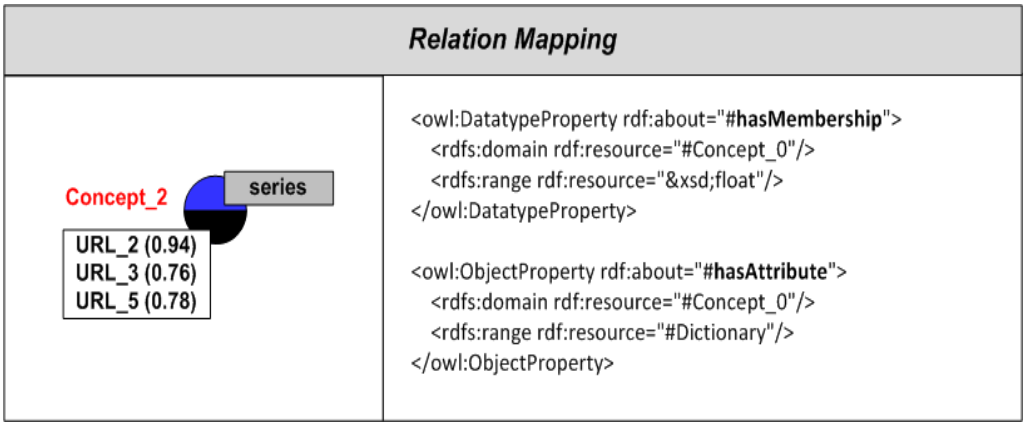



Figure 17. Relation Mapping Step

- *Class Mapping*: as said, each formal concept of the lattice becomes an ontology class. The construct *owl:Class* describes it. The proposed approach automatically produces a class name, identified by a specific progressive number. In Figure 18, the OWL code describes Concept_2 (evidenced in the lattice too) which has some value from the classes *SeriesDictionary* and *StudyDictionary*, for the property *hasAttribute*. This mapping allows Description Logic (DL) reasoner to infer which attributes are associated to a concept and vice versa. In general, this association enables DL reasoner to classify new objects (resources) according to their content (words). Once the classes are defined, an automatic labeling will be applied on the concepts, as detailed in the following sections.

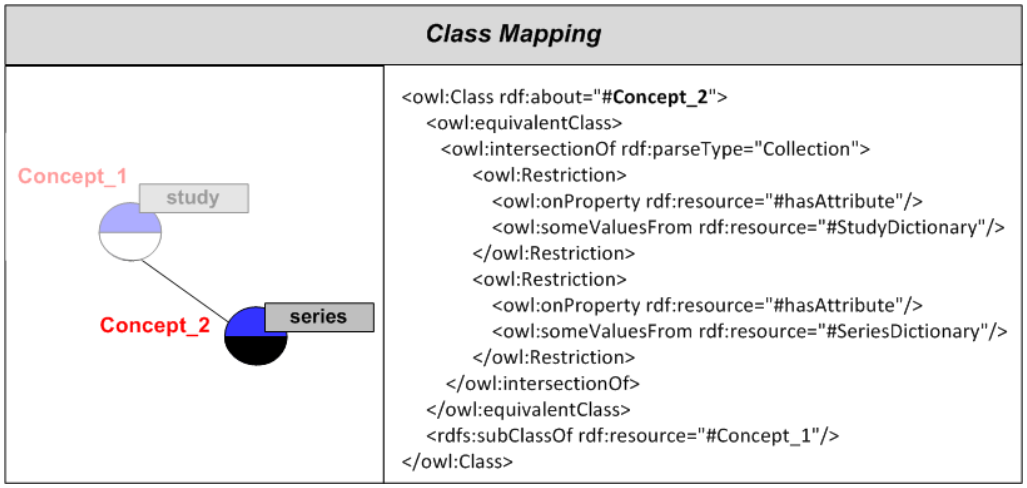



Figure 18. Class Mapping Step

- *Individuals Generation:* for each web resources in the extent of a concept, an instance of the corresponding ontology class is generated. The individuals of the class *Concept_2* of Figure 19 are URL_2, URL_3 and URL_5; furthermore they have associated two attributes (as evidenced in the lattice in Figure 2): the inherited attribute *study* and its own attribute *series*. In particular, the OWL code describing the individual URL_2 is shown.

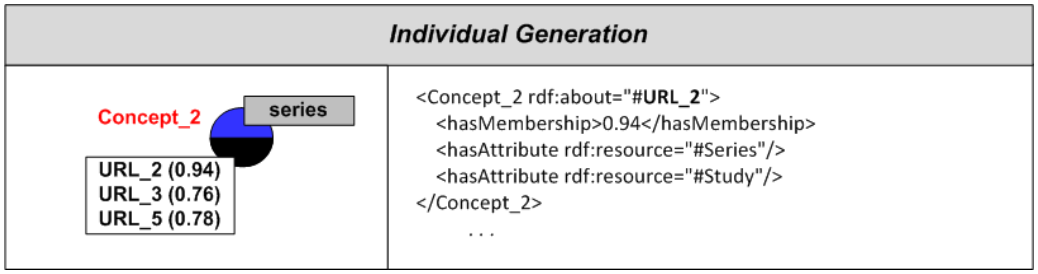



Figure 19. Individual Mapping Step

### *4.2.2 Semantic Annotation*

*Semantic Annotation* applies the general framework to annotate incoming resources. Obviously, the general framework has been enriched with new phases that are shown in Figure 20. The process is designed to manage different types of resources (e.g., textual information, images, etc.); through a data-driven processing it yields an OWL-based annotation, easily adaptable to an ad-hoc built ontology. The real potentiality is the generation of knowledge structuring in automatic way, without any compulsory human intervention.

The approach accomplished by the process can be applied to different information sources tuning each component to set up the specific data process, according to specific source format and type.

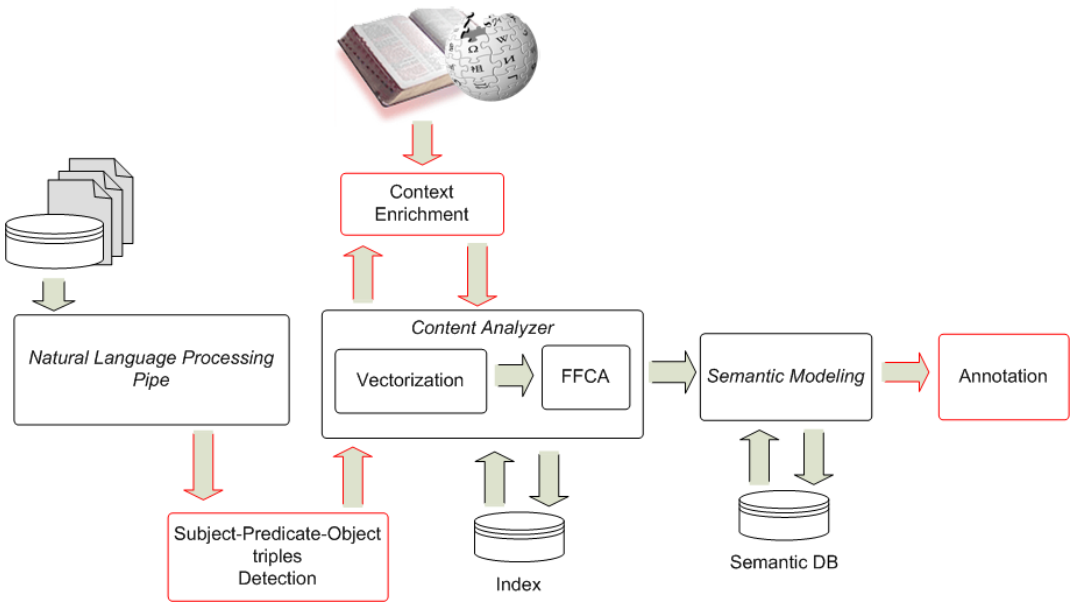



Figure 20. Semantic Annotation Logical View Process

Differentially from the logical view shown in Figure 10, this process adds three new phases (in red color in Figure 20) *Subject-Predicate-object triples Detection*, *Context Enrichment* and *Annotation*, each one accomplishes a specific task. Furthermore, the *Content Analyzer* process is adapted in order to perform different analysis on subject, predicate and object (i.e., different context/lattice for each one is building) returned from previous step. Specifically:

- *Subject-Predicate-object triples Detection.* This phase extracts triples with the form <subject, predicate, object> from the analysis of the given resource. Depending on the input resource and the type of semantic annotation, ad hoc data analysis is proactively set up. For instance, in the case of resources from the geographic domain, the semantic annotation refers to geographic maps, whose data are accessible by RSS-feeds. The *Subject-Predicate-object triples Detection* extracts triples from feeds content with the following semantics: <Location(lat, long), hasTypeInfo, Info>, that means that a specific location of the map has associated an information (from an RSS feed) of a certain type. Another example of input resources considers web pages. In this case, this module behaves as a usual annotation tool. Specifically, if the annotation regards the text content, the *Subject-Predicate-object triples Detection* generates triples such as <noun, verb, noun>, exploiting natural languages processing techniques. If the annotation refers to links among web pages, the module explores hyperlink among pages to generate triple such as <pageX, anchorName, pageY>. Additional details can be provided in the instantiation of the process to a specific application domain on relative resources. Thus, according to nature of the processed resources, an ad-hoc component must be designed, in order to generate and gather suitable triples.

- *Context Enrichment.* This phase enrich the context of the input element. Thus, for each triple, subject and object are processed in order to generate a *context enrichment* of them. They collect additional information from external (on-line) sources, in order to augment the meaning (context) of the subjects and objects. The context enrichment is an activity which strictly depends on the type of processed resources. For instance, let us consider the previous example about geographic maps, the *Context Enrichment* adds information about the *Location(lat, long)*, such as: the name of the city or country associated to that *location*.

- *Annotation.* This phase generates an ontology which represents a structured semantic annotation of input resources. In particular, it gets all the processed data from the analyzers in form of lattices generated by FFCA modeling. Exploiting the collected data and structures, this phase accomplishes fuzzy *Relational Concept Analysis* (described in section 2.4), a model to enhance the conceptual structures of FCA, with additional relations among concepts. In fact, it generates an RCA lattice that reveals also no-hierarchical relationship. After the generation of the lattice, a mapping process is set up to transform the lattice into OWL ontology. Result is the generation of an ontology which represents a structured semantic annotation of input resources.

The process is applicable to several kinds of resources. Obviously, according to the specific resource format, some ad-hoc tailored arrangements are necessary for processing the resources. The Chapter 7 is devoted to deeply depict the sketched process through all its steps by presenting an example framed into an insightful application case study, i.e. automatic semantic text annotation.

### *4.2.3 Information Retrieval*

The general framework has been applied in order to support *Information Retrieval* activity.

The process of Information Retrieval foresees two main activities: *Training* and *Query Processing*. Training phase is essentially the general framework with the ontology extraction described in Section 4.2.1. Query Processing browse the ontology in order to provide pertinent results. In particular, Query Processing re-adapts the standard measures of the precision, recall and F-measure [62] to the applicative context. The goal indeed is to get a synthetic measure which represents how each resource (i.e., individual belonging to a class) is semantically relevant with respect to a user query. To do this, we use the F-measure which is based on a harmonic mean function on the values of precision and recall. This way, a ranked list of resources is the result to a given query.

Specifically, the activities are:

- *Training*, that is used to analyze incoming data and to consequently build the ontology.

  Figure 21 details the Training component, showing its processing activities. Mainly, the process is based on the general framework explained in Section 4.1. In particular, there is a pipe of activities: *Natural Language Processing Pipe,*

*Content Analyzer* and *Semantic Modeling*. At the end of the process the extracted knowledge is stored in the Semantic Data Base. Training is crucial to enable the Query Processing process of the next component.

- *Query Processing* is the activity that browses the ontology according to the user's query in order to discover relevant results among training data. Figure 22 emphasizes the main activities foreseen in this process. The input query is in the form of free text or concept based query (e.g., fragment of taxonomy, etc.). Furthermore, a confidence *threshold* is specified in input in order to get the desired precision. The Query Processing guarantees a comparative evaluation between the given query and the formal concepts of the extracted ontology in order to discover the similarity in the meaning between the respective concepts. Specifically, following sub tasks are involved:
  - *FFCA-based Classification:* this task computes the similarity value between query concepts and ontology concepts. More formally, let us define:
    - $D = \{d_1, d_2, \ldots, d_n\}$ the whole set of dictionaries (see Section 4.1.1) defined for each attribute (terms and keywords), $A = \{a_1, a_2, \ldots, a_n\}$ of a given fuzzy formal context;
    - $I = \{i_1, i_2, \ldots, i_t\}$ the whole set of individuals (resources) of a given OWL ontology;
    - $C_i = (D_i, I_i)$ i-th ontology class (where $I_i \subset I$ represents the individuals and $D_i \subset D$ the synonym dictionaries associated to attributes of $C_i$);
    - $Q_i = \{q_1, q_2, \ldots, q_m\}$ a query, i.e. a set of attributes (i.e., terms) to search in the lattice.

    Thus, given an ontological class $C_i$ we define a local precision $P_i$ and recall $R_i$

$$P_i = \frac{|Q_i \cap A_j|}{|A_j|} \qquad R_i = \frac{|Q_i \cap A_j|}{|Q_i|}$$

    where,

    - $A_j$ is a set of all the attributes in the dictionaries associated to the attributes of $C_i$;
    - $P_i$ can be seen as a measure of *exactness* or *fidelity* of ontological class $C_i$. It is computed as ratio between the number of *relevant* attributes of dictionaries in $C_i$ and all the attributes of associated to $C_i$.
    - $R_i$ represents a measure of *completeness* of ontological class $C_i$ and it is given as the number of *relevant* attributes of dictionaries in $C_i$ divided by the total number of query terms.

The intersection is computed by applying Wu & Palmer similarity [63]. The F-measure value $F_i$ relative to the class $C_i$ is computed as follows:

$$F_i = 2 \times \frac{P_i \times R_i}{P_i + R_i}$$

Final result of this computation is the list of the F-measure values $F = \{F_1, \dots, F_s\}$ on all the ontological classes in the given ontology, given a query.

- *Filtering & Ranking*: now, for each individual of ontology, $i_k \in I$ we calculated its own *score* as follows:

$$score(i_k) = \sum_{j=1}^{s} (\mu_j(i_k) \times F_j)$$

where $\mu_j(i_k)$ represents the membership of resource $i_k$ mapped in the ontology (*hasMembership*, see Section 4.1.3). This synthetic value represents how this resource (i.e., individual) is relevant with respect to the given query. The result set is composed of $i_k$ whose $score(i_k)$ is greater than input confidence threshold.

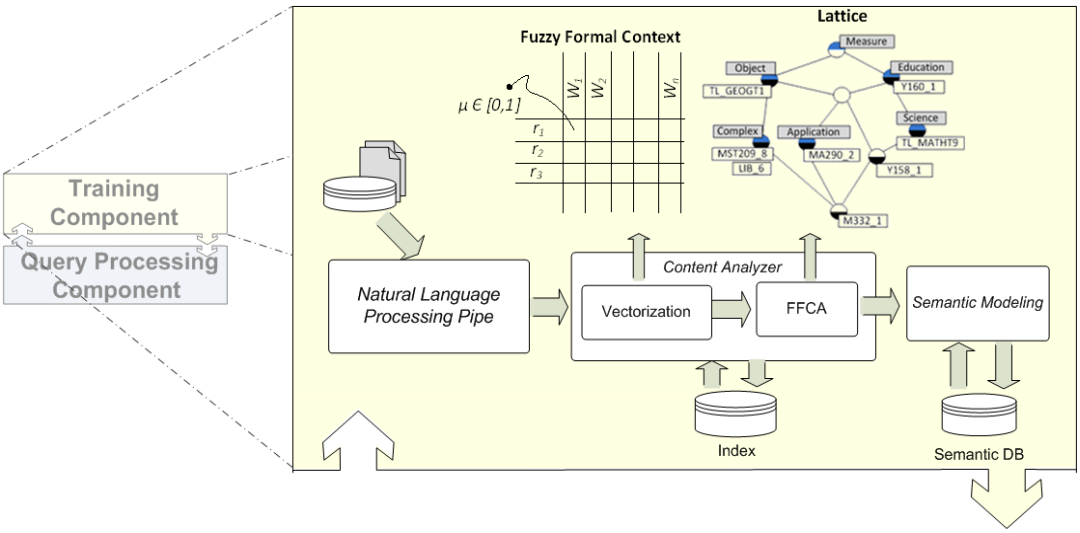



Figure 21. Workflow of Training activity.

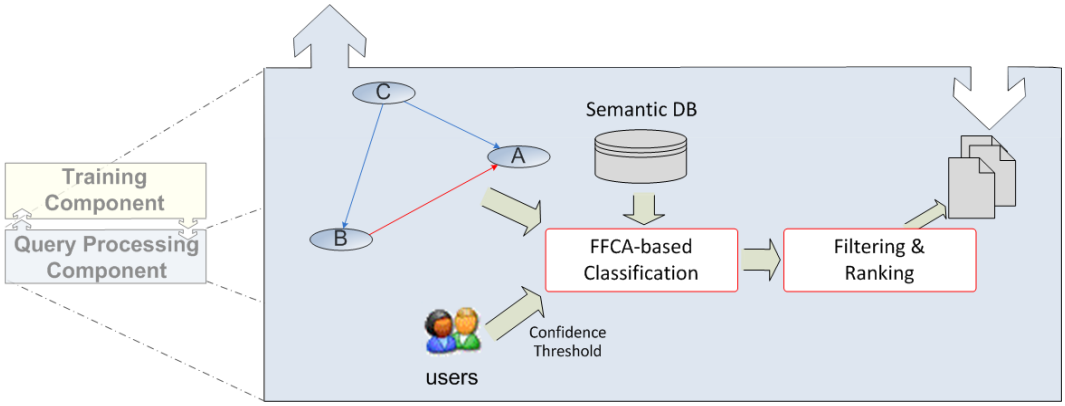



Figure 22. Workflow of Query Processing activity

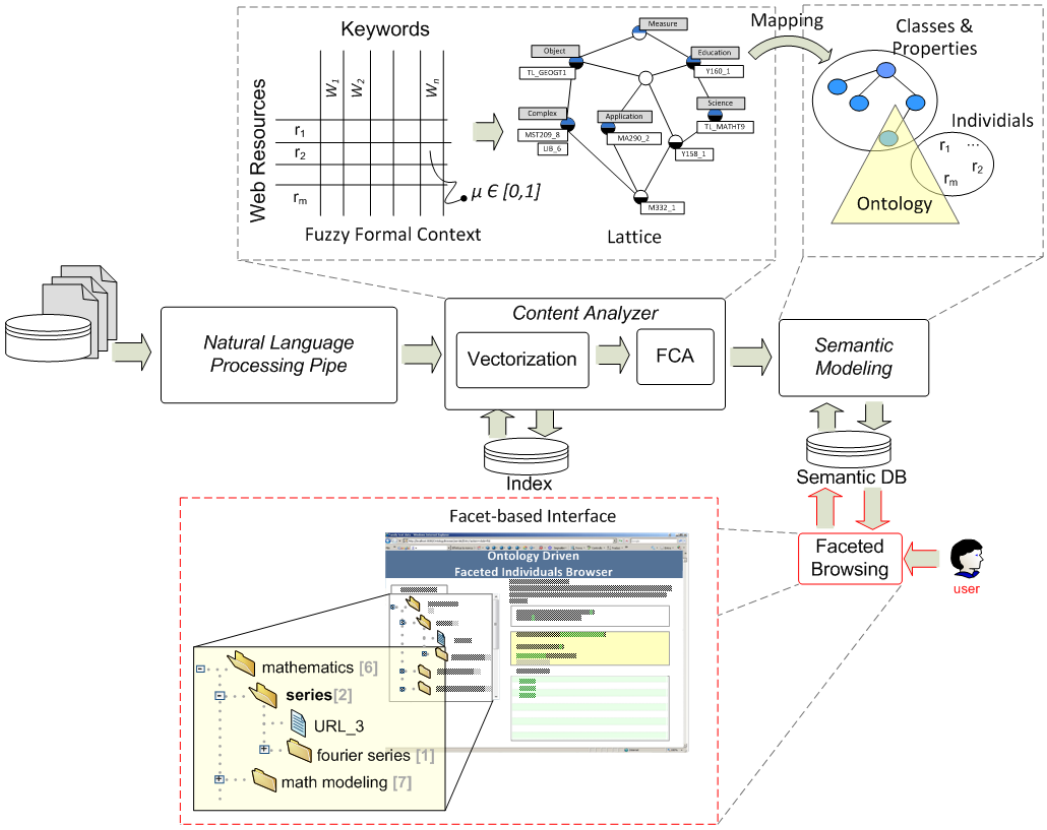



Figure 23. Faceted Browsing Logical View Process

### *4.2.4 Faceted browsing*

The faceted browsing [64] is an efficient technique for accessing a collection of information (or resource content) through filtering of search space. The goal of this data exploration technique is to restrict the search space to a set of relevant resources. Unlike a simple traditional hierarchical category scheme, users have the ability to drill down to concepts based on multiple dimensions. These dimensions are called facets and represent important characteristics of the information units. Each facet has multiple restriction values and the user selects a restriction value to constrain relevant items in the information space. A known project, called SWED[19] exploits the facet navigation; it uses a number of thesauri, ontologies and lists to categorize publish and arrange data in its directory web site. The facet based navigation can improve Semantic Web approaches which make Web-based data more accessible via the use of Community Portals – i.e. Web-portals that provide customized 'views' of information [65]. Usually, the facets have to be defined by domain knowledge expert and the resources have to be manually classified respect to that knowledge model. In this research work, the faceted browsing is dynamically carried out according to the content of analyzed resources.

Specifically, *Faceted Browsing* is strictly related to the general framework (see Section 4.1) and to the Ontology and Taxonomy Extraction (see Section 4.2.1). In particular, the overall process is sketched in Figure 23. The facet-based representation allows exploration among concepts and instances. Specifically, the new phase, i.e. *Faceted Browsing,* phase takes as input the OWL representation of the extracted lattice and produces a graphical representation, through the multifacet-based GUI.

Figure 23 provides a sketched overview of the whole process of mapping starting from the lattice representation to the ontology tree. The lattice is mapped into an ontology schema and relative population, through OWL language, as described above. Then, the OWL ontology, represented by a navigation tree, will be the browsable structure in the interface defined. Similar to Protegè editor [66], the proposed tree structure enables the multiple inheritances, by duplicating the concepts whenever they appear in a path between concepts. This generates different views of subsumption paths in the tree.

Specifically, Figure 24, the mapping process is evidenced for the class in the OWL ontology, named *Concept_2*. Note that *Concept_2* is from the mapping of the formal concept *c2* in the lattice; thus it is translated into a concept of the ontology tree. The own individuals of *Concept_2* appear, by clicking for expanding the relative node in the tree (see , Figure 24). Let us note that only resource named URL_3 is a proper object of the *Concept_2*. Then, URL_2 and URL_5 appear in the Concept_5 which is a specialization of the Concept_2.

Moreover, each node/concept of the ontology tree shows, in squared parenthesis, the number of proper individuals, i.e., the instances associated to that concept.

Let us remark that we use the term "ontology tree" to refer the ontology generated by the mapping process even though it is a hierarchy.

[19] www.**swed**.org.uk/

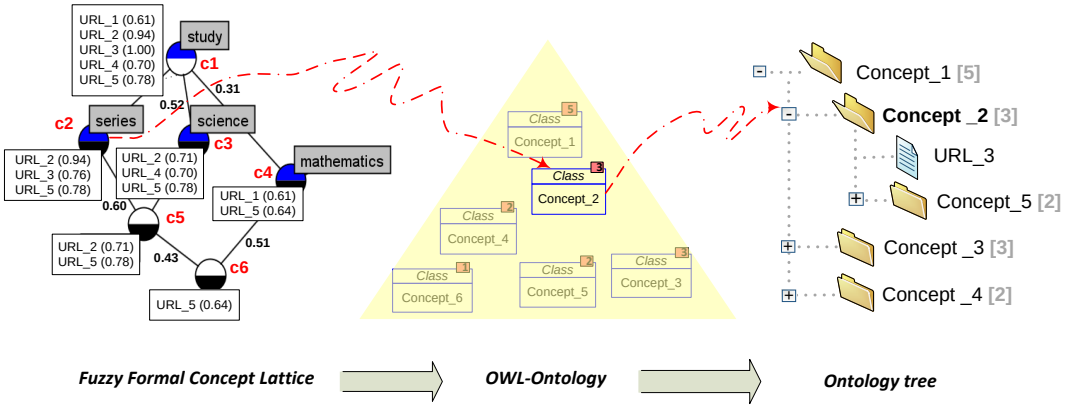



Figure 24. Ontology tree generated by OWL-based lattice representation

# Part III: Case Studies

# Chapter 5

# Automatic Faceted Browsing and Ontology-based Retrieval of web resources

This chapter presents an ontology-based retrieval approach that supports data organization and visualization and provides a friendly navigation model. This work exploits some of the research objectives described in Section 4.2. In particular, the methodologies evaluated here are:

- Ontology and Taxonomy Extraction (see Section 4.2.1), in order to extract formal and reusable model for the knowledge representation, which supports advanced queries and visualization procedures;
- Information Retrieval (see Section 4.2.3), in order to support ontology-based information retrieval of web resources according to flexible user query;
- Faceted Browsing (see Section 4.2.4), in order to provide an intuitive graphical interface that enable the browsing of the ontology concepts as well as the exploration of the relationships and the population;

In particular, the chapter is organized as follows. Section 5.1describes the user interface for the facet-based ontology exploration. Section 5.2 is devoted to presenting experimental results. Finally, conclusions close the chapter.

## 5.1 Faceted browsing of web resources

Due to the lattice-generated ontology, the built ontology tree reveals intrinsic relationships among attributes or objects, through the exploration of concepts and relative population. This allows a flexible navigation among its concepts and provides many different criteria to explore and retrieve its data.

More emphasis to the facet-based navigation is given by a visual query paradigm [68]: the user builds gradually a query by exploring the concepts/subconcepts in the ontology tree and, selecting a concept, new constraints are added: each action executed on the interface is a step for the query construction. Moreover, the user sees intermediate results of query while he's browsing the tree.

In order to facilitate the exploration and the navigation of an ontology as well as its own individuals (i.e. resources), a user-friendly interface has been designed. Figure 25 shows the graphical interface of proposed system. It is composed of two main parts: on the left, the labeled ontology tree for the exploration and, on the right, a web page which provides details about the concept selected in the ontology tree. Once selected a concept on the tree, additional details about it appear on the right hand. In fact, as shown in Figure 25, by clicking on

the concept named "*mathematics"*, on the tree, on right hand of the interface this concept appears with its descendants: all the subconcepts and individuals.

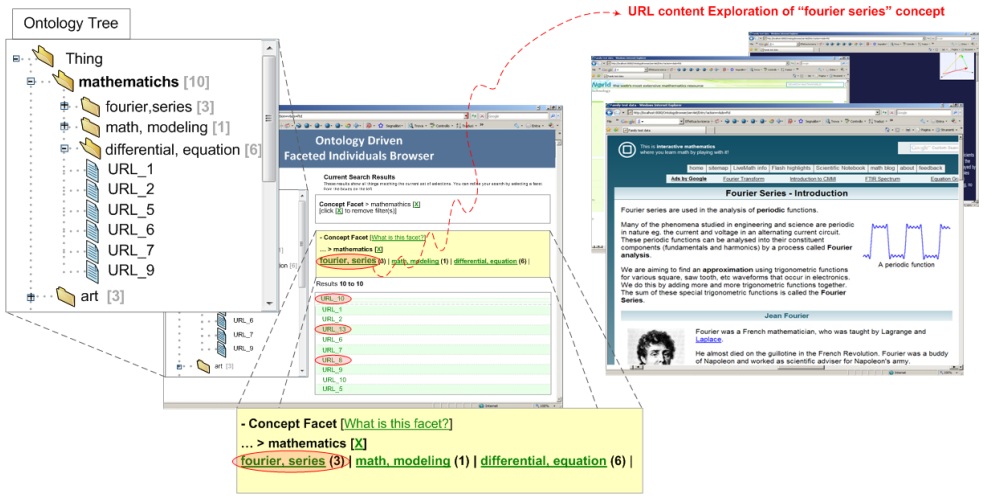



Figure 25. Ontology representation through facet-based Web Interface

The navigation can proceed forward by considering the subconcepts of "*mathematics"* (frame in the foreground, at the bottom of Figure 25, "*fourier, series", "math, modeling"* and "*differential, equation"*) and the relative individuals (in the lower frame).

Note that, in this case study the application domain is the Web thus, the individuals, i.e., instances are the URLs of web pages that represent the concept. Thus, in Figure 25 the individuals of the concept labeled "*fourier, series"* i.e., some effective web pages are emphasized. The frame in foreground, at the bottom of Figure 25 represents a facet of "*mathematics"* in the ontology tree. This facet can be interpreted as an implicit query on the system which returns all pages related to mathematics; in particular, the system not only returns all the related pages, but classifies them according some elicited categories, in this case, three: *"fourier, series", "math, modeling"* and *"differential, equation"*. In summary, the ontology tree provides a general view of the global structure, i.e., objects and relative individuals; details about an object appear on the right hand. In particular, clicking on a concept, the first sub level of structure is shown in the frame in the foreground, at the bottom of Figure 25, the lower frame instead, provides the list of all the involved objects, descendant of the object.

The synergy between semantic technologies and fuzzy data analysis allows an automatic characterization of the subject domain and its categorization with respect to ontology concepts.

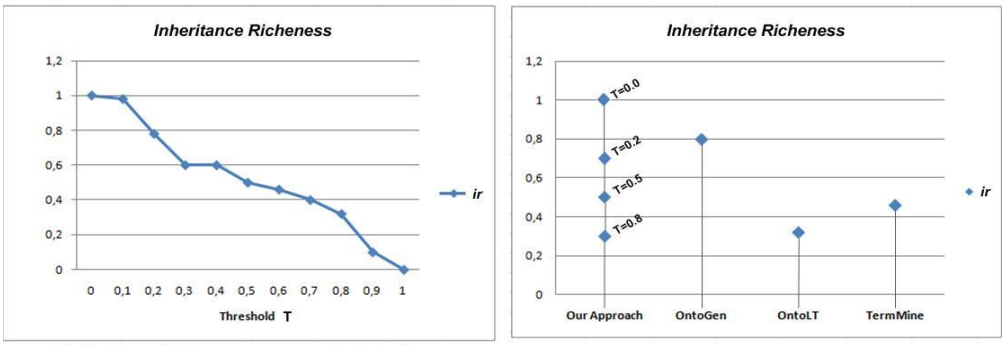
Moreover, the fuzzy FCA based construction of the ontology tree enables the generation of a family of ontologies that may have a deep and specialized structure or just a high level abstraction conceptualization. In fact, by varying the threshold $T$, related to fuzzy context (see Chapter 2), some relations between objects and attributes are discarded, thus a different ontological structure is generated (additional details are given in the following section). This scalability is one of the main required properties in facet-based navigation, especially when the dataset has a considerable size.

Finally, the ontology tree generated by proposed approach *can be exploited* as a controlled vocabulary. Generally, cataloging or indexing systems (e.g. libraries and product catalogues, etc.) use controlled vocabularies, i.e. they index all items using a constrained set of terms (e.g. in the Dewy decimal system for classification of books, a book about zoology will be classified under the term "Zoological Sciences"). The use of controlled vocabularies simplifies the consistency maintenance of the classification and makes it easier to find relevant items.

## 5.2 System validation and experimental results

The approach has been tested on a collection of web resources extracted from OpenLearn Project[20], a public, online accessible repository of learning materials. OpenLearn provides course materials from the Open University manually arranged in categories, according to the main educational subjects and courses.

The goal is to automatically elicit a categorization of collected materials, and then to evaluate the retrieval performance by comparison with OpenLearn categories. The sample is composed of 488 web resources. Let us outline the analysis of the computed ontology which can be achieved at two different abstraction levels: on one hand, by analyzing the categorization coming from the ontology structuring, generated by the lattice; on the other hand, by comparing the meaning of concepts (by the analysis of the enclosed resources and terms) and the naming to the OpenLearn categories.




*(a) Inheritance Richeness curve evaluated with our approach by varying Threshold T.*

*(b) Inheritance Richeness comparison.*

Figure 26. Inheritance Richness tendency by varying the threshold value

### *5.2.1 Analysis of the extracted ontology structure*

This framework achieves a fair arrangement of web resources through the extraction of the ontology tree, and supports hierarchical exploring and query processing. Specifically, two are the main contributions:

[20] http://openlearn.open.ac.uk/

- A mechanism for automatically generating web resource classification according to the conceptualization form the ontology;
- A tool for querying the tree in a hierarchical or free-form manner.

Literature is rich of many methods that extract ontologies but do not annotate the content during the process of ontology extraction. The proposed approach instead, produces the annotation of the resources. By exploiting the Formal Concept Analysis theory, objects are grouped by in the concepts of the fuzzy lattice. Each concept is described (i.e., annotated) by the attributes associated to the resources in that concept.

Because the ontology built by fuzzy FCA is a taxonomic structure, it is difficult to compare the proposed framework with other tools for ontology extraction that extract relations among concepts. Comparisons with other approaches would be based on the analysis of the generated ontology structure in term of resources distribution, class hierarchy depth and wideness.

We exploit the measure of *inheritance richness (briefly **ir**)* [67] for the ontology structure analysis. The ***ir*** is a measure which describes the distribution of information across different levels of the inheritance tree or, in other words, the fan-out of parent classes. More formally:

**Definition 17.** The *inheritance richness (**ir**)* of a ontology schema is defined as the average number of subclasses per class:

$$\boldsymbol{ir} = \frac{\sum_{C_i \in C} |H^C(C_s, C_i)|}{|C|}$$

where $H^C(C_s, C_i)$ is the set of classes $C_S$ that are subclasses of $C_i$; $H^C$ is a hierarchy of classes ($H^C$ is a directed, transitive relation $H^C \subseteq C \times C$ which is also called class taxonomy) and $C$ is the set of all the classes.

Figure 26 shows the ***ir*** measure, according to the nature of the ontology structure. Specifically, Figure 26(a) shows the inheritance richness of the proposed ontology extraction system using a resources sample from OpenLearn, by considering different threshold $T$ values. Recall the fuzzy extension of the lattice is strictly related to a confidence threshold $T$ (see FFCA theory).

As said, fuzzy FCA allows the building of a family of fuzzy formal lattices by varying the threshold $T$. The lattice structure can change varying $T$, because some relations can be revealed or pruned, according to the values associated to the entries of the fuzzy formal context.

Compared with other tools, this model allows a flexible construction of an ontology which, by varying the threshold, can provide a high-level conceptualization of the modeling domain or a deep specialization making explicit all the relations. For instance, fixing the threshold $T = 0.5$ (Figure 26(a)) means to consider the ontology (lattice) generated by a fuzzy formal context whose values associated to the relative matrix are greater or equal 0.5. Figure 26(a) emphasizes that the ontology presents quite high values of ***ir*** when the threshold is smaller than 0.5. A high value of ***ir*** characterizes an "horizontal" ontology [67], i.e. an ontology with a small number of inheritance levels, where each class has a relatively large number of subclasses. This means a general high level representation of content.

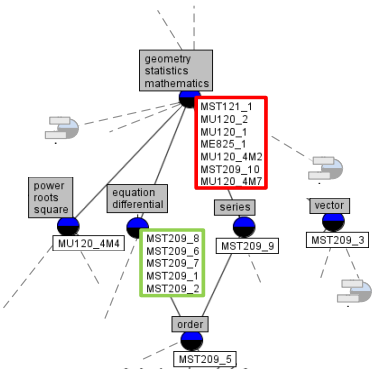
On the other hand, the ontology with a low ***ir*** is considered "vertical", i.e. it is composed by many inheritance levels and the classes have a small number of subclasses.

| | mathematics | statistics | geometry | differential | equation |
|---|---|---|---|---|---|
| MST121_1 | 1.00 | 0.83 | 1.00 | | |
| MU120_1 | 1.00 | 0.27 | 1.00 | | 0.17 |
| MU120_2 | 0.63 | 0.58 | 0.63 | | 0.19 |
| ME825_1 | 1.00 | 0.35 | 1.0 | | |
| MU120_4M2 | 1.00 | 0.83 | 1.0 | | |
| MST209_10 | 1.0 | 0.83 | 1.0 | | |
| MU120_4M7 | 1.0 | 0.83 | 1.0 | | |
| MST209_1 | 1.0 | 0.83 | 1.0 | 0.83 | 0.83 |
| MST209_2 | 0.4 | 0.67 | 0.75 | 1.0 | 1.0 |
| MST209_6 | 1.0 | 0.83 | 1.0 | 1.0 | 1.0 |
| MST209_7 | 0.88 | 0.75 | 0.88 | 1.0 | 1.0 |
| MST209_8 | 0.88 | 0.75 | 0.88 | 0.75 | 0.75 |

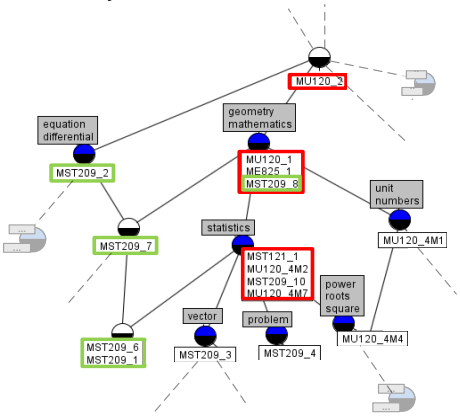
Fuzzy Formal Context with threshold T=0.2

| | mathematics | statistics | geometry | differential | equation |
|---|---|---|---|---|---|
| MST121_1 | 1.00 | 0.83 | 1.00 | | |
| MU120_1 | 1.00 | 0.27 | 1.00 | | 0.17 |
| MU120_2 | 0.63 | 0.58 | 0.63 | | 0.19 |
| ME825_1 | 1.00 | 0.35 | 1.0 | | |
| MU120_4M2 | 1.00 | 0.83 | 1.0 | | |
| MST209_10 | 1.0 | 0.83 | 1.0 | | |
| MU120_4M7 | 1.0 | 0.83 | 1.0 | | |
| MST209_1 | 1.0 | 0.83 | 1.0 | 0.83 | 0.83 |
| MST209_2 | 0.4 | 0.67 | 0.75 | 1.0 | 1.0 |
| MST209_6 | 1.0 | 0.83 | 1.0 | 1.0 | 1.0 |
| MST209_7 | 0.88 | 0.75 | 0.88 | 1.0 | 1.0 |
| MST209_8 | 0.88 | 0.75 | 0.88 | 0.75 | 0.75 |

Fuzzy Formal Context with threshold T=0.8




(a) Inheritance Richness = 0.78;




(b) Inheritance Richness = 0.32;

Figure 27. Fuzzy contexts with relative generated lattices (by varying a threshold T) and the computed inheritance richness (ir) values

A more accurate specialization is provided in some path from the concept root to the concept leaf.

Comparisons with other tools in term of the inheritance richness measure are shown in Figure 26(b). OntoGen[21], OntoLT[22] and TermMine[23] are the candidate tools. Let us evidence that, for these systems, the evaluation considers a fixed ontology, because no input thresholds exist. In general it is not possible to know a-priori which value of ***ir*** is the best one. Figure 26(b) shows that, with ***ir*** $\leq 0.5$ and the threshold $T \geq 0.5$, the proposed approach generates a vertical ontology comparable to value computed for TermMine, while similar value appear for OntoLT and the proposed approach when $T = 0.8$.

Figure 27 shows some sketched views of the lattice by considering two values of threshold $T$, 0.2 and 0.8 respectively (the relative fuzzy formal contexts computed by discarding the values below the given thresholds are shown too). Figure 27 evidences two different ontology structures: an horizontal one (with $T = 0.2$) with a more general conceptualization and a vertical, more specialized one (with $T = 0.8$) where resources left appear distributed in the concepts.

### *5.2.2 Analysis of the lattice consistency and retrieval performance*

A further analysis of the ontology tree focuses on resources classification and aims at estimating its consistency in term of concepts generated by the mapping. The extracted ontology is compared with the OpenLearn categories. In this study, we have considered the ontology generated from lattice by setting the threshold $T = 0.5$.

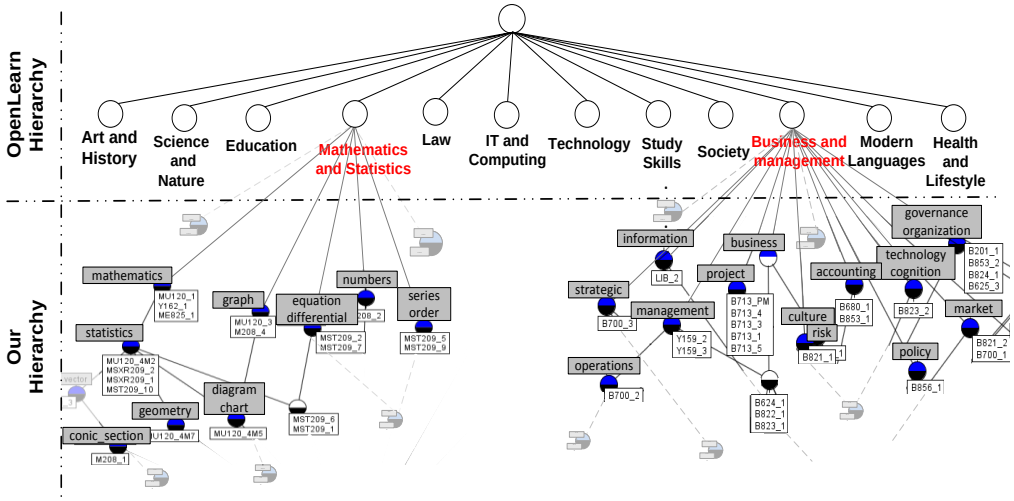



Figure 28 Example of concepts in the generated ontology and the corresponding OpenLearn categories

[21] http://ontogen.ijs.si/

[22] http://olp.dfki.de/OntoLT/OntoLT.htm

[23] http://protegewiki.stanford.edu/wiki/TerMine_Plugin

Figure 28 shows the ontology as a lattice-like structure rather than the navigation tree: this visualization allows us to facilitate the comparison with OpenLearn providing clear representation of the resources classification. From the analysis of the content of each ontology class, we have verified that 86% of whole collection of individuals (i.e., resources) finds a coherent classification in OpenLearn; i.e. the same groups of resources are collected together to form corresponding OpenLearn category. From this percentage, the 33% of resource appears in more specific classes. That means the ontology generated by lattice allows a natural refinement of classification of resources. Figure 28 sketches an example which shows some OpenLearn categories (at high level) and the relative specialization obtained by the proposed approach. For instance, the OpenLearn category *Mathematics and Statistics* appears more specialized in the proposed approach. Furthermore, by a comparison with OpenLearn categories, the remaining 14% of resources appears badly classified: the analysis of them reveals some semantic ambiguity in the content, because topics of different categories are gathered in a same category.

Let us evaluate the retrieval effectiveness on generated ontology by applying the information retrieval process defined in Section 4.2.3.

Table 11. Summary of training datasets and queries.

| Domain | Number of Resources | Number of Queries |
|---|---|---|
| **Art and History** | 65 | 3 |
| **Science and Nature** | 40 | 3 |
| **Mathematics and Statistics** | 37 | 2 |
| **Education** | 42 | 2 |
| **Law** | 40 | 2 |
| **IT and Computing** | 26 | 1 |
| **Technology** | 9 | 1 |
| **Study Skills** | 16 | 1 |
| **Society** | 85 | 2 |
| **Business and Management** | 65 | 2 |
| **Modern Languages** | 30 | 1 |
| **Health and Lifestyle** | 33 | 1 |
| **Tot.** | 488 | 21 |

The retrieval performance of the approach is assessed in terms of precision and recall measures, considering the analysis through *micro-average* of the individual precision-recall curves [62]. Let $\hat{Q} = \{Q_1, Q_2, \ldots, Q_n\}$ be a set of queries, $D$ all the relevant resources in $I_\alpha(O)$ for the given set of queries $Q$. For each query $Q_i$, we consider $\lambda = 20$ steps up to its maximum recall value and measure the number of relevant documents retrieved at each step $\lambda$.

Table 11 summarizes all the categories taken into account and the number of the relative resources and queries, exploited in the evaluation of the micro-average of the precision and the recall. In particular, twenty-one queries are analyzed in this study.

According to [62] the micro-averaging of recall and precision (at the generic step λ), is defined as follows:

$$Rec_\lambda = \sum_{Q_i} \frac{\left|R_{Q_i} \cap B_{\lambda,Q_i}\right|}{|R|} \qquad Prec_\lambda = \sum_{Q_i} \frac{\left|R_{Q_i} \cap B_{\lambda,Q_i}\right|}{|B_\lambda|}$$

where $R_{Q_i}$ is the set of relevant resources for a given query $Q_i$, $B_\lambda$ the set of retrieved resources at the step $\lambda$ and $B_{\lambda,Q_i}$ is the set of all relevant resources, retrieved at the step $\lambda$, for the query $Q_i$.

Figure 29 shows the tendency of the micro-average of recall/precision curve evaluated on the collection set, comparing the proposed approach with a known keyword based search engine called Lucene[24]. Precisely, three different ontology hierarchies have been built for different values of inheritance richness (***ir***). Let us note that the curve ***ir*** = 0.3 has associated an ontology with a vertical structure; in particular, with values of recall greater than 0.6, the precision is higher than the curve ***ir*** = 0.7 because the resources are distributed among ontology classes through more relations of specialization. On the contrary, in the case of ***ir*** = 0.7 the resources are arranged together, because each class of the ontology is richer of individuals (i.e., resources) and poor in the specialization.

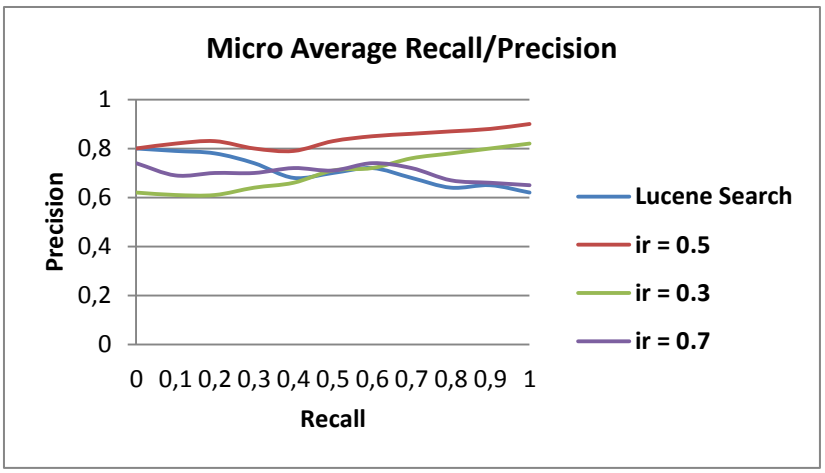



Figure 29. Micro-averaging precision/recall by varying the ir values

So, with different ontology hierarchy associated to different ***ir*** values, we have observed that the systems performance can change in term of data retrieval. The performance of web resources retrieval depends on the vertical or horizontal nature of the generated ontology by *varying* ***ir*** levels of the generated knowledge structure.

[24] http://lucene.apache.org

In particular, with the curve ***ir*** = 0.5, the precision/recall tendency is better than other cases (***ir*** = 0.3, ***ir*** = 0.7 and Lucene Search). Furthermore, the curve ***ir*** = 0.5 highlights that the precision is quite constant, it doesn't decrease meaningfully when the recall grows. The prevalence of this curve emphasizes that the best performances are obtained considering a median ***ir*** value, i.e. exploiting a right trade-off between the horizontal and vertical structures.

## 5.3 Conclusions

This chapter presents an approach that starting from the content analysis of a collection of web resources extracts ontology by applying methodology described in Section 4.2.1. The generated ontology is exploited as a retrieval model, as detailed in Section 4.2.3. Data have been clustered, organized, and visualized in different ways to support the user navigation. In particular, the methodology defined in Section 4.2.4 has been applied in order to provide faceted browsing of the ontology's individuals (i.e., web resources).

In brief, the main benefits are summarized as follows:

- concepts elicitation to support semi-automatically ontology design;
- automatic conceptualization of the knowledge embedded in web resources;
- automatic generation of web resource classification according to the conceptualization from the ontology;
- classification of the analyzed resources maintaining completely separated the rough data and semantic information.
- the fuzzy FCA model, which introduces more flexibility in the relation between object and attribute, yields families of lattices, according to the threshold fixed in the context description and the weight on the subsumption relation.
- support to query the tree in a hierarchical or free-form manner.

# Chapter 6

## Taxonomy Extraction applied to Enterprise Competency Management

In this new era, there has been an increasing development of social networks: nodes and links represent participants and their friendships, respectively; users can retrieve information from their friends or their friends' friends by propagating the request in the network. The user is in the center of the communication and sharing network: Flickr, Wikipedia, Del.icio.us, or YouTube, are grown thanks to the *User Generated Content* (briefly, UGC). The roles of people have drastically changed: from passive consumers of information to active collaborators, who create and share new content.

On the other hand, the profound changes in the global business environment, information technology and content management are enabling a global change in the e-market. A new model of organizations, called the Enterprise 2.0 [70] has opened up new methods for communication and conversations, and has transformed the way that companies share and access information, orchestrate resources and create value. It enables a common space for knowledge capture and sharing. Unlike information locked-up in email and documents, knowledge is easier to find and use when people actually need it, can be up to date, and it can be fully searched by all who have access. Enterprise 2.0 implements a multiparty "conversation" to share information and manage knowledge inside and outside the organization through blogs and wikis, social networking and tagging, rating systems, etc. These tools impose neither preconceived notions nor specific prerequisites: they just represent a common mean to link the individuals involved to participate, while they work together, share data and create networks of people with similar interests.

This chapter presents a workflow for manage and update employees' profiling according to the enterprise policy. The system provides a collaborative Enterprise 2.0 environment which traces the competencies and the skills of employees matured during their working activities through corporate blogs, enterprise wikis, forums etc. An automatic feedback is generated to support the human resources manager in the profiling revisions or to apply enterprise policies.

This work exploits some of the research objectives described in Section 4.2. In particular, the methodologies applied are:

- Ontology and Taxonomy Extraction (see Section 4.2.1), in order to conceptualize UGC in hierarchical manner;
- Information Retrieval (see Section 4.2.3), that is used to matchmake conceptualized UGC and existing enterprise taxonomy;

The chapter is organized as follows: Section 6.1 gives an overview of the competency management process, through its advantage and weakness. Successive Section 6.2 deepen

the workflow description, providing formal and specific details about each component. Section 6.3 provides a functional view of the system, through a sketched idea of an applicative scenario. Finally, conclusions close the chapter.

## 6.1 Competency Management: Advantages and Limitations

The Competency Management is considered as a set of processes that aim to identify, classify and manage competencies that employees need to perform specific tasks. Competency Management practical frameworks drive human resource managers to improve the results of an organization. Ensuring that there is the right person in the right position at the right time is one of the possible factors that enable organization processes improvement. Career management, recruitment, work assignment and team building are processes that get advantages from Competency Management. Nowadays, a wide use of Human Resource Management Systems (HRMSs), like SAP HR Module, is experienced within the organizations. These systems classify the competencies that are relevant in a given organization, to link them to employees' profiles and to organizational roles. Competencies are usually grouped into behavioral (e.g. communicating effectively, creative problem-solving, etc.) and technical manner and organized in different level hierarchy. The technical ones are domain-specific and change according to the mission of the organization in which are defined.

Furthermore, some complete specifications, covering all human resources management aspects, exist. HR-XML[25]. HRXML is a XML schema defined by the HR-XML Consortium in order to support standardized and practical exchange of information on competencies within a variety of business contexts. HR-XML also provides the properties useful to describe skills and competencies of each employee.

HRMSs suffer of (1) difficulty to constantly maintain employees' profiles up-to-date, (2) lack of native support to integration with Learning Management System, (3) lack of semantic interoperability useful to exchange competencies information across different cooperating organizations and (4) impossibility to exploit new Web 2.0 tools, like blog, wiki and forum, to share knowledge in order to retrieve important hidden information about employees. The first three issues are detailed in [71]. Conversely, the last issue is the main objective of the present work. In particular, the research idea is to exploit the user generated content (from wikis, blogs and forums), rating and other information coming from Web 2.0 enterprise tools in order to elicit knowledge about how employees use their competencies in order to foster enterprise collective knowledge and support or help the work of their colleagues. The elicited knowledge can be used to support the decision processes (e.g. training plans definition, work assignments, and so on) of the Human Resource managers. Suppose that both employees A and B have the competency X. A wrote twenty good rated blog posts in one year about X. Conversely, B wrote two poor rated blog posts in one year about X. The HRM can decide to assess the competency X for B but even though it is trivial to assign task that needs competency X to the employee A. Now, there are two main critical aspects to underline. The first aspect is how to extract relevant topics from user generated content. The second aspect is

[25] http://www.hr-xml.org/hr-xml

how to weight these topics and consequently match them with the competencies in the employees' profiles.

## 6.2 Workflow

The whole system is shown in Figure 30. The system can be mainly split in four macro-modules which wrap some specific components aimed to achieving explicit functionalities, detailed as follows:

- *UGC Semantic Modeling*: this module defines a common semantic 'wrapper' to represent all the UGC resources coming from different semantic tools into a simple unique RDF-based structuring format. All the UGC documents "wear" this semantic envelop which makes the processing of content homogeneous.
- *Taxonomy Extraction:* the module corresponds to the framework described in Section 4.2.1. Specifically, exploits a fuzzy extension of Formal Concept Analysis to elicit and structure the UGCs, in taxonomic way. The output is the generation of a taxonomy of new concepts elicited from the employees collaboration content.
- *Competency-Related Storage*: consists of ontologies which deal with the enterprise domain profiles, assets, objectives and markets and SKOS taxonomies about competencies domain.
- *Information Retrieval:* this module corresponds to framework described in Section 4.2.3. It provides hints or suggestions to human resources manager about the skills and competencies of enterprise employees, grown up in the social network activities. Matching terms or concepts represent a kind of recommendation to send the HR manager, in order to support about employees' competencies, in accordance with inside enterprise policies or strategies.

The framework, shown in Figure 30, evidences other components whose functionalities are not strictly related to the presented framework even though they exploit its outputs. Indeed, the hints provided by the *Information Retrieval* module are taken into account to intervene on the enterprise planning about learning strategies as well as organization activities, according to the enterprise policies. A human intervention is often required to manage the Human Resource Decision Support System (HR DSS).

### *6.2.1 UGC Semantic Modeling*

This module produces a unique, uniform format for the all the UGC resources coming from Social Web tools on the net. It wraps all different semantic formats used in the social activities, into a common representation, a 'semantic envelop', which provides a uniform structuring for the communication and processing of the data.

Specifically, the semantic envelop is composed by a selected set of classes and properties, coming from main ontologies and schemas exploited to represent resources in Web 2.0 and successive versioning. An example of a common wrapped document describing a blog entry is given in Listing 1.

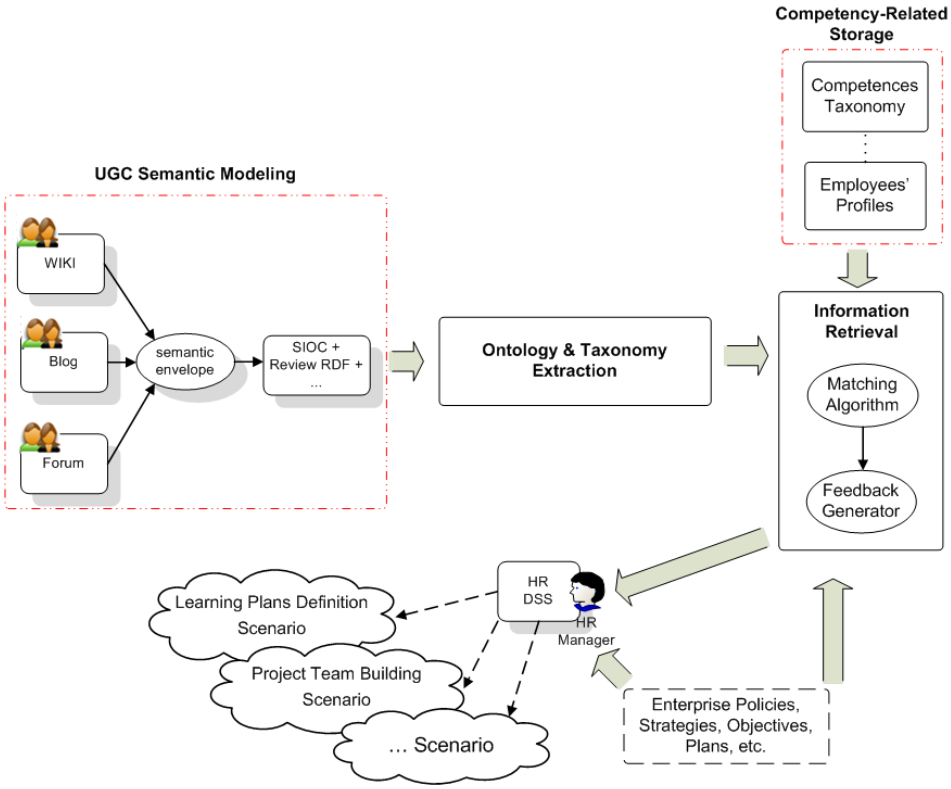



Figure 30. Workflow

Listing 1. An example of blog model

```
    <sioc:Post rdf:about="http://john.com/blog/2010/06/01/Programming-language-Java/">
        <dcterms:title>How to use hash table in Java?</dcterms:title>
        <dcterms:created>2010-06-01T09:33:30Z</dcterms:created>
        <sioc:has creator>
        <sioc:User rdf:about"http://john.com/" rdfs:label="John">
                <rdfs:seeAlso rdf:resource".../index.php?sioc type=user&amp;sioc id=1"/>
            </sioc:User>
        </sioc:has creator>
        <sioc:content>
    Class Hashtable is found in java.util package and is very useful data structure. If used sensibly, it can save time and can produce results efficiently. Hashtable class implements Cloneable, Map and Serializable interfaces …
        </sioc:content>
        <scot:hasTag rdfs:label="Programming Language" rdf:resource=
          "http://john.com/blog/category/programming-language/"/>
```

```
        <scot:hasTag rdfs:label="Java Tool" rdf:resource=
                "http://john.com/blog/category/java-tool/"/>
        <scot:hasTag rdfs:label="Hash Table" rdf:resource=
                 "http://john.com/blog/category/hash-table/"/>
        <rev:rating> 3.5 </rev:rating>
    </sioc:Post/>
```

All the UGC-based resources processed by *UGC Semantic Modeling* are then locally stored and indexed, as shown in Figure 31.

### *6.2.2 Taxonomy Extraction*

This module acquires the relevant data (wrapped in the UGC Semantic Modeling) and translates them into a digest form (i.e. a matrix-based representation), suitable to be mapped into the mathematical model. More specifically, the content of the property sioc:content which represents the textual content of the document is parsed at linguistic level, through preprocessing activities such as POS tagging, lemmatize and stop-word removal.

In this specific application domain, the model provides a knowledge structuring elicited by the documents produced by the employees in the social activities. Final result is a taxonomy arrangement of concepts and the subsumption relationships. As said, the property scot:hasTag is exploited to resolve ambiguity in the content of a resource, providing a synthetic description of the resource argumentation.

Let us observe the other properties in the semantic envelop are consulted to get the identity of documents' creators or just to recover/update the rating values when it is required.

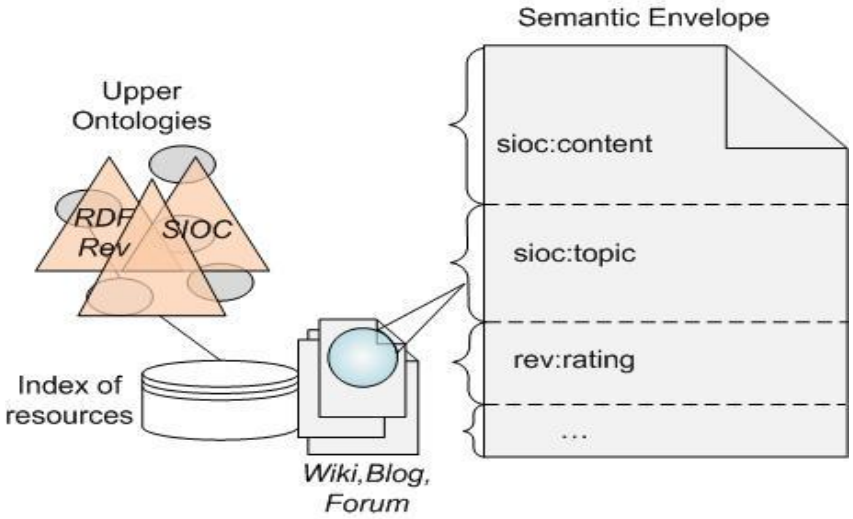



Figure 31. Uniform semantic envelop associated to the Web 2.0 resources.

### *6.2.3 Competency-Related storage*

In order to describe how competencies are associated to employees in enterprise human resource management systems we have to explain, at a conceptual level, two components (Figure 30): *Competencies Taxonomy* and *Employees' Profiles*. First of all, we need to define all competencies that are relevant in a specific organization. Typically, this task can be performed by using taxonomies. A novel approach to define domain specific taxonomies is based on the use of SKOS. An individual of the skos:Concept class represents a specific competency. The hierarchical relation between two competencies can be implemented by using the skos:narrower property. Listing 2 show how to use SKOS classes and properties to define competency taxonomy:

Listing 2. Example of classes and properties to define competency taxonomy

```
my:engineering rdf:type skos:Concept;
      skos:prefLabel "Engineering";
      skos:narrower my:electricalSystemsEngineering;
      skos:narrower my:mechanicalSystemsEngineering.
my:electricalSystemsEngineering rdf:type skos:Concept;
      skos:prefLabel "Electrical Systems Engineering".
my:mechanicalSystemsEngineering rdf:type skos:Concept;
      skos:prefLabel "Mechanical Systems Engineering";
...
```

Now, specific competencies have to be referenced by employees' profiles in order to assert the exact set of competencies owned by an employee. Other information like the proficiency level related to a specific competency for a given employee is also important. In order to define a conceptual representation of employees' competency profiles we could use some upper ontology like ResumeRDF[26].

ResumeRDF is an ontology developed in order to express on the Semantic Web the information contained in a resume, such as business and academic experience, skills, publications, certifications and so on. For instance, ResumeRDF provides cv:CV, cv:Person and cv:Skill that are classes we can use to link employees' profiles to specific SKOS competency taxonomies:

Listing 3. Example of class to link employees' profile to specific competency taxonomies

```
my:CV_Employee01 rdf:type cv:CV;
      cv:hasSkill my:ElectricalSystemsEngineering.

my:ElectricalSystemsEngineering rdf:type cv:Skill;
      skos:related my:electricalSystemsEngineering.
 ...
```

[26] http://rdfs.org/resume-rdf/

It's important to underline that the representation of competency-related storage we have provided follows a research trend consisting in modeling competency and in general enterprise aspects by using ontologies [72].

### *6.2.4 Information Retrieval*

This module is in charge to support decision about enterprise competencies management, providing hints or suggestions about the cumulate competencies of each employee. More specifically, this module achieves the conceptual matching between the concepts generated by the *Taxonomy Extraction* and the concepts from the *Competencies Taxonomy*, according to the specific *Employees' Profiles* in the *Competency-Related Storage*.

The goal is to guarantee a correct concept-based matching. The flow starts with *Taxonomy Extraction* module. It tracks the updating of the data driven taxonomy, coming from the formal lattice; in fact periodically, the formal context is completely re-generated, in accordance with the increasing social and collaborative activities in the enterprise.

When the *Taxonomy Extraction* module updates the taxonomy, the *Information Retrieval* module tries to get the matching between the concepts in the SKOS Competencies Taxonomy and the elicited, data-driven concepts in the lattice structure (see Figure 32). In particular, each concept name from the competencies taxonomy is compared to the all attributes names in each formal concept in the lattice. Thanks to Wordnet and Wu-Palmer similarity, a measure of relatedness among the two concepts is evaluated.

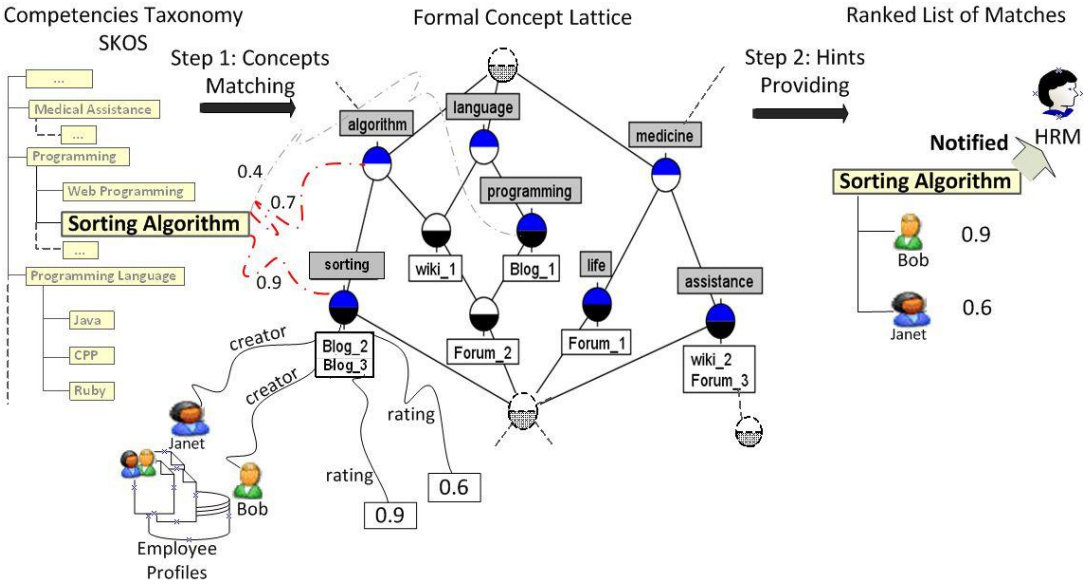



Figure 32. Concept Matching and a list of employees ranked according to their rating-based competency

Just to give a simple idea of the matching algorithm, let *C* be a formal concept in the lattice. Recall each formal concept is composed of a subset of object and attributes, according

to Definition 12. Thus, let us define $A = \{a_1, ..., a_n\}$ and $O = \{O_1, ... , O_l\}$ the sets of attributes (terms extracted by the selected properties in semantic envelop) and the set of objects (resources) of the Fuzzy Formal Context respectively. Then, let $P = \{p_1, p_2, ..., p_l\}$ be a concept of the competency taxonomy. A concept name in the competencies taxonomy could be represented by a composed linguistic expression (i.e. more than one term: for instance, Sorting Algorithm = {Sorting, Algorithm}).

The similarity between a concept of a taxonomy *P* and the formal concept *C* in the lattice, represented by *sim(P, C)* can be expressed by the following equation:

$$sim(P, C) = \frac{1}{|P|} \sum_{a_i \in C} \sum_{p_j \in P} sim_{WP}(a_i, p_j)$$

where $sim_{WP}(a_i, p_j)$ is the Wu-Palmer similarity.

Fixed a concept of the taxonomy $P_j$, the *Information Retrieval* module accomplishes the computation of the similarity between $P_j$ and each $C_i$ in the lattice *L*. Let us assume that $S(P, L) = \{sim(P, C_i) | \forall i: C_i \in L\}$.

Figure 32 shows an example (see Step 1: *Concepts Matching*): the SKOS concept *Sorting Algorithm* is candidate to match all the concepts of the lattice. Specifically, the SKOS concept matches with the concept *sorting, algorithm* and *programming* of the lattice.

Now, the next step is to get a list of resources whose content is related to the given SKOS concept. A simple idea of computation is given in Listing 4:

Listing 4. Example

```
Input: concept P from Competency Taxonomy;
       formal lattice L;
       fixed threshold τ

Output: ranked list of object (i.e. resources)
       select all the C_i in the lattice L s.t. sim(P, C_i) ≥ τ.
       for each selected concept C_i = (A_i, O_i)
            consider the objects o_k ∈ O_i.
            for each object o_k
                  compute τ_ik = μ_k × sim(P, C_i) × rating(o_k)
                    (where μ_k is given in Definition 3)
              rank all objects with respect to their own threshold τ_ik
```

The algorithm takes as inputs a concept from Competencies Taxonomy, a lattice and a threshold which represents the required minimal value similarity. Then it returns a ranked list of objects, i.e. some selected web resources of the *UGC Semantic Modeling* module. Let us suppose the threshold, required in the algorithm is fixed to 0.6. Let us observe, looking at Figure 32, resources named *Blog_2* and *Blog_3* (in the lattice representation) are the right candidates, whereas Blog 1 is eliminated, due to the fact its similarity value is below the threshold $\tau = 0.6$.

According to their semantic envelop, these resources are associated to their own creators through the property sioc:has_creator. In the given pseudo-code, the term *rating($o_k$)* represents the value associated in the property rev:rating, the social evaluation of user generated content. The *rating($o_k$)* intervenes in the computation of value $\tau_{ik}$ which describes how that resources is relevant for the given concept *P*. At this point, the *Information Retrieval* module can redact the list of employees which is associated to these resources (see Figure 32, Step 2: *Hints Providing*). After a further filtering based on enterprise strategies, coming from *Feedback Generator*, the *Information Retrieval* module sends the list and related information to the human resources manager, which takes it into account, for possible feedback to the employees' profiles or just to plan new enterprise strategies of learning and staff organization.

## 6.3 Team Building: a sample Scenario

Enterprise policies and marketing strategies often push the Human Resources Management Systems to improve the internal productivity and efficiency, according to the defined strategies and plans. Nevertheless, Human Resources Management Systems present some weakness in managing and exploiting employees' competency profiles, as described in Section 6.1, Figure 30. In particular, the expertise acquired by the employees during their work activities is often hidden in what is commonly called "tacit knowledge". As the Web 2.0 grows in enterprise contexts, employees' tacit knowledge can be found in the unstructured or semi-structured data produced by collaborative and social tools. This knowledge is completely skipped in the updating of employees' competency profiles. The proposed approach aims at introducing this aspect in order to refine the evaluation process of the employees' knowledge background, by exploiting the user generated content (from wikis, blogs, forums, etc.) they produced.

The Figure 32 emphasizes how the system can support this limitation. The ranked list of employees is given as suggestion to the Human Resources Decision Support System (see Figure 30) in order to improve or combine organizational choices about the selection of human resources to assign for a specific activity in a planned scenario.

Just to give an example, let us suppose the HR Manager is defining a team for a project on *Semantic Search Engines*. In particular the HR Manager is looking for a developer with the *Sorting Algorithm* competency for a specific project activity. Both Bob and Janet have this competency. The HR Manager has to decide the appropriate candidate to this role. The *Feedback Generator* shown in Figure 30 is in charge to filter the resulting list (Bob and Janet), according to the enterprise policies (in this case, it is needed to select the developer who used *Sorting Algorithm* competency in the best possible way). In the proposed example we obtain information about the way employees use their competencies helping their colleague by exploiting rating information. Rating is the social evaluation of user generated content. The *Feedback Generator* supports the HR DSS by filtering the results coming from the *Information Retrieval* module and it returns just Bob. Thus, the HR Manager obtains the appropriate candidate to work on the considered project activity.

## 6.4 Conclusion

Recent trends in Social Web emphasize the role of the Web as a platform for true collaborative activities: users become active collaborators, rather than passive viewers. This new vi-

sion of the Web, aimed at communicating, sharing knowledge, improves the collaborative activities among users as well as their skills and competencies. Enterprises recognize the added value provided by the social networks and are investing in the developing of internal plans and policies for supporting career and competencies management, exploiting tacit competencies coming from collaborative activities, rather than planning traditional learning activities. Enterprise 2.0 provides an alternative distributed workforce to share and manage knowledge inside and outside the organization through blogs, wikis, social tagging, rating systems, etc.

This chapter introduces a framework for dynamic refinement of user profiling by monitoring his competencies acquired during social activities. Refinements in the user profiling are driven by analysis of UGC deployed by users in the post and blog activities: a fuzzy extension of Formal Concept Analysis model supports the elicitation of implicit knowledge and the content structuring into a conceptual representation. The resulting hints refine employees' profiles as well as provide business choices in enterprise policies and strategies.

# Chapter 7

## Automatic Textual Resources Annotation

The promise of the emerging Semantic Web domain is that machine understandable semantics augmenting Web resources facilitate the information discovery and retrieval by making use of available semantic annotations and their underlining ontologies. Semantic annotation represents the core of Semantic Web technology: it bridges the gap between legacy non-semantic web resources descriptions to their elicited, formally specified conceptualization, converting syntactic structures into knowledge structures, i.e., ontologies. Most existing approaches and tools are designed to deal with manual, semi or automatic semantic annotation exploiting available ontologies through a pattern-based discovery of concepts.

Just to emphasize the actual benefits of semantic annotation, Figure 33 shows a simple interface for a geographic application that provides an integrated view of semantics and concrete data. It supports the user to find information and services in a specified area. The map environment is semantically annotated. That means the user can navigate the concepts in the tree structure (see Figure 33, on the right) and automatically the relative information (instances) will appear in the map. So, let us suppose that the user is looking for a hotel that provides free Wi-Fi access and if possible satellite television then he will explore the concepts in the tree, in order to discover the concept "hotel" and concepts related to "Wi-Fi" and "Sat TV". The fired concepts (particularly, its combination) will be visible on the map, and so the user, will get information about the hotels that better match his request.

The approach defined in this research work aims at generating automatic semantic annotation of web resources, without any prefixed ontological support: it is the generated annotation that provides concepts and relations, arranged into an ex-novo ontology. It achieves a modeling of a high level abstraction framework (described in Section 4.2.2) which can process resources from different sources (textual information, images, etc.) and generate an ontology-based annotation. Specifically, a data-driven processing reveals data and the intrinsic relationship among them (in form of triples), extracted by the resources content. On the basis of the discovered semantics, corresponding concepts and properties are modeled; then, through an OWL-based coding annotation, an ad-hoc ontology, is built.

The benefit is the generation of knowledge structuring in automatic way, without any compulsory human intervention. The approach relies on the mathematical modeling of Fuzzy FCA and Fuzzy RCA (see Sections 2.3 and 2.4, respectively) which allow the extraction of data and relations in form of triples. Particularly, the fuzzy extension of these two theories enables a straightforward flexibility to the building of the ontology which describes a "structured" annotation.

The remaining chapter is organized as follows. Section 7.1 describes a textual resource annotation application. Some experimentations are shown in Section 7.2, 7.3, 7.4. Then, Section 0 provides an outlook on main semantic annotation approaches. Conclusion closes the chapter.

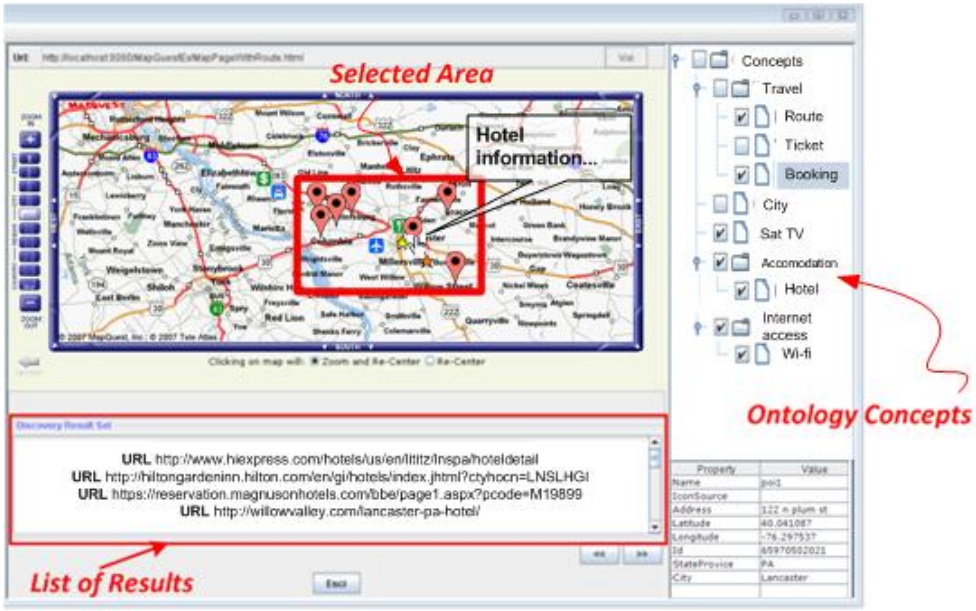



Figure 33. Example of map semantic annotation

## 7.1 A context-dependent application: textual resources annotation

In the domain of textual resources, the framework (described in Section 4.2.2) proposed in this research work is instantiated to achieve a textual semantic annotation. According to the overall process described in previous section, this instantiation should generate an ontology that reveals the under laying semantics (concepts and relations) of textual resources.

Figure 34 shows a snapshot of the actual user interface, which evidences how the textual information are analyzed and translated into an ontology.

In particular, at the top-left side of interface you can type your text; at the top-right, system shows a tree of elicited ontology; at the bottom-right there is a representation of relations. Finally, at the bottom-left of interface, there is an area dedicated to parameters setting that affect the context enrichment of involved attributes.

From interface in Figure 34, let us consider the following fragment of text (from *Plain Text* area):

"Virgil has written Aeneid. Dante Alighieri wrote Divine Comedy. Tolkien was born in England. Tom lives in Tokyo."

Applying the text parsing (defined in Section 4.2.2) to given sentences, following triples <subject, predicate, object> are returned:

- < Virgil, has written, Aeneid >
- < Dante Alighieri, wrote, Divine Comedy >
- < Tolkien, was born, England >
- < Tom, lives, Tokyo >

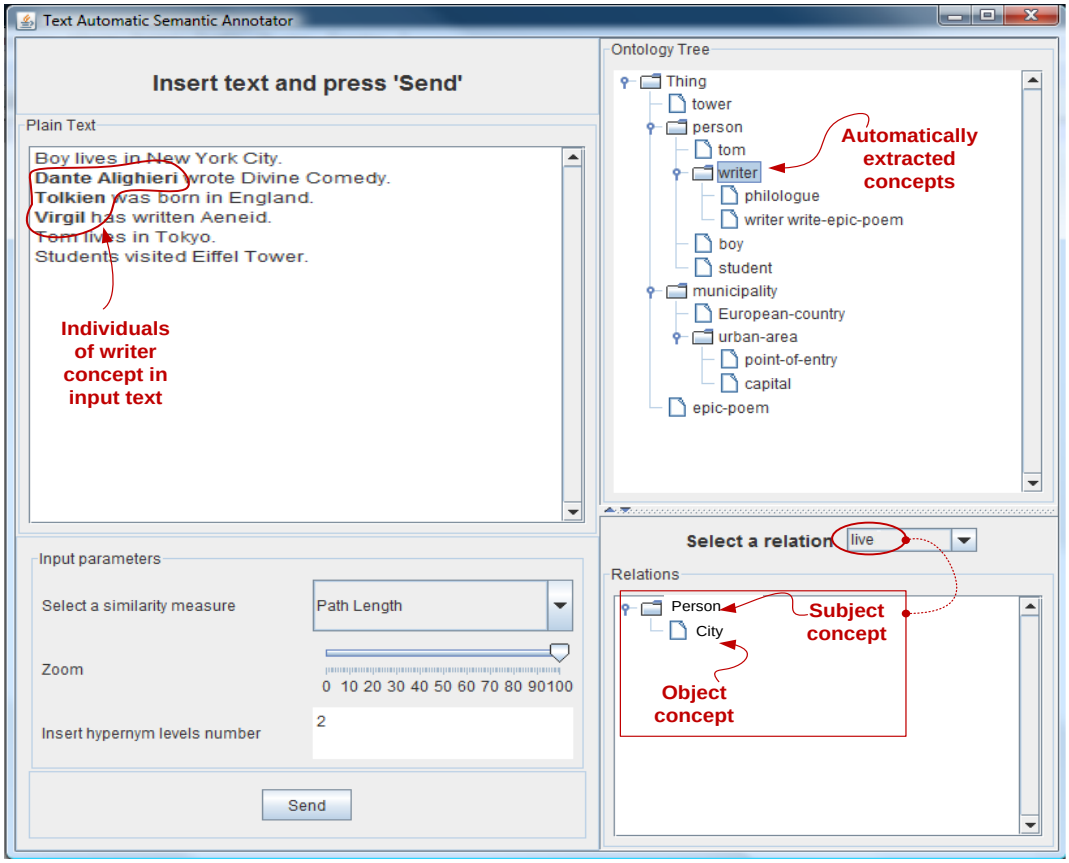



Figure 34. Application screenshot

To discover text meaning, we make crossing researches of elements definition into Wordnet lexical database and Wikipedia free encyclopedia. The synergic use of WordNet and Wikipedia which provides named entities like people, organizations, geographic locations, books, songs, products, etc., increase application performances [73]. The result is unambiguous definition of the given element (describing the element property).

A described in Section 4.2.2, exploiting the theory behind the FFCA, individual fuzzy formal contexts are built for subject and object elements.

Table 12 and Table 13 show a fuzzy version of the formal contexts associated to subjects (i.e., *domain context*) and objects (i.e., *range context*) elements gathered by the previous example.

Table 12. Domain context

| OBJECT | ATTRIBUTES | | | | | |
|---|---|---|---|---|---|---|
| | **writer** | **author** | **poet** | **person** | **philologue** | **philologist** |
| Virgil | 0.95 | 0.95 | 1 | 0.93 | 0 | 0 |
| Dante Alighieri | 0.95 | 0.95 | 1 | 0.93 | 0 | 0 |
| Tolkien | 1 | 1 | 0.47 | 0.94 | 1 | 1 |
| Tom | 0.33 | 0.33 | 0.25 | 1 | 0.2 | 0.2 |

Table 13. Range context

| OBJECT | ATTRIBUTES | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **epic _poem** | **heroic _poem** | **poem** | **country** | **territorial _division** | **national _capital** | **city** | **region** |
| Aeneid | 1 | 1 | 0.95 | 0 | 0 | 0 | 0 | 0 |
| Divine Comedy | 1 | 1 | 0.95 | 0 | 0 | 0 | 0 | 0 |
| England | 0 | 0 | 0 | 1 | 0.94 | 0.2 | 0.2 | 0.5 |
| Tokyo | 0 | 0 | 0 | 0.2 | 0.25 | 1 | 0.5 | 0.5 |

Once built the fuzzy formal contexts (see Table 12 and Table 13) for domain and range, a fuzzy formal lattice can be generated. Figure 35 and Figure 36 emphasize the hierarchical structure generated by the lattices.

In order to simplify the description, we have presented the domain and range contexts associated to the subject and object as separate entities. The combination of domain and range context allows us to gather more properties and, thereby, to construct a more accurate classification of text. In addition although subject and object play different roles, they represent the same type of information (they are both nouns). In particular, this combination is exploited in the modeling of predicates.

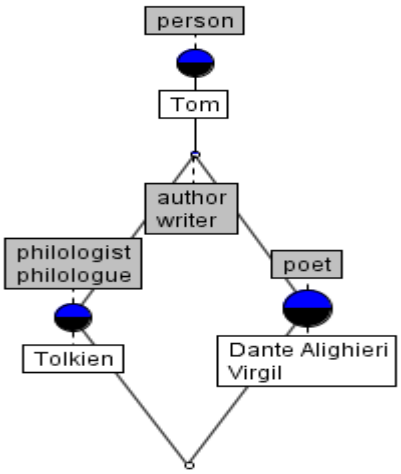



Figure 35. Domain Lattice

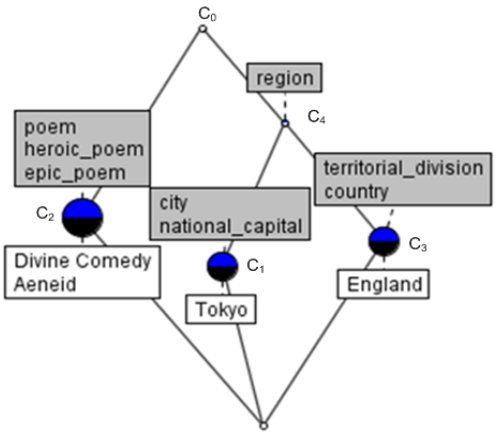



Figure 36. Range Lattice

Like for subjects and objects, the *Content Analyzer* produces a fuzzy formal context for predicates, where verbs are objects (rows) and the attributes (columns) come from the combination of domain and range contexts, as shown in Table 14 (all the verbs are converted into infinite form). Furthermore, membership values are inherited from domain and range context of specific verb. For example, *bear* have same *Tolkien*'s attribute values as domain ones, and *England's* attribute values as range ones.

Table 14. Relation context

| | ATTRIBUTE | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **DOMAIN** | | | | | | **RANGE** | | | | | | | |
| **OBJECT** | **writer** | **author** | **poet** | **person** | **philologue** | **philologist** | **epic_poem** | **heroic_poem** | **poem** | **country** | **territorial_division** | **national_capital** | **city** | **region** |
| write 1 | 0.95 | 0.95 | 1 | 0.93 | 0 | 0 | 1 | 1 | 0.95 | 0 | 0 | 0 | 0 | 0 |
| write 2 | 0.95 | 0.95 | 1 | 0.93 | 0 | 0 | 1 | 1 | 0.95 | 0 | 0 | 0 | 0 | 0 |
| bear | 1 | 1 | 0.47 | 0.94 | 1 | 1 | 0 | 0 | 0 | 1 | 0.94 | 0.2 | 0.2 | 0.5 |
| live | 0.33 | 0.33 | 0.25 | 1 | 0.2 | 0.2 | 0 | 0 | 0 | 0.2 | 0.25 | 1 | 0.5 | 0.5 |

Figure 37 shows the lattice built from the context described in Table 14. This lattice comes from the relations (i.e., predicates). Let us note that since predicates $write_1$ and $write_2$ have same infinite form and, most importantly, they are described in the same manner in the context, they are grouped in the same concept lattice.

Next step achieves the semantic annotation through Fuzzy RCA.

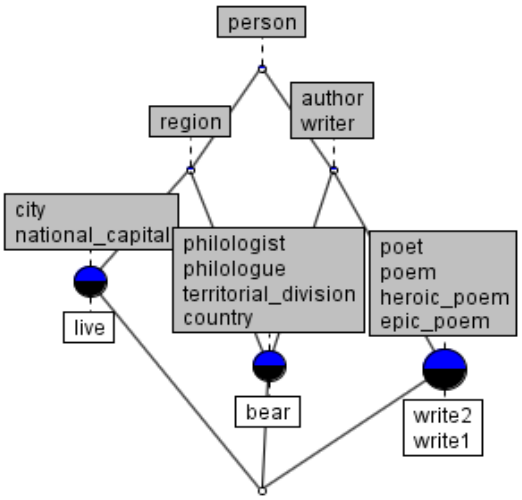



Figure 37 - Relations lattice

Then a new special attribute (relational attribute) with the form *predicate:rangeConcept* is added for that domain object that is connected to the range concept. Table 15 shows added attributes. Fuzzy membership of attributes is calculated as the inverse of Euclidean distance between memberships of predicate and range objects (if there was a matching between predicate and subject objects).

Table 15. RCA context

| OBJECT | ATTRIBUTES | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **writer** | **author** | **poet** | **person** | **philologue** | **philologist** | **write: C2** | **bear: C3** | **live: C1** |
| Virgil | 0.95 | 0.95 | 1 | 0.93 | 0 | 0 | 1 | 0 | 0 |
| Dante Alighieri | 0.95 | 0.95 | 1 | 0.93 | 0 | 0 | 1 | 0 | 0 |
| Tolkien | 1 | 1 | 0.47 | 0.94 | 1 | 1 | 0 | 1 | 0 |
| Tom | 0.33 | 0.33 | 0.25 | 1 | 0.2 | 0.2 | 0 | 0 | 1 |

The result is depicted into Figure 38: on the left there is resulting RCA lattice which refers old *Range lattice* (on the right), through *write*, *bear* and *live* relations.

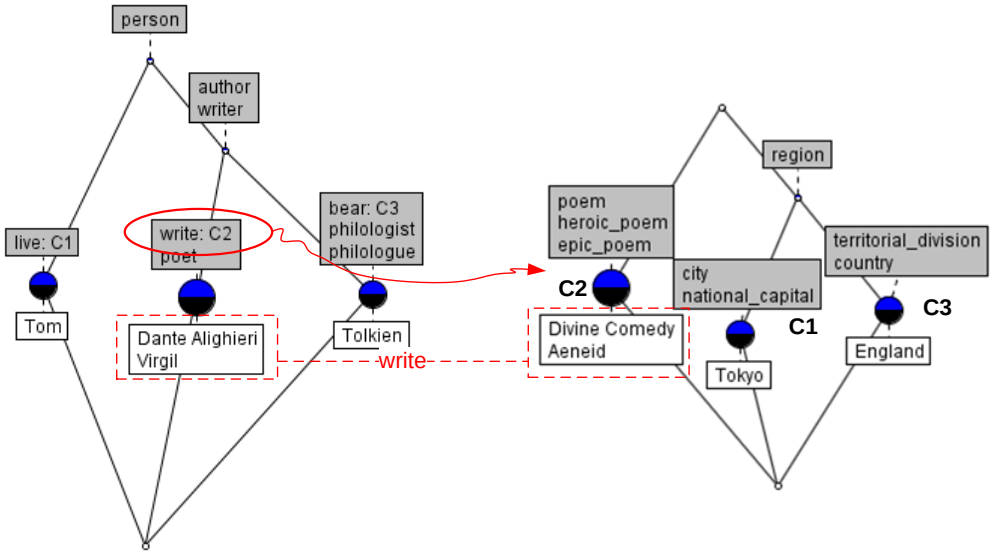



Figure 38. RCA lattice

This lattice describes, through a conceptual structure, the semantic annotation computed from the analysis of the input textual sentences. In fact, the fuzzy RCA lattice can be mapped to a *fuzzy ontology* through a semantic representation in OWL. In particular, datatype property has exploited to model simple attribute of concept, and object property to model specific

RCA attribute. Considering the interface in Figure 34, the RCA lattice generated by the textual data (on the left) is mapped in the ontology (on the right).

Furthermore, in Figure 34, ontology classes are named through a labeling function which selects an ontology name among most representative datatype properties. In the example, when user selects the *writer* class in the ontology, the words in the P*lain Text area: Dante Alighieri*, *Virgil* and *Tolkien* which represent its individuals are highlighted. Moreover, if the user selects a relation from scroll-box (in the bottom-right of the interface), all the involved subject and object concepts are shown in the relative panel. In the example, the predicate *live* involves *Person* class as subject, and *City* as object.

On the bottom-left of the interface, there is the Input Parameter area. It is possible to set some specific parameters, described as follows:

- *Similarity measure*: favorite similarity measure to perform similarity between attributes, during Context Enrichment phase. Selected knowledge based similarity measures are described in [74];
- *Zoom*: classification accuracy value. In Figure 34, this parameter is expressed in percentage by moving a scroll-bar. It represents an attribute-based filtering: if the Zoom value assumes maximal value (in figure it is equal to 100%), then all attributes collected in the Context Enrichment phase are involved in the ontology generation and specifically, in their classification into ontology classes; otherwise, only the attributes whose membership values are greater than zoom value are included.
- *Hypernym level number*: number of requested hypernym levels to add during Context Enrichment. A hypernym is a more generic word than given one. Larger is the number of hypernyms, more generic is the final classification into ontology classes.

These parameters setting influences the generation of the ontology and the relative population. Figure 39, Figure 40 and Figure 41 show some examples of setting of the zoom and hypernym levels and the relative impact of the ontology generation. For instance, in Figure 39, fixing zoom level to 70%, the ontology generated by the text in the interface, consists of only three classes.

For explicative purposes, Figure 39 shows for each concept, the relative instances colored with the same color. For instance, yellow square describes *poet* class whose individuals are *Dante Alighieri* and *Virgil*. Likewise, *city* class is delimited by green square as well as its individuals: *New York City* and *Tokyo*.

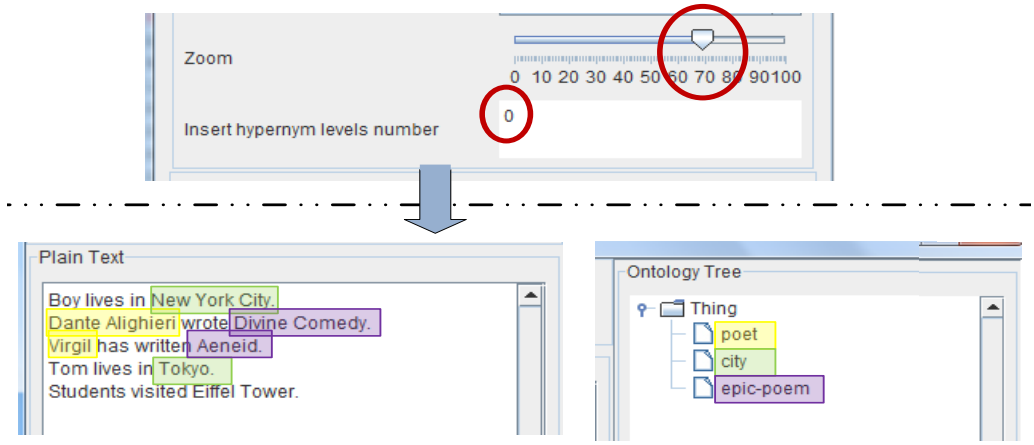



Figure 39 - Annotation results with Zoom equal to 70% and Hypernym level is equal to 0

In Figure 40, the hypernym level has been set to 3 (instead of 0) leaving the same zoom value. The resulting ontology has only a class (*person)* that is quite general and has associated all the proper names in the text (without no further specialization). Let us note this represents a different semantic annotation of the given text.

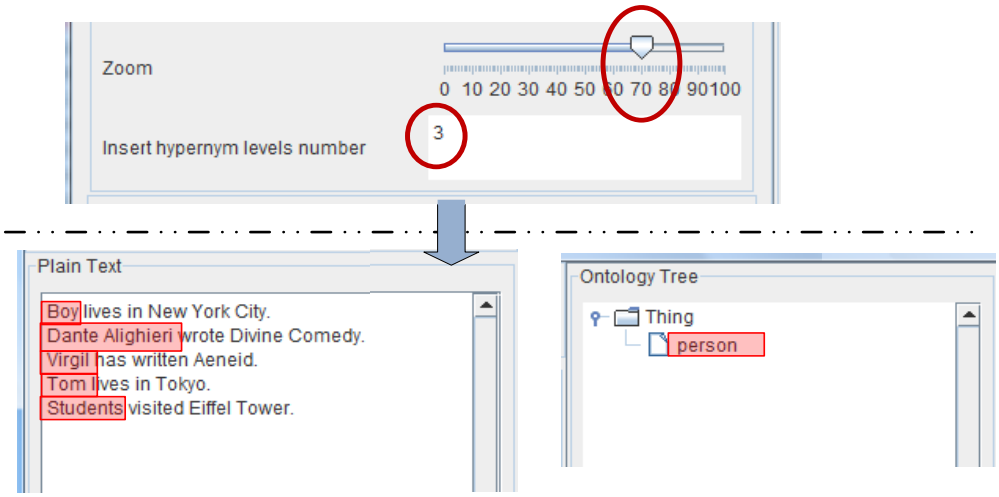



Figure 40. Annotation results with Zoom equal to 70% and Hypernym level equal to 3

Finally, in Figure 41 a new combination of the two parameters is given. The zoom is maximal, whereas the hypernym level is still 3. The resulting ontology is more accurate: all subjects and objects of input text are semantically annotated and a deeper details is given in terms of class specialization.

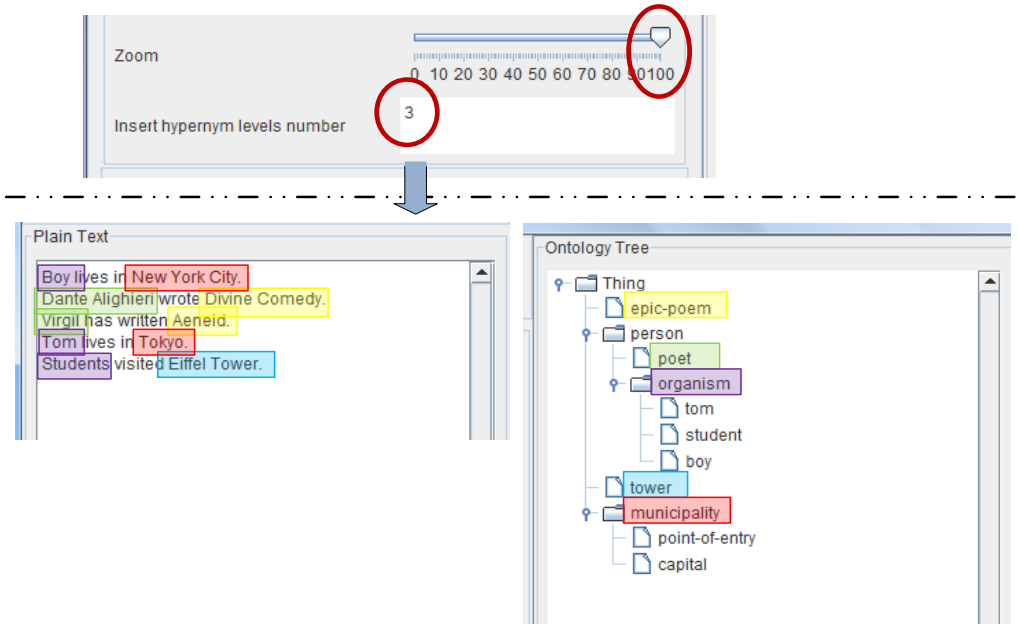



Figure 41. Annotation results with Zoom equal to 100% and Hypernym level equal to 3

## 7.2 Experimental Evaluation

The experimentation of this approach basically consists of demonstrating the proposed methodology is efficacy to generate conceptualization associated to the analyzed text. To reach this issue, the experimental results aim at showing the effectiveness of the proposed framework (instantiated for the semantic textual annotation) in terms of text categorization performances. The goal is to measure the collocation, classification or in general the distribution of the words extracted by both the text and Context Enrichment phase with respect to the classes/concepts of the generated ontology. Specifically, some specific measures (detailed in the next section) [67] are taken into account to study the ontology structure generated in accordance with the setting of some parameters in *Input parameters* area of the interface (Figure 34). In fact, *Zoom* and *Hypernym levels* are *"*tuned*"* in order to find straightforward performance of this semantic annotation tool.

The experimentation consists initially of studying the generated ontology according these criteria in order to evaluate the quality of its structure and the distribution of its population (with respect to the initial textual data). Then, to complete the analysis of tool, a comparison with other approaches will be done too, on the basis of standard Information Retrieval (IR) measures.

## 7.3 Ontology structure evaluation

The ontology structure strictly depends on the degree of generality or specificity of the text classification: an ontological class with a deep detail in term of subclass relations provides more specialization in the classification of words computed from the input text, if compared with classes with no nesting of subclass, due to the flat classification of all the words inside it.

This kind of ontology evaluation is achieved by exploiting some specific metrics given in [67] and described as follows:

- *Attribute Richness* (AR): evaluates quality of ontology design; it is defined as the average of attributes (slot) associated to each class in the resulting ontology. Intuitively, an high level of AR implies that the ontology conveys a rich level of information related to the knowledge domain.
- *Average Population* (P): measures the distribution of individuals (words) across the classes in the resulting ontology. It indicates how much the instances of knowledge base can be representative of the ontology structure: for instance, a low average number of instances per class evidences a poor coverage of the defined classes and then, the resulting ontology probably presents a not well-defined structure (with respect to the knowledge domain).
- *Inheritance Richness* (IRs) is a measure which describes the distribution of information across different levels of the ontology structure, i.e., the fan-out of parent classes. An high value of IR outlines an *horizontal* ontology structure, i.e. an ontology with a small number of inheritance levels and a relatively large number of subclasses. An ontology with a low IR instead, is *vertical*, because it is composed of many inheritance levels and the classes have a small number of subclasses.

These measures have been computed for the ontology generated by proposed framework instantiated for textual annotation. Figure 42 shows the tendency of these measures, with respect to some setting of *Zoom* and *Hypernym levels* parameters. Let us note AR values increase, as the values of Zoom and the *Hypernym* increase too. In other words, increasing the number of words and *hypernyms* involved in the classes specification implies that the resulting ontology can represent the associated knowledge domain with much amount of pertinent information. Similar consideration can be done for IR measure: increasing *Zoom* and *Hypernym,* means to build an ontology that grows up in horizontal way (low specialization level).

Moreover, let us observe the tendency of AR and IR do not present relevant changes by increasing hypernym level more than 2.

Differently, P curve tends to be constant or even decrease when *Hypernym* and *Zoom* increase, respectively. In particular, the number of hypernyms does not affect P. These result coherently demonstrates that increasing the involved attributes the structure of the ontology improves in the amount of instance for class, but not in the amount of classes.

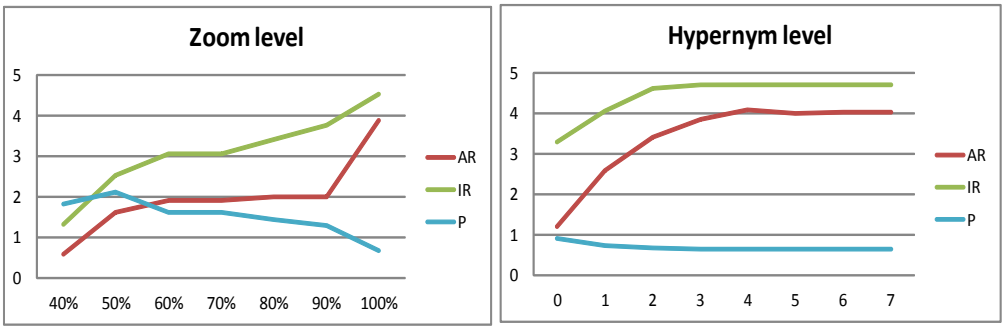



Figure 42 Ontology quality evaluation

## 7.4 Text categorization performances

The second step of experimental evaluation is aimed to assess the performance of the proposed application with respect to the text classification. Evaluation is conducted by comparing proposed tool with AlchemyGrid[27]. AlchemyGrid is a Named Entity Recognition system that is able to categorize text according to large number of specific classes, such as City, Country, Person, etc. It has been selected for the compared experimentation, because it provides online demo useful to perform experimental evaluation, in an easy and immediate way. The text corpus selected by considering generic sentences talking about general interest. The goal is to extract triples (subject-verb-object) from textual information. We have exploited about fifty documents of different types: textual documents (with extension txt, docx, etc.) and web pages (HTML-like) for a whole of about 1 megabyte, which is then translated in simple text. After a pre-processing activity aimed at discarding complex sentences (only simple phrases composed of a subject-verb-object are extracted), the final text has been given as an input to the AlchemyGrid demo and to proposed application.

To evaluate the effectiveness of both applications to classify words from input corpus into right classes (notice the relevance of each word for a class is given by a priori human annotation), we exploit standard measures of retrieval effectiveness (i.e., Precision, Recall, F-Measure).

According to the study of ontology metrics (Figure 42), the value of *Hypernym* level parameter is fixed to 2. The Zoom parameter, instead assumes different values in order to find best performances in terms of text categorization. Specifically, the more accurate value, comparable with AlchemyGrid is obtained by varying the Zoom in the range [80, 100] %.

Table 16 emphasizes the performance in terms of Precision (P), Recall (R) and F-measure (F) with respect to the more relevant extracted classes. In particular, the zoom is fixed to 90% in this experimentation (and hypernym levels to 2). In nutshell, Table 16 emphasizes the results of proposed application generally outperform AlchemyGrid.

[27] http://www.alchemyapi.com/api/demo.html

Table 16 – Comparison of the proposed system with AlchemyGrid annotation

| | *Movie* | | | *Sport* | | | *City* | | | *Country* | | | *Person* | | | *Average* | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | P | R | F | P | R | F | P | R | F | P | R | F | P | R | F | P | R | F |
| Automatic Annotator | 1 | 0.5 | 0.7 | 1 | 1 | 1 | 1 | 1 | 1 | 0.9 | 0.8 | 0.8 | 0.9 | 0.9 | 0.9 | **0.9** | **0.8** | **0.9** |
| Alchemy Grid | 0 | 0 | 0 | 1 | 0.5 | 0.7 | 0.5 | 0.3 | 0.4 | 0.7 | 0.6 | 0.75 | 0.5 | 0.3 | 0.4 | **0.5** | **0.3** | **0.4** |

## 7.5 Related Work

Annotating is, very generally speaking, the act to attach data to some other piece of data. The first annotation usage was motivated by the human exigency to add extra information to information itself in order to provide a more accurate description of it. Nowadays, the widespread availability of semantic annotations is strictly tied to the Semantic Web. Thus a more specific definition claims that annotation establishes, within some context, a (typed) relation between the annotated data and the annotating data. The annotation becomes formal, because exploits coding based on machine-understandable formal language. The last evolution of annotation is semantic: it exploits an ontology to guarantee a common understanding through a shared conceptualization. In last years, many approaches achieve semantic annotation. Manual annotation provides user-friendly interactive tool to manually support creation and maintenance of ontology-based markups.

An example of manual annotation tool is Onto-Mat-Annotizer[28]. It is an interactive tool for the Web pages text annotation. It represents the concrete implementation of the CREAM (CREAting Metadata for the Semantic Web) framework [75]: an annotation framework suitable to allow for the easy and comfortable creation of relational metadata. Then, M-OntoMat-Annotizer[29] (M stands for Multimedia), extends the CREAM framework and its reference implementation, Onto-Mat-Annotizer, with the Visual Descriptor Extraction (VDE) tool, in order to allow low-level feature annotation. It is used as an annotation tool for web pages and acts as the basis of an ontology engineering environment.

It supports manual annotation of video and image data by indexers with little multimedia experience by automatic extraction of low level features that describe objects in the content.

[28] http://annotation.semanticweb.org

[29] http://www.acemedia.org/aceMedia/results/software/m-ontomat-annotizer.html

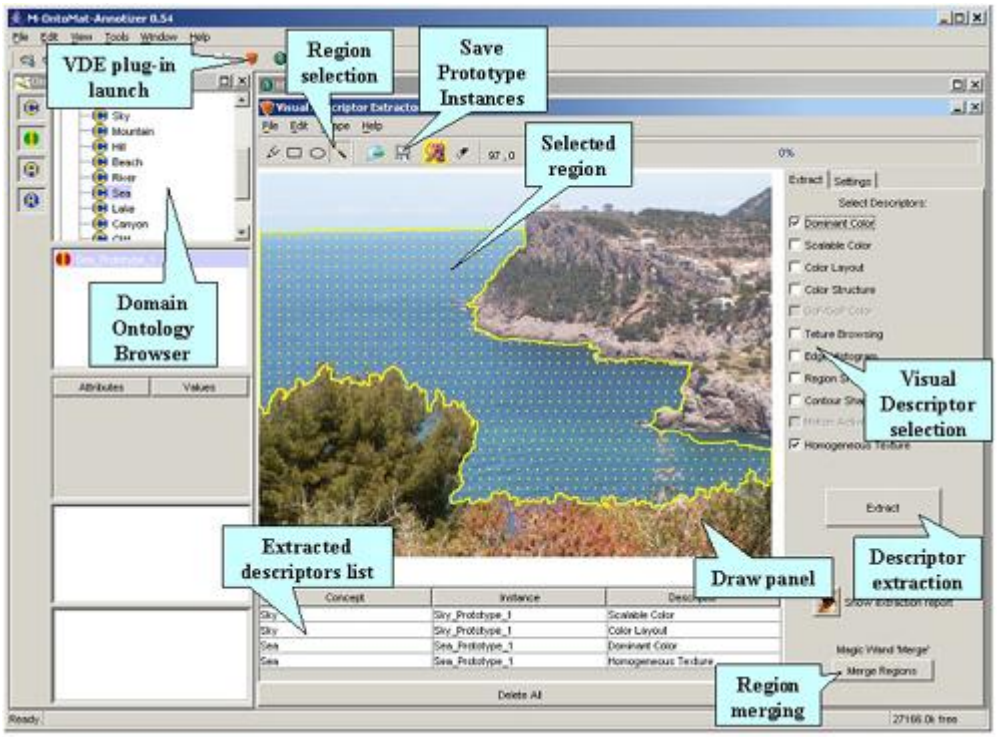



Figure 43 - M-Ontomat Annotizer

Figure 43 shows an interface example of M-Ontomat-Annotizer: a user can select image portions and manually annotate them on the base of chosen ontology.

In general, manual annotation results often quite expensive, and cannot provide multiple perspectives of a data source in correspondence with the different users needs. Moreover, manual semantic annotation has lead to a knowledge acquisition bottleneck [14]. This problem has been overcome by semiautomatic annotation that provides a balanced modeling between the suggestion of annotation and the human approval of the extracted annotation. A complete automation implies a wide scalability and a reduction of the burden of annotation of new resources. Automatic annotation collects the benefit of an improved retrieval and interoperability through a common framework for the integration of information from heterogeneous sources. In literature there are many platforms that provide semi or fully automated semantic annotation services. In particular, according to [76], [77] we classify semantic annotation approaches according the schema given in Figure 44.

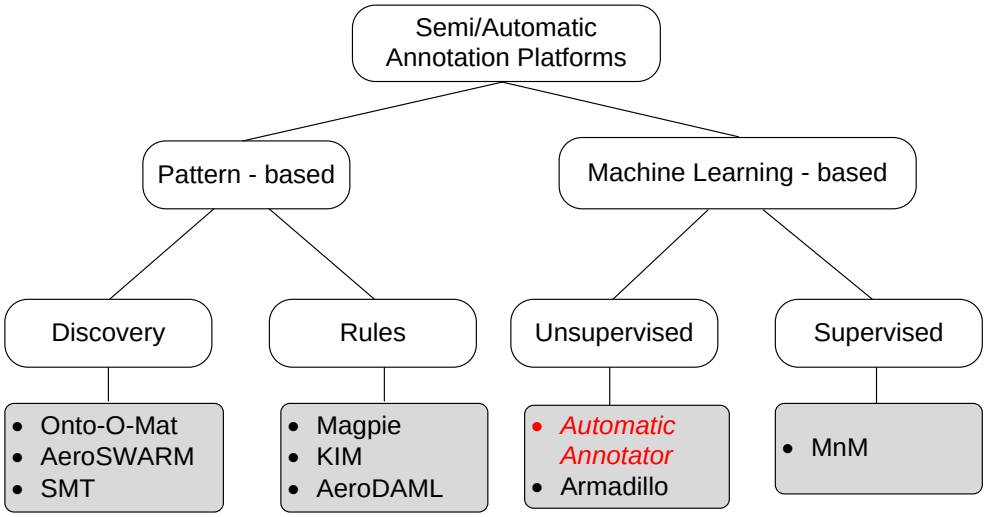



Figure 44. Classification of semantic annotations approaches

Among the knowledge based semi-automatic annotation tool, SMT [78] is a semi-automatic tool for markup of documents that combines commercial text extraction tools and manual annotation in a predefined templates in order to produce consistent OWL annotations. The user reads a preprocessed document, chooses a relevant template and fills it in. The main feature of SMT is the automatic OWL-based annotations, even the annotation process involves a high degree of user assistance.

KIM[30] platform provides a Knowledge and Information Management (KIM) infrastructure for automatic semantic annotation, indexing, and retrieval of unstructured and semi-structured contents. It performs information extraction based on the ontology and a massive knowledge base. In KIM, semantic annotation is considered as the process of assigning to the entities their semantic descriptions, provided by ontology. The platform is based on GATE[31] (General Architecture for Text Engineering). GATE is an infrastructure for development software components based on natural languages. GATE system provides functionalities such as annotation of textual documents both manually and automatically. Figure 45 shows KIM at-work: it semantically annotates input text (on the right) exploiting fixed ontology (on the left).

[30] http://www.ontotext.com/kim

[31] http://gate.ac.uk/

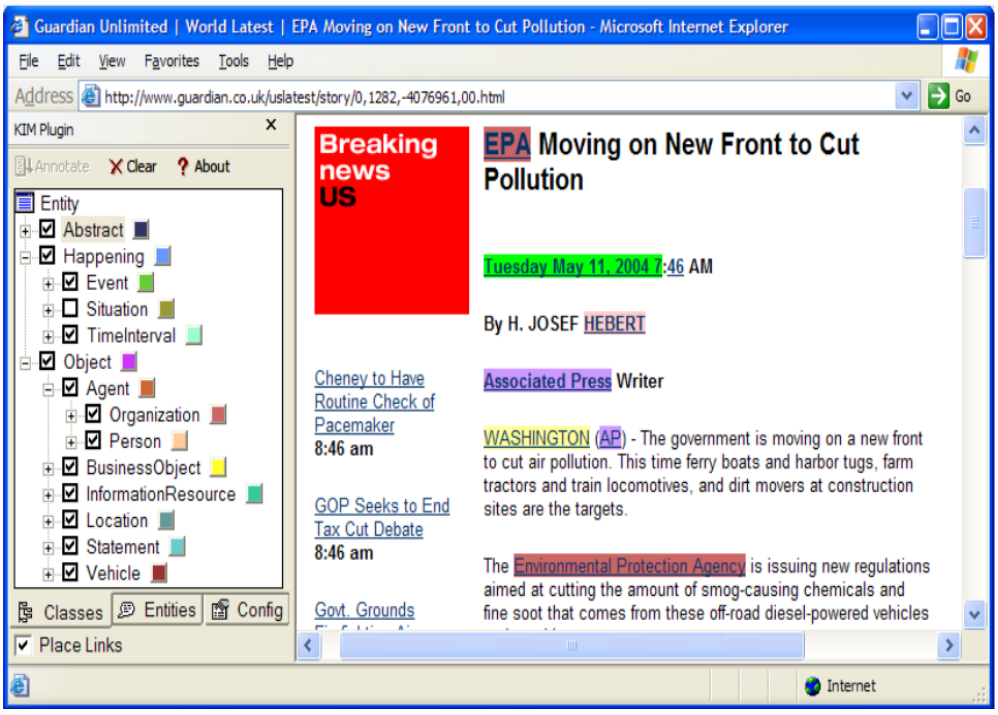



Figure 45 - KIM tool

Semantic annotation in MnM [79] is guided by an initial training phase to generate rules. AeroSWARM[32] is an automatic tool for annotation using OWL ontologies based on the DAML annotator AeroDAML [80]. It has both a client server version and a Web enabled demonstrator in which the user enters a URI and the system automatically returns a file of annotations on another web page. Magpie [81] is a browser that enables an ontology based semantic markup system for on-the-fly annotation of web documents.

Similar to [82], the automatic annotator framework proposed in this research work can be classified among the unsupervised machine learning based approaches (see Figure 44) due to the natural learning from FCA techniques [83], even though they are usually considered as a method of data analysis. In particular, with respect to the existing approaches, it does not work on pre-built ontologies but generates an ad-hoc ontology elicited by the parsed information, through the annotation process described in Section 4.2.2. This potential is strictly related to its theoretical approach that exploits mathematical, relational-based modeling, i.e., Formal and Relational Concept Analysis. Although there are some studies based on RCA [84], [85], actually, platforms or approaches with a similar theoretical modeling are not comparable to the research herein discussed.

[32] AeroSWARM project page (http://ubot.lockheedmartin.com/ubot/hotdaml/aeroswarm.html accessed on 2 August 2004).

## 7.6 Conclusion

The semantic annotation is a key activity in the Semantic Web technology that, adding formal semantics (metadata, knowledge) to the web content for the purpose of more efficient access and management of web resources, promotes the semantic interoperability.

This work achieves an automatic semantic annotation of web resources. It exploits a data-driven approach based on fuzzy FCA and RCA that allow the elicitation of concepts and relationship in the resources content. The discovered structure is translated into an ontology that collects the semantic annotation extracted from resources. Thus, differently by usual approach the ontology is generated, rather than exploited to capture the semantics in the annotation process.

The semantic textual annotation seems to exhibit interesting results in terms of experimental results, and anticipates promising performances in term of text categorization. The main aspects that reveal the effectiveness of this approach are listed as follows:

- Capacity to classify all words in the text: even the approach works with simple sentences, all the words of the text used in the experimentation are well-classified.
- No pre-established category definition: the application does not use pre-established categories/classes, but compute them on-the-fly.
- Accurate specialization: the application can intercept more specific classes than predefined, known *named entities* through class-subclass relations.
- Customizable specificity level of classification: as a consequence of the elicitation of subsumption relations, the instances of the generated ontologies appear adequately classified, with an accurate detail level.
- Relations design: by the analysis of textual content, different types of relations can be revealed and opportunely modeled.

# Chapter 8

## Ontology based information retrieval applied to e-Learning Recommendations

Typically, e-learning represents an application of Information Technologies (and in particular Internet Technologies) for the development of learning processes, enabling the production and the fruition of educational content at anytime and from anywhere. Recently, in an alternative definition, the "e-learning" is an individual or collaborative group activity where both synchronous and asynchronous communication may be employed. In this context, the diversity of students' background is one of the most important issues. Enrolled students come from many different linguistic, cultural, and academic backgrounds; hence the conventional e-course materials cannot always meet different students' needs. Thus, it is evident that *one curriculum for all* is no longer suitable for the e-learning environments. The aforementioned statement suggests that a great expectation for personalized e-learning is raising [86].

Nowadays, the Semantic Web technologies are considered the most promising solutions to effectively organize and manage available e-learning resources, meeting the peculiar requirements of both teachers and students. Furthermore, in the e-learning domain, an increasing role is given to the knowledge modeling through metadata-based standards [87], although problems of incompatibility due to heterogeneous metadata descriptions might be avoided by using ontologies as a conceptual backbone [88]. On other hand, the Semantic Web approach for the personalization of e-learning processes is also tightly coupled with the availability of great volumes of reusable educational content. The increasing number of IEEE LOM-compliant Learning Object Repositories (e.g. MERLOT[33] with 10607 public objects stored, eRIB[34] with 49761, EdNA Online[35] with 30300, etc.) demonstrates that having more available learning objects means multiply the opportunities to better satisfy the learners' preferences. Furthermore, in the last years, the spreading of Web 2.0 [89] has involved also the e-learning field. Tools like blogs (used to share ideas), wikis (used as a way to construct knowledge in a collaborative way), podcast (used to distribute multimedia files over the Internet) and other web sharing applications (e.g. Flickr, YouTube, del.icio.us, etc.) are exploited by Internet communities in order to work and make business but also to teach and learn. The coherent utilization of the aforementioned tools in e-learning processes is called *e-learning 2.0.*

[33] www.merlot.org

[34] www.edusource.ca

[35] edna.edu.au

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns="http://my.netscape.com/rdf/simple/0.9/">
        <channel>
                <title> Dot Org</title>
                <link>http://www.organiz.org</link>
                <description> Organization web site</description>
        </channel>
        <image>
                <title>Organization</title>
                <url>http://www. organiz.org/images/logo.gif</url>
                <link>http://www. organiz.org</link>
        </image>
        <item>
                <title>New Status Updates</title>
                <link>http://www. organiz.org/status/</link>
        </item>
</rdf:RDF>
```

Figure 46. . A sample of RSS channel

The personalized e-learning experiences can really improve the e-learning processes [90]. It becomes more effective and efficient when a great number of educational content requires be dynamically filtering and assembling with respect to learners' preferences and cognitive states. Then, in the Semantic Web environment, the personalization process should be driven by learners'exigencies and personalized learning content should go to learners with push logic mechanisms. The idea consists of exploiting the Web as a prominent source of educational content for supporting and improving personalized e-learning processes. In order to reach this goal, some important issues will be faced: (a) managing the several types of educational content the Web systems offer (*interoperability*), (b) extracting from the Web and providing to students only educational content that are relevant with respect to the subjects currently treated within their e-learning experiences and suitable for their profiles (*contextualization and personalization*) and (c) driving the Web search activities (for relevant educational content) in order to identify promising Web zones in which to find educational valued content about a domain of interest (*vertical search*). The approach defined in this work proposes:

- The adoption of a standard publication language, like *RSS (Really Simple Syndication)* [89], as a "lingua franca" used to simplify the information management (e.g., extraction, filtering, classification, delivery, etc.) and to solve the *Interoperability* problem.

  RSS (RDF site syndication - Rich Site Summary - Really Simple Syndication) is an Extensible Markup Language (XML) file which is used by sites for syndication of their articles on the Internet. It has been exploited extensively in news sites and weblogs to explain the content and related information of a web page. The XML-based format of RSS is simple. It mainly consists of a "channel" which contains a list of "items" described by a title, a link, a short description (or sum-

mary), a publication date, etc. An example is shown in Figure 46, which illustrates main parts of an RSS feed. Each RSS feed is composed of one <channel> tag. It includes information encompassed by specific tags such as <title>, <link>, <description>. An RSS channel contains one or more items. The <item> element usually has a few elements to describe the content.

This choice simplifies the architectural design assuming to (a) search only for Web Repositories and Sites that publish their content using RSS feeds, (b) use RSS fields values in order to perform filtering, classification, etc. and (c) exploit RSS feeds readers and aggregators [18] to delivery educational content to students.

- The introduction of a mathematical model based on the FFCA which allows the structuring of the educational content through a lattice representation, contextualized to the e-learning argumentation. Through the building of the relative lattice contexts, FCA enables the representation of the relationships between feeds and topics of learning objects. Thus, the system exhibits only content which matches the user experiences.

In literature, several works deal with similar issues [91], [92]. In particular, these works make reference to a class of systems that are known as *recommender systems* and apply them to enrich learning experiences.

The proposed approach in this research work provides an automatic mechanism to build the learning context that represents learner's current needs, cognitive state and preferences. Final result is a personalized learning experience through ad-hoc educational paths and the elicitation of the feeds content to provide further sources of study and, at the same time, support the generation of customized recommendations. Specifically, main advantages produced by the improvement of the personal learning experience can be listed as follows:

- The improvement of the personal learning environment providing quality content (coming from sources approved by teachers/tutors), related to the interest of learners.
- The support to self-learning. The system presents new, continuous content stream to learners without teachers' (tutors') interventions during the "active" learning experience.
- The obsolescence risk for learning objects is almost non-existent, thanks to RSS feeds which provide content constantly updated and allow for keeping update also the learning objects stored in more static repositories.
- The system provides learners both content that directly refers to their interests and content that is semantically related to their interests. The exploration of concepts semantically related to the current learning objectives tend to increase students ability to remember what they have learned, to enlarge new knowledge, and to transfer it to new tasks more effectively than passive approaches.

The chapter presents an e-learning recommender system which enables a contextualized RSS-feeds fruition to support students in their learning path. Section 8.1 describe a Web

platform using in this case study. Section 8.2 introduces a sample scenario for the user interaction with the system. Section 8.3 details the experimental results. Finally conclusions close the chapter.

## 8.1 Intelligent Web Teacher

The Intelligent Web Teacher (IWT) is an e-Learning Web-based platform whose distinctive features are the construction and delivery of personalized e-learning experiences through the execution of specific algorithms [93]. Using these algorithms it is possible to generate courses tailored to a class, to a specific group and even to single learners. The foundation of IWT is the *Learning Model* described in [94]. The *Learning Model* allows to automatically generate a Unit of Learning (i.e., a course, a module or a lesson structured as a sequence of Learning Activities represented by Learning Objects and/or Learning Services) and to dynamically adapt it during the learning process according to the learner's preferences and cognitive state (personalization process). A Unit of Learning (UoL), during its execution, represents what we have previously named e-learning experience. In IWT, the piece of the educational domain that is relevant for the e-learning experience we want to define, concretize and broadcast is formalized in a machine-understandable way. The used mechanism is named ontology, i.e., an engineering artifact, constituted by a specific vocabulary used to describe a certain reality, plus a set of explicit assumptions regarding the intended meaning of the vocabulary words.

In the IWT approach, the vocabularies are composed by terms representing subjects (or concepts within an e-learning ontology) that are relevant for the frame of the educational domain we want to model. Subjects are associated to other subjects through a set of three conceptual relations: *HasPart* (in brief *HP*) that is a part-of relation, *IsRequiredBy* (in brief *IRB*) that is an order relation and *SuggestedOrder* (in brief *SO*) that is a "weak" order relation. The ontologies constructed following the few aforementioned informal rules are named e-learning ontologies [95].

Furthermore, Learning Objects are associated to subjects within a specific e-learning ontology by means the relation *Explain* (in brief *Exp*). E-learning experiences are defined as: (i) a set of *Target Concepts* (*TC*), i.e., the set of high-level concepts to be transmitted to the learner; (ii) a *Learning Path* (*LP*), i.e., an ordered sequence of atomic concepts (subjects) that is necessary to explain to a learner in order to let him/her learn *TC*.

Given the personalization on a particular learner, the sequence does not contain subjects already "learn" (i.e., known with a grade greater than the fixed threshold) by that learner (the information is managed by means the *Learner Model*); (iii) a *Presentation* (*PR*), i.e., an ordered list of learning objects that the learner has to use in order to acquire knowledge about subjects included in *LP*. Figure 47 shows an overview of the personalized e-learning experience definition process foreseeing the selection of the *TC* (performed by the teacher), the automatic extraction of *LP* (performed by the IWT algorithms) and the automatic binding between Learning Objects and *LP* concepts (performed by the IWT optimization algorithms).

The exclusion of the subject $C_4$ and the selection, for instance, of the learning object $LO_2$ in the place of $LO_1$ is due to the personalization process that takes into account the cognitive state and the learning preferences of the current involved learner. IWT, like other e-Learning systems [96], exploits Knowledge Representation techniques and languages in order to model educational domains for e-learning purposes.

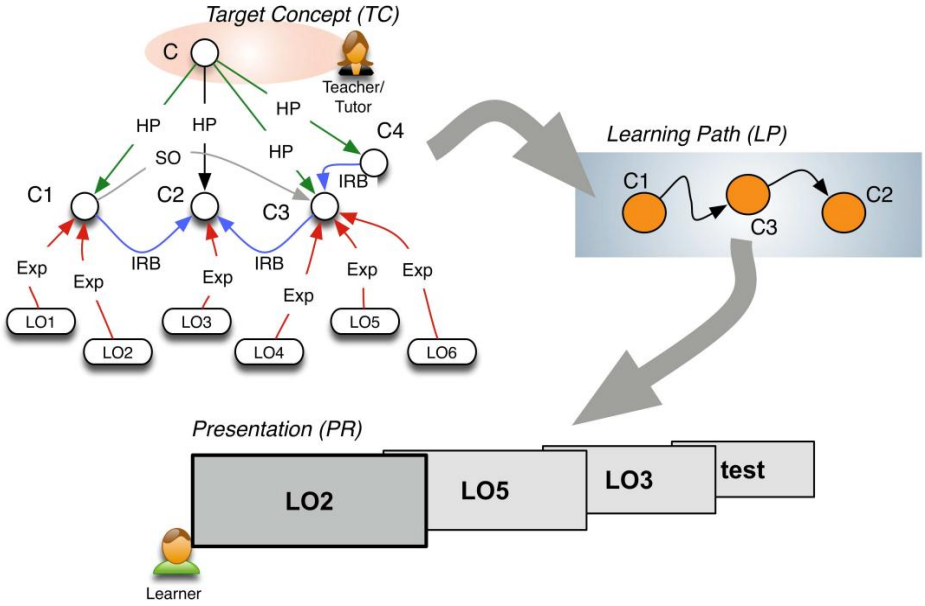



Figure 47. The IWT e-Learning Experiences definition process

Moreover, IWT uses the aforementioned explicit knowledge representation to personalize (in automatic way) the learners' experiences. In fact, it accomplishes the personalization task with high performances in terms of response-time and provides an ad-hoc e-learning reasoning engine that works on knowledge base composed of educational domain ontologies and user profiles.

## 8.2 A sample scenario

In order to provide an overview of the integrated system, a sample scenario is depicted through the main steps a teacher and a learner accomplish during, respectively, the definition and the fruition of the learning experience.

Let us suppose, Robert, a full professor of Mathematics, is preparing a personalized experience using IWT and Mark, one of the undergraduate students, has to attempt the course of Mathematics in e-learning mode. Robert defines an ontology which represents the knowledge that is relevant for the learning module he is preparing in his course (see Figure 48: *Ontology design*). He fixes the learning objectives and related parameters and then, he may suggest also a list of RSS feeds (by providing URL format) which include additional related learning material for deepening study on course topics by learners (Figure 48: *Additional RSS sources*). The list is processed in asynchronous mode; the immediate modifications have no effect on the system: a query does not produce new results until the Training component process the last feeds (Figure 48: *System Training Process*). The system alerts Robert through a notification when the training phase is finished and new feed content has been embedded into the lattice. At this point, Robert can submit his query, for instance, *Differential Equation* to the system. On the other hand, Mark navigates the learning objects or-

ganized as personalized learning path extracted by the IWT algorithms, in order to maintain the cognitive state and learning preferences of Mark.

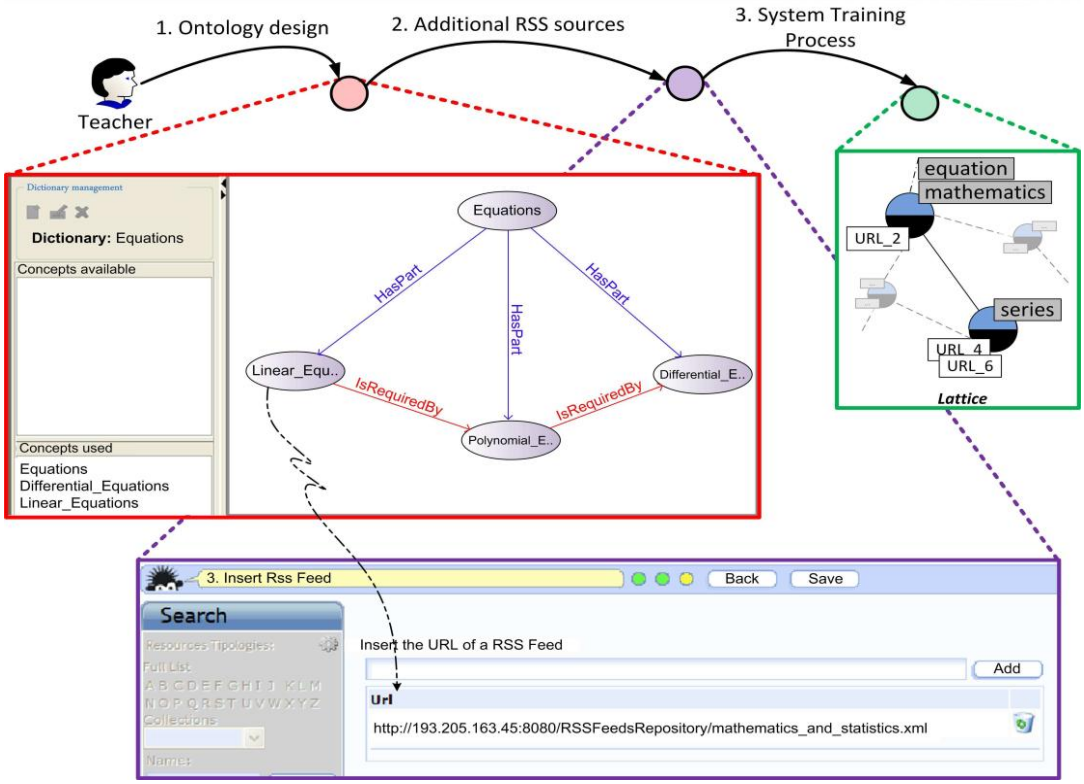



Figure 48. Preparing an RSS-based learning experience in IWT

In particular, as Mark is registered to the Mathematics Course (managed by Robert), he follows the proposed personalized sequence of learning objects in the context of mathematics, as shown in Figure 49 where the main snapshots of the system are evidenced. During the learning objects navigation (Figure 49: *Learning Objects navigation*), Mark can select the Differential Equation and loads related content (Figure 49: *Learning Object Access*). Then, he can exploit the additional links which provide on-the-fly additional learning material about the subject. Indeed, Mark can choose to access the additional educational material by following the link *Suggested Insights*. IWT prepares a query in a transparent way and sends it to the Query/Answering component (Figure 49: S*ystem Q/A Process)*. The query is formalized as the active subject in Mark's course fruition (i.e., the learning context). The Query/Answering component executes the query and returns results to Mark as an aggregate view of posts (extracted from RSS Feeds) that is relevant for his current study.

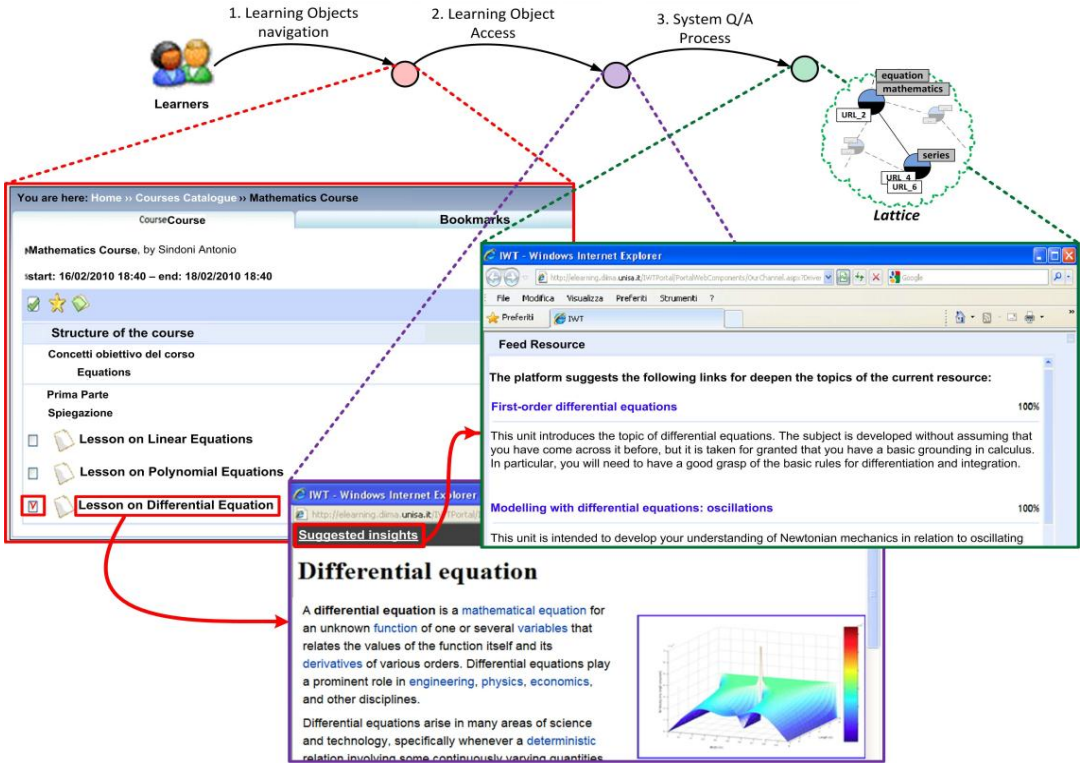



Figure 49. Executing an RSS-based learning experience in IWT

## 8.3 Experimental Results

The proposed system consists of an asynchronous Web oriented application, which allows users concurrent accesses. The asynchronous nature of the system enables users' interaction, without waiting for a reply, when a query is submitted. Unfortunately, the training process requires some time for the start-up loading, due the onerous activities like the text parsing of the new or updated feeds, the keyword extraction, the building of the context-based matrix and finally the lattice generation. Thus, in order to guarantee feed updating, the lattice is re-generated periodically (i.e., once a day); meanwhile, the user can use the last updated version of lattice for his interactions.

The system provides good query/answering performance: for instance, considering a lattice composed of 500 nodes (viz. fuzzy concepts) and a query with two terms, the average response time is about 4s. Table 17 gives the response time, considering different parameters setting (nodes of lattice and terms for each query).

Table 17. System query/answering response time

| # Nodes of Lattice | # Terms of the Query | Time (second) |
|---|---|---|
| 615 | 2 | 4,36 |
| 615 | 3 | 6,86 |
| 430 | 2 | 2,76 |
| 430 | 3 | 3,34 |
| 205 | 2 | 1,23 |
| 205 | 3 | 1,87 |

The approach has been validated on a collection of RSS-Feeds, coming from OpenLearn Project. Precisely, a sample of 589 RSS-Feed has been selected.

After the processing (described in Section 4.2.1) of these feeds collection, the resulting lattice evidences the most of RSS-Feeds are grouped consistently in the appropriate categories. Let us notice the assessment of the produced lattice is not an immediate activity and requires a careful analysis of the generate concepts (through their associated objects and attributes). Thus, from an accurate analysis of the sample content, the approach seems to elicit categories which are representative of the given sample; in particular let us observe the lattice produces a more specialized classification of feeds. Indeed, some feeds are placed in formal concepts, which are subclasses of other concepts, associated to more general OpenLearn categories. That evidences the hierarchical structure of the lattice, through a specialization of the original OpenLearn categories.

More specifically, 88% of whole collection of feeds is classified coherently with respect to OpenLearn's categories and the 32% of this percentage appears in some specialized categories. This is due to the high amount of feeds contained in formal concepts, then the nature of the lattice promotes their specialization in deeper sub-classes.

The remaining of the whole collection (i.e., 12% of the feeds) is misclassified: the analysis of these feeds reveals a certain ambiguity in the content, which does not guarantee an accurate classification.

After the processing (described in Section 4.2.1) of these feeds collection, the resulting lattice evidences the most of RSS-Feeds are grouped consistently in the appropriate categories. Let us notice the assessment of the produced lattice is not an immediate activity and requires a careful analysis of the generate concepts (through their associated objects and attributes). Thus, from an accurate analysis of the sample content, the approach seems to elicit categories which are representative of the given sample; in particular let us observe the lattice produces a more specialized classification of feeds. Indeed, some feeds are placed in formal concepts, which are subclasses of other concepts, associated to more general OpenLearn categories. That evidences the hierarchical structure of the lattice, through a specialization of the original OpenLearn categories.

Table 18. OpenLearn categories vs. some FFCA-based specializations

| OpenLearn Category | Concept | Rss-feed ID |
|---|---|---|
| **Mathematics and Statistics (39)** | Differential Equation (4) | MST209_8,MST209_1, MST209_6 |
| | Series (2) | MST209_9, MST209_5 |
| | Geometry Mathematics (4) | MU120_1, ME624_1, Y162_1, ME825_1 |
| | … | … |
| **Business and Management (34)** | Market (2) | B821_2, B700_1 |
| | Project Management(4) | B713_1, B713_3, B713_4, B713_5 |
| | Management (34) | Y159_2, B700_2 |
| | … | … |
| **Law (10)** | Parliament (1) | W100_2 |
| | Privacy (1) | W100_6 |
| | … | … |
| **Technology (34)** | Engineering Technology (5) | T207_1, T207_2, T837_1, T839_1, LIB_10 |
| | Managing Complexity (2) | T306_1, T306_2 |
| | … | … |

In particular, increasing the threshold $\alpha = 0.7$ some relevant feeds have been lost. The second case ("Differential System", Figure 50-b) emphasizes good performance in terms of the precision, when the threshold values are in the range [0.5, 0.8] and the recall, when threshold values are smaller than 0.7. The last case considers a more general subject ("Mathematics", Figure 50-c); as a consequence, the resulting tendency of the precision is almost good, by varying the threshold levels; in particular the threshold value greater than 0.7 guarantees all the relevant feeds are retrieved. The recall instead tends to decrease, when the threshold value increases; this evidences that the higher the threshold is, the smaller number of relevant feeds is retrieved. That means a high threshold applies a fine-grained filtering on the feeds in the lattice, according to their membership values.

A further query/answering performance has been evaluated comparing the approach defined with OpenLearn. Precisely, we have submitted the same query to the both systems in order to compare the results. Generally, the approach works better than OpenLearn when the query is composed of two terms at least, differently from OpenLearn which returns more relevant feeds with a single word query. The reasons are due to the fact the approach achieves a hierarchy of concepts, thanks to the lattice structure and the textual parsing of the collected feeds enables the extraction of linguistic details. This way, a specific query (i.e., composed of some words) finds more pertinent answers exploiting the approach rather than a system with a flat categorization.

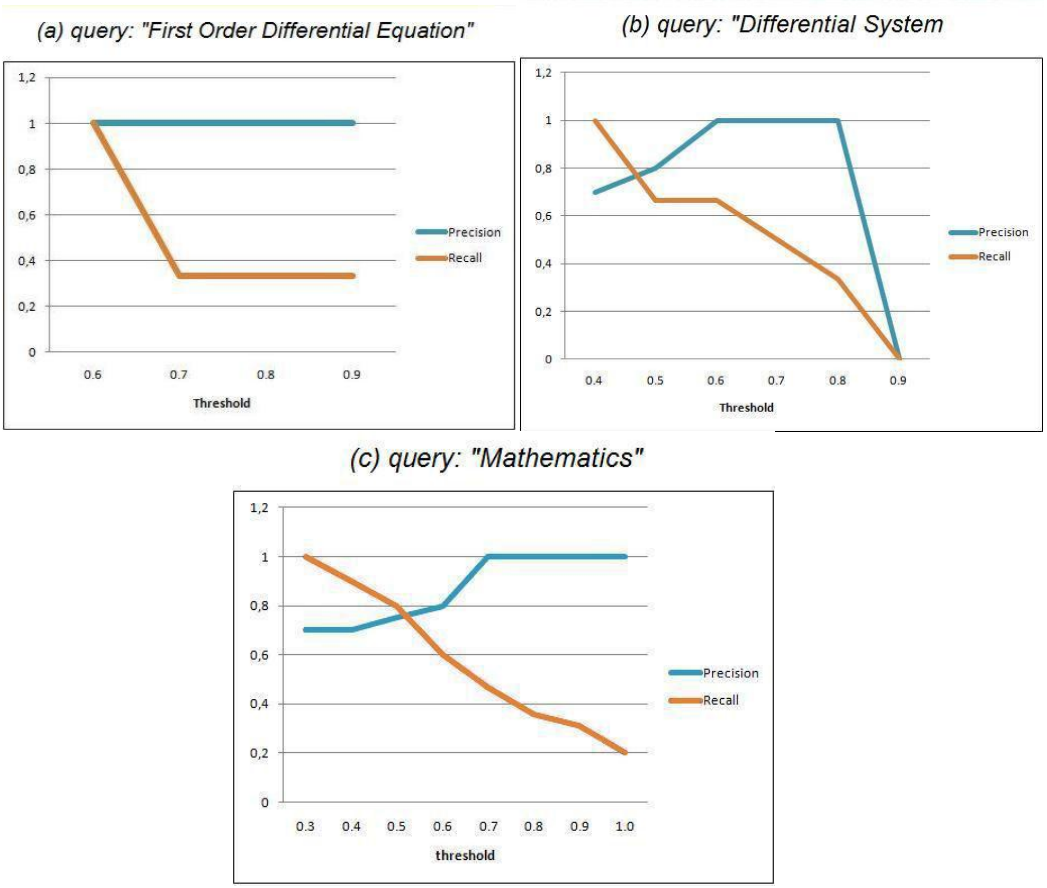



Figure 50. Precision/recall evaluation, given the queries

Just to give an idea, Figure 51 compares the query/answering performance of both systems; each query includes the term "equation" and incrementally adds a to the query expression: Precisely, "equation"', "differential equation", "differential equation system", "differential equation system model" are the queries submitted to the systems. Let us observe the proposed approach produces better results with queries composed of two and three terms, confirming the good performance due to the hierarchy-based data classification. Moreover, the approach can return feeds whose textual description does not contain explicitly terms specified in the query: the parsing activity extracts additional words that are related to the query (i.e., synonyms, hyperonyms, etc.). The analysis of the list returned by the query emphasizes the relevance and the specificity of the retrieved feeds.

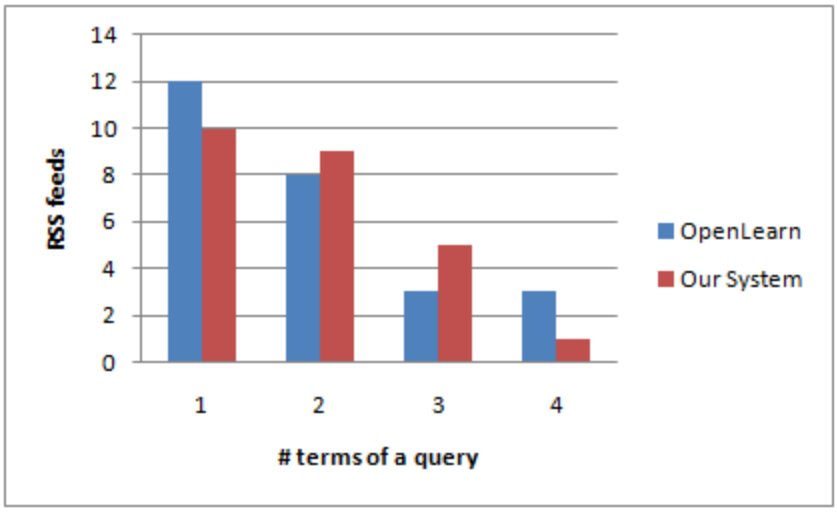



Figure 51. Comparative query/answering performance with an incremental query

## 8.4 Related Works

Nowadays, the Semantic Web technologies are considered the most promising solutions to effectively organize and manage available "e-learning" resources, meeting the peculiar requirements of both teachers and students.

In literature, several works deal with recommendation-based approaches in the e-Learning domain. In [97], the authors define two classes of recommender systems: *Knowledge-based Recommender Systems and Social Recommender Systems*. Systems belonging to the first class use ontologies to manage content available for recommendations. Recommendations occur on the basis of matches between user profiles attributes and ontologies. This kind of system is also called *Attribute-based Recommender System*. The systems belonging to the second class exploit the concept of users' similarity (the system recommends a content to a user if a similar user has enjoyed this content or if he/she has already enjoyed similar content) and use one of the following filtering techniques: *User-based Collaborative Filtering* assumes that two users are similar if they similarly rate the same content; *Stereotype Filtering* assumes that two users are similar if they have similar profiles; *Item-based Filtering* assumes that content similarly rated are similar. We think that Social Recommender Systems (although they are very useful in informal learning processes) are not adequate when formal learning activities are delivered so we have investigated the integration of several techniques in order to define the proposed approach that can be considered as a Knowledge-based Recommender System. In particular, the criticism about the ontology construction, underlined in [97], is relaxed by the use of FFCA technique in order to automatically build the conceptualization of the contents to be considered during the recommendation process.

In [98], the authors combine an *Ontology Content-based Recommender System* with an *Interest-based Recommender System*. Content is analyzed and classified using an ontology that is automatically built using the same content. Moreover, recommendations are forwarded to learners on the basis of their current activity (the subject they are studying). Conceptu-

ally, the proposed approach is very similar to the approach presented in [98], although the research work leverages on a more refined interest-based technique granted by the learning model used in IWT (see Section 8.1) where the *learning path* is represented by both a content layer and a concept layer. By exploiting this technique, the proposed system always guesses the concept (and its related concepts) that a learner is interested in. Then, using the concept layer, the proposed system can implicitly deduce learners' current interests and accurately query the recommender system.

In [91], the authors synergically exploit four techniques in order to define their recommendation approach: Cont*ent-based, Collaborative Filtering, Rule-based and Demographic-based*. The originalities with respect to other research are represented by the introduction of *Rule-based* and *Demographic-based* techniques. The first one applies a filter to recommendations on the basis of rules attached to some attributes coming from user profiles. The second one is used in order to suggest content with respect to some demographic properties of the users (e.g. geographical region, age, etc.). Furthermore, in [91] is introduced the idea of *Teacher's Recommendation* that is proposed also by proposed approach, where teachers suggest to the system the Feed URLs to be used as content sources. Then, the feeds are fragmented (into RSS posts) and analyzed. This approach provides a solution to filter the content used for recommendations where formal learning activities are executed. While in [91] the recommended contents are Learning Objects, proposed approach is focused on RSS posts recommendations (extracted by sources suggested by teachers) because IWT already provides the personalization of learning object sequences. So, proposed approach provides richer formal learning experiences by exploiting several controlled content sources with respect to other approaches.

In [92], the authors use Mining and Information Retrieval techniques in order to provide a *content-based recommender system*. Their approach exploits a *learner model* and a *content model* that are constructed to support the recommendation system. In the proposed approach, the *learner model* is already provided by the IWT system (see Section 8.1) and is constantly maintained updated by IWT algorithms that are exploited in order to personalize learning experiences.

With respect to other works, the proposed approach achieves a synergic application of the lightweight ontologies and the fuzzy FCA theory. Ontologies guarantee the modeling of educational domains and support algorithms for the automatic construction of personalized learning paths. On the other hand, Fuzzy FCA enables the extraction of contextual knowledge embedded in the RSS Feeds content and arranges it in a hierarchical structure.

Moreover, educational ontologies are used to deduce and formalize the *learning context* exploited by a specific learner during his learning experience. The learning context is used to automatically query the Fuzzy FCA-based system in order to select all relevant content (RSS posts) and present them to the interested learner. Let us stress the system response is tailored on a specific learner, i.e., takes into account the corrisponding learning context, derived by a customization of an educational ontology portion. Similar approach is presented in [99] where crisp FCA modeling is exploited to produce learning concept hierarchies from a text corpus. In [100] fuzzy FCA becomes a kind of conceptual clustering and solves the high sparsity problem of user rating matrix. In particular, a fuzzy concept is viewed as a fuzzy cluster of users with similar interests, while in the proposed approach, a fuzzy concept represents an actual concept arising from the feeds content.

The work presented in [101] instead, aims at discovering and visualizing the domain ontology from the on-line messages created by the learners rather than exploiting the building of automatically raising an ad-hoc taxonomy (coming from FCA model) of educational resources. Their ontology discovery method is a fuzzy taxonomy generation based on subsumption relations among extracted concepts. Similarly to the proposed approach, each concept is expanded: in [101] WordNet-based synonymy relations are added, while, in the proposed approach, similarity measures based on the sense, defined in WordNet, are associated to each concept.

Many other approaches in literature, deal with similar problems. Just for giving some examples, in [102] a zigzag structure has been exploited to define relations among information. The approach allows the user to create personalized concept maps and semantic interconnection among web resources. In [103] instead, a hyperbolic structure substitutes the traditional concept map diagrams for visualizing concept spaces in the educational course. Differently, the proposed approach aims at eliciting intrinsic relationships among the given data and resources, without any human intervention. Moreover, as said, the fuzzy extension of this model reveals a degree of interrelations (i.e., an approximate subsumption) between linked concepts. The resulting fuzzy lattice reveals knowledge-based, hierarchical dependences elicited from the feeds content, emphasizing the taxonomic nature of this structure.

## 8.5 Conclusions

This chapter presents a work that achieves a system which provides, in push logic, suitable, contextualized and personalized RSS-based educational content. The system improves the personalized learning process in IWT through the application of Fuzzy Formal Concept Analysis. The integration of methodology described in Section 4.1 in IWT overcomes some limitations of existing e-learning Recommender Systems. In particular, the proposed approach (i) provides automatic mechanism to build the learning context that represents learning current needs, cognitive state and preferences and (ii) handles linguistic issues (e.g., synonymy, meronymy, etc.) improving content selection. A conceivable future development foresees also an extension, by defining a crawler which will carry out a specific "focused" spidering of the web. The crawler will act identifying e-learning community sites and will be able to discriminate relevant web pages whose content have educational value and are related to the given subjects of interest. Furthermore, it could provide a solution to the *Vertical Search* problem, pointed above.

# Chapter 9

## Ontology based information retrieval applied to Disease Diagnosis

Traditional approaches to the medical diagnosis practice have many drawbacks, like as: the huge growth of biomedical information has made difficult the retraining for the individual doctors; the poor dissemination of effective research results; and so on.

New trend is the Evidence-Based Medicine (EBM) which aims to apply the best available evidences gained from the scientific methods to clinical decision making. The challenge of EBM is to define a systematic approach to integrate research results with clinical expertise and patient preferences, and exploit them during the medical diagnosis. So, it is necessary to provide a suitable model to structure medical results and to perform the knowledge spreading and sharing.

There are several ongoing efforts aimed at developing formal models of medical knowledge and reasoning to design decision support systems. These efforts have focused on representing content of clinical guidelines and their logical structure. Semantic Web technologies and ontologies are enabling elements to achieve these aims. In fact, today ontologies are assuming increasingly important role in the area of knowledge based decision support systems by introducing capabilities in terms of logic based reasoning.

This work presents ODINO, a multilingual web based application that addresses the aim to use semantic web technologies in order to support medical practices through an effective user interface. In particular, ontologies are used to model available medical diseases features (e.g., skin diseases), symptomatologies, treatment protocols and so on. The relations between symptomatologies (i.e., symptoms and signs) and available diseases are represented by using fuzzy labels resulting from the analysis of medical expertise included in [104]. Standard formalisms, like OWL and SKOS, are used to specify domain knowledge and controlled vocabularies, i.e., diseases, symptomatology, active ingredients and clinical tests according to standard specifications, like as ICD-9-CM[36].

Ontologies, controlled vocabularies and information retrieval techniques are exploited to provide typical capabilities of Semantic Web portals [105] and medical decision support. Some of the main features of the system are:

- disease catalogue browsing including images, symptoms and signs, treatments, etc. to support rapid training;
- preliminary medical diagnosis, indeed medical knowledge querying by specifying symptoms, signs and complications, to find eligible diseases;

[36] International Statistical Classification of Diseases, 9th Revision, Clinical Modification

- faceted search of diseases by enabling multi-criteria selections (i.e., symptomatologies, complications, active ingredients, etc.) to support differential diagnosis.

ODINO results are explained to physician by highlighting symptoms/signs/complications that match the retrieved diagnosis and by suggesting other important features of founded diseases.

The chapter is organized as follows. Section 9.1 describes the knowledge layer of ODINO. The medical decision support methodology is described in Section9.2. Section 9.3 details features of system and provides the results of the case study. Then, Section 9.5 introduces some related works in the applicative domain. Finally, there are conclusions of the chapter.

## 9.1 Knowledge Layer

ODINO is a knowledge based system that provides capabilities in terms of clinical decision support system and medical semantic web portal. In particular, ODINO makes an intensive usage of ontologies, controlled vocabularies and other standards typical in the medical domain. This section provides details about medical knowledge modeling of ODINO.

### *9.1.1 Technologies and Standards*

Semantic technologies (i.e., OWL and SKOS) have been used to represent medical diseases. On the other hand, knowledge developed in ODINO has been aligned with international standard classification of medical diseases available in ICD9-CM. In particular, in this work, W3C's recommendation has followed to build ontologies and construct knowledge bases according to the OWL-DL restrictions. A Medical Disease Ontology has been defined as described in Section 9.1.3. SKOS is used in the proposed system together with OWL in order to express and share knowledge about medical domain. In particular, ODINO exploits SKOS to write vocabularies and taxonomies, like as: symptomatologies, diseases, drugs, etc. (as described in Section 9.1.2). On the other hand, OWL is used to represent the formal model of disease ontology by defining axioms and constraints. The International Statistical Classification of Diseases and Related Health Problems (ICD) is a way to code and organize diseases and a wide variety of signs, symptoms, abnormal findings, complaints, and external causes of injury or disease, published by the World Health Organization[37] (WHO). It assigns a unique up to six characters long code to every health condition. In this work, the ninth version of ICD (ICD-9) has been used. In particular, ODINO uses ICD9 - Clinical Modification (ICD-9-CM) that provides additional morbidity details than ICD9. ICD9-CM has been applied to align controlled vocabularies deployed during the knowledge base design. This alignment provides shareability of the knowledge base deployed for ODINO.

[37] http://www.who.int/en/

### *9.1.2 Controlled Vocabularies & Taxonomies*

Some controlled vocabularies and taxonomies (i.e., diseases, symptomatology, clinical tests and active ingredients) have been deployed in ODINO by using SKOS technology. Generally, following SKOS properties have been used to represent vocabularies:

- *preferredLabel* (in both Italian and English languages), to define the favorite name of concept;
- *alternateLabels* (in both Italian and English languages), to associate alternative terms to the same concept in order to specify synonym, and so on;
- *hiddenLabel*, to associate the right ICD-9-CM code to the defined concepts (i.e., disease, symptom, etc.) taking into account results from the alignment process;
- *exactMatch,* to eventually specify relation between concepts with equivalent meaning;
- *broader* and *narrower* to specify hierarchical relations among concepts. These properties are enabling elements to create taxonomies and to support faceted navigation as described below in Section 9.3.3;
- *prefSymbol* and *altSymbol* properties, to join one or more images to concept.

In particular, *exactMatch* allows to maintain ICD9-CM coding and to update the knowledge base with alignment to others classification systems (such as ICD-10 or ICD-11).

Controlled vocabularies and taxonomies developed in ODINO are:

- *Symptomatology* – that contains definitions of concept concerning symptoms and signs, like as: paresthesia, anhidrosis, plaque of psoriasis, etc. Listing 5 shows an example of SKOS concept modeling a symptomatology and their hierarchical relations;

Listing 5. An *example* of symptom specification in SKOS

```
<owl:Thing rdf:about="#Sin157">
    <rdf:type rdf:resource="&skos;Concept"/>
    <skos:prefLabel xml:lang="en">Hand swelling</skos:prefLabel>
    <skos:prefLabel xml:lang="it">Edema delle mani</skos:prefLabel>
    <skos:altLabel xml:lang="en">Hand edema</skos:altLabel>
    <skos:hiddenLabel>729.81</skos:hiddenLabel>
    <skos:prefSymbol>edema1.jpg</skos:prefSymbol>
    <skos:altSymbol>edema2.jpg</skos:altSymbol>
    <skos:inScheme rdf:resource="#Symptomatology "/>
</owl:Thing>
```

- *Disaeses* – that contains medical diseases, their characteristics and hierarchical relations like as: leprosy, malaria, AIDS, etc. In particular, broader and narrower properties, in SKOS, allow defining a hierarchy of diseases;

- *Clinical Tests* – that models clinical and laboratory tests useful in diagnostic process, like as: haemochrome, urinalysis, specific blood tests for syphilis, etc.

- *Active ingredients* – that defines the drugs suitable for a specific disease treatment, like as: Dapsone, Rifampicin, Antacids, etc. Since, ICD9 doesn't contain any drug classification, this vocabulary is not aligned with it.

### *9.1.3 Medical Disease Ontology*

This section describes main aspects of knowledge models developed in ODINO. An ontology named *Medical Disease Ontology* has been defined. Through a set of properties and concepts, this ontology models disease characteristics and relations. In particular, *Medical Disease Ontology* includes the definition of a correlation degree between disease and its symptomatology by using fuzzy labels. These properties lead system's diagnosis processes and aid disease information consultation.

One of the main classes defined is the concept of Disease which identifies condition of medical disease that causes pain, dysfunction, distress, social problems, and/or death, to afflicted person. Figure 52 shows a sketch of the ontology by illustrating the relations between *Disease*, *Symptom/Sign* and *Complication* classes. In particular:

- *Symptom/Sign* is the class identifying symptom or sign that influences a disease. Many instances of *Symptom/Sign* may be associated to the same *Disease*. So, the relation *"has symptom"* hasn't any cardinality restriction.

- *Complication* is a medical disease or symptom that represents an unfavorable evolution of disease, health condition or medical treatment. Analogously with *"has symptom",* the relation *"has complication"* hasn't any cardinality restriction.

*Complication* and *Symptom/Sign* are themselves related with other classes:

- *Evidence,* that represents the bridging element with the *Symptomatology* SKOS vocabulary;

- *Specificity* and *Frequency,* that allow to indicate how much *Complication* or *Symptom/Sign* implies the presence of that disease.

Furthermore, as shown in Figure 53, *Disease* is also related with *Clinical test* class. *Clinical test* identifies each possible clinical, laboratory and diagnostic test. Analogously with *Symptom/Sign* and *Complication*, the model foresees the specification of *Precision* degree between a *Clinical test* and a *Disease* (see Figure 53). In particular, the *Precision* defines how much a *Clinical test* result determines the presence of specific disease.

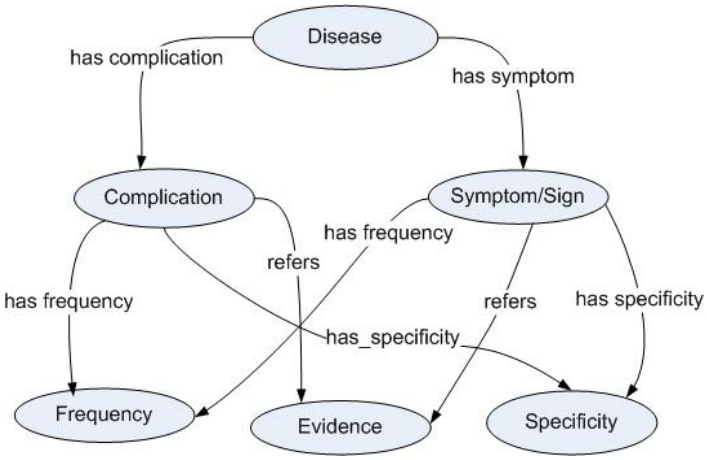



Figure 52. A sketch of Medical Disease Ontology: Disease, Complication and Symptom/Sign.

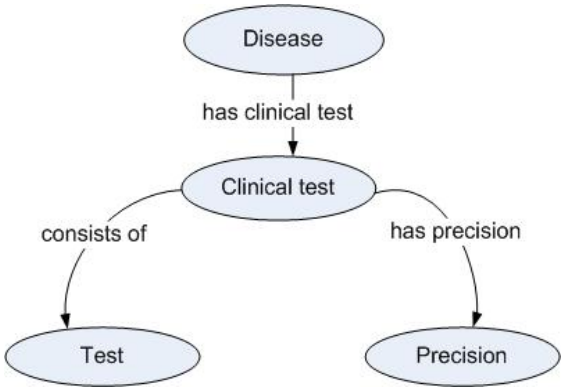



Figure 53. A sketch of some classes of *Medical Disease Ontology*: *Disease* and *Clinical test*.

*Frequency, Specificity* and *Precision* are enumeration classes defined by using *oneOf* construct available in OWL. More precisely, we have defined, for each of them, some individuals that identify fuzzy labels. Values assumed by these relations are obtained from the analysis of medical practice and experience included in [104]. Specifically:

- *Frequency* - is the rate with a symptom/sign or complication is present in a given disease. Identified levels are: *very frequent, very frequent/frequent, frequent, frequent/occasional, occasional*;
- *Specificity* - is the measure with which manifestation implies a specific disease. Identified levels are: *high, high/medium, medium, medium/low, low;*
- *Precision* - is the level with which *Clinical test* result allows to include or exclude a diagnosis. Identified levels are: *high, high/medium, medium, medium/low, low*.

Furthermore, the model foresees other concepts and relations useful to define additional information of diseases, and recommendation, etiology, epidemiology, transmission of disease, prophylaxis, and so on.

Code in

Listing 6 shows an individual definition of Disease as an example. In particular, "Sin189" is of type *Evidence* and is associated with "*Medium/Low*" specificity and "*Frequent*" frequency degree to disease "*Mal002*". Furthermore, we have associated a complication (*infertility*) and a *NMR (Nuclear Magnetic Resonance)* test.

As detailed in the following section, *Medical Disease Ontology* is the baseline to build the medical diagnosis methodology.

Listing 6. *Medical Disease Ontology*: individual disease example

```
<owl:Thing rdf:about="&Symptomatology ;Sin189">
  <rdf:type rdf:resource="#Evidence"/>
</owl:Thing>
...
<owl:Thing rdf:about="#Sem_fever">
   <rdf:type rdf:resource="#Symptom/Sign"/>
   <refers rdf:resource="&Symptomatology ;Sin189"/>
   <frequency rdf:resource="#Frequent"/>
   <specificity rdf:resource="#Medium-Low"/>
</owl:Thing>
…
<owl:Thing rdf:about="&Diseases;Mal002">
   <rdf:type rdf:resource="#Disease"/>
   <has_semeiotics rdf:resource="#Sem_fever"/>
   <has_complication rdf:resource="#Com_infertility"/>
   <has_clinical_test rdf:resource="#Acc_RMN"/>
</owl:Thing>
```

## 9.2 Medical Diagnosis Methodology

### *9.2.1 Mathematical model to support preliminary diagnosis*

One of the main features supported by ODINO is preliminary medical diagnosis. During medical practices, physician submits a query by specifying one or more evidences that may be symptoms, signs and/or complications. So, the system evaluates correlation degree among incoming query and available diseases by means of an approach based on Information Retrieval techniques and FFCA [106]. Furthermore, system is capable to use relations among concepts in order to augment query results by using a Description Logic reasoner.

As described above, *Specificity* and *Frequency* are the main features to include or exclude diagnosis results. Specifically, a mathematical model of *Disease* and its correlation with *Symptom/Sign* and *Complication* may be obtained in order to perform FFCA.

The FFCA, informally, exploits a matrix that represents the fuzzy relation between objects of the input data (i.e., *Diseases*) and some attributes (i.e., *Symptoms/Signs* and *Complications*). The relation is calculated by assessing both *Frequency* and *Specificity* degree between *Disease*, *Symptom/Sign* and *Complication*. In particular, the value of relation, $\mu_{i,j}$, between $Disease_i$ and $Symptoms/Signs_j$ or $Complications_j$ is calculated by performing a linear combination (a weighted sum) of *Specificity* and *Frequency* with parameters $\lambda_1$ e $\lambda_2$. These parameters consist of constant values chosen based on medical experience, depicted in [104], which shows that, during preliminary diagnosis, *Frequency* is less important than *Specificity*. Formally:

$$\mu_{i,j} = (Specificity * \lambda_1) + (Frequency * \lambda_2), \lambda_1 > \lambda_2$$

Taking into account the matrix created according to these criteria, FFCA arrange data in a corresponding lattice [106]. Specifically, diseases that share symptoms/signs or complications are arranged together. As argued, the lattice is used to retrieve concepts closer to the incoming query in order to retrieve eligible set of diseases.

### *9.2.2 Identification of candidate disease*

In this process the Information Retrieval described in Section 4.2.3 is applied to get a synthetic value which represents how each resource is relevant with respect to the user query. The score represents correlation degree between query and available diseases. Then, a ranked list of diseases may be retrieved in order to answer to a given query.

## 9.3 Case Study

Actually ODINO is used in INMP and in the Hospital of San Gallicano, Rome.

Taking into account the experimental results obtained, next subsections detail main features of ODINO. ODINO has been tested with dermatological disease knowledge. Defined vocabularies include: 700 symptoms/signs; 150 diseases; 125 clinical tests; 260 active ingredients.

### *9.3.1 Disease Catalogue Browsing.*

This feature accomplishes the aim to provide rapid training on specific diseases. In fact, ODINO knowledge base includes many multimedia information, such as: definition and historical data, etiological agents, disease transmission, large set of pictures revealing specific symptomatology, complications, clinical tests and treatment protocols, and so on.

### *9.3.2 Preliminary medical diagnosis*

This functionality allows to assist physician during preliminary diagnosis practice. Physician starts the process by selecting some clinical manifestations (i.e., symptoms, signs and/or complications). As shown in Figure 54, ODINO supports selection through two alternative interactions, namely: by browsing ICD-9 tree and specifying ICD-9 categories; by using text based search.

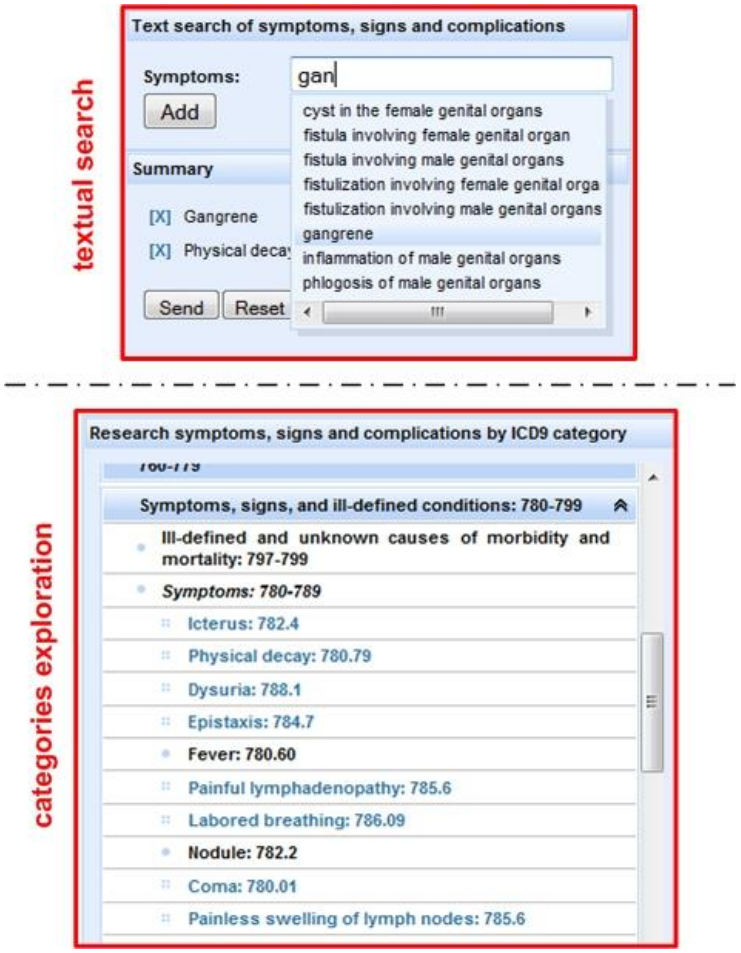



Figure 54. Selection of clinical manifestation by mean textual search (with autocomplete) or categories exploration

The matching algorithm (described in Section 9.2.2) allows the system to retrieve eligible diseases. So, taking into account clinical manifestations selected by the user (e.g., *Gangrene* and *Physical decay*), as highlighted in Figure 55, the system retrieves a ranked list of eligible diagnosis. In particular, the system shows correlation degree useful to rank the results (Figure 55(a)) and it allows to access to motivations of each result (in Figure 55(b)).

Result motivations consist of hints to the physicians about distinctive symptomatology of retrieved disease. These hints may be useful to suggest eventually escaped factors. For example, as shown in Figure 56, system highlights: submitted symptoms in the request (red outlined box) and *pathognomonic signs* (green outlined box) for the retrieved disease (i.e., in the picture *"Venereal ulcer"*). This is useful information because the presence of a *pathognomonic sign* means, beyond any doubt, that particular disease is present.

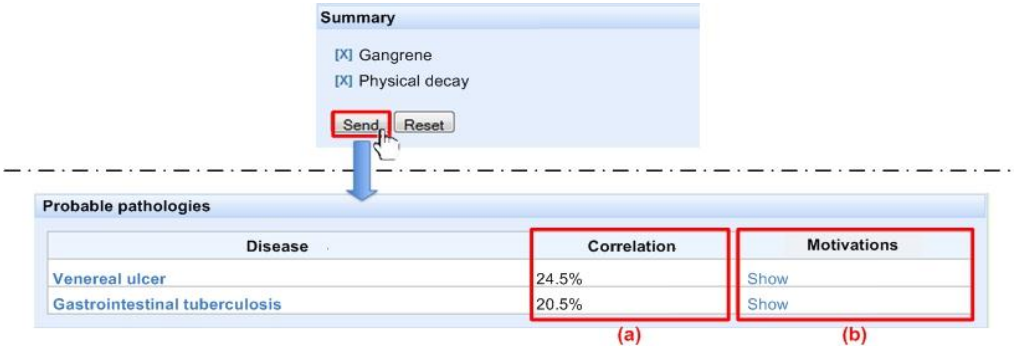



Figure 55. Preliminary diagnosis retrieved results: (a) correlation degree, (b) links to motivations of results

In order to validate preliminary diagnosis performed by ODINO, several queries and relevance sets have been defined with physician experts of dermatological diseases.

In particular, through the analysis of retrieval performance, we have measured: *Average Uninterpolated Precision* (AUP) [107] and *Precision*. The AUP is defined as the ratio between the sum of the precision value at each point of hierarchical structure (or node of lattice) where a relevant item appears, and the total number of relevant items.

The performance for Precision and AUP are shown in Figure 57, considering a different number *N* of the clinical manifestations (e.g., *Fever*, *Headache*, *Anemia*, etc.). Let us note that, for *N* larger than 6, the values of Precision and AUP provide good performance results (between 0.7 and 1). Considering these results we can say that knowledge base and methodology are enough selective for available diseases.

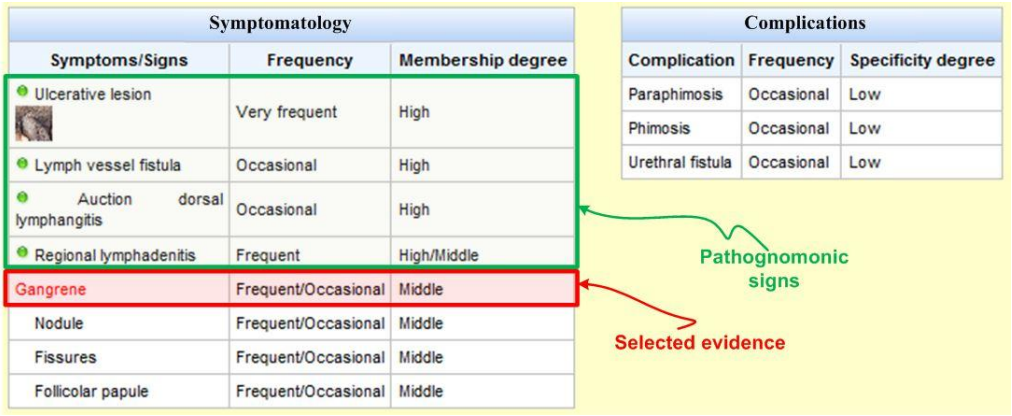



Figure 56. Example of result motivation interface

More generally, by analyzing system usage and physician expectancy, experimental results reveal that 86% of diseases retrieved by the system are coherent with results expected by the physicians; just the remaining 14% of results emerge to be too vague. Specifically, by considering wrong results, we have identified that the problem is strictly related to ambiguous clinical manifestations. More precisely, two weakness have been revealed: cases in

which huge number of clinical manifestations characterize the same disease (i.e., *AIDS*); and cases in which the same clinical manifestation characterizes many disease with similar *Frequency* and *Specificity* degree (i.e., *Fever*).

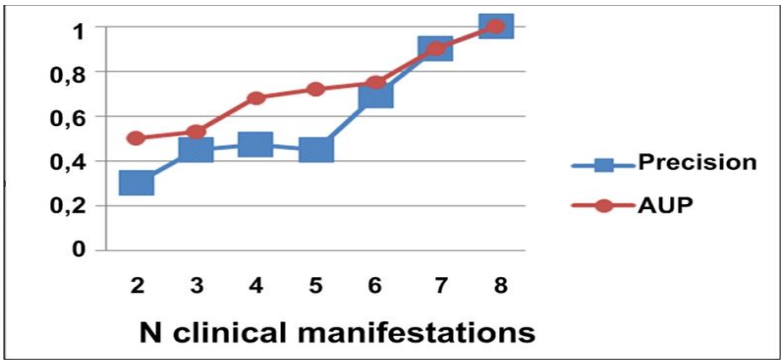



Figure 57. Performance evaluation on Precision and AUP.

### *9.3.3 Faceted search*

ODINO also provides a faceted search of diseases through taxonomy constraints (i.e., Symptomatology, etc.). Faceted search is an exploration technique for structured datasets based on the facet theory.

As described in [108], in faceted search the information space is partitioned using orthogonal conceptual dimensions of the data. Each dimension is called facet and represents important disease partitioning feature. The facet has multiple restriction values and the user selects a restriction value to filter relevant items in the information space. The facet based approach can be directly mapped to navigation in semi-structured data, as well as in *Medical Diseases Ontology* model deployed in ODINO. Specifically, available facets are: symptoms, complications, clinical tests and active ingredients. All of these facets are associated with controlled vocabulary defined according to the SKOS formalism.

Obviously, items that ODINO filters step-by-step are diseases. Figure 58 shows an example of faceted search in ODINO. The left side of Figure 58 shows constraint "step-by-step" selection to filter diseases (constraints are highlighted with box outlined in green color). At each step of the facet-based search, the system returns an ordered list of eligible diagnosis allowing also further filtering.

With respect to text based search, a faceted search is more useful because allows navigation of an unknown dataset through system suggested restriction values at each step. Additionally, facets provide an intuitive user interface eliminating the need to write exact queries; and prevent empty query results by including restriction values that certainly lead to true results.

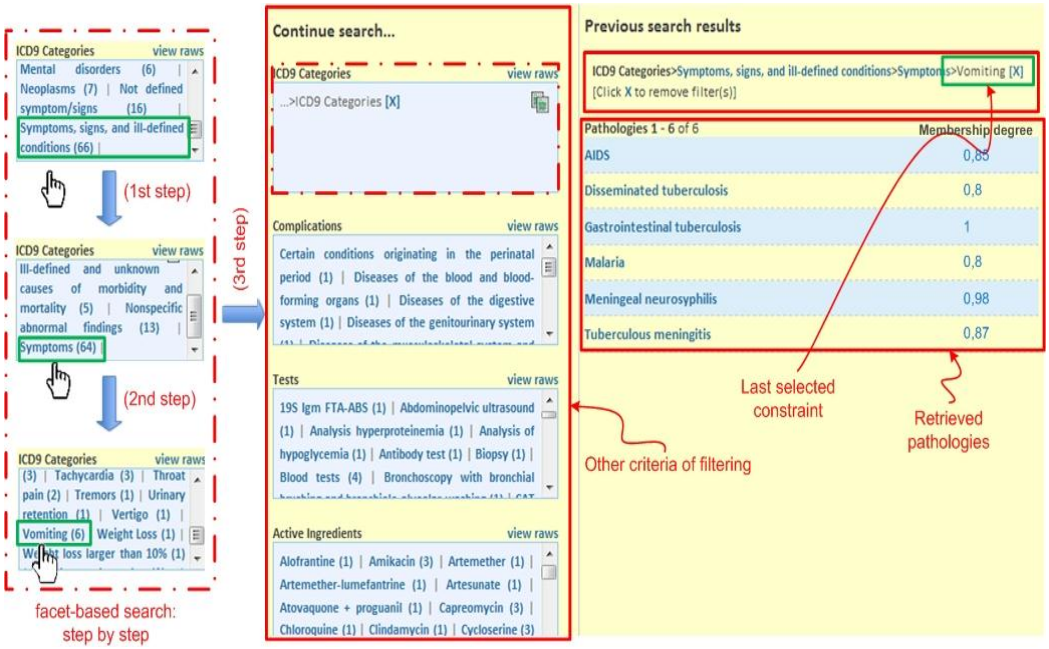



Figure 58. An example of faceted search of diseases

## 9.4 Related Works

In the past years, a great deal of artificial intelligence research has been directed towards the development of expert systems for problem solving in medical diagnosis domain. As examples of developed medical expert systems, we mention: Mycin [109], DXplain [110], Puff [111], Cadiag2 [112], Gideon [113]and Casnet [114]. In particular, interesting approach is used in Cadiag2 that exploits fuzzy set theory to model medical concepts, and fuzzy logic to emulate diagnostic processes.

As argued in [115], [116], fuzzy sets offer linguistic label that well approximate medical texts. In addition, fuzzy logic provides reasoning methods capable of making approximate inference [117], [118]. So, fuzzy set theory provides appropriate basis for the development of a computer-based diagnosis system [119]. Nevertheless, there are some problems related with these expert systems consisting in their limited [120], [121] flexibility, adaptability, extendibility and cooperation capability. Strictly related to the proposed approach, some works, such as [122] and [123], highlight benefits of mashing semantic and fuzzy logic in healthcare. Analogously, the proposed approach, ODINO exploits: Semantic Web formalism to represent fuzzy relations between diseases and symptomatologies; and soft computing techniques to arrange diseases that share these relations.

Conversely, solutions like MEDBOLI [124] and ODDIN [125] use Semantic Web Technologies to develop a software that allows users to make diagnosis. In fact, since in medical discipline homogeneity of terminology is mostly problematic, the semantic technologies can be exploited to make known machine-readable latent relationships [126]. Ontologies allow users to understand meaning of each element, and improve reasoning [127], [128]. Regarding efforts which apply semantic techniques, initiatives such as OpenGalen [129] should be

mentioned, a not-for-profit organization which provides downloadable open source medical terminology. Other initiatives include, for example, OBO Foundries [130], a collaborative experiment among developers of science-based ontologies. Within the scope of this research there are many resources in use such as Biological Ontologies [131], Ontology-based Support for Human Disease Study and Medical Ontologies to support human disease research and control [132], "relations in biomedical ontologies" [133], and SNOMED CT, a concept-oriented controlled vocabulary [134].

However, today, semantic annotation of Web content with metadata is not very common on health websites, but it is sometimes mandated by, e.g., government standards. The annotation processes are often tedious and require capability in the use of large vocabularies, such as Medical Subject Headings (MeSH)[38]. In this perspective, one of the major contribution of ODINO is that diseases are semantically annotated by using typical standards and controlled vocabularies of medical domain, as well as ICD9. These standards are exploited in order to provide feature of faceted based navigation of diseases. Furthermore, thanks to the benefits of semantic technologies usage (flexibility and configurability of the system), faceted navigation supports multi-criteria selection to discover right diseases. In this sense, ODINO was inspired by previous semantic portals such as SWED [105], the MultimediaN E-culture demonstrator [134] and HealthFinland [135]. Another contribution of ODINO is the definition of sharable *Medical Disease Ontology* model capable to represent relation degree between disease and symptoms coming from statistical analysis included in the book [104].

## 9.5 Conclusion

This chapter presents a project named ODINO, an ontology based web application that provides features to daily support medical practices. In particular, ODINO exploits Semantic Web formalisms in order to model medical knowledge. On the other hand, Fuzzy Formal Concept Analysis and ontology inference are applied to support preliminary medical diagnosis.

The availability of rich set of multimedia information allows to provide rapid training feature. Moreover, thanks to the semantic modeling of knowledge base, a faceted search of diseases has been developed.

Benefits deriving from the applied approach are especially related to the reusability of the knowledge layer, scalability and flexibility of the main features, like as faceted search.

Future works are related to introduce support to improve the exploitation of multimedia information, available in the knowledge base, during faceted search and preliminary diagnosis.

[38] http://www.nlm.nih.gov/mesh/

# Chapter 10

# Conclusion and Future Work

This chapter closes the thesis work by describing a short summary. Furthermore, Section 10.2 describes future challenges.

## 10.1 Summary

This research work addresses methodology for automatic knowledge extraction taking into account text corpus (e.g., file txt, pdf, etc.). In particular, extracted knowledge has been translated into an ontology artifact by using semantic web formalism, such as: OWL and RDF. This is obtained by defining methodology of Ontology Extraction in order to map Fuzzy Lattice into the ontology structure. Furthermore, this methodology has been extended and applied to different research objectives: Semantic Annotation, Information Retrieval and Faceted Browsing. The methodologies defined exploit common sense knowledge, i.e. Wikipedia and Wordnet.

These methodologies have been applied to different case study. Specifically:

- *Ontology/Taxonomy Extraction*: to conceptualize content included in RSS feeds in the context of e-learning available from the web directory (e.g., OpenLearn[39], Merlot[40], etc.).
- *Information Retrieval*: applied to build a search engine of RSS Feed in the context of e-learning, and a medical diagnosis system in order to support disease discovery starting from symptoms and signs.
- *Facet Browsing*: applied to support general web resources organization and User Generated Content (UGC) in order to support the enterprise competency management.
- *Semantic Annotation:* applied to the annotation plain text, but the framework is applicable to other types of multimedia resources.

## 10.2 Future Work

In order to address interoperability and evolution of the extracted knowledge models, the future challenges go in the following directions:

[39] http://openlearn.open.ac.uk/

[40] http://w[3]ww.merlot.org/merlot/index.htm

- *Ontology alignment*. Study, definition and development of the approaches for extracting approximate matching in order to harmonize heterogeneous ontology conceptualization. The result of a matching operation is the evaluation of relation between two ontologies. Concepts matching enable us to augment knowledge discovery performances.
- *Ontology Merging.* Study, definition and development of the approach s to create ontology from two or more source ontologies. The new ontology will unify and in general replace the original source ontologies. By the merging of the concepts ontology, we intend to support knowledge Extraction applications through reduction of redundancy.

# Bibliography


[1] B. Fortuna, M. Grobelnik, D. Mladenic. OntoGen: Semi-automatic Ontology Editor. HCI International 2007, Beijing, July 2007.

[2] Paul Buitelaar and Daniel Olejnik and Michael Sintek. A Protege Plug-in for Ontology Extraction from Text Based on Linguistic Analysis. In Proceedings of the 1st European Semantic Web Symposium (ESWS), 2004.

[3] Navigli R., Velardi P., Gangemi A. Ontology Learning and its application to automated terminology translation. IEEE Intelligent Systems, vol. 18:1, January/February 2003.

[4] Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer, and D. Wenke, OntoEdit: Collaborative ontology development for the Semantic Web, The Semantic Web - ISWC 2002, First International Semantic Web Conference, Sardinia, Italy, June 9-12, Proceedings (Ian Horrocks and James A. Hendler, eds.), Lecture Notes in Computer Science, vol. 2342, Springer, 2002.

[5] Faure D., Nédellec C. and Rouveirol C. Acquisition of Semantic Knowledge using Machine learning methods: The System ASIUM. Technical report number ICS-TR-88-16, 1998.

[6] Cimiano, P., and Volker, J. 2005. A framework for ontology learning and data-driven change discovery. In Proc. Of the NLDB'2005.

[7] M. F. Lopez, A.G. Perez, "Overview and Analysis of Methodologies for Building Ontologies", In Knowledge Engineering Review, 17(2), 2002.

[8] Lopez, M. F., & Perez, A. G. (2002). Overview and analysis of methodologies for building ontologies. Knowledge Engineering Review, 17(2).

[9] Uschold, M., & Grüninger, M. (1996). Ontologies principles methods and applications. Knowledge Engineering Review, 11(2).

[10] Grüninger, M., & Fox, M. S. (1995). Methodology for the design and evaluation of ontologies. In IJCAI'95, Workshop on basic ontological issues in knowledge sharing, Montreal.

[11] Cho, W. C. and Richards, D. Ontology construction and concept reuse with formal concept analysis for improved web document retrieval. Web Intelli. and Agent Sys. 5, 1 (Jan. 2007), 109-126.

[12] C. Carpineto and G. Romano, Exploiting the Potential of Concept Lattices for Information Retrieval with CREDO Journal of Universal Computer Science, 2004, volume 10, (8), pp. 985-1013.

[13] C. De Maio, G. Fenza, V. Loia, S. Senatore, "Towards an automatic Fuzzy Ontology generation" FUZZ-IEEE 2009, ICC Jeju, Jeju Island, Korea, 20-24 August, 2009.

[14] Maedche, A. and Staab, S. Ontology Learning for the Semantic Web. IEEE Intelligent Systems, 16 (2). 72-79, 2001

[15] S. Pollandt, Fuzzy-Begriffe: Formale Begriffsanalyze unscharfer Daten. Berlin-Heidelberg: Springer-Verlag, 1996

[16] Q. T. Tho, S. C. Hui, A. C. M. Fong, and T. H. Cao, "Automatic Fuzzy Ontology generation for Semantic Web", IEEE Transactions on Knowledge and Data Engineering, Vol. 18(6), pp. 842- 856, 2006.

[17] Didier Dubois, Henri Prade. Possibility theory and formal concept analysis in information systems. Proc. 13th International Fuzzy Systems Association World Congress IFSA-EUSFLAT 2009, Lisbon, July 20-24, 2009.

[18] Ganter, B., & Wille, R. (1999). Formal concept analysis: Mathematical foundations. Berlin, Heidelberg: Springer.

[19] L.A. Zadeh, "Fuzzy Logic and Approximate Reasoning," Synthese, vol. 30, pp. 407-428, 1975.

[20] A. Burusco, R. Fuentes-Gonzalez, Construction of the L-Fuzzy concept lattice, Fuzzy Sets and Systems 97 (1998) 109icss,

[21] R. Belohlavek, Fuzzy Galois connections, Math. Logic Quarterly 45 (4) (1999) 497–504.

[22] R. Belohlavek, Lattices of fixed points of fuzzy Galois connections, Math. Logic Quarterly 47 (2001) 111(1998)

[23] R. Belohlavek, Fuzzy Relational Systems, Foundations and Principles, Kluwer, New York, 2002.

[24] Rouane-Hacene M., Huchard M., Napoli A., and Valtchev P. A proposal for combining formal concept analysis and description logics for mining relational data. In Kuznetsov S.O and Schmidt S., editors, Proceedings of the 5th International Conference on Formal Concept Analysis (ICFCA 2007), Clermont-Ferrand, LNAI 4390, pages 5165. Springer, Berlin, 2007;

[25] Bendaoud R., Rouane-Hacene M., Toussaint Y., Delecroix B., and Napoli A. Textbased ontology construction using relational concept analysis. In Flouris G. and d'Aquin M., editors, Proceedings of the International Workshop on Ontology Dynamics, Innsbruck (Austria), pages 5568, 2007.

[26] Kuznetsov, S.O., Obiedkov, S.A.: Comparing performance of algorithms for generating concept lattices. J. Exp. Theor. Artif. Intelligence 14(2/3), 189-216 (2002).

[27] Bordat, J.P., Calcul pratique du treillis de Galois d'une correspondance, Math. Sci. Hum.,1986, no. 96, pp. 31–47.

[28] Ganter, B. and Reuter, K., Finding All Closed Sets: A General Approach, Order, 1991, vol. 8, pp. 283-290.

[29] Kuznetsov, S.O., A Fast Algorithm for Computing All Intersections of Objects in a Finite Semi-lattice, Automatic Documentation and Mathematical Linguistics, vol. 27, no. 5 (1993) 11–21.

[30] Lindig, C., Algorithmen zur Begriffsanalyse und ihre Anwendung bei Softwarebibliotheken, (Dr.-Ing.) Dissertation, Techn. Univ. Braunschweig, 1999.

[31] Nourine L. and Raynaud O., A Fast Algorithm for Building Lattices, Information Processing Letters, vol. 71, 1999, 199-204.

[32] Norris, E.M., An Algorithm for Computing the Maximal Rectangles in a Binary Relation, Revue Roumaine de Mathématiques Pures et Appliquées, 1978, no. 23(2), pp. 243–250.

[33] Godin, R., Missaoui, R., and Alaoui, H., Incremental Concept Formation Algorithms Based on Galois Lattices, Computation Intelligence, 1995.

[34] Van der Merwe, D., Obiedkov, S.A., Kourie, D.G.: AddIntent: A New Incremental Algorithm for Constructing Concept Lattices. In: Eklund, P. (ed.) ICFCA 2004. LNCS (LNAI), vol. 2961, pp. 372–385. Springer, Heidelberg (2004)

[35] Semantic Wave 2006: Executive Guide to the Business Value of Semantic technologies. Semantic Interoperability Community of Practice (SICoP), 2006.

[36] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, and Eve Maler. Extensible markup language (XML) 1.0 (second edition). Technical report, www.w3c.org, 2000.

[37] Resource description framework (RDF) model and syntax specification. Technical report, www.w3c.org, February 1999.

[38] Dan Brickley and R.V. Guha. RDF vocabulary description language 1.0: RDF Schema. Technical report, www.w3c.org, April 2002.

[39] Mike Dean, Dan Connolly, Frank van Harmelen, James Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein. Web ontology language (OWL) reference version 1.0. Technical report, www.w3c.org, 2002.

[40] Powell, M. Nillson, A. Naeve, P. Johnston, and T. Baker. Dcmi abstract model. Web Page, June 2007. http://dublincore.org/documents/abstract-model/

[41] DCMI Usage Board. Dcmi metadata terms. Web Page, January 2008. http://dublincore.org/documents/dcmi-terms/

[42] Dan Brickley and Libby Miller. The friend of a friend (FOAF) project | FOAFproject. http://www.foaf-project.org/.

[43] J. Golbeck and M. Rothstein. Linking social networks on the web with foaf: A semantic web case study. In AAAI, pages 1138–1143, 2008.

[44] Edd Dumbill. XML watch: Finding friends with XML and RDF. http: //www.ibm.com/developerworks/xml/library/x-foaf.html.

[45] Dan Brickley and Libby Miller. FOAF vocabulary specification. http://xmlns.com/foaf/spec/.

[46] Peter Mika. Flink: Semantic web technology for the extraction and analysis of social networks. Journal of Web Semantics, 2005.

[47] Uldis Bojars and John Breslin. sioc-project.org | Semantically-Interlinked online communities. http://sioc-project.org/

[48] U. Bojars, J. G. Breslin, A. Finn, and S. Decker. Using the semantic web for linking and reusing data across web 2.0 communities. Web Semantics: Science, Services and Agents on the World Wide Web, 6(1):21–28, 2008

[49] Uldis Bojars and John Breslin. SIOC core ontology specification. http://rdfs.org/sioc/spec/.

[50] U. Bojars, B. Heitmann, and E. Oren. A prototype to explore content and context on social community sites. In SABRE Conference on Social Semantic Web (CSSW 2007), 2007

[51] S. Jupp, S. Bechhofer, and R. Stevens. SKOS with OWL: don't be full-ish! In Fifth International workshop on OWL Experiences and Directions, 2008.

[52] Greene, B., & Rubin, G. (1971). Automatic grammatical tagging of English, Technical report. Department of Linguistics, Brown University, Providence, Rhode Island.

[53] Klein, S., & Simmons, R. (1963). A computational approach to grammatical coding of English words. JACM 10.

[54] Cutting, D., Kupiec, J., Pederson, J., & Sibun, P. (1992). A practical part-of-speech tagger. In Proceedings of the third conference on applied 5atural language processing (pp. 133–140), ACL, Trento, Italy.

[55] Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In Proceedings of the international conference on new methods in language processing (pp. 44–49), Manchester, UK.

[56] Porter, M. F. (1980). An algorithm for suffix stripping. Program, 14, 130–137.

[57] Porter, M. F. (2001). Snowball: A language for stemming algorithms. <http://snowball.tartarus.org/texts/introduction.html>.

[58] Hooper, R., & Paice, C. (2005). The Lancaster stemming algorithm. <http://www.comp.lancs.ac.uk/computing/research/stemming/> Accessed 18.04.06.

[59] Ramos, J. (2003). Using TF-IDF to determine word relevance in document queries. First International Conference on. Machine Learning.

[60] Salton, G., & Buckley, C. (1987). Term weighting approaches in automatic text retrieval. Technical report. UMI order number: TR87-881. Cornell University.

[61] Zhou, B., Hui, S. C., and Chang, K., "A formal concept analysis approach for Web Usage Mining" in Intelligent information Processing II, Z. Shi and Q. He, Eds. Springer-Verlag, London, 437-441, 2005.

[62] C.J. Van Rijsbergen, Information Retrieval, second ed., Dept. of Computer Science, University of Glasgow, 1979.

[63] Wu Z., Palmer M., Verb Semantics and Lexical Selection, In Proc. of the 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, New Mexico, 1994.

[64] K.-P. Yee, K. Swearingen, K. Li, and M. Hearst. Faceted metadata for image search and browsing. In CHI. 2003.

[65] Paul Shabajee Summary Report from SWARA Survey of Biodiversity/Wildlife Information in the UK, - HP Technical Report, 2004

[66] J. Gennari, M.A. Musen, R.W. Fergerson, W.E. Grosso, M. Crubézy, H. Eriksson, N.F. Noy, S.W. Tu, The Evolution of Protègè: An Environment for Knowledge-Based Systems Development, Tech. Rep. SMI-2002-0943, Stanford University, 2002

[67] Samir Tartir, I. Budak Arpinar, Michael Moore, Amit P. Sheth, and Boanerges Aleman-Meza. Ontoqa: Metric-based ontology quality analysis. Proceedings of IEEE Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources, November:45_53, 2005.

[68] Gibbins, N., Harris, S., Dix, A., & Schraefel, M. C. (2004). Applying mspace interfaces to the semantic web. Tech. rep. 8639, ECS, Southampton.

[69] Yang, Y.: An evaluation of statistical approaches to text categorization. Journal of Information Retrieval, 1999, Vol. 1 (1/2) 67–88

[70] A. P. McAfee. "Enterprise 2.0: The Dawn of Emergent Collaboration", "MIT Sloan Management Review", 47:3, pp. 21–28 2006.

[71] G. Acampora, M. Gaeta, F. Orciuoli, P. Ritrovato, "Exploiting Semantic and Social Technologies for Competency Management", submitted to The 10th IEEE International Conference on Advanced Learning Technologies - ICALT 2010, Sousse (Tunisia), July 5-7, 2010.

[72] V. Radeski and Z. Dika and F. Trichet "CommOn: A Framwork for Developing Knowledge-Based Systems Dedicated to Competency-Based Management". In 28th International Conference on Information Technology Interfaces, Cavtat-Dubrovnik, pages. 419-424, 2006

[73] Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: A Core of Semantic Knowledge. In: WWW 2007. 16th international World Wide Web conference, ACM Press, New York

[74] Jason Michelizzi. Semantic relatedness applied to all words sense disambiguation. Master's thesis, Graduate School of the University of Minnesota, 2005.

[75] Siegfried Handschuh and Steen Staab. Authoring and annotation of web pages in cream. 2002

[76] Lawrence Reeve and Hyoil Han. 2005. Survey of semantic annotation platforms. In Proceedings of the 2005 ACM symposium on Applied computing (SAC '05), Lorie M. Liebrock (Ed.). ACM, New York, NY, USA, 1634-1638. DOI=10.1145/1066677.1067049 http://doi.acm.org/10.1145/1066677.1067049

[77] V. Uren, P. Cimiano, J. Iria, S. Handschuh, M. Vargasvera, E. Motta, and F. Ciravegna. Semantic annotation for knowledge management: Requirements and a survey of the state of the art. Web Semantics: Science, Services and Agents on the World Wide Web, 4(1):14–28, January 2006.

[78] Kettler, B., Starz1, J., Miller, W., Haglich, P.: A Template-Based Markup Tool for Semantic Web Content. In Proc. of the 4th Int. Semantic Web Conf. (ISWC 2005), LNCS, SpringerVerlag, Vol. 3729 (2005) 446–46

[79] Vargas-Vera M., Motta E., Domingue J., Lanzoni M., Stutt A., Ciravegna F. (2003) MnM: A Tool for Automatic Support on Semantic Markup, KMi Technical Report, TR Number133, Sept. 2003.

[80] Kogut P., Holmes W. (2001) AeroDAML: Applying Information Extraction to Generate DAML Annotations from Web Pages. In Workshop on Knowledge Markup and Semantic Annotation. At 1st International Conference on Knowledge Capture (K-CAP 2001), Victoria, B.C., Canada.

[81] Domingue, Dr John, Dzbor, Dr Martin, Motta, and Prof Enrico. Magpie: Browsing and navigating on the semantic web. Proceedings ACM Conference on Intelligent User Interfaces (IUI), pages 191{197, January 2004. Portugal. [cited at p. 2, 8, 9, 13]

[82] F. Ciravegna, S. Chapman, A. Dingli, Y. Wilks, Learning to harvest information for the Semantic Web, in: Proceedings of the 1st European Semantic Web Symposium, May 10–12, 2004, Heraklion, Greece, 2004.

[83] Kuznetsov, Sergei, Machine Learning and Formal Concept Analysis - Concept Lattices, Lecture Notes in Computer Science, 2004, Volume 2961/2004, Springer Berlin / Heidelberg

[84] R. Wille, R. Hoberg, and V. Beeh. Relational Concept Analysis: Semantic Structures in Dictionaries and Lexical Databases. Shaker Verlag,Aachen, 1998.

[85] R. Bendaoud, A. Napoli, and Y. Toussaint. Formal concept analysis: A unified framework for building and refining ontologies. Knowledge Engineering: Practice and Patterns, April:156_171, 2008.

[86] Chen W., Hayashi Y., Jin L., Mitsuru I., Mizoguchi R., An Ontologybased Intelligent Authoring Tool, 6th Int. Conference on Computers in Education, pp. 41–49.

[87] Brase J. and Nejdl W., Ontologies and Metadata for eLearning, Handbook on Ontologies, 2004, pp. 555–574.

[88] Stojanovic L., Staab S., Studer R., eLearning based on the Semantic Web, In WebNet2001 - World Conference on the WWW and Internet, Orlando, Florida, USA, 2001.

[89] Murugesan, S. Understanding Web 2.0, IT Professional, vol.9, no.4 pp. 34–41, 2007

[90] Gascuena, J. M., Fernandez-Caballero, A., Gonzalez, P. Domain Ontology for Personalized E-Learning in Educational Systems. ICALT IEEE Computer Society (2006) 456-458

[91] Itmazi, J. A., Gea Meg´ıas, M., Using Recommendation Systems in Course Management Systems to recommend Learning Objects, Int. Arab J. Inf.Technol., 5 (3), 2008, 234-240

[92] Khribi, M. K., Jemni, M., & Nasraoui, O., Automatic Recommendations for E-Learning Personalization Based on Web Usage Mining Techniques and Information Retrieval. Educational Technology & Society, 12 (4), 2009, 30-42

[93] G. Acampora, M. Gaeta, and V. Loia, Hierarchical optimization of personalized experiences for e-learning systems through evolutionary models, Neural Computing & Applications.

[94] Albano G., Gaeta M., Ritrovato P., IWT: an innovative solution for AGS e-Learning model, International Journal of Knowledge and Learning, vol. 3, no.2, pp. 209–224, 2007.

[95] M. Gaeta, P. Ritrovato, F. Orciuoli, Advanced Ontology Management System for Personalised e-Learning, Knowledge-Based Systems, 22 (2009), pp. 292-301

[96] Sicilia, M. A., Garcia-Barriocanal, E., On the Convergence of Formal Ontologies and Standardized e-Learning, Journal of Distance Education Technologies 3(2),2005, 13-29.

[97] Drachsler, H., Hummel, H. G. K., Van Den Berg, B., Eshuis, J., Berlanga, A. J., Recommendation strategies for e-learning: preliminary effects of a personal recommender system for lifelong learners, 2007

[98] Zhuhadar, L., Nasraoui, O., Wyatt, R., Romero, E., Multi-language Ontology-Based Search Engine, ACHI, 2010, 13-18

[99] Cimiano, P., Hotho, A., Staab, S. ”Learning concept hierarchies from text corpora using formal concept analysis”. Journal of Artificial Intelligence Research 24, 305-339, 2005

[100] Fang P., Zheng S. “A Research on Fuzzy Formal Concept Analysis Based Collaborative Filtering Recommendation System,” Knowledge Acquisition and Modeling, 2009. KAM ’09. Second International Symposium on , vol.3, no., pp.352-355, Nov. 30 2009-Dec. 1 2009

[101] Lau R. Y. K., Chung A. Y. K., Song D., Huang Q. “Towards Fuzzy Domain Ontology Based Concept Map Generation for E-Learning” in Advances in Web Based Learning - ICWL 2007, pp. 90-101, 2008

[102] Dattolo A. and Luccio F. “Formalizing a model to represent and visualize concept spaces in e-learning environments”. Proceedings of the 4th Webist International conference, May 4-7, Funchal, Madeira, Portugal, 2008, pp. 339-346.

[103] Cassidy, K.,Walsh, A. and Coghlan, B. “Using Hyperbolic Geometry for Visualization of Concept Spaces for Adaptive eLearning”. A3H: 1st Inter. Workshop on Authoring of Adaptive & Adaptable Hypermedia, June 20, 2006, Dublin, Ireland (2006).

[104] A. Morrone. GLOBAL DERMATOLOGY ricerca clinica e logica matematica in Medicina delle Migrazioni, Manuale pratico.

[105] D. Reynolds, P. Shabajee, S. Cayzer, Semantic Information Portals, in: Proceedings of WWW 2004, Alternate track papers & posters, ACM Press, New York, New York, 2004.

[106] De Maio, C., Fenza, G., Loia, V. and Senatore, S. (2010), Knowledge structuring to support facet-based ontology visualization. International Journal of Intelligent Systems, 25: 1249–1264. doi: 10.1002/int.20451.

[107] N.Nanas, V.Uren and A. de Roeck, “Building and Applying a Concept Hierarchy Representation of a User Profile", In Proceedings of the 26th annual international ACM SIGIR Conference on Research and Development in Information Retrieval, ACM Press, 2003.

[108] Eyal Oren, Renaud Delbru, and Stefan Decker. Extending Faceted Navigation for RDF Data. ISWC 2006, volume 4273 of Lecture Notes in Computer Science, chapter 40, pages 559–572.

[109] E. H. Shortliffe. Computer-Based Medical Consultations: MYCIN. Elsevier, New York, 1976.

[110] Barnett GO, Cimino JJ, Hupp JA, Hoffer EP. DXplain: an evolving diagnostic decision-support system. JAMA 1987;258:67-74.

[111] J. S. Aikins, J. C. Kunz, E. H. Shortliffe, R. J. Fallat. Puff: An expert system for interpretation of pulmonary function data. Comput. Biomed. Res., 3(16):199–208, 1983.

[112] K. Adlassing. Cardiag 2 expert system. IEEE Transactions on Systems, Man and Cybernetics, SMC-16(2), 1986.

[113] Edberg SC. Global infectious diseases and epidemiology network (GIDEON): a world wide Web-based program for diagnosis and informatics in infectious diseases. Clin Infect Dis. 2005; 40(1): 123-126.

[114] S. M. Kulikowski, C. A. Weiss. Representation of expert knowledge for consultation: the CASNET and EXPERT projects. Artificial Intelligence in medicine. Szolovits, P. (Ed.), Boulder: Westview Press, pages 21–56, 1982.

[115] H. Bossel, S. Klaczko, and N. Muller, A fuzzy-algorithmic approach to the definition of complex or imprecise concepts, in Systems Theory in the Social Sciences, , Eds. Stuttgart: Birkhauser Verlag, 1976, pp. 202-282.

[116] L. A. Zadeh, Linguistic variables, approximate reasoning and dispositions, Med. Iniform, vol. 8, pp. 173-186, 1983.

[117] L. A. Zadeh, Lotfi A., Outline of a new approach to the analysis of complex systems and decision processes, IEEE Trans. Svst., Man, Cybern., vol. 3, pp. 28-44, 1973.

[118] R. E. Bellman and L. A. Zadeh, Local and fuzzy logics, memo. ERL-M584, Electronics Research Laboratory, College of Engineering, University of California, Berkeley 94720, May 11, 1976.

[119] K. -P. Adlassnig, A survey on medical diagnosis and fuzzy subsets, in Approximate Reasoning in Decision Analysis. M. M. Gupta and E. Sanchez Eds. New York: North-Holland, 1982, pp. 203-217.

[120] B. Iantovics, C. Chira, D. Dumitrescu. Principles of the Intelligent Agents. Casa Cartii de Stiinta Press, Cluj-Napoca, 2007.

[121] J. Kuhl, E.J. Graham. Esagent: Expert system control of simulated agent-based mobile robots. Intelligent Systems Research Laboratory, Technical Report TR-ISRL-04-02 University of Louisville, Louisville, 2004.

[122] Acampora, Giovanni; Lee, Chang-Shing; Wang, Mei-Hui; , FML-Based Ontological Agent for Healthcare Application with Diabetes. *Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT '09.* vol.3, no., pp.413-416, 15-18 Sept. 2009.

[123] Chang-Shing Lee; Mei-Hui Wang; Acampora, G.; Loia, V.; Chin-Yuan Hsu; , "Ontology-based intelligent fuzzy agent for diabetes application," *Intelligent Agents, 2009. IA '09. IEEE Symposium on* , 2009.

[124] Rodríguez A., Mencke M., Alor Hernandez G., Posada Gómez R. and J. M. Gomez., Medboli: Medical Diagnosis Based on Ontologies and Logical Inference. The Third International Conference on Digital Society, ICDS 2009 (accepted, to appear), 2008

[125] García-Crespo Á., Rodríguez González A., Mencke M., J. M. Gomez., Colomo Palacios R.: ODDIN: Ontology-driven differential diagnosis based on logical inference and probabilistic refinements. Expert Syst. Appl. 37(3): 2621-2628 (2010)

[126] Fuentes-Lorenzo, D., Morato, J., & Gómez, J. M. (2009). Knowledge management in biomedical libraries: A Semantic Web approach. Information Systems Frontiers, 11(4), 471–480.

[127] García Sanchez, F., Fernandez Breis, J., Valencia García, R., Gómez, J. M., & Martínez- Bejar, R. (2008). Combining Semantic Web technologies with multiagent systems for integrated access to biological resources. Journal of Biomedical Informatics, 41(5), 848–859.

[128] Gomez, J. M., Colomo-Palacios, R., Mayoral, M. R., & Garcia-Crespo, A. (2008). Microarray information and data integration using SAMIDI. In J. R. Rabuñal, J. Dorado, & A. Pazos (Eds.), Encyclopedia of artificial intelligence. Hershey, PA: IGI Global.

[129] Rector, A. L., Rogers, J. E., Zanstra, P. E., & Van Der Haring, E. (2003). OpenGALEN: Open source medical terminology and tools. In Proceedings of the American medical informatics association symposium 2003 (pp. 982–985).

[130] Smith, B., Ashburner, M., Rosse, C., Bard, J., Bug, W., Ceusters, W., et al. (2007). The OBO Foundry: Coordinated evolution of ontologies to support biomedical data integration. Nature Biotechnology, 25(11), 1251–1255.

[131] Lambrix, P., Tan, H., Jakoniene, V., & Strömbäck, L. (2007). Biological ontologies. In C. J. O. Baker & K. H. S. Cheung (Eds.), Semantic Web revolutionizing knowledge discovery in the life sciences (pp. 85–89). Berlin: Springer.

[132] Hadzic, M., & Chang, E. (2005). Medical Ontologies to support human disease research and control. International Journal of Web and Grid Services, 1(2), 139–150.

[133] Smith, B., Ceusters, W., Klagges, B., Köhler, J., Kumar, A., Lomax, J., et al. (2005). Relations in biomedical ontologies. Genome Biology, 6(5), 1425–1433.

[134] Schulz S, Hanser S, Hahn U, Rogers, J. The semantics of procedures and diseases in SNOMED CT, Methods Inf Med. 2006; 45(4): 354-358.

[135] Suominen, O., Hyvönen, E., Viljanen, K. and Hukka, E.: HealthFinland-A National Semantic Publishing Network and Portal for Health Information. Web Semantics: Science, Services and Agents on the World Wide Web, 7(4), pp. 287--297 (2009).